

Visualização Gráfica Interativa dos Catálogos de Curso

D. D. Souza R. N. Máximo B. B. N. França

Relatório Técnico - IC-PFG-19-28
Projeto Final de Graduação
2019 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Visualização Gráfica Interativa dos Catálogos de Curso

Douglas Delgado de Souza, Renato Noronha Máximo, Breno Bernard Nicolau de
França

Instituto de Computação Universidade Estadual de Campinas (UNICAMP), Caixa Postal 6176
13083-970 Campinas-SP, Brasil

d088963@dac.unicamp.br, r176675@dac.unicamp.br, breno@ic.unicamp.br

Resumo. Neste projeto partimos de uma necessidade da comunidade da Unicamp e desenvolvemos uma visualização gráfica para os catálogos de cursos. O projeto partiu de alguns conceitos visuais introduzidos pelo GDE e conta com grande interatividade, trazendo uma experiência rica para quem utiliza o sistema para planejar o curso. Foi apresentado para futuros possíveis usuários e teve feedback positivo. O sistema desenvolvido pode ainda ser expandido para atender a outras tarefas cotidianas da comunidade.

Palavras-Chave: Catálogos de Cursos, Proposta de Currículo, GDE, Visualização Gráfica.

1. Introdução

Cursos de graduação universitários são usualmente estruturados com base em uma matriz curricular, considerando o perfil desejado do egresso, os conteúdos e competências esperados desse egresso. Assim, o currículo é estruturado de forma a organizar em disciplinas os elementos da sua matriz. Na Universidade Estadual de Campinas (UNICAMP), é comum o uso de catálogos de cursos para orientação, referência e evolução dos cursos de graduação e pós-graduação. Nesses catálogos, o currículo é acompanhado de uma proposta de cumprimento sugerida pela coordenação do curso. Esta proposta serve de guia para que os alunos possam se planejar ao longo de sua vida acadêmica com o fim de cumprir os requisitos do curso, e a universidade possui mecanismos para permitir que esse guia seja seguido rigorosamente.

Existem, entretanto, fatores que podem dificultar o andamento regular de um curso, deixando o aluno em uma situação onde não há a possibilidade de seguir exatamente a proposta de cumprimento do currículo. Dentre tais fatores temos: reprovações, transferências internas e externas, mudanças de catálogo, intercâmbios, preferências pessoais por horários, entre outros. A reorganização das disciplinas a serem cursadas

geralmente é um trabalho difícil e recorrente. A cada novo semestre o aluno precisa replanear os semestres seguintes para que possa cumprir o currículo como deseja e no menor tempo possível. Neste planejamento, diversos fatores devem ser levados em conta, dentre eles o oferecimento das disciplinas, o cumprimento de seus pré-requisitos e o limite de créditos semestrais.

Atualmente, os alunos da UNICAMP já contam com uma ferramenta voltada ao planejamento semestral do curso. Este sistema, denominado GDE [1], foi desenvolvido e é mantido por alguns ex-alunos da Unicamp e conta com uma visualização gráfica para o catálogo de curso. Essa ferramenta é muito útil, mas possui algumas limitações, tendo em vista o contexto para o qual foi desenvolvida.

O GDE dá acesso à proposta para cumprimento de currículo e aos detalhes individuais de cada disciplina, mas sua interface possui problemas de usabilidade e a visualização gráfica dos cursos tem pouca integração com o restante das informações, como os detalhes das disciplinas.

Outra limitação do GDE é fato de exigir que o usuário tenha algum vínculo com a UNICAMP e esteja autenticado. Isso impede que alunos externos tenham acesso às suas funcionalidades, deixando-os limitados a obter as informações dos cursos a partir do site da diretoria acadêmica (DAC), com informações estáticas e pouco interativo.

Em outra perspectiva, considerando as coordenações de curso e professores, a atualização dos catálogos dos cursos não é uma tarefa simples e não existe ferramenta que auxilie nesse processo. Por exemplo, decisões como a remoção de uma disciplina do catálogo requerem que se verifique o impacto causado em todas as disciplinas seguintes, conferindo se os conteúdos necessários são ou podem ser ensinados em outras disciplinas. A inclusão de novas disciplinas é outro exemplo de mudança que demanda esforço, pois diversas checagens devem ser feitas a fim de garantir o cumprimento de seus pré-requisitos e o balanceamento da carga horária semestral.

Neste trabalho, desenvolvemos uma ferramenta para consulta aos catálogos dos cursos da UNICAMP. Essa ferramenta tem como objetivo ser de simples utilização, publicamente acessível, útil e visualmente agradável. Denominada GradView, ela é uma nova visualização gráfica e interativa dos catálogos de cursos da UNICAMP e deverá beneficiar futuros ingressantes, alunos advindos de transferência externa e professores

da própria UNICAMP, pois facilitará a visualização do vetor [2] de cada disciplina e sua cadeia de pré-requisitos, tornando mais fácil a comparação entre grades curriculares e o planejamento de novas grades.

