

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIA DA COMPUTAÇÃO

Matheus Paulon Novais

Geração e Análise de Grafos Curriculares para Planejamento Acadêmico

Florianópolis
2025

Matheus Paulon Novais
Geração e Análise de Grafos Curriculares para Planejamento Acadêmico

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Graduação em Ciência da Computação.

Florianópolis, 2025.

Prof. Lúcia Helena Martins Pacheco, Dra.
Coordenadora do Curso

Banca Examinadora:

Gustavo Zambonin, Me.
Orientador
Universidade Federal de Santa Catarina

Prof. Jean Everson Martina, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Rafael de Santiago, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Antonio Carlos Mariani, Me.
Avaliador
Universidade Federal de Santa Catarina

Matheus Paulon Novais

Geração e Análise de Grafos Curriculares para Planejamento Acadêmico

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Gustavo Zambonin, Me.

Coorientador: Prof. Jean Everson Martina, Dr.

Florianópolis

2025

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Novais, Matheus Paulon
Geração e Análise de Grafos Curriculares para
Planejamento Acadêmico / Matheus Paulon Novais ;
orientador, Gustavo Zambonin, coorientador, Jean Everson
Martina, 2025.
50 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Ciências da Computação, Florianópolis, 2025.

Inclui referências.

1. Ciências da Computação. 2. Geração de grafos. 3.
Planejamento acadêmico. 4. Organização curricular. I.
Zambonin, Gustavo. II. Martina, Jean Everson. III.
Universidade Federal de Santa Catarina. Graduação em
Ciências da Computação. IV. Título.

AGRADECIMENTOS

Agradeço ao meu orientador, Gustavo, por todo o auxílio durante a produção desse trabalho e a paciência que foi necessária para isso. Ao professor Jean e aos professores orientadores da banca pela correção e pelas aulas que tive durante todo esse período de minha vida acadêmica. Nesse mesmo âmbito, é impossível deixar de agradecer a Universidade Federal de Santa Catarina que permitiu todo o desenvolvimento, pessoal e profissional, junto ao aprendizado possível para chegar onde cheguei e a todos os professores que tive nessa jornada.

À família Beirão-Cabrera que me acolheu em sua casa durante boa parte da minha graduação, dando-me a honra de ter duas famílias durante esse período e ao filho deles, Matheus, que além de compartilhar o mesmo nome, me abrigou em sua casa e tive o prazer de compartilhar a sua amizade durante esses anos.

Aos meus amigos, começando por aqueles que desde o IFSC estão presentes em todos os momentos, seja no grupo que se mantém vivo desde uma matéria de artes ou partilhando a graduação comigo saibam que os considero enormemente. Aqueles que moraram juntos na rua Joe Collaço, meu mais sincero abraço e obrigado. Os dois tansos que atualmente moram comigo sabem a sua importância nisso tudo, mas ainda sim, os agradeço por viver no mesmo teto e estendo essa graça a todos que compõem a Trupe. Para não ficar mais longo, um grande obrigado a todos os amigos que fiz durante essa graduação, cada um de vez tem uma parte nisso.

Ao meu cachorro que me recebe todas as vezes como sendo a primeira vez, e aos dois gatos que cercam minha mesa e meu quarto, obrigado pelo apoio incondicional.

Por fim, expresso minha profunda gratidão aos meus pais, Zirlândio e Denise, a quem devo tudo. Nada disso seria possível sem o apoio de vocês, e dedico meus esforços para retribuir, ainda que minimamente, a grandiosidade do que fizeram por mim. À minha irmã, Beatriz, pelo companheirismo desde sempre. Dirijo também meu agradecimento a toda a minha família, que constitui o alicerce fundamental da minha existência.

“No dia que eu morrer, eu quero o meu caixão pintado de verde e branco como
meu coração”(Mancha Verde)

RESUMO

Compreender a progressão de um aluno em um curso pode ser um desafio complexo quando se consideram apenas os nomes das disciplinas e seus respectivos códigos dispostos linearmente em um currículo estático. A visualização dessas informações e a garantia das interdependências entre as disciplinas tornam-se ainda mais desafiadoras. Para abordar esse problema, existem formas visuais de demonstrar o currículo de um curso, como o uso de grafos, onde os vértices representam as disciplinas, possivelmente com seus códigos, e as arestas indicam as dependências de requisitos. A forma clássica de realizar essa visualização envolve a manipulação manual de cada disciplina e suas dependências dentro de alguma ferramenta gráfica, o que é um processo trabalhoso e suscetível a erros. O objetivo deste trabalho é desenvolver uma aplicação que gere automaticamente esses grafos curriculares, proporcionando uma visualização clara e substancial. A aplicação utiliza o progresso acadêmico individual do estudante para mapear seu estado atual e oferece sugestões para a organização dos futuros semestres. Estas sugestões são geradas por algoritmos que otimizam a seleção de disciplinas com base nos pré-requisitos já cumpridos e nas restrições definidas pelo aluno. A ferramenta desenvolvida deve facilitar a visualização do currículo, mas também auxiliar na tomada de decisões estratégicas por parte dos alunos, contribuindo para um planejamento acadêmico mais eficiente e personalizado.

Palavras-chave: Geração de grafos. Organização curricular. Planejamento acadêmico.

LISTA DE FIGURAS

Figura 1 – Diagrama de fluxo das disciplinas e seus pré-requisitos com setas indicando dependência.	19
Figura 2 – Exemplo de XML do Histórico Escolar Digital.	28
Figura 3 – Representação gráfica do exemplo de mapeamento do currículo.	34
Figura 4 – Representação gráfica do exemplo de mapeamento do currículo.	36
Figura 5 – Identificação das disciplinas disponíveis com base no progresso do aluno.	39
Figura 6 – Resultado da seleção de disciplinas com base na carga horária e valores definidos.	40
Figura 7 – Consulta para o curso de Ciência da Computação.	42
Figura 8 – Consulta de turmas oferecidas para disciplinas.	43
Figura 9 – Consulta de turmas oferecidas para matérias sugeridas.	45

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO	14
1.2	MÉTODO DE PESQUISA	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	TEORIA DE GRAFOS	17
2.1.1	Grafos, Direções e ciclos	17
2.1.2	Ordenação de grafos	18
2.1.3	Coloração de grafos e Gestão de Conflitos de Horário	20
2.1.4	Método do Caminho Crítico	20
2.2	GRAFOS CURRICULARES	22
2.2.1	Formalização do Grafo Curricular como um Grafo Acíclico Direcionado	23
2.2.2	Aplicações ao Planejamento Acadêmico	23
2.2.3	Análise Curricular aplicada a Currículos	25
2.3	DIPLOMA DIGITAL	26
2.3.1	Definição, Validade Jurídica e Segurança Tecnológica	26
2.3.2	A Padronização de Dados como Pilar da Interoperabilidade Acadêmica	27
2.3.3	Impacto e Sinergia com a Geração Automatizada de Grafos Curriculares	28
3	DESENVOLVIMENTO DA SOLUÇÃO	31
3.1	MODELAGEM DO CURRÍCULO ACADÊMICO	31
3.1.1	Fundamentação da Abordagem em Grafo	31
3.1.2	Modelagem do Currículo para um Grafo Direcionado	32
3.1.3	O Esquema do Grafo Curricular: Nós e Relações	32
3.2	VISUALIZAÇÃO DO PROGRESSO DO ESTUDANTE	34
3.2.1	O Controle Curricular como Fonte de Dados do Aluno	35
3.2.2	Extração de Dados e Estruturação do Progresso	35
3.2.3	Mapeamento do Progresso no Grafo Curricular	35
3.3	GERAÇÃO DE SUGESTÕES PARA O SEMESTRE	36
3.3.1	Identificação de Disciplinas Cursáveis	37
3.3.2	Heurística para Valorar Disciplinas	37
3.3.3	Otimização da Grade Semestral	38
4	IMPLEMENTAÇÃO REAL DA APLICAÇÃO	41
5	CONCLUSÃO	47
	REFERÊNCIAS	49

A **ARTIGO 51**

B **CODIGO DA APLICAÇÃO 59**

1 INTRODUÇÃO

No ano de 2022, mais de 36 mil alunos se matricularam na Universidade Federal de Santa Catarina (INFORMACAO(DPGI/SEPLAN), 2023), junto a esse registro são definidas as disciplinas que serão realizadas por cada um desses alunos. A partir do segundo semestre de graduação surge uma das problemáticas do estudante, durante o período de recesso deve-se escolher suas matérias baseadas em recomendações do seu currículo e de sugestões dadas por colegas e professores para isso é necessário ler um documento extenso em qual é citado cada pré-requisito, código e possivelmente carga horária necessária para cursar cada matéria e fazer o pedido para a instituição confirmar.

A organização tradicional da grade curricular, com nomes e códigos lineares, dificulta a compreensão das interdependências e pré-requisitos entre as disciplinas, levando a escolhas equivocadas e atrasos na conclusão do curso. A falta de ferramentas de planejamento eficientes também impede a criação de cronogramas de estudo otimizados, prejudicando o aprendizado e o tempo dos alunos. Soluções tecnológicas, como a geração automatizada de grafos curriculares, podem facilitar a visualização e o planejamento da trajetória acadêmica, permitindo que os estudantes tomem decisões informadas sobre seu percurso educativo.

Para o estudante, a ausência de uma representação estrutural clara exige um esforço manual de mapeamento de pré-requisitos, o que pode resultar em escolhas que não otimizam a trajetória até a conclusão do curso. Nesse contexto, a aplicação de modelos formais, como a teoria dos grafos, oferece uma abordagem estruturada para traduzir essas dependências em um formato computacionalmente analisável. Diante dessa dificuldade de visualização, este estudo propõe o desenvolvimento de um gerador automático de grafos curriculares. A ferramenta visa, portanto, aplicar essa fundamentação teórica para auxiliar os alunos na complexa tarefa de navegar por essa rede de interdependências, facilitando a compreensão da estrutura do curso e a tomada de decisões estratégicas para otimizar seu aprendizado.

Uma ideia básica de um grafo curricular é definir seus vértices como representações das disciplinas de um currículo e as arestas como os pré-requisitos entre elas. Vértices são unidades ou entidades, enquanto arestas indicam as conexões entre pares de vértices (SANTIAGO, 2023), facilitando a visualização das relações e interdependências entre as matérias. Para aprimorar a organização visual e a clareza, conceitos de grafos podem ser aplicados para agrupar disciplinas por semestre ideal, área de conhecimento ou status (concluída, em andamento, pendente), facilitando a identificação de caminhos críticos e a distinção entre diferentes fases do curso.

Além de proporcionar uma visualização clara e intuitiva das relações entre as matérias, a aplicação deverá incorporar os dados oficiais de disciplinas e turmas, cedidos pela Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação (SeTIC). A ferramenta utilizará o controle curricular do próprio aluno para preencher dinamicamente seu progresso, permitindo a geração de sugestões para a organização dos futuros semestres com base em sua trajetória individual. Com esses dados, a ferramenta oferecerá um panorama completo do currículo, auxiliando na construção de um plano de estudos eficiente e personalizado.

1.1 OBJETIVO

Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de uma aplicação computacional para a geração automática de grafos curriculares. A ferramenta se propõe a converter a estrutura de disciplinas e pré-requisitos de um curso, comumente apresentada em formatos lineares, em um modelo de dados baseado em grafos. O sistema permitirá a visualização gráfica das dependências entre as matérias e o mapeamento do progresso individual de um estudante sobre essa estrutura. A partir dessa representação, a aplicação funcionará como uma ferramenta de auxílio ao planejamento acadêmico, apresentando a estrutura do curso e as possíveis trajetórias com base nas regras curriculares vigentes.

Objetivo Específicos

- Demonstração visual dos grafos curriculares criados;
- Visualização do progresso do estudante;
- Sugerir matérias para semestres futuros.

Escopo do Trabalho

Desenvolvimento de uma aplicação que automatiza a criação de grafos curriculares, utilizando de ferramentas visuais para a representação das interdependências entre as disciplinas. O trabalho deve também incluir funcionalidades que ajudem o estudante com a criação de um horário para sua matrícula e a compreensão da sua passagem pela universidade.

Não é do escopo do trabalho se tornar uma ferramenta padrão para todas as universidades, tendo em vista a variação da organização do currículo. Também não é de interesse uma conexão direta com o Sistema de Controle Acadêmico da Graduação (CAGR), para a realização automática de matrícula sendo a ideia de uma ferramenta auxiliar e não substitutiva.

Trabalhos Relacionados

O desafio de visualizar e planejar trajetórias em currículos complexos tem motivado o desenvolvimento de diversas ferramentas. O projeto *Curricular Analytics* (HEILEMAN et al., 2018) estabeleceu um *framework* importante para quantificar a complexidade dos cursos, utilizando métricas de centralidade para identificar gargalos no fluxo dos estudantes. No campo da visualização, Macnair et al. (MACNAIR et al., 2020) demonstraram que interfaces interativas melhoram significativamente a compreensão dos requisitos quando comparadas aos catálogos tradicionais.

Uma iniciativa de destaque é a plataforma *Curricular Analytics* (University of Illinois Chicago, 2024), da University of Illinois Chicago, que aplica métricas quantitativas à análise estrutural de cursos. A ferramenta avalia a complexidade curricular através de indicadores como o *Blocking Factor* (impacto dos pré-requisitos no bloqueio de disciplinas futuras) e o *Delay Factor* (comprimento do caminho crítico para a graduação). Essas métricas permitem identificar gargalos e validar o impacto de reformas acadêmicas, reforçando a eficácia do uso de grafos para modelar e otimizar a progressão discente.

No contexto nacional e especificamente voltado para a realidade da UFSC, Mota (MOTA, 2019) desenvolveu o “Guru da Matrícula”, um sistema multi-institucional projetado para superar as limitações de simuladores anteriores, como o MatrUFSC. A ferramenta foca na montagem da grade horária semestral, oferecendo validação automática de pré-requisitos e detecção de conflitos de horário. Embora tenha avançado na verificação de regras acadêmicas e no suporte à rematrícula imediata, sua abordagem privilegia a simulação da grade semanal.

Este trabalho se diferencia dessas iniciativas ao propor uma mudança de perspectiva: em vez de focar apenas na grade do próximo semestre, utiliza-se a Geração Automatizada de Grafos Curriculares para visualizar a trajetória completa do curso. A ferramenta aqui proposta processa o histórico escolar individual para mapear o progresso sobre o grafo de dependências (DAG), permitindo um planejamento de longo prazo e a identificação de caminhos ótimos para a integralização, e não apenas a resolução de conflitos de horário semanais.

1.2 MÉTODO DE PESQUISA

A abordagem inicial para o trabalho tem como base uma pesquisa bibliográfica sobre a teoria dos grafos e suas aplicações no planejamento acadêmico, incluindo tópicos como análise curricular, ordenação topológica e algoritmos de otimização. Simultaneamente, é realizada uma pesquisa documental para compreender a estrutura dos currículos universitários, utilizando fontes de dados institucionais para analisar a organização das disciplinas, suas cargas horárias e as regras de pré-requisitos. Com essa pesquisa compilada, é construído um modelo formal do currículo como um Grafo Acíclico Direcionado (DAG), estabelecendo uma estrutura computacional para as funcionalidades subsequentes da aplicação.

A implementação das funcionalidades de planejamento acadêmico inicia-se com o tratamento dos dados brutos de disciplinas e turmas fornecidos pela SETIC. Esta fase preliminar envolve a coleta, limpeza e estruturação dessas informações para formar uma base de dados consistente e processável, que servirá de alicerce para as operações subsequentes da aplicação. A partir dessa base de dados formalizada, o escopo do trabalho avança para a aplicação de modelos de otimização sobre a estrutura do grafo. Um dos primeiros passos nesta etapa é, então, a escolha e a verificação de heurísticas que podem auxiliar na modelagem do problema real de seleção de disciplinas, traduzindo os modelos teóricos em regras práticas para o sistema de recomendação.

Demonstração visual dos grafos curriculares criados

Neste estágio, ocorrerá a parte mais visualmente impactante do projeto. Será necessário testar diferentes maneiras de apresentar os grafos, explorar possíveis algoritmos para sua geração e avaliar os protótipos junto com orientadores e opiniões externas. Espera-se um estudo aprofundado na teoria de problemas de grafos conhecidos, adaptando-os para a aplicação. A escolha da melhor visualização e algoritmos garantirá uma ferramenta intuitiva e eficiente para os usuários.

Visualização do progresso do estudante

Esta etapa crucial começa com o processamento dos dados individuais fornecidos pelo usuário, como seu controle curricular ou histórico de disciplinas. O desafio consiste em desenvolver um mecanismo que mapeie com precisão essas informações sobre o grafo curricular completo do curso. Esse mapeamento permitirá a diferenciação visual do estado de cada disciplina, como, por exemplo, atribuindo cores distintas para matérias já concluídas, em andamento e pendentes. Devido às possíveis diferenças e nuances entre as estruturas curriculares de cada curso, é necessário um trabalho de validação para garantir que a representação gráfica seja um reflexo fiel da trajetória acadêmica do aluno, proporcionando um panorama claro.

Sugerir matérias para semestres futuros

Nesta etapa, o processo avança da verificação de pré-requisitos para a geração de sugestões para os semestres futuros, através da aplicação de modelos teóricos. O sistema empregará uma heurística definida para a seleção de disciplinas, que considera o limite de créditos do semestre como uma restrição fundamental. Essa abordagem buscará maximizar uma métrica de progresso, a qual será detalhada posteriormente. Adicionalmente, o algoritmo poderá incorporar conceitos do Método do Caminho Crítico para identificar e priorizar disciplinas que são cruciais para o avanço no curso. O resultado será um conjunto de recomendações que respeita as dependências e é gerado com base nesses critérios predefinidos.

2 FUNDAMENTAÇÃO TEÓRICA

A Seção 2.1 deste capítulo aborda os conceitos fundamentais da teoria de grafos, seguido pela formalização do conceito de grafos curriculares e sua aplicação na modelagem de estruturas acadêmicas, na Seção 2.2. Adicionalmente, será apresentada uma análise sobre a iniciativa do Diploma Digital na Seção 2.3, detalhando o ecossistema de dados padronizados que ele estabelece e sua importância para a interoperabilidade em sistemas de informação educacionais. Por fim, será ainda brevemente explorada a sinergia entre esses domínios, demonstrando como a padronização de dados promovida pelo Diploma Digital atua como a tecnologia habilitadora para a geração e análise preditiva dos grafos curriculares, conectando a teoria matemática aos objetivos práticos deste trabalho.

2.1 TEORIA DE GRAFOS

Nesta seção, estabelece-se o alicerce matemático necessário para a representação e análise computacional de estruturas acadêmicas. A discussão inicia-se com as definições fundamentais de grafos, refinando-se até o conceito de Grafos Acíclicos Direcionados (DAGs), estrutura escolhida para modelar os currículos. Subsequentemente, exploram-se algoritmos essenciais para a navegação e organização dessa estrutura, como a ordenação topológica para o sequenciamento de estudos e a coloração de grafos para a gestão de conflitos de horário. Por fim, são apresentados modelos de otimização combinatória, especificamente o Método do Caminho Crítico e o Problema da Mochila.

2.1.1 Grafos, Direções e ciclos

Para saber o que é um grafo curricular deve-se primeiro entender o que é um grafo.

Definição 2.1.1. (DIESTEL, 2025) Um *grafo* $G = (V, E)$ é um par de dois conjuntos de forma que $E \subseteq V \times V$, de forma que os elementos de E sejam subconjuntos de V com 2 elementos. [...] Os elementos de V são de vértices (nodos ou pontos) do grafo G , os elementos de E são arestas (ou linhas).

Durante este mesmo parágrafo de seu livro, Diestel observa que a representação mais comum de um grafo se dá através do desenho de pontos, representando os vértices, e linhas conectando esses pontos, representando as arestas. Essa visualização intuitiva é fundamental para a compreensão da estrutura de um grafo. No entanto, quando se aplica essa teoria ao contexto de currículos acadêmicos, a simples representação de vértices e arestas se mostra insuficiente. Isso ocorre porque currículos, especialmente os acadêmicos, exigem uma ordem específica de disciplinas, considerando pré-requisitos e sequências lógicas de matérias que devem ser seguidas.

Para essa necessidade de ordem e sequência deve-se considerar um tipo particular de grafo que incorpora a noção de orientação. É nesse ponto que os *grafos orientados* se tornam

relevantes. Para entender sua definição, retomamos o conceito fundamental de grafo, mas agora introduzimos a ideia de que suas arestas possuem direções, podem assim ser chamadas de arcos.

Definição 2.1.2. (SANTIAGO, 2023) Um *grafo orientado* é definido como uma dupla $G = (V, A)$, onde V continua representando o conjunto de vértices, mas A passa a ser o conjunto de arcos. Cada arco em A é um par ordenado (u, v) , com ambos $(u, v \in V)$.

Esse par ordenado representa um arco que se origina no vértice u e termina no vértice v , indicando uma direção específica na relação entre esses vértices. Essa representação com arcos direcionados é crucial para modelar relações que possuem uma ordem ou dependência, como as disciplinas em um currículo.

Um grafo orientado é o suficiente para representar um currículo acadêmico, oferecendo uma visualização clara das dependências e pré-requisitos entre as disciplinas. No entanto, é possível refinar essa representação, fechando ainda mais a sua categorização. Ao longo da trajetória de um estudante em uma universidade, é fundamental observar que não existe, em geral, um percurso circular de matérias, ou seja, não há uma sequência que, partindo de uma disciplina específica, retorne a ela após passar por outras. Essa característica fundamental é capturada pelo conceito de aciclicidade em teoria dos grafos.

Definição 2.1.3. (BONDY; MURTY, 1979) Um grafo é denominado *acíclico* quando não contém nenhum ciclo. Um ciclo é um caminho de comprimento positivo que começa e termina no mesmo vértice, com todos os outros vértices intermediários sendo distintos entre si e da origem.

Com isso é definido a forma de representação de um currículo acadêmico como, pelo menos, um Grafo Acíclico Direcionado (DAG). Essa abordagem garante que exista uma direção iniciada pelo ingresso de um estudante em sua faculdade seguindo até a sua formatura, mostrando que existe um fluxo unidirecional de dependências entre as matérias, avançando de pré-requisitos para disciplinas mais avançadas, sem retornos.

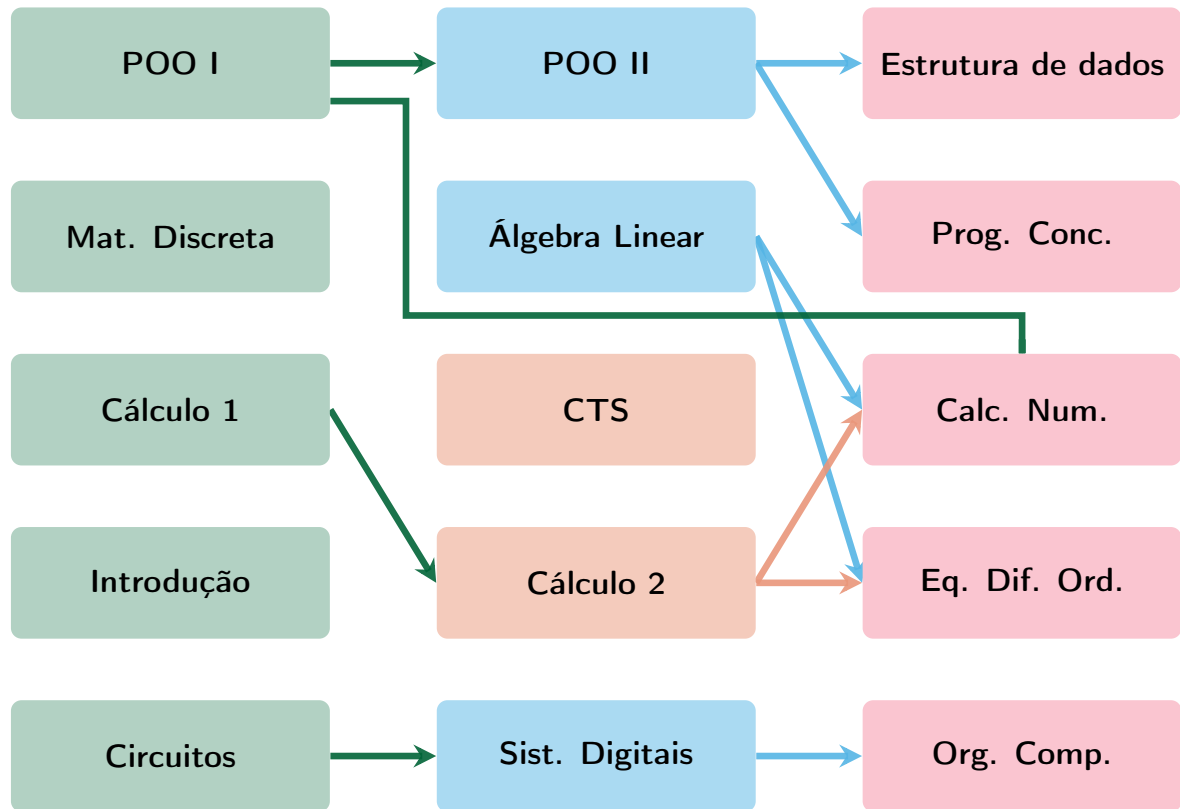
2.1.2 Ordenação de grafos

Uma vez que um currículo é modelado como um Grafo Acíclico Direcionado (DAG), um problema fundamental é determinar uma sequência válida para cursar as disciplinas. Uma solução para este problema é a Ordenação Topológica, um processo que organiza os vértices de um DAG em uma sequência linear que respeita todas as dependências. A exigência de que o grafo seja acíclico é fundamental, pois a presença de um ciclo tornaria impossível satisfazer a relação de precedência (por exemplo, um ciclo $u \rightarrow v \rightarrow u$ implicaria que u deve vir antes de v , e v deve vir antes de u).

Definição 2.1.4. (SANTIAGO, 2023) Uma *ordenação topológica* de um DAG $G = (V, A)$ é uma sequência linear de todos os seus vértices tal que, para cada arco dirigido $(u, v) \in A$, o vértice u aparece antes do vértice v na sequência.

A exigência de que o grafo seja acíclico é fundamental, pois a presença de um ciclo tornaria impossível satisfazer a relação de precedência (por exemplo, um ciclo $u \rightarrow v \rightarrow u$ implicaria que u deve vir antes de v , e v deve vir antes de u).

Figura 1 – Diagrama de fluxo das disciplinas e seus pré-requisitos com setas indicando dependência.



Exemplo 2.1.1. Na Figura 1, do currículo do curso de Ciência da Computação (UFSC/INE, 2025), para a identificação do estado de cada disciplina, adota-se uma legenda de cores. A cor verde é atribuída às disciplinas já concluídas, a cor azul refere-se às disciplinas que estão sendo cursadas no semestre vigente, enquanto a cor laranja indica disciplinas do semestre que não foram escolhidas, e, a rosa indica as disciplinas que o discente planeja cursar no semestre subsequente.

É possível notar questões como a não-realização da disciplina *Cálculo 2* resulta na não-existência de um arco válido para as disciplinas de *Cálculo Numérico* e *Equações Diferenciais Ordinárias*. Mas já a não-realização da matéria de *Ciência, Tecnologia e Sociedade* não impede que existe uma sequência válida de uma disciplina da primeira fase para alguma da terceira fase. O problema de encontrar uma sequência válida para cursar todas as disciplinas é, portanto, o problema de encontrar uma ordenação topológica para esse grafo curricular.

O resultado do algoritmo fornecerá uma ou mais sequências possíveis de disciplinas que um aluno pode seguir para completar o curso sem violar nenhum pré-requisito. Este problema é análogo a Técnica de Avaliação e Revisão de Programas (PERT) e ao sequenciamento de

tarefas(SANTIAGO, 2023), onde certas atividades devem ser concluídas antes que outras possam começar. Uma das aplicações mais intuitivas da ordenação topológica é no planejamento de uma grade curricular de um curso. Nesse cenário, as disciplinas são representadas como vértices e os pré-requisitos como arcos dirigidos.

2.1.3 Coloração de grafos e Gestão de Conflitos de Horário

A coloração de grafos é uma ferramenta versátil para agrupar e particionar elementos em um sistema complexo. No contexto acadêmico, ela pode ser aplicada a dois problemas distintos que utilizam grafos fundamentalmente diferentes: a visualização do currículo e a resolução de conflitos de horário. Para evitar ambiguidades com outras técnicas na literatura, é importante especificar que a abordagem utilizada é a *coloração própria de vértices*.

Definição 2.1.5. (DIESTEL, 2025) A *coloração própria* de um *vértex* de um grafo $G = (V, E)$ é um mapeamento de $c : V \rightarrow S$ de forma que $c(v) \neq c(w)$ quando v e w sejam adjacentes. Os elementos do conjunto S são chamados de *cores* disponíveis.

Enquanto a coloração do DAG curricular pode ser usada para fins de visualização (por exemplo, agrupar disciplinas por semestre ideal ou área de conhecimento), um desafio mais complexo é a gestão de horários. A montagem de uma grade semestral viável não depende do grafo de pré-requisitos, mas de um *grafo de conflito* (KELL, 2010).

Este segundo grafo, $G_{conflito} = (V', E')$, é construído de forma que cada turma ofertada em um semestre é representada por um vértice $v' \in V'$. Uma aresta $(u', v') \in E'$ é então adicionada se, e somente se, as turmas representadas por u' e v' possuem um conflito de horário, ou seja, são ofertadas em horários que se sobrepõem.

O problema de encontrar uma grade de horários sem choques é, então, reduzido ao problema de encontrar uma coloração própria para o grafo de conflito. Nesse modelo, cada cor representa um slot de tempo distinto. Uma coloração válida garante que quaisquer duas turmas com a mesma cor não são adjacentes, significando que não têm conflito e podem ocorrer no mesmo horário. O número mínimo de cores necessárias, o número cromático $\chi(G_{conflito})$, corresponde ao número mínimo de slots de tempo necessários para acomodar todas as turmas sem sobreposição (DIESTEL, 2025). É crucial destacar que a análise de planejamento acadêmico envolve, no mínimo, dois grafos distintos: o DAG curricular para sequenciamento de pré-requisitos e o grafo de conflito para alocação de horários.

2.1.4 Método do Caminho Crítico

É possível encontrar o caminho de custo máximo em um grafo, o que é um problema NP-difícil. No entanto, ao trabalhar com um Grafo Acíclico Direcionado (DAG) e utilizando o conceito de ordenação topológica, conforme a Definição 2.1.4, pode-se aplicar o Método do Caminho Crítico (KELLEY JR.; WALKER, 1959).

Esse método oferece uma estrutura rigorosa que ajuda a modelar uma situação de projetos, onde é separado o *planejamento* (sequência lógica das atividades) da *programação* (alocação de tempo e recursos). Divide-se a fase de planejamento do projeto nas seguintes etapas: a definição das atividades, seus tempos e recursos; a construção do grafo representando as atividades, seus tempos de execução e as interdependências entre as tarefas; o cálculo do caminho crítico; e a programação temporal das tarefas.

Cada atividade desse projeto será então um arco no grafo, enquanto os vértices serão usados para estabelecer precedências entre as atividades. Há duas regras para a construção do grafo. Primeiro, cada atividade é representada por apenas um arco. Segundo, para manter a corretude das relações de precedência, precisa-se responder às seguintes perguntas:

- i. Quais atividades devem preceder imediatamente a atividade atual;
- ii. Quais atividades devem vir depois da atividade atual;
- iii. Quais atividades devem ocorrer concorrentemente da atividade atual;

É possível, portanto, realizar cálculos para determinar a duração total do projeto e classificar as atividades em críticas e não-críticas (TAHA, 2006). Sendo uma atividade classificada como crítica quando sua execução é restrita a um intervalo de tempo predefinido, cujo início e fim são fixos e equivalem à sua duração. Um *evento* é um ponto no tempo que marca a conclusão de certas tarefas e o início de outras (SANTIAGO, 2023). E_j é o tempo mais cedo da ocorrência de um evento j , T_j é o tempo mais tarde da ocorrência de um evento j , e D_{ij} é a duração da atividade representada pelo arco (i, j) (TAHA, 2006).

Para calcular o caminho crítico são necessários dois passos: *forward-pass*, que define E_j e o *backward-pass* que define T_j .

Definição 2.1.6. *Forward-pass* é definido como:

<pre> 1: Inicializa $E_1 \leftarrow 0$ 2: for $E_j \in V_j$ do 3: $E_j \leftarrow \max_{v \in N^-(j)} \{T_v - D_{jv}\}$ 4: end for </pre>	$\triangleright$ indicando que o evento inicial é o ponto de partida
--	--

Após a conclusão do *forward-pass*, é iniciado o *backward-pass* em cima do vértice n , representando todas as atividades já concluídas.

Definição 2.1.7. *Backward-pass* é definido como:

<pre> 1: Inicializa $T_n \leftarrow E_n$ 2: for E_j until $E_1 = T_1 = 0$ do 3: $E_j \leftarrow \min_{v \in N^+(j)} \{T_v - D_{jv}\}$ 4: end for </pre>	$\triangleright$ indicando que o evento inicial é o ponto de partida
--	--

A dessas ideias e definições é possível auxiliar em uma visão clara da estrutura de dependências de um curso e adaptar de forma que forneça uma ferramenta quantitativa para o progresso de um estudante, com foco em evitar atrasos e dificuldades maiores que o comum durante o desenvolvimento acadêmico.