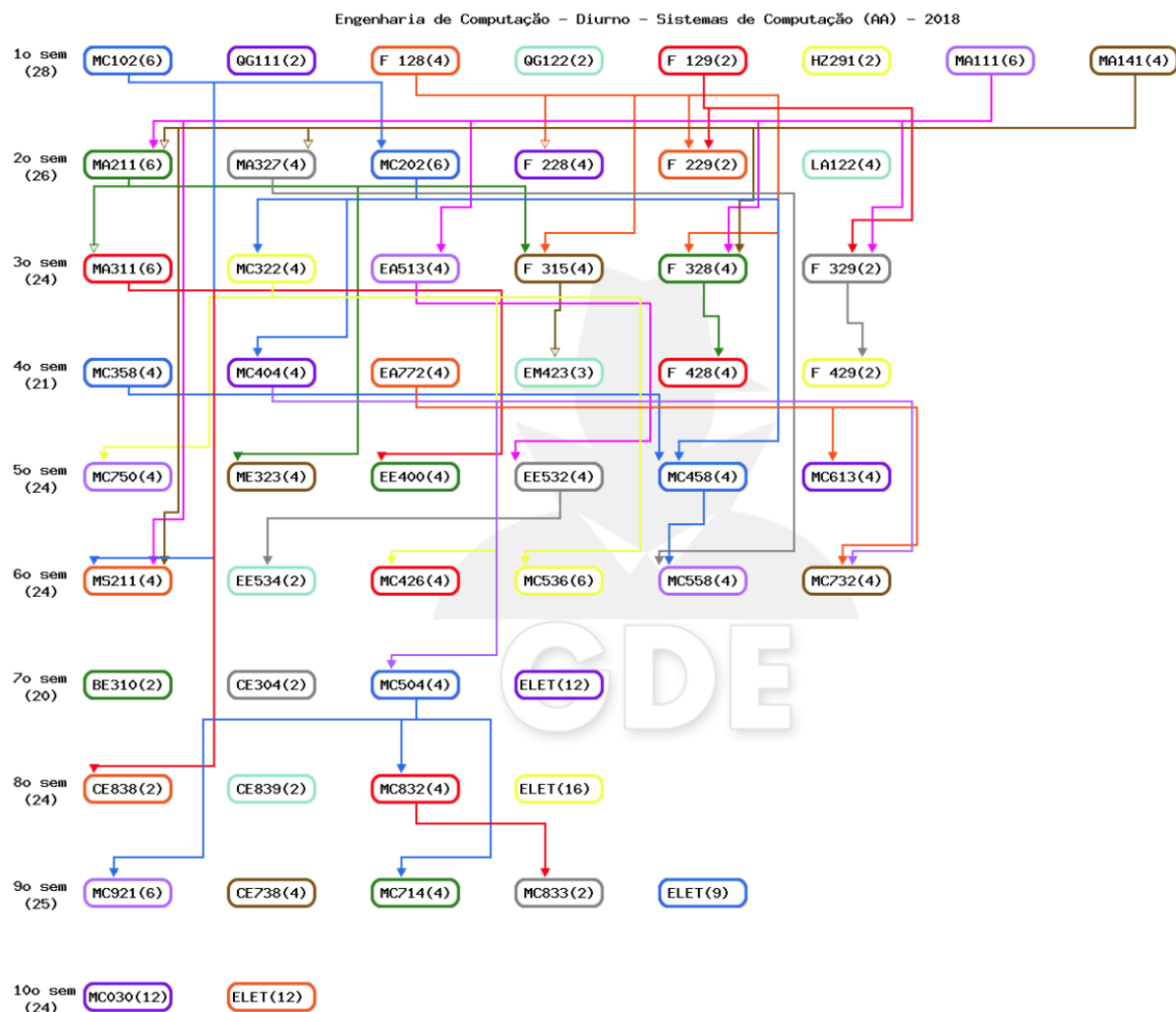
Esta ferramenta poderá ser integrada ao atual site da DAC, sendo uma nova opção oficial na consulta dos catálogos de cursos, e é simples o bastante para ser modificada e implementada em outras universidades brasileiras, ajudando também as suas comunidades.

O restante deste trabalho está organizado da seguinte forma. Na seção 2, apresentamos os conceitos gerais do funcionamento da ferramenta e na seção 3 discutimos sua implementação e funcionamento. Em seguida, na seção 4, apresentamos a avaliação da solução e finalizamos o texto com as conclusões e trabalhos futuros na seção 5.

2. O GDE

Este trabalho se inspira nas qualidades do GDE para desenvolver uma versão com novas funcionalidades e maior interatividade. Um dos maiores destaques do GDE é a sua proposta de visualização gráfica dos catálogos de curso, à qual o próprio sistema se refere como “Árvore de disciplinas”.

Neste formato, um currículo é representado pela a organização das disciplinas em um grafo, onde as disciplinas são representadas pelos nós e as arestas representam as relações de pré-requisitos entre as disciplinas. Neste grafo, os semestres são empilhados verticalmente em camadas, estando o primeiro semestre no topo, como apresentado na Fig. 1.