Problema da mochila

A seleção de disciplinas para um semestre, sob a restrição de uma carga horária máxima, pode ser modelada como o Problema da Mochila (*Knapsack Problem*). Este é um problema clássico de otimização combinatória, cujo nome deriva da situação análoga ao preenchimento de uma mochila com um conjunto de itens, cada um com peso e valor distintos. O objetivo é escolher um subconjunto de itens que maximize o valor total, sem que a soma de seus pesos ultrapasse a capacidade máxima da mochila (PISINGER; TOTH, 1998).

Existem diversas variantes deste problema, mas a mais diretamente aplicável ao planejamento de matrículas, onde cada disciplina é um item único que pode ser escolhido ou não, é a variante *Problema da Mochila 0/1* (0/1 Knapsack Problem).

Definição 2.1.8. (PISINGER; TOTH, 1998) Dados n itens, cada um com um peso w_i e um valor v_i , e uma mochila com capacidade máxima C , o objetivo é selecionar um subconjunto de itens para maximizar o valor total $\sum_{i=1}^n v_i x_i$, sujeito à restrição de que o peso total $\sum_{i=1}^n w_i x_i$ não exceda a capacidade C , onde $x_i \in \{0, 1\}$ indica se o item i é incluído ou não.

A aplicação deste modelo ao planejamento acadêmico traduz o desafio da matrícula em um problema de otimização estruturado. A *mochila* simboliza o semestre letivo, enquanto sua *capacidade* é o limite máximo de créditos permitido. As disciplinas disponíveis para matrícula são os *itens* a serem escolhidos, cada uma com um peso correspondente à sua carga horária. O *valor* de cada disciplina é uma métrica de utilidade estratégica, definida de acordo com os objetivos do estudante. A aplicação detalhada deste modelo é explorada na Seção 2.2.2.

2.2 GRAFOS CURRICULARES

A aplicação dos conceitos de teoria dos grafos, apresentados na Seção 2.1, ao domínio do planejamento acadêmico permite uma abordagem quantitativa para a análise de estruturas curriculares. Nesse contexto, o *Grafo Curricular* é formalizado como a estrutura de dados para a modelagem de um currículo. A partir dessa formalização, estabelecem-se as conexões entre algoritmos de grafos e problemas de otimização com as tarefas de sequenciamento de disciplinas e de elaboração de grades semestrais. Finalmente, introduz-se o campo da Análise Curricular, que constitui o arcabouço teórico para a quantificação da complexidade estrutural dos currículos, visando informar a tomada de decisão acadêmica.

2.2.1 Formalização do Grafo Curricular como um Grafo Acíclico Direcionado

A estrutura de um currículo acadêmico, com suas intrincadas dependências e sequências obrigatórias, encontra na teoria dos grafos uma representação matemática natural e poderosa. O *Grafo Curricular*, também conhecido na literatura como Rede de Pré-requisitos de Cursos (Course-Prerequisite Network - CPN) (STAVRINIDES; ZUEV, 2023), é formalmente definido como uma aplicação direta do conceito de Grafo Acíclico Direcionado (DAG), Seção 2.2.

Definição 2.2.1. (ZUEV; STAVRINIDES, 2025), (MACNAIR; OUTROS, 2020) Na modelagem, o grafo $G = (V, A)$ é constituído por dois conjuntos:

1. V é o conjunto de vértices(ou nós), onde cada vértice $v \in V$ representa uma disciplina única do currículo acadêmico.
2. A é o conjunto de arcos (ou arestas direcionadas), onde um arco $(u, v) \in A$ representa uma relação de pré-requisito, indicando que a disciplina u deve ser concluída com sucesso antes que o estudante esteja apto a cursar a disciplina v .

A propriedade “acíclica” é uma condição inerente e fundamental para a validade de um grafo curricular. A existência de um ciclo, como por exemplo uma sequência de dependências $u \rightarrow v \rightarrow \dots \rightarrow u$, implicaria em uma contradição lógica: uma disciplina seria, direta ou indiretamente, um pré-requisito para si mesma. Tal estrutura circular tornaria impossível a conclusão do currículo, pois não haveria um ponto de partida válido para a sequência de disciplinas envolvidas no ciclo. Portanto, a aciclicidade do grafo não é apenas uma conveniência de modelagem, mas uma restrição fundamental que valida a própria coerência lógica de um plano de curso.

Essa definição estabelece o grafo curricular como um modelo estrutural do currículo acadêmico que representa o fluxo de conhecimento (ZUEV; STAVRINIDES, 2025). Cada caminho possível do grafo, de um vértice sem arcos de entrada (uma disciplina inicial) a um vértice sem arcos de saída (uma disciplina terminal), representa uma sequência potencial de aprendizado que um estudante pode seguir.

2.2.2 Aplicações ao Planejamento Acadêmico

A representação de um currículo como um grafo acíclico direcionado permite a aplicação direta de um vasto repertório de algoritmos e modelos teóricos da ciência da computação para resolver problemas fundamentais do planejamento acadêmico. A determinação de uma sequência válida de disciplinas para a graduação e a otimização da seleção de disciplinas para um semestre são duas tarefas centrais para qualquer estudante que podem ser formalmente mapeadas para problemas clássicos da teoria dos grafos e da otimização combinatória.

Sequenciamento de Disciplinas e Ordenação Topológica

O desafio de determinar uma sequência válida de disciplinas que um estudante pode seguir para completar seu curso, respeitando todas as regras de pré-requisito, é formalmente análogo ao problema de encontrar uma ordenação topológica para o grafo curricular correspondente. Conforme introduzido na subseção Subseção 2.1.2, uma ordenação topológica de um DAG é uma sequência linear de todos os seus vértices, tal que para cada arco direcionado de um vértice u para um vértice v , u aparece antes de v na sequência (MEINKE; BAUER, 1976).

No contexto acadêmico, cada ordenação topológica do grafo curricular representa uma trajetória de graduação válida, ou seja, uma lista sequencial de todas as disciplinas do curso que um estudante pode seguir sem violar dependências (MEINKE; BAUER, 1976). É importante notar que, para a maioria dos currículos, não existe uma única ordenação topológica. A existência de múltiplas ordenações topológicas distintas não representa uma ambiguidade do modelo, mas sim a representação formal da flexibilidade e das escolhas disponíveis para o estudante.

Durante sua jornada, o estudante pode encontrar com várias disciplinas cujos pré-requisitos já foram satisfeitos, podendo escolher a ordem que irá cursá-las. O conjunto de todas as ordenações topológicas possíveis define o espaço completo de todas as trajetórias válidas para conclusão do curso. Assim, o desafio do planejamento acadêmico pode ser visto não apenas como a busca por uma trajetória válida, mas como a navegação neste espaço de possibilidades para encontrar uma trajetória que seja ótima segundo critérios individuais (AJANOVSKI, 2019).

Otimização da Seleção de Disciplinas para um Semestre

A tarefa de um estudante de selecionar um conjunto de disciplinas para cursar em um semestre, a partir de um conjunto de opções cujos pré-requisitos já foram cumpridos, pode ser modelada como uma variante do Problema da Mochila (Seção 2.1.4).

Definição 2.2.2 (0/1 Knapsack Problem). (ATASHBAR; RAHIMI, 2011) O **Problema da Mochila 0/1** é um problema de otimização combinatória. Dado um conjunto de n itens, onde cada item i (para $i = 1, \dots, n$) possui um peso $w_i > 0$ e um valor $v_i > 0$, e uma mochila com capacidade máxima W , o objetivo é determinar quais itens incluir na mochila.

Pode-se representar a escolha de cada item por uma variável de decisão binária $x_i \in \{0, 1\}$, onde $x_i = 1$ se o item i for selecionado e $x_i = 0$ caso contrário. Com o objetivo de:

$$\text{maximizar } Z = \sum_{i=1}^n v_i x_i$$

Sujeito à restrição de capacidade:

$$\sum_{i=1}^n w_i x_i \leq W$$

Ao aplicar este modelo à seleção de disciplinas, estabelece-se a seguinte analogia:

1. **Mochila:** O semestre acadêmico a ser planejado.
2. **Capacidade da Mochila (W):** A carga horária máxima permitida para o semestre (ou limite de créditos).
3. **Itens:** O conjunto de todas as disciplinas para as quais o estudante está apto a se matricular.
4. **Peso de cada item (w_i):** A carga horária (ou número de créditos) da disciplina i .
5. **Valor de cada item (v_i):** Uma métrica de utilidade ou benefício associada à disciplina i (por exemplo, prioridade acadêmica, interesse do aluno, ou um valor derivado da sua posição no grafo de pré-requisitos).

O objetivo é selecionar um subconjunto de disciplinas (itens) cuja soma das cargas horárias (pesos) não exceda a capacidade do semestre, de forma a maximizar a soma dos seus valores (MUSTAPA; SALAHUDDIN; HASHIM, 2022). A definição do “valor” de uma disciplina é o aspecto mais complexo e subjetivo desta modelagem. Este valor não é uma propriedade intrínseca da disciplina, mas uma função das metas e prioridades do estudante. Pode representar o progresso no curso (o número de novas disciplinas que serão desbloqueadas após sua conclusão), o interesse pessoal do aluno, a relevância para objetivos de carreira, a taxa histórica de aprovação, ou uma combinação ponderada de múltiplos fatores. Isso transforma o problema de uma otimização de critério único para um problema de decisão multicritério, onde a função de “valor” precisa ser personalizada para refletir as preferências individuais.

Encontrar a solução ótima exata pode ser computacionalmente custoso se o número de disciplinas elegíveis for grande. Nesse cenário, Algoritmos Gulosos (*Greedy Algorithms*) oferecem uma abordagem heurística eficiente para encontrar uma solução de alta qualidade em tempo hábil (XIAO, 2024). Uma estratégia gulosa comum consiste em calcular uma razão de eficiência para cada item (disciplina), como uma razão de “valor” por horas aula necessárias. As disciplinas são então ordenadas de forma decrescente por essa razão, e o algoritmo seleciona iterativamente as disciplinas do topo da lista, desde que elas ainda “caibam” na mochila (não excedam a carga horária restante), até que nenhuma outra disciplina possa ser adicionada. Embora essa abordagem não garanta a solução globalmente ótima, ela pode eventualmente produzir um planejamento semestral satisfatório.

2.2.3 Análise Curricular aplicada a Currículos

A modelagem de currículos como grafos é a base para um campo de estudo formal conhecido como Análise Curricular (*Curricular Analytics*) (HEILEMAN et al., 2018). Este campo, desenvolvido principalmente por Heileman, Abdallah e seus colaboradores, estabelece um arcabouço para quantificar a complexidade de currículos e relacionar essa complexidade aos resultados de sucesso do estudante (KRASNOV, 2025). A Análise Curricular utiliza a teoria

dos grafos e análise de redes para transformar a estrutura de pré-requisitos de um programa acadêmico em um objeto mensurável, permitindo uma avaliação objetiva e baseada em dados.

A hipótese central que norteia este campo é a existência de uma correlação inversa entre a complexidade estrutural de um currículo e o sucesso dos estudantes. Currículos com estruturas mais simples, com menos dependências e maior flexibilidade, tendem a apresentar melhores taxas de retenção e graduação em tempo hábil (CAROLINA, 2025). É crucial distinguir entre a complexidade estrutural, que se refere à organização dos cursos e suas interdependências (o foco da Análise Curricular), e a complexidade instrucional, que se refere à dificuldade do conteúdo ou à qualidade do ensino, sendo esta última mais qualitativa e difícil de mensurar objetivamente (HANSEN; NEMETH; STEWART, 2024). Ao focar na estrutura, a Análise Curricular fornece métricas acionáveis que podem guiar reformas curriculares com o objetivo de remover barreiras desnecessárias à progressão do estudante.

2.3 DIPLOMA DIGITAL

A transformação digital no ensino superior brasileiro tem sido impulsionada por diversas iniciativas, dentre as quais se destaca a instituição do Diploma Digital. Contudo, analisar esta inovação meramente como a substituição de um documento físico por um equivalente eletrônico seria subestimar seu alcance e impacto. O Diploma Digital é, na verdade, o elemento mais visível de uma estratégia mais ampla e profunda do Ministério da Educação (MEC) para estabelecer um ecossistema nacional de dados acadêmicos padronizados e interoperáveis (MEC, 2021). Esta seção se propõe a analisar criticamente esta iniciativa, desdobrando-a em três eixos: a definição do Diploma Digital e as tecnologias que garantem sua validade e segurança; a padronização de dados como pilar para a interoperabilidade, e, finalmente, a sinergia direta entre este novo ecossistema de dados e os objetivos deste trabalho.

2.3.1 Definição, Validade Jurídica e Segurança Tecnológica

O Diploma Digital é formalmente definido como um documento nato-digital, ou seja, um artefato cuja geração, emissão, armazenamento e validação ocorrem integralmente em meio eletrônico. Esta característica o distingue fundamentalmente de uma simples digitalização de um documento em papel, pois sua essência não é uma imagem, mas um conjunto de dados estruturados cuja validade jurídica é equivalente ao documento físico (MEC, 2021). A base legal para esta equiparação e para a própria existência do Diploma Digital foi estabelecida por um conjunto de normativas do MEC, com destaque para a Portaria nº 554 (MEC, 2019), que dispõe especificamente sobre a emissão e o registro de diplomas de graduação por meio digital pelas Instituições de Ensino Superior (IES) pertencentes ao Sistema Federal de Ensino.

A confiabilidade deste documento, crucial para sua aceitação e para a viabilidade de sistemas que o consomem, é assegurada por uma robusta cadeia de confiança que se inicia no arcabouço legal e se concretiza em tecnologias criptográficas consolidadas. O primeiro elo

dessa cadeia é a própria legislação, que não apenas institui o documento, porém exige também a implementação de uma estrutura tecnológica específica e rigorosa. O segundo elo é a exigência explícita do uso da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil) para a assinatura digital de todos os documentos (ITI, 2025).

A ICP-Brasil é o padrão nacional para certificação digital, garantindo que as assinaturas eletrônicas no diploma sejam autenticadas por uma hierarquia de confiança auditada e regulamentada pelo governo. A regra exige que os representantes legais das instituições de ensino (emissora e registradora) usem certificados digitais A3 ou superiores. Esses certificados são armazenados em dispositivos físicos (como *tokens* ou cartões inteligentes), garantindo alta segurança contra fraudes e protegendo a chave privada (MEC, 2019). O terceiro e último elo da cadeia de confiança é a aplicação de primitivas criptográficas para garantir as propriedades essenciais de um documento digital: *autenticidade, integridade e irrefutabilidade*.

A autenticidade é garantida pela assinatura digital ICP-Brasil, que vincula inequivocamente as identidades dos signatários ao documento. A integridade é assegurada pelo fato de que qualquer alteração no conteúdo do documento após a assinatura a invalidaria. Para complementar, a legislação exige o uso do carimbo de tempo (*timestamp*), também no padrão ICP-Brasil. O carimbo de tempo é um selo eletrônico que atesta a data e a hora exatas em que o documento foi assinado, provendo uma evidência temporal legalmente válida e impedindo que a validade de uma assinatura seja contestada no futuro, mesmo que o certificado do signatário venha a expirar.

Algumas soluções tecnológicas, como a desenvolvida pela Rede Nacional de Ensino e Pesquisa (RNP), adicionam ainda uma camada de segurança e transparência por meio do registro em *blockchain*, que cria um histórico distribuído e imutável das emissões (MEC, 2021).

2.3.2 A Padronização de Dados como Pilar da Interoperabilidade Acadêmica

Embora a segurança criptográfica seja fundamental, o aspecto mais relevante da iniciativa para sistemas como o descrito é a padronização da estrutura da informação acadêmica. Essa iniciativa estabelece um ecossistema de registros acadêmicos digitais inter-relacionados, todos seguindo um mesmo padrão. Dentre os documentos que compõem este ecossistema, dois são de importância capital para este projeto: o *Histórico Escolar Digital* e o *Currículo Escolar Digital*.

A existência desses documentos padronizados representa uma solução para um desafio dos sistemas de informação educacionais no Brasil: a falta de interoperabilidade. Tradicionalmente, cada IES mantinha os dados acadêmicos de seus alunos em sistemas de gestão próprios, com modelos de dados e formatos de arquivo proprietários, criando silos de informação que impediam a troca de dados e a realização de análises em larga escala (MEC, 2021).

A base tecnológica para esta padronização é a adoção do XML (*Extensible Markup Language*) como formato para a representação dos dados, e do XSD (*XML Schema Definition*) como mecanismo para a validação de sua estrutura. O XML permite que os dados sejam armazenados de forma hierárquica e semanticamente descrita.

Figura 2 – Exemplo de XML do Histórico Escolar Digital.

```

1 <HistoricoEscolar>
2   <Aluno>...</Aluno>
3   <ComponenteCurricular>
4     <Codigo>MTM3110</Codigo>
5     <Nome>Calculo 1</Nome>
6     <Status>Aprovado</Status>
7   </ComponenteCurricular>
8   <ComponenteCurricular>
9     <Codigo>MTM3120</Codigo>
10    <Nome>Calculo 2</Nome>
11    <PreRequisitos>
12      <Componente>MTM3110</Componente>
13    </PreRequisitos>
14    <Status>Pendente</Status>
15  </ComponenteCurricular>
16 </HistoricoEscolar>

```

Fonte: O autor.

Mais importante, o MEC disponibiliza um conjunto de arquivos XSD que funcionam como um contrato formal e rigoroso, definindo quais elementos devem compor cada documento, seus tipos de dados, sua cardinalidade e suas inter-relações. Esquemas como o *Leiaute de Histórico Escolar* e o *Leiaute de Currículo Escolar* garantem a conformidade sintática e estrutural dos arquivos XML gerados por qualquer IES do país (MEC, 2025). Essa uniformidade permite que sistemas de diferentes instituições possam intercambiar e processar informações acadêmicas de forma automatizada e confiável, sem a necessidade de adaptadores para cada fonte de dados.

2.3.3 Impacto e Sinergia com a Geração Automatizada de Grafos Curriculares

Primeiramente, a conformidade obrigatória com um esquema XSD permite o desenvolvimento a partir de um conjunto de dados determinístico e robusto. Esta padronização de software pode, por exemplo, facilitar o processamento de forma automática e confiável o arquivo XML do histórico escolar de um aluno para popular sua representação individual em um grafo curricular, marcando as disciplinas já cursadas, as notas obtidas e as que estão disponíveis para serem cursadas. Este processo automatizado substitui a necessidade de entrada manual de dados, que é lenta e suscetível a erros, ou o desenvolvimento de complexos e frágeis *web scrapers* para extrair informações de portais acadêmicos com estruturas proprietárias. Portanto, é possível garantir a fidedignidade dos dados de entrada, que é um requisito fundamental para a precisão do grafo e das sugestões geradas pelo sistema.

Em segundo lugar, e de forma mais estratégica, a adoção massiva deste padrão em todo o território nacional resultará na geração de um volume sem precedentes de registros acadêmicos em formato uniforme. Quando devidamente anonimizados e agregados, esses

milhões de arquivos XML se convertem em um repositório de dados de valor inestimável para a mineração de dados educacionais e a análise de trajetórias acadêmicas.

Ao analisar este grande volume de dados longitudinais, como discutido na Seção 2.2, o sistema pode além de fazer a simples verificação de pré-requisitos e oferecer também outras análises, como por exemplo, identificar gargalos curriculares, que são disciplinas com altas e recorrentes taxas de reprovação que obstruem o fluxo dos estudantes; descobrir trajetórias de sucesso, mapeando as sequências de disciplinas mais comuns entre os alunos que concluem o curso com bom desempenho e no tempo previsto, consequentemente aprimorando a qualidade das recomendações de matrículas.

A capacidade de análise de dados históricos introduz uma abordagem complementar ao planejamento e aconselhamento acadêmico. Os sistemas que operam exclusivamente com base nas regras formais do currículo oficial seguem um modelo prescritivo, que informa ao aluno quais disciplinas ele pode ou não cursar. O ecossistema de dados criado pela iniciativa do Diploma Digital possibilita a transição para um modelo descritivo e preditivo, fundamentado no comportamento coletivo de um grande volume de estudantes. A análise dos dados agregados descreve as trajetórias reais dos alunos ao longo do currículo, permitindo a identificação de padrões de sucesso e dificuldade.

Com base nisso, o sistema pode gerar previsões e oferecer um aconselhamento contextualizado, evoluindo de uma afirmação como “Você está apto a cursar a disciplina X” para uma recomendação como “Com base na análise de trajetórias de estudantes com perfil similar, cursar a disciplina X neste semestre está associado a uma maior probabilidade de sucesso, enquanto a disciplina Y, embora permitida, apresenta um histórico de desafios para alunos neste ponto do curso”. A transição de um planejamento baseado em regras para um planejamento orientado por dados torna-se uma realidade prática, tendo na padronização de dados acadêmicos promovida pelo MEC a sua principal tecnologia habilitadora.

3 DESENVOLVIMENTO DA SOLUÇÃO

Este capítulo detalha a arquitetura conceitual e o desenvolvimento da ferramenta de planejamento curricular, desde a modelagem dos dados institucionais até a implementação do algoritmo de recomendação. O foco reside na justificativa das decisões de projeto e na aplicação dos fundamentos teóricos para resolver os desafios práticos da demonstração de currículos, visualização de progresso e planejamento curricular.

3.1 MODELAGEM DO CURRÍCULO ACADÊMICO

3.1.1 Fundamentação da Abordagem em Grafo

A escolha de um modelo de dados em grafo se deu pela sua capacidade de representar nativamente entidades e as relações entre elas. Conforme discutido anteriormente, um currículo acadêmico pode ser concebido como uma rede onde as disciplinas são os nós e as dependências (como pré-requisitos) são as arestas que definem o fluxo de progressão possível para um estudante, como na Definição 2.2.1. Um banco de dados orientado a grafos armazena essa abstração de forma isomórfica, tratando tanto os vértices quanto as arestas como entidades de primeira classe. Essa correspondência direta elimina a complexidade translacional requerida por outros paradigmas de armazenamento.

Em uma abordagem com modelo relacional, a representação de tais conexões ocorreria de forma indireta. Seria necessária uma tabela para as disciplinas e uma tabela de junção adicional para mapear cada relação de pré-requisito, associando pares de identificadores de disciplinas. A principal limitação desta abordagem é exposta ao se realizar consultas que necessitam percorrer múltiplos níveis de relacionamento.

Para cada nível de profundidade na cadeia de pré-requisitos, a consulta exigiria uma operação de junção adicional. O custo computacional de tais operações tende a crescer de forma significativa com a complexidade da consulta, uma vez que o sistema de gerenciamento de banco de dados pode precisar processar grandes conjuntos de dados intermediários. Embora construções recursivas possam formular essas buscas, por exemplo, a tarefa de identificar o *caminho crítico* de um curso, determinar a sequência mais longa de dependências exige uma análise completa da conectividade do grafo, para qual este modelo de dados não é otimizado.

Alternativamente, um modelo de dados orientado a documentos representaria cada disciplina como uma estrutura autocontida, na qual as referências aos pré-requisitos seriam incorporadas, por exemplo, como uma lista de identificadores. Tal estrutura é performática para a recuperação de informações de uma única disciplina. Contudo, para realizar consultas que exploram os relacionamentos, como determinar todas as disciplinas com um determinado pré-requisito, seria necessário percorrer múltiplos documentos, passando a complexidade da junção de dados do banco para a camada de aplicação. Nesse paradigma, os relacionamentos não são entidades primárias, o que torna a análise da rede uma tarefa computacionalmente custosa.

A adoção de um banco de dados de grafos, portanto, não se justifica apenas pela eficiência na representação e consulta de dados conectados. De forma mais estratégica, essa escolha habilita a aplicação direta do arcabouço teórico apresentado no Capítulo 2. Algoritmos como a Ordenação Topológica, definido na Subseção 2.1.2, essencial para determinar sequências de graduação válidas, e o Método do Caminho Crítico, definido na Subseção 2.1.4, utilizado para identificar disciplinas cruciais ao avanço do estudante, dependem de operações de travessia de grafo. A capacidade de executar essas análises de forma nativa e eficiente no próprio banco de dados é uma consequência direta da escolha deste paradigma, alinhando a implementação tecnológica com os objetivos analíticos do trabalho.

3.1.2 Modelagem do Currículo para um Grafo Direcionado

A dependência entre as disciplinas foi modelada utilizando grafos direcionados (Definição 2.1.3). Por exemplo, tendo uma aresta $(A) \rightarrow (B)$ com o rótulo como Pré-Requisito indicando que a disciplina A é um pré-requisito para a disciplina B , estabelecendo uma ordem obrigatória de progressão. Essa estrutura transforma a grade curricular em um Grafo Acíclico Dirigido (DAG), conceito central discutido na Seção 2.1 e na Definição 2.1.3.

A persistência desse modelo foi projetada para ser atômica e livre de redundâncias. A decisão conceitual foi garantir que, para cada currículo de um curso específico, existisse um e apenas um grafo correspondente. Ao adicionar ou atualizar informações, o sistema não simplesmente insere novos dados, mas verifica a existência de cada nó e aresta antes de criá-los. Essa abordagem idempotente assegura a integridade e a consistência do grafo ao longo do tempo, um requisito essencial para a confiabilidade das consultas e sugestões. A consulta para salvar o grafo, por exemplo, utiliza uma chave composta pelo ID do currículo e pelo código do curso para garantir a unicidade de cada estrutura curricular no banco de dados.

Com o currículo formalizado como um grafo, torna-se possível aplicar algoritmos para quantificar e extrair informações sobre a sua topologia. Análises de conectividade e centralidade, por exemplo, permitem identificar programaticamente as disciplinas que possuem maior impacto no fluxo do curso, seja por servirem como pré-requisito para um grande número de outras matérias (alto grau de saída), ou por funcionarem como pontos de conexão entre diferentes áreas da grade (alta centralidade de intermediação). Essa capacidade de interrogar a estrutura do grafo é o que fundamenta as heurísticas e os algoritmos de recomendação.

3.1.3 O Esquema do Grafo Curricular: Nós e Relações

Para que o modelo de grafo seja computacionalmente interpretável e consistente, é essencial formalizar sua estrutura através de um esquema. Este esquema define as categorias de entidades que existem no domínio e os tipos de conexões que podem existir entre elas. A seguir, são detalhadas as definições dos nós (vértices), que representam as entidades, e das arestas (relações), que descrevem suas interações.

Definição 3.1.1. Os nós (vértices) do grafo, que representam as entidades do domínio, são definidos com os seguintes tipos:

1. **Disciplina:** Uma disciplina com seus atributos, como código, nome e carga horária.
2. **Currículo:** Estrutura de uma grade curricular de um curso em um determinado período.
3. **Turma:** Representa a oferta de uma disciplina em um semestre letivo, com informações como professor, horários e vagas.

Os nós definidos acima representam as entidades fundamentais do ecossistema. No entanto, isoladamente, eles são apenas repositórios de atributos. A verdadeira expressividade e poder do modelo de grafo emergem das arestas que os conectam. As relações a seguir são as que dão estrutura e significado ao grafo curricular.

Definição 3.1.2. As arestas (relações), que descrevem as interações entre os nós definidos em 3.1.1, são definidas com os seguintes tipos:

1. **Pré-Requisito:** Uma relação de um nó Disciplina para outro, indicando uma relação de pré-requisito. A direção da aresta estabelece a ordem de precedência obrigatória.
2. **Faz parte de:** Conecta um nó Disciplina a um Currículo, estabelecendo uma relação de pertencimento. Assim a mesma disciplina pode fazer parte de múltiplos currículos.
3. **Oferece:** Liga uma Disciplina às suas turmas ofertadas (Turma), conectando a estrutura estática do currículo com a dinâmica da oferta semestral.

Exemplo 3.1.1. Considere um fragmento de dados de um currículo de Ciência da Computação (código ‘208’, ID ‘20071’), cujos dados são inspirados na estrutura do Diploma Digital (Fig. 2). Temos então quatro UnidadeCurricular: MTM3110 - ‘Cálculo 1’, MTM3120 - ‘Cálculo 2’, INE5202 - ‘Cálculo Numérico’, MTM3131 - ‘Equações Diferenciais’.

A estrutura de RequisitoCurricular especificaria que MTM3101 é um PreRequisito para MTM3102, e que MTM3102 é, por sua vez, um PreRequisito para MTM3103 e MTM3104. Adicionalmente, para a oferta do segundo semestre do ano de 2025, existiria uma Turma com CodigoTurma de número 1234 associada à disciplina MTM3102.

Tradução dos Nós (Definição 3.1.1):

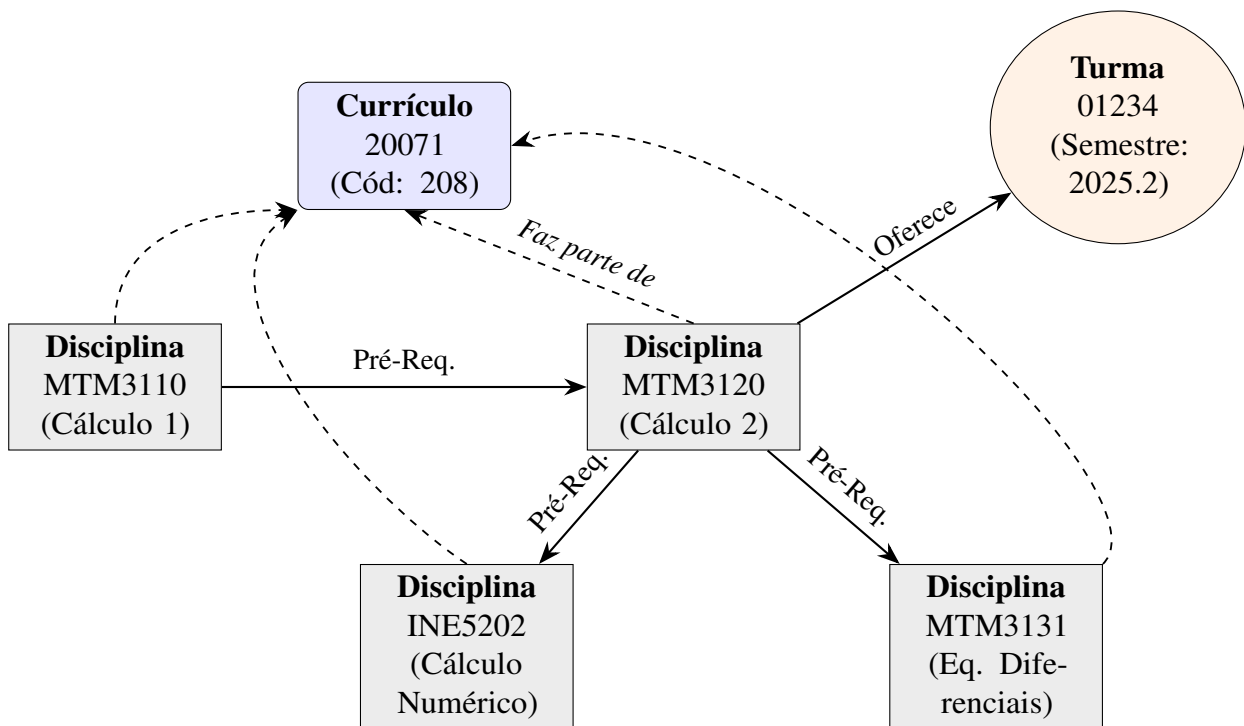
Um nó Currículo seria criado para representar a grade, com propriedades como id: ‘20071’, codigoCurso: ‘208’. Seriam criados quatro nós de Disciplina, correspondentes às seguintes disciplinas: MTM3110 – Cálculo 1, MTM3120 – Cálculo 2, INE5202 – Cálculo Numérico e MTM3131 – Equações Diferenciais. Um nó Turma seria criado para a oferta: codigoTurma: ‘01234’, semestre: ‘2025.2’.

Tradução das Relações (Definição 3.1.2):

Foram estabelecidas diferentes relações entre as entidades, de acordo com sua natureza semântica. Primeiramente, criaram-se quatro relações do tipo *Faz parte de*, responsáveis por associar cada disciplina ao currículo correspondente, conforme segue: MTM3101, MTM3102, MTM3103 e MTM3104 foram todas vinculadas ao Currículo 20071.

Em seguida, definiram-se três relações do tipo *Pré-Requisito* para modelar as dependências entre disciplinas: a disciplina MTM3101 é pré-requisito de MTM3102; a disciplina MTM3102 é pré-requisito tanto de MTM3103 quanto de MTM3104. Por fim, criou-se uma relação do tipo *Oferece*, que conecta a disciplina MTM3102 à sua respectiva oferta semestral, representada pela turma 01234. A Fig. 3 ilustra a representação visual deste micro-grafo no banco de dados.

Figura 3 – Representação gráfica do exemplo de mapeamento do currículo.



Fonte: O autor.

3.2 VISUALIZAÇÃO DO PROGRESSO DO ESTUDANTE

Com o grafo curricular estabelecido, a etapa subsequente é determinar a posição e o percurso já realizado por um estudante específico dentro deste mapa. Esta fase foca na tradução do histórico acadêmico individual em um formato de dados que possa ser diretamente sobreposto ao grafo, contextualizando-o para permitir a geração de recomendações personalizadas.

3.2.1 O Controle Curricular como Fonte de Dados do Aluno

A fonte de dados selecionada para representar a jornada do aluno é o Controle Curricular. Este documento, é a intersecção entre a estrutura curricular oficial do curso e o histórico escolar do estudante. Ele contém o registro de todas as disciplinas e seus respectivos status: cursadas e aprovadas, em andamento, dispensadas por equivalência e pendentes. A escolha deste documento se deve à sua natureza consolidada, servindo como um registro que reflete o conjunto de nós no grafo curricular que o aluno já “visitou” ou está em processo de visitar.

O ponto de partida para a contextualização do progresso é a submissão do Controle Curricular pelo estudante. O sistema foi projetado para que esta interação seja a mais direta possível, tratando o documento como fonte para a análise. Uma decisão de projeto fundamental para a integridade do processo foi a de automatizar a identificação de metadados, como o código do curso e o identificador do currículo, extraíndo-os diretamente do conteúdo do arquivo.

3.2.2 Extração de Dados e Estruturação do Progresso

Uma vez que o arquivo PDF é recebido pelo servidor, ele é submetido a um processo de análise para extrair as informações relevantes. Esta tarefa é delegada a um componente de software especializado, um parser, que é invocado pelo sistema principal. A função deste componente é analisar o texto e a disposição dos elementos no documento para identificar os códigos das disciplinas e as palavras-chave que determinam seus respectivos status.

O desafio desta etapa reside na variabilidade de formatação entre os Controles Curriculares de diferentes cursos. A ausência de um padrão unificado para estes documentos exige que o parser seja capaz de lidar com diferentes layouts. Uma padronização futura, como a adoção do Diploma Digital, Seção 2.3, poderia simplificar este processo, fornecendo dados já estruturados.

O produto final do processo de análise é uma representação computacional e organizada do progresso do aluno. Essa estrutura de dados traduz as informações textuais do documento em um formato que o sistema pode interpretar, criando um inventário claro da jornada acadêmica do estudante. Nele, cada disciplina identificada é classificada e agrupada em categorias distintas que definem seu status, como cursadas, dispensadas ou em andamento e que metadados como curso, currículo e a quantidade de horas aula mínima, máxima e média. Essa categorização é fundamental, pois é a partir desta representação lógica e estruturada que os algoritmos subsequentes poderão compreender o estado do aluno e tomar decisões.

3.2.3 Mapeamento do Progresso no Grafo Curricular

A etapa final desta contextualização é o mapeamento do objeto de progresso, obtido na fase anterior, com o grafo curricular persistido. O sistema utiliza os códigos das disciplinas extraídos do documento para identificar os nós (Disciplinas) correspondentes no grafo. Para os algoritmos de recomendação, as disciplinas nas categorias cursadas, dispensadas e em andamento

são funcionalmente equivalentes, pois todas representam pré-requisitos que estão sendo ou já foram satisfeitos.

Observação 3.2.1. *O subgrafo de progresso do aluno é o conjunto de nós do grafo curricular que representam as disciplinas já cursadas, dispensadas ou em andamento por um estudante. Ele não constitui uma nova estrutura de dados, mas uma projeção do grafo principal que expressa o estado atual do aluno dentro do currículo, servindo como base para a determinação das disciplinas que podem ser cursadas nos próximos semestres.*

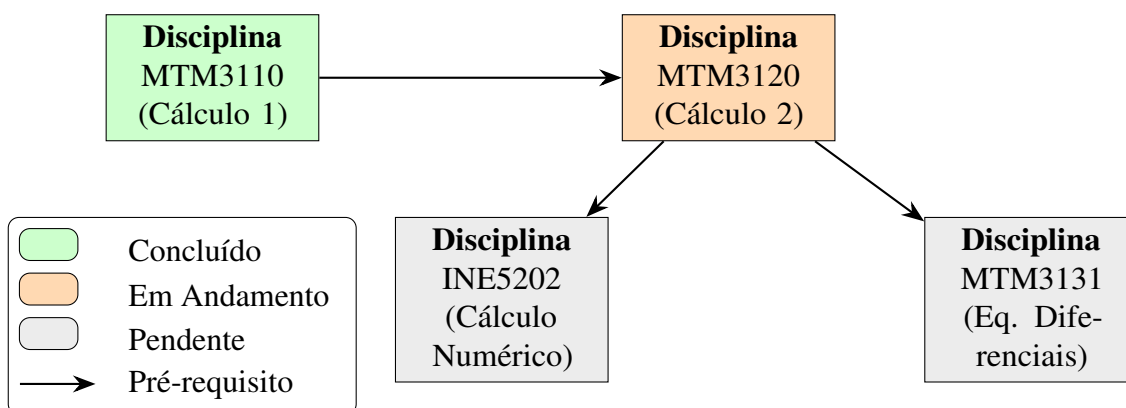
É a partir deste contexto que o sistema realiza as travessias no grafo para determinar os próximos passos possíveis na trajetória acadêmica do aluno, como será detalhado na Seção 3.3.

Exemplo 3.2.1. Continuando o cenário apresentado no Exemplo 3.1.1, suponha que um aluno deste currículo faça o envio do seu Controle Curricular. O documento detalha o progresso do discente, indicando um status de “Aprovado” para a disciplina MTM3110 (Cálculo 1) e “Em andamento” para MTM3120 (Cálculo 2). As disciplinas INE5202 (Cálculo Numérico) e MTM3131 (Eq. Diferenciais), por sua vez, constam ambas com o status “Pendente”.

O processo de análise, descrito na Seção 3.2, interpreta este documento e o traduz em um estado de progresso. Os nós **Disciplina** no grafo são então contextualizados. O nó MTM3110 é marcado como “concluído”, enquanto o MTM3120 é marcado como “em andamento”. Para os algoritmos de recomendação (detalhados na Seção 3.3), ambos os status são tratados como pré-requisitos satisfeitos. Os demais nós permanecem no estado “pendente”.

A Fig. 4 ilustra visualmente este mapeamento de estado sobre o grafo do currículo.

Figura 4 – Representação gráfica do exemplo de mapeamento do currículo.



Fonte: O Autor.

3.3 GERAÇÃO DE SUGESTÕES PARA O SEMESTRE

A fase final da solução consiste na aplicação de uma sequência de algoritmos para sintetizar as informações do modelo de grafo e do progresso do aluno em uma sugestão de matrícula concreta e otimizada. Este processo é executado quando o sistema recebe uma

requisição com o contexto do aluno, o semestre desejado e suas restrições pessoais. A geração é dividida em três etapas computacionais: a filtragem de disciplinas elegíveis, a ponderação de seu valor estratégico e, por fim, a otimização da grade horária.

3.3.1 Identificação de Disciplinas Cursáveis

O primeiro passo é determinar o universo de nós do tipo *Disciplina* que o estudante está apto a cursar. Esta operação é executada através de um algoritmo que utiliza o subgrafo de progresso do aluno, Observação 3.2.1, como seu estado inicial. O sistema inspeciona todos os nós *Disciplina* que possuem uma relação *Faz parte de* com o nó *Currículo* do estudante.

Para cada *Disciplina* candidata, o algoritmo inspeciona todas as suas relações de entrada do tipo *Pré-Requisito*. Uma *Disciplina* é considerada disponível se, e somente se, todos os nós de origem dessas relações (ou seja, todas as suas *Disciplinas* que eram pré-requisitos) já estiverem contidos no conjunto de disciplinas que compõem o subgrafo de progresso do aluno. Essa verificação garante que apenas as *Disciplinas* cujas dependências foram integralmente satisfeitas sejam consideradas para as etapas seguintes. O resultado desta fase é uma lista de todas as *Disciplinas* elegíveis para matrícula.

3.3.2 Heurística para Valorar Disciplinas

Com a lista de *Disciplinas* disponíveis, é necessário um critério para diferenciar sua importância relativa, visto que nem todas contribuem igualmente para o avanço no curso. Para quantificar essa relevância, foi desenvolvida uma heurística própria.

Esta heurística é calculada através de uma nova travessia no grafo, partindo de cada uma das *Disciplinas* disponíveis. Para cada *Disciplina*, o algoritmo explora suas relações de saída do tipo *Pré-Requisito* para determinar quantas outras *Disciplinas* ela desbloqueia. A métrica é então ponderada para dar mais peso aos desbloqueios diretos.

Definição 3.3.1. Um desbloqueio pode ser de duas formas:

1. Diretos: Conta-se o número de *Disciplinas* para as quais a *Disciplina* em análise serve como *Pré-Requisito* direto. Este valor recebe um peso maior na fórmula final.
2. Indiretos: Número de *Disciplinas* que tem como *Pré-Requisito* das *Disciplinas* desbloqueadas diretamente (segundo nível de dependência). Este recebe um peso menor.

A soma ponderada desses dois valores resulta em uma pontuação numérica que representa o impacto de cada *Disciplina*. Sendo assim, *Disciplinas* que atuam como “chaves” para múltiplos ramos do *Currículo* recebem uma pontuação maior, enquanto *Disciplinas* terminais, que não são *Pré-Requisito* para nenhuma outra, recebem pontuação zero.

3.3.3 Otimização da Grade Semestral

A etapa final é a montagem de uma grade horária que utilize os nós Turma ofertados para o semestre, respeitando as restrições do aluno e maximizando o valor estratégico. Este problema foi modelado como uma otimização combinatória, análoga ao Problema da Mochila 0/1. Dado a Definição 2.1.8, o objetivo é selecionar um conjunto de Turmas que maximize a soma dos “valores definidos” para cada Disciplina, sem que a soma de suas cargas horárias exceda a capacidade da mochila e sem que haja conflitos de horário.

A definição do valor de cada item (v_i) é um componente crucial, pois ele combina a importância estratégica da Disciplina com a conveniência para o aluno. O valor base é o “valor definido” da Disciplina associada, que é multiplicado por um fator de ponderação para priorizar a estratégia de progressão. A este valor é somado um “pontuação de compatibilidade”.

Essa pontuação de compatibilidade aplica penalidades numéricas para refinar a seleção. Pontos são subtraídos do valor total da Turma caso seus horários de aula coincidam com os dias da semana que o estudante especificou como indesejados. Seguindo a mesma lógica, uma penalidade distinta também é aplicada se o turno em que a Turma é ofertada (como matutino, vespertino ou noturno) for um dos períodos que o estudante explicitamente indicou que deseja evitar. O resultado é um valor final ajustado para cada Turma, refletindo tanto sua importância estratégica quanto sua conveniência pessoal, e é este valor que o algoritmo tentará maximizar.

Para resolver este problema de otimização, aplica-se um algoritmo de solução para o Problema da Mochila 0/1, frequentemente implementado por meio de programação dinâmica. O algoritmo então processa o conjunto de Turmas candidatas para encontrar a combinação que maximiza a soma do valor total ajustado, sujeito a múltiplas restrições:

- i. A soma dos pesos (w_i), ou seja, as cargas horárias, não deve exceder a capacidade da mochila (a carga horária máxima semanal);
- ii. Nenhuma Turma selecionada pode ter conflito de horário com outra;
- iii. Apenas uma Turma por Disciplina pode ser incluída na solução final.

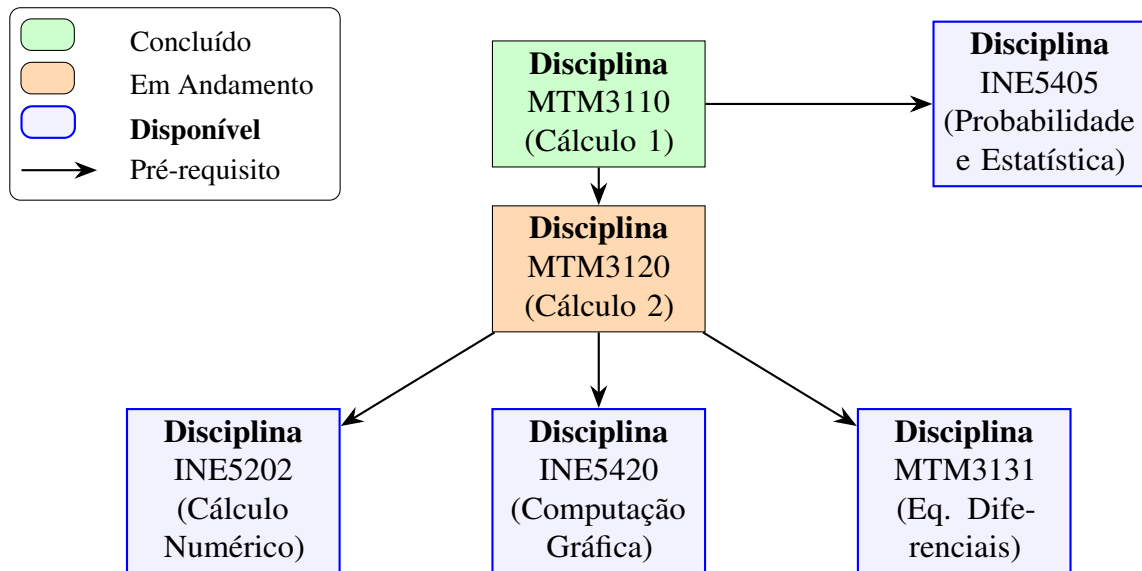
Exemplo 3.3.1. Tomando como base o progresso do aluno apresentado no Exemplo 3.2.1, e considerando um grafo curricular ligeiramente expandido, o primeiro passo da geração de sugestões é identificar as disciplinas elegíveis. Para esta análise, o grafo é incrementado com as disciplinas INE5405 (“Probabilidade e Estatística”), que possui Pré-Requisito de MTM3110, e INE5420 (“Computação Gráfica”), com Pré-Requisito de MTM3120.

O algoritmo de travessia (detalhado na Subseção 3.3.1) analisa então as dependências. As disciplinas INE5202 (Cálculo Numérico) e MTM3131 (Eq. Diferenciais) exigem MTM3120; como esta se encontra “Em Andamento” (status considerado satisfeito para fins de pré-requisito), ambas se tornam disponíveis. De forma análoga, INE5405 (Probabilidade e Estatística) exige MTM3110, que já está “Concluído”, tornando-a também disponível.

Finalmente, INE5420 (Computação Gráfica) também depende de MTM3120 (‘Em Andamento’), o que a torna igualmente elegível. Portanto, para este aluno, o conjunto de disciplinas disponíveis para cursar no próximo semestre seria: INE5405, INE5202, INE5420 e MTM3131.

A Fig. 5 ilustra este resultado, destacando as disciplinas disponíveis no grafo.

Figura 5 – Identificação das disciplinas disponíveis com base no progresso do aluno.



Fonte: O autor.

Exemplo 3.3.2. A partir do conjunto de cinco disciplinas disponíveis, identificado no Exemplo 3.3.1, a etapa seguinte é aplicar a heurística do “Valor Definido da Disciplina” (detalhada na Subseção 3.3.2) e, em seguida, resolver o Problema da Mochila para selecionar a grade final.

Para este exemplo, vamos atribuir a cada disciplina disponível uma carga horária (peso) e um “Valor Definido” (valor), conforme a tabela abaixo. Quanto maior esse valor, indica que a disciplina desbloqueia mais caminhos futuros no currículo.

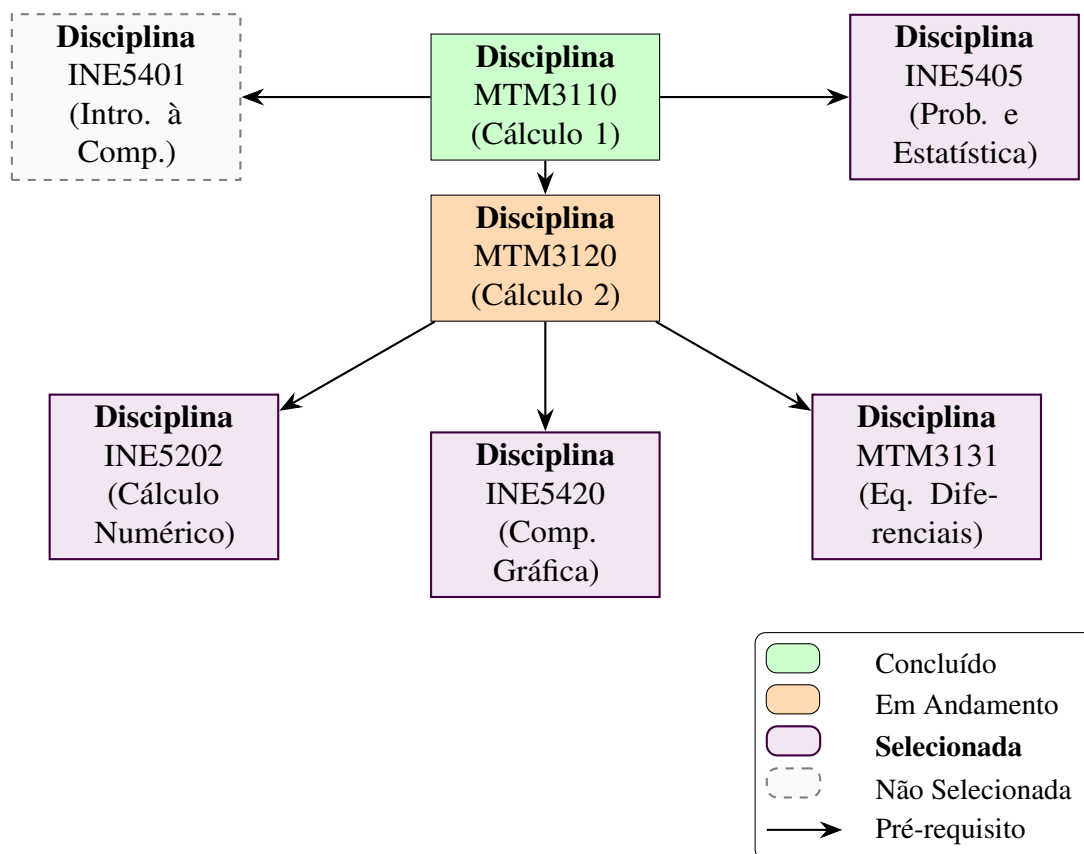
Código	Disciplina	Carga Horária (h)	Valor Definido
INE5202	Cálculo Numérico	4	2
MTM3131	Eq. Diferenciais	4	0
INE5420	Computação Gráfica	4	0
INE5405	Prob. e Estatística	5	4
INE5401	Introdução a Comp.	2	0

Tabela 1 – Disciplinas disponíveis com suas respectivas cargas horárias e valores definidos.

Suponha que o aluno defina uma carga horária máxima (capacidade da mochila) de **18 horas** para o semestre. O algoritmo do Problema da Mochila 0/1 (Subseção 3.3.3) buscará a combinação de disciplinas que maximiza a soma dos valores definidos sem ultrapassar as 18 horas. A solução ótima para este cenário seria selecionar as seguintes disciplinas: **Cálculo Numérico, Eq. Diferenciais, Prob. Estatística e Computação Gráfica.**

Esta combinação resulta em uma carga horária total de **17 horas** e um valor definido total de **6**, sendo a maior pontuação possível dentro do limite de **18 horas**. A disciplina INE5401 não são selecionadas, pois sua inclusão, mesmo que coubessem na carga horária, resultaria em uma carga horária obtida mais baixa. A Fig. 6 ilustra este resultado, diferenciando as disciplinas selecionadas das não selecionadas.

Figura 6 – Resultado da seleção de disciplinas com base na carga horária e valores definidos.



Fonte: O Autor.

4 IMPLEMENTAÇÃO REAL DA APLICAÇÃO

Os capítulos anteriores estabeleceram a fundamentação teórica (Capítulo 2) e o design conceitual da solução (Capítulo 3). Este capítulo detalha a implementação técnica, descrevendo como os conceitos de modelagem em grafo, processamento de dados e visualização interativa foram traduzidos em um sistema funcional. A arquitetura foi concebida em três camadas principais: o *banco de dados* (persistência), o *backend* (lógica de negócios e processamento) e o *frontend* (interface e visualização do usuário). O código para a aplicação chamada de **Planejador Acadêmico Unificado Livre ONLINE (PAULON)** pode também ser acessado no GitHub do Repositório do Código.

BANCO DE DADOS

A escolha da tecnologia de persistência foi uma decisão central para a viabilidade do projeto. Conforme justificado na Subseção 3.1.1, a natureza dos dados curriculares, onde as conexões entre entidades são fundamentais, levou à adoção do Neo4j. Trata-se de um banco de dados orientado a grafos, projetado especificamente para armazenar e consultar dados cuja estrutura é baseada em conexões.

Esta abordagem permitiu uma implementação direta do modelo de dados definido na Subseção 3.1.3. Entidades como Disciplina, Currículo e Turma foram implementadas como nós no grafo. As conexões entre elas, como PRÉ-REQUISITO, FAZ PARTE DE e OFERECE, foram implementadas como relações direcionadas. A conexão da aplicação com o banco de dados é gerenciada por um *driver* específico.

O uso do Neo4j simplifica as consultas de dependência. A linguagem de consulta *Cypher* permite a travessia nativa do grafo para executar operações que são centrais para este trabalho, como a identificação de pré-requisitos.

PIPELINE DE DADOS E ESTRUTURAÇÃO

O *backend* da aplicação atua como a camada central de lógica e processamento, executando duas funções essenciais. A primeira é a ingestão e estruturação dos dados, que transforma as informações curriculares de diversas fontes em um formato coeso e persistente dentro do banco de dados Neo4j. A segunda é atuar como um servidor de *API*, que responde às solicitações do *frontend* para consulta e análise desses dados.

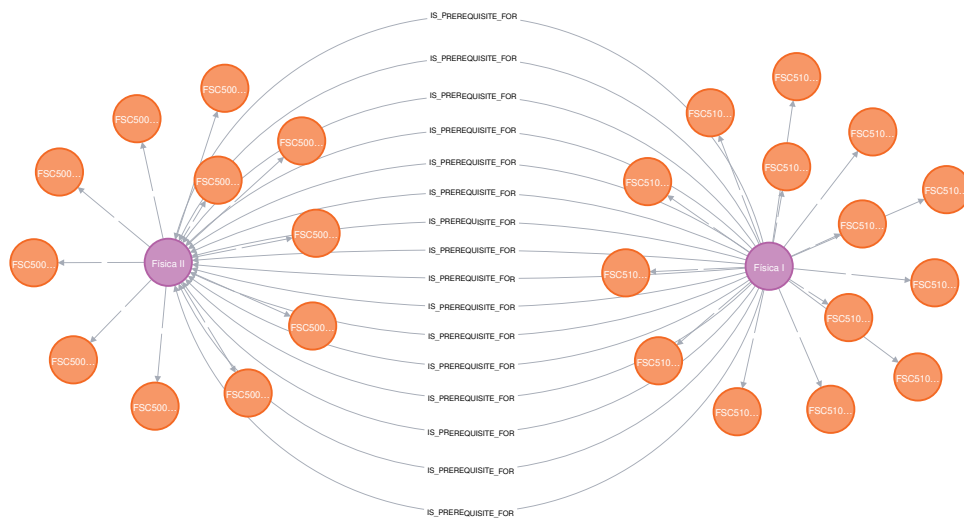
Coleta e Estruturação de Dados

O processo de alimentação do banco de dados inicia-se fora do servidor principal, através de *scripts* de coleta e formatação. Esta etapa, por vezes referida como ETL (Extra-

Este parser não processa o JSON diretamente, ele itera sobre um conjunto de *handlers* especializados. Cada *handler*, como o *BachelorDegreeCurriculumHandler*, é responsável por verificar se consegue interpretar a estrutura de um determinado arquivo JSON. Se um *handler* reconhece o formato, ele assume a responsabilidade de traduzir o JSON em um conjunto de objetos de modelo de dados, como *CurriculumGraph* e *CourseNode*.

Estes objetos, que representam o grafo em memória, são então enviados para o módulo *CurriculumRepository*. Este repositório atua como a camada de abstração final da persistência, isolando a lógica de negócio da lógica de armazenamento. Ele obtém uma sessão de escrita a partir do *Neo4jDriverFactory* e transforma os objetos do modelo em consultas *Cypher*. A operação de salvamento é transacional: primeiro, executa uma consulta em lote para criar ou atualizar (MERGE) todos os nós de *Disciplina* e *Currículo*. Somente após a conclusão desta etapa, uma segunda transação em lote é executada para criar os relacionamentos (como PRÉ-REQUISITO). Ele transforma os objetos do modelo em consultas *Cypher* para criar ou atualizar os nós e relações no Neo4j, garantindo que a estrutura de dados definida na Subseção 3.1.3 seja corretamente persistida.

Figura 8 – Consulta de turmas oferecidas para disciplinas.



Fonte: O autor.

Visualização e Interação

A camada de *frontend* é o principal ponto de interação do usuário com o sistema. Ela foi desenvolvida utilizando a biblioteca *ReactFlow*, uma ferramenta voltada para a renderização de aplicações interativas baseadas em nós e arestas.

Para garantir que o grafo seja apresentado de forma legível e organizada, a biblioteca *dagre* é utilizada. Ela calcula automaticamente um layout hierárquico para os nós, organizando as disciplinas visualmente de acordo com seus semestres sugeridos e dependências.

A implementação utiliza um componente de nó customizado, o *CourseNode*, para exibir as informações de cada disciplina. Alinhado aos objetivos da Seção 3.2, o componente

CurriculumDiagram aplica estilos visuais distintos aos nós com base no progresso do aluno. São utilizados estilos específicos para diferenciar disciplinas concluídas, em andamento ou pendentes, permitindo a imediata visualização do estado acadêmico do estudante. Além da visualização estática, o ReactFlow habilita a interatividade. Funções como as que permitem ao usuário clicar em uma disciplina para destacar visualmente todas as suas dependências (pré-requisitos) ou as disciplinas que dependem dela (pós-requisitos), auxiliando na análise de caminhos críticos.

Camada de Aplicação e Lógica de Negócios

A comunicação entre a camada de visualização (ReactFlow) e a camada de dados (Neo4j) não é direta. Ela é intermediada por uma camada de aplicação, implementada como *API routes* dentro do *framework* Next.js. Este *backend-for-frontend* é responsável por executar a lógica de negócios e as consultas ao banco de dados.

Esta camada de API serve a duas funções distintas. A primeira é a de consulta simples ao grafo, implementada por *endpoints* como `app/api/graph/route.js` e específicos como `app/api/graph/prerequisites/[nodeId]/route.js`. Estes *endpoints* recebem solicitações da interface do usuário e executam consultas *Cypher* para buscar os nós, arestas e caminhos necessários para renderizar a visualização do progresso, conforme descrito na Seção 3.2.

A segunda função é a execução da lógica de sugestão, que é a implementação prática da Seção 3.3. O *endpoint* `app/api/sugestoes/route.js` recebe o progresso do aluno e suas restrições (carga horária, dias a evitar). Ele então orquestra uma série de consultas e algoritmos: identifica as disciplinas cursáveis (Subseção 3.3.1), aplica a heurística de valoração (Subseção 3.3.2) e, por fim, seleciona o conjunto ótimo de turmas.

Processamento do Histórico do Aluno

Uma implementação central para a personalização da ferramenta é a extração de dados do Controle Curricular do aluno, conforme proposto na Subseção 3.2.1. Este processo inicia-se no *frontend* com o componente PdfUploader, que envia o arquivo PDF para o *endpoint* `app/api/upload/route.js`.

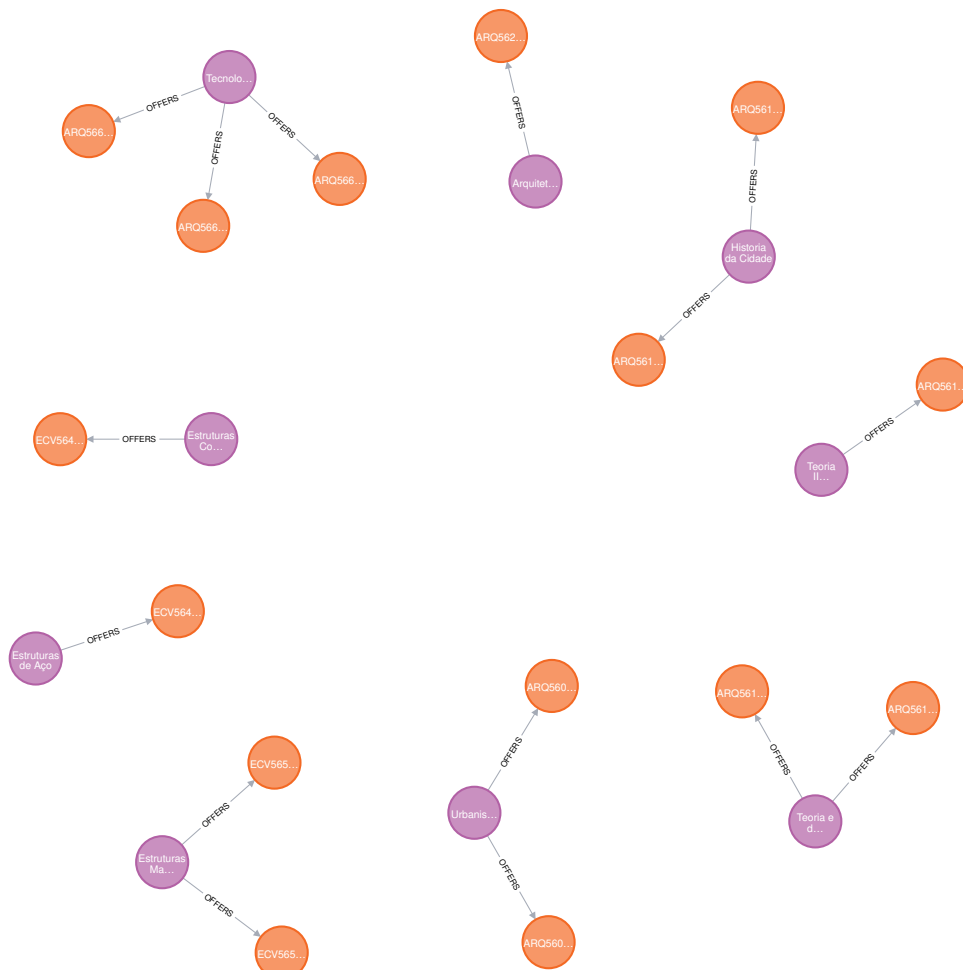
Devido à complexidade da análise de PDFs, a solução implementada adota uma arquitetura poliglota. O *endpoint* da API (Node.js) não processa o arquivo diretamente; ele invoca um processo filho para executar um *script* Python, o `pdf_parser.py`. Este *script* utiliza bibliotecas de extração de texto e expressões regulares para analisar o conteúdo do documento, superando o desafio da variabilidade de formatação (Subseção 3.2.2). O *parser* identifica os códigos das disciplinas e as classifica em listas (concluídas, em andamento, dispensadas), retornando um objeto JSON estruturado para a camada de API. Este JSON é então armazenado no estado da sessão do usuário, servindo como entrada para todas as operações de visualização e sugestão.

Geração da Grade Horária e Resolução de Conflitos

A etapa final do processo de sugestão é a tradução da lista de disciplinas otimizadas em uma grade horária viável. Esta funcionalidade implementa na prática a Otimização da Grade (Subseção 3.3.3), combinando-a com a teoria de Gestão de Conflitos de Horário (Subseção 2.1.3). Os dados das turmas reais, com seus respectivos horários e professores, são importados para o Neo4j pelo *pipeline* de dados, criando os nós Turma e as relações OFERECE (Subseção 3.1.3).

O algoritmo de sugestão em `app/api/sugestoes/route.js` executa uma otimização que não apenas maximiza o “valor” das disciplinas (Problema da Mochila), mas também seleciona um conjunto de nós Turma específicos que não possuam sobreposição de horários entre si e que respeitem os dias e turnos indesejados pelo aluno. O resultado desta computação é uma grade horária completa e sem conflitos, que é então enviada ao *frontend* e renderizada visualmente para o usuário através do componente `WeeklyScheduleView`.

Figura 9 – Consulta de turmas oferecidas para matérias sugeridas.



Fonte: O Autor.

5 CONCLUSÃO

Este trabalho abordou o desafio que estudantes universitários enfrentam na visualização e planejamento de suas trajetórias acadêmicas, um processo frequentemente dificultado por estruturas curriculares lineares e pela complexidade na identificação de interdependências entre disciplinas. A proposta central foi o desenvolvimento de uma aplicação computacional para a geração automática de grafos curriculares, visando facilitar a compreensão da estrutura do curso e auxiliar na tomada de decisões estratégicas.

Para atingir este objetivo, o projeto fundamentou-se na teoria dos grafos para modelar o currículo como um Grafo Acíclico Direcionado (DAG), onde as disciplinas são representadas por nós e os pré-requisitos por arestas. A solução foi implementada utilizando uma arquitetura de três camadas: um banco de dados orientado a grafos para a persistência eficiente dos relacionamentos, uma lógica para o processamento e análise dos dados, e uma interface de usuário interativa para a renderização visual.

Os objetivos específicos do trabalho foram alcançados. A aplicação demonstra visualmente os grafos curriculares criados, permitindo uma clara visualização das dependências. O sistema processa o Controle Curricular do estudante para mapear seu progresso individual, diferenciando disciplinas concluídas, em andamento ou pendentes. Além disso, a ferramenta gera sugestões de matrículas para semestres futuros, aplicando uma heurística de valoração de disciplinas e otimizando a seleção através de um modelo análogo ao Problema da Mochila 0/1.

Em suma, a implementação do sistema valida a viabilidade técnica da aplicação de modelos baseados em grafos para a gestão de trajetórias acadêmicas. A ferramenta automatiza a verificação de pré-requisitos, um processo anteriormente manual, e instrumentaliza dados institucionais para fornecer uma visão sistêmica da graduação. Ao converter grades curriculares estáticas em estruturas de dados dinâmicas e interativas, o trabalho estabelece uma base tecnológica funcional que permite aos estudantes transitar de uma leitura passiva de catálogos para um planejamento ativo e fundamentado de sua formação.

Trabalhos Futuros

A evolução natural desta pesquisa passa, primeiramente, pela plena integração com os padrões de interoperabilidade estabelecidos pela iniciativa do Diploma Digital. A substituição dos atuais mecanismos de extração de dados de arquivos PDF pela ingestão direta de documentos XML validados via XSD eliminaria a fragilidade associada à variabilidade de formatações. Essa padronização não apenas auxiliaria a coleta de dados individuais, mas também habilitaria a escalabilidade da ferramenta para outras instituições de ensino.

Com a consolidação de uma base de dados padronizada, abre-se a perspectiva para a aplicação de técnicas de Mineração de Dados Educacionais e Aprendizado de Máquina. A arquitetura atual pode ser expandida para incorporar a análise de dados históricos, permitindo a transição de um modelo puramente prescritivo para um modelo preditivo, conforme comentado

na Subseção 2.2.3. A análise de grandes volumes de trajetórias acadêmicas permitiria identificar padrões de sucesso e risco, sugerindo caminhos ótimos baseados em probabilidades estatísticas de aprovação e retenção.

Nesse contexto, o sistema de recomendações pode evoluir para considerar conjuntos de disciplinas frequentemente cursadas em simultâneo com êxito. Diferente da abordagem atual, que trata as disciplinas individualmente dentro do Problema da Mochila, métodos baseados em regras de associação ou filtragem colaborativa poderiam sugerir *clusters* de matérias que possuem sinergia pedagógica ou logística, capturando o conhecimento tácito dos estudantes que não está explícito no grafo curricular oficial.

Por fim, para garantir a efetividade das soluções propostas, é imprescindível a validação das heurísticas desenvolvidas através de métodos empíricos. A realização de estudos de caso e a aplicação de questionários (*surveys*) junto ao corpo discente permitirão avaliar a qualidade subjetiva das recomendações e a usabilidade da visualização em grafos. O feedback coletado nessas pesquisas servirá como base para o refino dos pesos utilizados nos algoritmos de otimização, alinhando as sugestões matemáticas às necessidades reais e à satisfação dos estudantes.

REFERÊNCIAS

- AJANOVSKI, V. V. Body of knowledge explorer: Long-term student guidance across the computer-science domain. In: **Proceedings of the 19th Koli Calling International Conference on Computing Education Research**. New York, NY, USA: Association for Computing Machinery, 2019. (Koli Calling '19). ISBN 9781450377157. Disponível em: <https://doi.org/10.1145/3364510.3364531>.
- AMAZON. **What is ETL? — Extract, Transform, Load Explained**. 2025. <https://aws.amazon.com/what-is/etl/>. Acessado em 27 out. 2025.
- ATASHBAR, N. Z.; RAHIMI, F. Optimization of educational systems using knapsack problem. **International Journal of Machine Learning and Computing**, p. 552–555, 01 2011.
- BONDY, J. A.; MURTY, U. **Graph Theory with Applications**. [S.l.]: North-holland, 1979.
- CAROLINA, U. of S. **Curricular Analytics — University Advising Center**. 2025. Acessado em 13 de out. de 2025. Disponível em: https://sc.edu/about/offices_and_divisions/advising/university_advising_center/curricular_analytics/.
- DIESTEL, R. **Graph Theory**. [S.l.]: Springer, 2025.
- GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, MA, USA: Addison-Wesley, 1995.
- HANSEN, J.; NEMETH, A.; STEWART, J. Extending curricular analytics to analyze undergraduate physics programs. **Phys. Rev. Phys. Educ. Res.**, American Physical Society, v. 20, p. 020143, Nov 2024. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.20.020143>.
- HEILEMAN, G. L. et al. Curricular analytics: A framework for quantifying the impact of curricular reforms and pedagogical innovations. **arXiv preprint**, 2018. Disponível em: <https://arxiv.org/abs/1811.09676>.
- INFORMACAO(DPGI/SEPLAN), D. de Gestao da. **UFSC em Numeros**. [S.l.], 2023. Disponível em: <http://dpgi-seplan.ufsc.br/>. Acesso em: 08 jun. de 2024.
- ITI. **ICP-Brasil**. 2025. <https://www.gov.br/iti/pt-br/assuntos/icp-brasil>. Publicado originalmente em 27/06/2017. Atualizado em 07/05/2025 13h07. Acessado em: 07 out. 2025.
- KELL, B. **21-110: Conflict Graphs**. 2010. Disponível em: <https://www.math.cmu.edu/~bkell/21110-2010s/conflict-graphs.html>.
- KELLEY JR., J. E.; WALKER, M. R. Critical-Path Planning and Scheduling. In: **Proceedings of the 1959 Eastern Joint Computer Conference**. [S.l.: s.n.], 1959. p. 160–173.
- KRASNOV, D. **CurricularAnalytics: A vignette for the R package CurricularAnalytics**. 2025. Acessado em 13 de out. de 2025. Disponível em: <https://cran.r-project.org/web/packages/CurricularAnalytics/vignettes/CurricularAnalytics.html>.
- MACNAIR, S. et al. Curriculum visualization. In: **2020 ASEE Virtual Annual Conference Content Access**. ASEE Conferences, 2020. Disponível em: <https://stevemacn.github.io/resources/papers/Curri-Curriculum-Visualization-ASEE-2020.pdf>.

MACNAIR, S.; OUTROS. Curriculum visualization. In: **Proceedings of the ASEE Annual Conference & Exposition 2020**. [s.n.], 2020. Acessado em 13 de out. de 2025. Disponível em: <https://stevemacn.github.io/resources/papers/Curri-Curriculum-Visualization-ASEE-2020.pdf>.

MEC. PORTARIA Nº 554, DE 11 DE MARÇO DE 2019. **Diário Oficial da União**, mar. 2019. Disponível em: https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/66544171/do1-2019-03-12-portaria-n-554-de-11-de-marco-de-2019-66543842.

MEC. **Diploma Digital**. 2021. <https://www.gov.br/mec/pt-br/diploma-digital>. Acessado em 5 jul. 2025.

MEC. **Pacote XSD v1.05 — Diploma Digital**. 2025. Acessado em 8 de out. de 2025. Disponível em: <https://www.gov.br/mec/pt-br/diploma-digital/dados>.

MEINKE, J. G.; BAUER, C. R. Topological sorting as a tool in curriculum planning. **SIGCSE Bull.**, Association for Computing Machinery, New York, NY, USA, v. 8, n. 3, p. 61–66, jul. 1976. ISSN 0097-8418. Disponível em: <https://doi.org/10.1145/952991.804757>.

MOTA, F. J. **Guru da Matrícula: Um simulador de matrículas multi-institucional e com suporte a análise de pré-requisitos**. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Catarina, Florianópolis, 2019. Bacharelado em Ciências da Computação.

MUSTAPA, M.; SALAHUDDIN, L.; HASHIM, U. Automated study plan generator using rule-based and knapsack problem. **International Journal of Advanced Computer Science and Applications**, v. 13, 01 2022.

PISINGER, D.; TOTH, P. Knapsack problems. In: _____. **Handbook of Combinatorial Optimization: Volume1–3**. Boston, MA: Springer US, 1998. p. 299–428. ISBN 978-1-4613-0303-9. Disponível em: https://doi.org/10.1007/978-1-4613-0303-9_5.

SANTIAGO, R. de. **Anotacoes para a Disciplina de Grafos**. [S.l.], 2023. Disponível em: <https://www.inf.ufsc.br/~r.santiago/downloads/INE5413.pdf>. Acesso em: 08 jun. de 2024.

STAVRINIDES, P.; ZUEV, K. Course-prerequisite networks for analyzing and understanding academic curricula. **Applied Network Science**, v. 8, 04 2023.

TAHA, H. A. **Operations Research: An Introduction**. 8th. ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2006. ISBN 0-13-188923-0.

UFSC/INE. **Currículo de Ciência da Computação**. 2025. <https://cco.ufsc.br/curriculo/>. Acessado em 05 jul. 2025.

University of Illinois Chicago. **Curricular Analytics Tools**. 2024. Acessado em: 07 dez. 2025. Disponível em: <https://curricularanalytics.uic.edu/>.

XIAO, J. The application of greedy algorithm in the course selection and scheduling system of college students. **Applied and Computational Engineering**, v. 75, p. 48–53, 07 2024.

ZUEV, K.; STAVRINIDES, P. Breadth, depth, and flux of course-prerequisite networks. **arXiv preprint**, 2025. Disponível em: <https://arxiv.org/html/2506.23510>.

A ARTIGO

Geração e Análise de Grafos Curriculares para Planejamento Acadêmico

Matheus Paulon Novais¹, Gustavo Zambonin¹, Jean Everson Martina¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

matheus.novais@grad.ufsc.br, gustavo.zambonin@posgrad.ufsc.br, jean.martina@ufsc.br

Abstract. *Understanding student progression in higher education is a complex challenge, often hindered by the linear and textual format of traditional curriculum grids. This lack of structural clarity complicates the identification of dependencies and prerequisites. This work proposes the development of a computational tool for the automatic generation of curriculum graphs. By modeling the curriculum as a Directed Acyclic Graph (DAG) and applying the Knapsack Problem to the course recommendation process, the system allows students to visualize their specific academic path and optimize future semester choices. The solution, utilizes Neo4j for graph persistence and processes student history to provide personalized, conflict-free schedule suggestions.*

Resumo. *Compreender a progressão de um aluno em um curso superior é um desafio complexo, frequentemente dificultado pelo formato linear e textual das grades curriculares tradicionais. Essa falta de clareza estrutural complica a identificação de dependências e pré-requisitos. Este trabalho propõe o desenvolvimento de uma ferramenta computacional para a geração automática de grafos curriculares. Ao modelar o currículo como um Grafo Acíclico Dirigido (DAG) e aplicar o Problema da Mochila ao processo de recomendação de disciplinas, o sistema permite que os estudantes visualizem sua trajetória específica e otimizem as escolhas dos futuros semestres. A solução, utiliza Neo4j para persistência em grafos e processa o histórico escolar para oferecer sugestões de grade personalizadas e livres de conflitos.*

1. Introdução

No ano de 2022, mais de 36 mil alunos se matricularam na Universidade Federal de Santa Catarina [de Gestão da Informação (DPGI/SEPLAN) 2023], junto a esse registro são definidas as disciplinas que serão realizadas por cada um desses alunos. A partir do segundo semestre de graduação surge uma das problemáticas do estudante, durante o período de recesso deve-se escolher suas matérias baseadas em recomendações do seu currículo e de sugestões dadas por colegas e professores para isso é necessário ler um documento extenso em qual é citado cada pré-requisito, código e possivelmente carga horária necessária para cursar cada matéria e fazer o pedido para a instituição confirmar.

A organização tradicional da grade curricular, com nomes e códigos lineares, dificulta a compreensão das interdependências e pré-requisitos entre as disciplinas, levando a escolhas equivocadas e atrasos na conclusão do curso. A falta de ferramentas de planejamento eficientes também impede a criação de cronogramas de estudo otimizados,

prejudicando o aprendizado e o tempo dos alunos. Soluções tecnológicas, como a geração automatizada de grafos curriculares, podem facilitar a visualização e o planejamento da trajetória acadêmica, permitindo que os estudantes tomem decisões informadas sobre seu percurso educativo.

Para o estudante, a ausência de uma representação estrutural clara exige um esforço manual de mapeamento de pré-requisitos, o que pode resultar em escolhas que não otimizam a trajetória até a conclusão do curso. Nesse contexto, a aplicação de modelos formais, como a teoria dos grafos, oferece uma abordagem estruturada para traduzir essas dependências em um formato computacionalmente analisável. Diante dessa dificuldade de visualização, este estudo propõe o desenvolvimento de um gerador automático de grafos curriculares. A ferramenta visa, portanto, aplicar essa fundamentação teórica para auxiliar os alunos na complexa tarefa de navegar por essa rede de interdependências, facilitando a compreensão da estrutura do curso e a tomada de decisões estratégicas para otimizar seu aprendizado.

Uma ideia básica de um grafo curricular é definir seus vértices como representações das disciplinas de um currículo e as arestas como os pré-requisitos entre elas. Vértices são unidades ou entidades, enquanto arestas indicam as conexões entre pares de vértices [de Santiago 2023], facilitando a visualização das relações e interdependências entre as matérias. Para aprimorar a organização visual e a clareza, conceitos de grafos podem ser aplicados para agrupar disciplinas por semestre ideal, área de conhecimento ou status (concluída, em andamento, pendente), facilitando a identificação de caminhos críticos e a distinção entre diferentes fases do curso.

Além de proporcionar uma visualização clara e intuitiva das relações entre as matérias, a aplicação deverá incorporar os dados oficiais de disciplinas e turmas, cedidos pela Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação (SeTIC). A ferramenta utilizará o controle curricular do próprio aluno para preencher dinamicamente seu progresso, permitindo a geração de sugestões para a organização dos futuros semestres com base em sua trajetória individual. Com esses dados, a ferramenta oferecerá um panorama completo do currículo, auxiliando na construção de um plano de estudos eficiente e personalizado.

2. Fundamentação Teórica e Trabalhos Relacionados

A fundamentação matemática para a representação de estruturas acadêmicas reside no conceito de grafos, definidos inicialmente como pares de conjuntos compostos por vértices e arestas [Diestel 2025]. Dada a natureza sequencial do aprendizado, essa definição é refinada para Grafos Acíclicos Direcionados (DAGs), nos quais a orientação dos arcos denota precedência e a ausência de ciclos é mandatória conforme definem Bondy e Murty [Bondy and Murty 1979]. Nesse contexto, o Grafo Curricular é formalizado como um DAG onde o conjunto de vértices representa as disciplinas ofertadas e o conjunto de arcos expressa as relações de pré-requisito entre elas [Zuev and Stavrinides 2025, MacNair and outros 2020]. Essa propriedade de aciclicidade assegura a coerência lógica do plano de ensino, pois a existência de qualquer ciclo implicaria uma dependência circular impossível de ser resolvida pelo estudante.

Além do sequenciamento lógico garantido pelo DAG, o planejamento acadêmico envolve problemas de alocação de recursos e gestão de conflitos que exigem abordagens

algorítmicas distintas. A construção de grades horárias sem choques utiliza a técnica de coloração de vértices em grafos de conflito [Kell 2010], onde o número cromático determina o mínimo de horários necessários. Paralelamente, para otimizar o tempo de integralização, aplica-se o Método do Caminho Crítico (CPM) [Kelley and Walker 1959], que permite identificar as cadeias de disciplinas que determinam a duração mínima do curso e onde atrasos são inadmissíveis.

No que tange à seleção estratégica de disciplinas para um semestre específico, este trabalho modela a decisão do estudante como uma instância do Problema da Mochila 0/1 (*Knapsack Problem*) [Pisinger and Toth 1998]. Nessa analogia, o semestre letivo possui uma capacidade limitada definida pela carga horária máxima, e as disciplinas disponíveis são itens com pesos (créditos) e valores (utilidade estratégica). O objetivo torna-se maximizar o valor acadêmico agregado sem violar o limite de créditos, o que transforma uma escolha subjetiva em um problema de otimização combinatória passível de resolução via algoritmos gulosos ou exatos [Zandi Atashbar and Rahimi 2011].

O desenvolvimento dessas ferramentas insere-se no campo do *Curricular Analytics* [Heileman et al. 2018], que propõe métricas quantitativas como fatores de bloqueio e atraso para avaliar a complexidade curricular. Embora trabalhos como o de Mota [Mota 2019] tenham avançado na automação da grade horária semanal através de ferramentas como o “Guru da Matrícula”, observa-se uma lacuna na integração entre a resolução de conflitos imediatos e o planejamento de longo prazo. A presente proposta busca preencher esse espaço ao utilizar a visualização completa do grafo para guiar o estudante por trajetórias ótimas, diferenciando-se pela análise global do fluxo do curso em detrimento de uma visão puramente semestral.

A viabilidade prática dessa modelagem automatizada é garantida pela recente padronização de dados promovida pela iniciativa do Diploma Digital e do ecossistema do Ministério da Educação [MEC 2021]. A adoção de padrões abertos como XML e esquemas XSD rigorosos [MEC 2025] assegura a interoperabilidade entre sistemas e a fidedignidade dos dados históricos. Isso permite que a geração dos grafos curriculares seja alimentada por dados estruturados e confiáveis.

3. Desenvolvimento da solução

A escolha de um modelo de dados em grafo se deu pela sua capacidade de representar nativamente entidades e as relações entre elas. Um banco de dados orientado a grafos armazena essa abstração de forma isomórfica, tratando tanto os vértices quanto as arestas como entidades de primeira classe. Essa correspondência direta elimina a complexidade translacional requerida por outros paradigmas de armazenamento.

Em uma abordagem com modelo relacional, a representação de tais conexões ocorreria de forma indireta. Seria necessária uma tabela para as disciplinas e uma tabela de junção adicional para mapear cada relação de pré-requisito, associando pares de identificadores de disciplinas. A principal limitação desta abordagem é exposta ao se realizar consultas que necessitam percorrer múltiplos níveis de relacionamento.

Alternativamente, um modelo de dados orientado a documentos representaria cada disciplina como uma estrutura autocontida, na qual as referências aos pré-requisitos seriam incorporadas, por exemplo, como uma lista de identificadores. Tal estrutura é performativa para a recuperação de informações de uma única disciplina.

A adoção de um banco de dados de grafos, portanto, não se justifica apenas pela eficiência na representação e consulta de dados conectados. De forma mais estratégica, essa escolha habilita a aplicação direta de algoritmos como a Ordenação Topológica, essencial para determinar sequências de graduação válidas, e o Método do Caminho Crítico utilizado para identificar disciplinas cruciais ao avanço do estudante, dependem de operações de travessia de grafo. A capacidade de executar essas análises de forma nativa e eficiente no próprio banco de dados é uma consequência direta da escolha deste paradigma, alinhando a implementação tecnológica com os objetivos analíticos do trabalho.

3.1. Modelagem do Currículo para um Grafo Direcionado

A dependência entre as disciplinas foi modelada utilizando grafos direcionados. Por exemplo, uma aresta $(A) \rightarrow (B)$ com o rótulo, *Pré-Requisito*, indicando que a disciplina *A* é um pré-requisito para a disciplina *B*, estabelecendo uma ordem obrigatória de progressão, transformando a grade curricular em um Grafo Acíclico Dirigido (DAG).

A persistência desse modelo foi projetada para ser atômica e livre de redundâncias. A decisão conceitual foi garantir que, para cada currículo de um curso específico, existisse um e apenas um grafo correspondente. Ao adicionar ou atualizar informações, o sistema não simplesmente insere novos dados, mas verifica a existência de cada nó e aresta antes de criá-los. Essa abordagem idempotente assegura a integridade e a consistência do grafo ao longo do tempo, um requisito essencial para a confiabilidade das consultas e sugestões. A consulta para salvar o grafo, por exemplo, utiliza uma chave composta pelo ID do currículo e pelo código do curso para garantir a unicidade da estrutura no banco de dados.

Com o currículo formalizado como um grafo, torna-se possível aplicar algoritmos para quantificar e extrair informações sobre a sua topologia. Análises de conectividade e centralidade, por exemplo, permitem identificar programaticamente as disciplinas que possuem maior impacto no fluxo do curso, seja por servirem como pré-requisito para um grande número de outras matérias (alto grau de saída), ou por funcionarem como pontos de conexão entre diferentes áreas da grade (alta centralidade de intermediação).

3.2. O Esquema do Grafo Curricular: Nós e Relações

Para que o modelo de grafo seja computacionalmente interpretável e consistente, é essencial formalizar sua estrutura através de um esquema. Este esquema define as categorias de entidades que existem no domínio e os tipos de conexões que podem existir entre elas.

As entidades do domínio são mapeadas no grafo através de três tipos de vértices: a *Disciplina*, contendo atributos essenciais como código e carga horária; o *Currículo*, que estrutura a grade do curso vigente em determinado período; e a *Turma*. Já as interações entre os nós são representadas por três arestas distintas: o *Pré-Requisito*, que liga nós de *Disciplina* estabelecendo a precedência obrigatória; o *Faz parte de*, que conecta a *Disciplina* ao *Currículo*, permitindo o compartilhamento de matérias entre cursos; e a relação *Oferece*, que vincula a *Disciplina* às suas *Turmas*.

4. Visualização do progresso do estudante

Com o grafo curricular estabelecido, a etapa subsequente é determinar a posição e o percurso já realizado por um estudante específico dentro deste mapa. Esta fase foca na tradu-

ção do histórico acadêmico individual em um formato de dados que possa ser diretamente sobreposto ao grafo, para permitir a geração de recomendações personalizadas.

4.1. O Controle Curricular como Fonte de Dados do Aluno

A fonte de dados selecionada para representar a jornada do aluno é o Controle Curricular. Este documento, é a intersecção entre a estrutura curricular oficial do curso e o histórico escolar do estudante. Ele contém o registro de todas as disciplinas e seus respectivos status: cursadas e aprovadas, em andamento, dispensadas por equivalência e pendentes. A escolha deste documento se deve à sua natureza consolidada, servindo como um registro que reflete o conjunto de nós no grafo curricular que o aluno já “visitou” ou está em processo de visitar. O ponto de partida para a contextualização do progresso é a submissão do Controle Curricular pelo estudante. O sistema foi projetado para que esta interação seja a mais direta possível, tratando o documento como fonte para a análise.

4.2. Extração de Dados e Estruturação do Progresso

Uma vez que o arquivo PDF é recebido pelo servidor, ele é submetido a um processo de análise para extrair as informações relevantes. O desafio desta etapa reside na variabilidade de formatação entre os Controles Curriculares de diferentes cursos. A ausência de um padrão unificado para estes documentos exige que o parser seja capaz de lidar com diferentes layouts. Uma padronização futura, como a adoção do Diploma Digital, poderia simplificar este processo, fornecendo dados já estruturados.

O produto final do processo de análise é uma representação computacional e organizada do progresso do aluno. Cada disciplina identificada é classificada e agrupada em categorias distintas que definem seu status, como *cursadas*, *dispensadas* ou *em andamento* e que metadados como *curso*, *currículo* e a quantidade de horas aula mínima, máxima e média.

A etapa final desta contextualização é o mapeamento do objeto de progresso, obtido na fase anterior, com o grafo curricular persistido. O sistema utiliza os códigos das disciplinas extraídos do documento para identificar os nós (*Disciplinas*) correspondentes no grafo. Para os algoritmos de recomendação, as disciplinas nas categorias cursadas, dispensadas e em andamento são funcionalmente equivalentes, pois todas representam pré-requisitos que estão sendo ou já foram satisfeitos.

O **subgrafo de progresso do aluno** é o conjunto de nós do grafo curricular que representam as disciplinas já cursadas, dispensadas ou em andamento por um estudante. Ele não constitui uma nova estrutura de dados, mas uma projeção do grafo principal que expressa o estado atual do aluno dentro do currículo, servindo como base para a determinação das disciplinas que podem ser cursadas nos próximos semestres.

5. Geração de sugestões para o semestre

A fase final da solução consiste na aplicação de uma sequência de algoritmos para sintetizar as informações do modelo de grafo e do progresso do aluno em uma sugestão de matrícula concreta e otimizada. Este processo é executado quando o sistema recebe uma requisição com o contexto do aluno, o semestre desejado e suas restrições pessoais. A geração é dividida em três etapas computacionais: a filtragem de disciplinas elegíveis, a ponderação de seu valor estratégico e, por fim, a otimização da grade horária.

5.1. Identificação de Disciplinas Cursáveis

O primeiro passo é determinar o universo de nós do tipo *Disciplina* que o estudante está apto a cursar. Esta operação é executada através de um algoritmo que utiliza o subgrafo de progresso do aluno, como seu estado inicial. O sistema inspeciona todos os nós *Disciplina* que possuem uma relação *Faz parte de* com o nó *Currículo* do estudante.

Para cada *Disciplina* candidata, o algoritmo inspeciona todas as suas relações de entrada do tipo *Pré-Requisito*. Uma *Disciplina* é considerada disponível se, e somente se, todos os nós de origem dessas relações (ou seja, todas as suas *Disciplinas* que eram pré-requisitos) já estiverem contidos no conjunto de disciplinas que compõem o subgrafo de progresso do aluno. Essa verificação garante que apenas as *Disciplinas* cujas dependências foram integralmente satisfeitas sejam consideradas para as etapas seguintes. O resultado desta fase é uma lista de todas as *Disciplinas* elegíveis.

5.2. Heurística para Valorar Disciplinas

Com a lista de *Disciplinas* disponíveis, é necessário um critério para diferenciar sua importância relativa, visto que nem todas contribuem igualmente para o avanço no curso. Para quantificar essa relevância, foi desenvolvida uma heurística própria calculada através de uma nova travessia no grafo, partindo de cada uma das *Disciplinas* disponíveis. Para cada *Disciplina*, o algoritmo explora suas relações de saída do tipo *Pré-Requisito* para determinar quantas outras *Disciplinas* ela desbloqueia. A métrica é então ponderada para dar mais peso aos desbloqueios diretos.

A avaliação do potencial de desbloqueio é composta por duas dimensões. A primeira considera os desbloqueios *Diretos*, contabilizando o número de *Disciplinas* para as quais a matéria em análise atua como *Pré-Requisito* imediato; a este valor é atribuído um peso maior na fórmula final. A segunda dimensão abrange os desbloqueios *Indiretos*, disciplinas que exigem como pré-requisito aquelas desbloqueadas diretamente, aos quais se atribui um peso menor no cálculo.

A soma ponderada desses dois valores resulta em uma pontuação numérica que representa o impacto de cada *Disciplina*. Sendo assim, *Disciplinas* que atuam como “chaves” para múltiplos ramos do *Currículo* recebem uma pontuação maior, enquanto *Disciplinas* terminais, que não são *Pré-Requisito* para nenhuma outra, recebem pontuação zero.

5.3. Otimização da Grade Semestral

Este problema foi modelado como uma otimização combinatória, análoga ao Problema da Mochila 0/1. O objetivo é selecionar um conjunto de *Turmas* que maximize a soma dos “valores definidos” para cada *Disciplina*, sem que a soma de suas cargas horárias exceda a capacidade da mochila e sem que haja conflitos de horário.

A definição do valor de cada item (v_i) é crucial, pois combina a importância estratégica da *Disciplina* com a conveniência para o aluno. O valor base é o “valor definido” da *Disciplina* associada, que é multiplicado por um fator de ponderação para priorizar a progressão, este valor é somado a uma “pontuação de compatibilidade”. Essa pontuação de compatibilidade aplica penalidades para refinar a seleção, são subtraídos do valor total da *Turma* caso seus horários de aula coincidam com os dias da semana

que o estudante especificou como indesejados. Seguindo a mesma lógica, uma penalidade distinta também é aplicada se o turno em que a *Turma* é ofertada for um dos períodos que o estudante explicitamente deseja evitar. O resultado é um valor final ajustado para cada *Turma*, refletindo tanto sua importância estratégica quanto sua conveniência pessoal, e é este valor que o algoritmo maximizará.

6. Implementação da ferramenta

A estrutura do sistema divide-se em três camadas essenciais que compreendem a persistência de dados, a lógica de negócios e a interface de visualização. No nível de persistência, a solução adota o banco de dados orientado a grafos Neo4j. Esta abordagem permite o mapeamento direto de entidades como *Disciplina* e *Currículo* em nós, enquanto as dependências, a exemplo de pré-requisitos, são modeladas como relações direcionadas. A utilização da linguagem de consulta *Cypher* viabiliza a execução eficiente de travessias no grafo, fundamentais para a análise de dependências acadêmicas.

O processamento e estruturação dos dados ocorrem no *backend*, que opera inicialmente através de um *pipeline* de ingestão (ETL). *Scripts* externos padronizam dados institucionais brutos em arquivos JSON, separando informações estáticas de ofertas dinâmicas de turmas. A persistência dessa lógica utiliza o componente *GraphParser*, baseado no padrão de projeto *Chain of Responsibility*, para converter os arquivos em objetos de memória e, subsequentemente, armazená-los no banco de dados através de operações transacionais gerenciadas pelo *CurriculumRepository*.

A interação do usuário é concretizada na camada de *frontend* utilizando a biblioteca ReactFlow, que, em conjunto com o algoritmo de layout *dagre*, renderiza o grafo curricular de forma hierárquica e organizada. A interface visual aplica estilizações para refletir o estado acadêmico do estudante, diferenciando disciplinas concluídas, em curso ou pendentes, além de oferecer interatividade para a exploração de pré-requisitos.

A comunicação entre a interface e os dados é mediada por uma camada de API desenvolvida em Next.js, que atua como um *backend-for-frontend*. Esta camada não apenas serve dados para visualização, mas também orquestra a lógica complexa de sugestão de matrículas e o processamento de históricos escolares. Para a extração de dados do histórico em PDF, o sistema emprega uma arquitetura poliglota, invocando *scripts* em Python para realizar a análise textual e estruturar as informações do progresso do aluno. Por fim, o sistema integra a otimização pedagógica com a gestão de conflitos de horário para gerar uma grade horária viável e otimizada.

7. Conclusão

A pesquisa e o desenvolvimento realizados neste trabalho evidenciam que a complexidade inerente ao planejamento acadêmico pode ser mitigada através do uso de ferramentas tecnológicas adequadas. A aplicação proposta cumpriu seu objetivo de transformar a visualização curricular, convertendo listas estáticas e pouco intuitivas em grafos dinâmicos e interativos. A modelagem baseada em Grafos Acíclicos Dirigidos mostrou-se não apenas viável, mas superior às abordagens tradicionais para a representação de dependências acadêmicas.

A implementação do algoritmo de sugestão de matrículas, fundamentado em otimização combinatória, representa um avanço significativo na autonomia discente. Ao

automatizar a ponderação entre a importância estratégica de cada disciplina e as restrições de carga horária, o sistema oferece um suporte à decisão que é, ao mesmo tempo, matematicamente rigoroso e personalizado à realidade de cada aluno.

Os trabalhos futuros devem focar na expansão das capacidades de integração e análise de dados. A adoção dos padrões de interoperabilidade definidos pela iniciativa do Diploma Digital, com a ingestão direta de arquivos XML validados via XSD, é um passo prioritário para eliminar a dependência de formatos de arquivo não estruturados como o PDF. Além disso, a acumulação de dados históricos em um formato de grafo padronizado abre perspectivas promissoras para a aplicação de técnicas de Aprendizado de Máquina. Seria possível evoluir de um modelo puramente prescritivo, baseado nas regras oficiais, para um modelo preditivo, capaz de identificar padrões de sucesso e risco com base no desempenho real de ex-alunos, sugerindo trajetórias que maximizem a eficiência temporal, e também as taxas de aprovação e retenção.

Referências

- Bondy, J. A. and Murty, U. (1979). *Graph Theory with Applications*. North-holland.
- de Gestao da Informacao(DPGI/SEPLAN), D. (2023). Ufsc em numeros. Data report, UFSC.
- de Santiago, R. (2023). Anotacoes para a disciplina de grafos. Annotations, UFSC.
- Diestel, R. (2025). *Graph Theory*. Springer.
- Heileman, G. L., Abdallah, C. T., Slim, A., and Hickman, M. (2018). Curricular analytics: A framework for quantifying the impact of curricular reforms and pedagogical innovations. *arXiv preprint*.
- Kell, B. (2010). 21-110: Conflict graphs.
- Kelley, Jr., J. E. and Walker, M. R. (1959). Critical-Path Planning and Scheduling. In *Proceedings of the 1959 Eastern Joint Computer Conference*, pages 160–173.
- MacNair, S. and outros (2020). Curriculum visualization. In *Proceedings of the ASEE Annual Conference & Exposition 2020*. Acessado em 13 de out. de 2025.
- MEC (2021). Diploma Digital. <https://www.gov.br/mec/pt-br/diploma-digital>. Acessado em 5 jul. 2025.
- MEC (2025). Pacote xsd v1.05 — diploma digital. Acessado em 8 de out. de 2025.
- Mota, F. J. (2019). Guru da matrícula: Um simulador de matrículas multi-institucional e com suporte a análise de pré-requisitos. Bacharelado em Ciências da Computação.
- Pisinger, D. and Toth, P. (1998). *Knapsack Problems*, pages 299–428. Springer US, Boston, MA.
- Zandi Atashbar, N. and Rahimi, F. (2011). Optimization of educational systems using knapsack problem. *International Journal of Machine Learning and Computing*, pages 552–555.
- Zuev, K. and Stavrinides, P. (2025). Breadth, depth, and flux of course-prerequisite networks. *arXiv preprint*.

B CODIGO DA APLICAÇÃO

Alternativamente ao Repositório do Github da aplicação, o código pode também ser acessado pelo Códigos UFSC.