Obs: Caso existam dois conjuntos de pré-requisitos presentes integralmente no catálogo, apenas um deles será exibido.

Fig. 1: Visualização “Árvore / Integralização” do GDE.

Outra característica interessante do GDE é realçar a cadeia de pré-requisitos quando o cursor do mouse repousa sobre o nó de uma disciplina, conforme apresentado na Fig. 2.

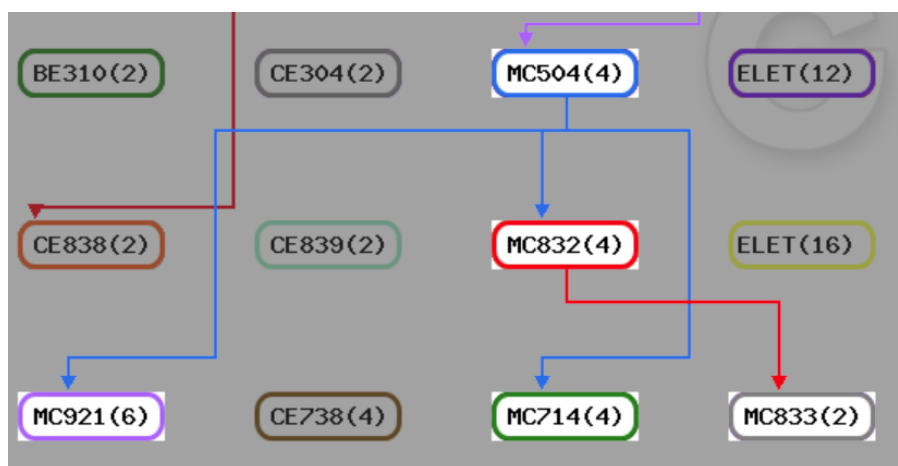


Fig. 2: Destaque para a cadeia de disciplinas ao *hover* do mouse no GDE.

O nome da disciplina também aparece, e ao clique do mouse algumas opções são exibidas, que levam a outras páginas do sistema.

Neste trabalho propomos uma repaginação da “árvore de disciplinas”, conforme descrito a seguir.

3. Proposta de Solução

Seguindo o mesmo formato introduzido pelo GDE projetamos uma ferramenta alternativa para a visualização dos catálogos de curso no formato de grafo. A ferramenta planejada conta com as seguintes características:

1. O sistema deve oferecer suas funcionalidades sem necessidade de autenticação ou controle de acesso;
2. O sistema deve permitir a seleção de um catálogo, curso e modalidade;
3. O sistema deve exibir as disciplinas em um grafo e organizadas por semestre;
4. O sistema deve prover uma visualização que evidencie a cadeia de pré-requisitos;
5. O sistema deve permitir a visualização de todas as informações públicas de uma disciplina a partir da visualização gráfica do catálogo;
6. O sistema deve permitir reorganizar as disciplinas nos semestres interativamente;
7. O sistema deve checar e validar, após cada alteração no catálogo, as restrições associadas às disciplinas, como seus pré-requisitos, e prover *feedback* imediato da factibilidade da alteração realizada no currículo;
8. O sistema deve oferecer uma busca para encontrar os conteúdos previstos em uma disciplinas para um dado curso/modalidade;
9. O sistema deve permitir visualizar as informações da disciplina evidenciando a relação de carga teórica/prática/laboratórios/orientadas/à distância/eletivas;

10. O sistema deve permitir visualizar as informações evidenciando os créditos das disciplinas e balanceamento de créditos nos semestres;
11. O sistema deve permitir identificar a interdependência entre as disciplinas;
12. O sistema deve permitir visualizar os blocos de disciplinas eletivas, exibindo os conjuntos de disciplinas que podem ser escolhidas;
13. O sistema deve permitir mover as disciplinas dos blocos de eletivas para a estrutura proposta do currículo, as quais também são submetidas às checagens de pré-requisitos;
14. O sistema deve permitir remover disciplinas eletivas previamente movidas para estrutura proposta do currículo;
15. O sistema deve prover uma identificação visual específica para os pré-requisitos parciais;
16. O sistema deve apresentar as sugestões de cumprimento das disciplinas eletivas nos semestres com os créditos indicados;
17. O sistema deve organizar as disciplinas nos semestres de forma a minimizar o emaranhamento entre os pré-requisitos, melhorando a visualização;
18. O sistema deve conter os catálogos dos últimos 9 anos disponíveis para consulta (2012-2020);
19. O sistema deve possuir uma visualização agradável, organizada e de fácil utilização.

4. Metodologia

4.1. Planejamento das funcionalidades

Primeiramente, identificamos as funcionalidades existentes no GDE. Ainda, por meio de conversas com outros alunos e professores, identificamos potenciais novas funcionalidades que poderíamos implementar.

Já com um conjunto inicial de funcionalidades planejadas fizemos uma reunião de *brainstorming*, na qual esclarecemos melhor qual seria o objetivo do sistema a ser desenvolvido. A Fig. 3 apresenta algumas preocupações identificadas nesse momento e organizadas por tipos de usuários da ferramenta.

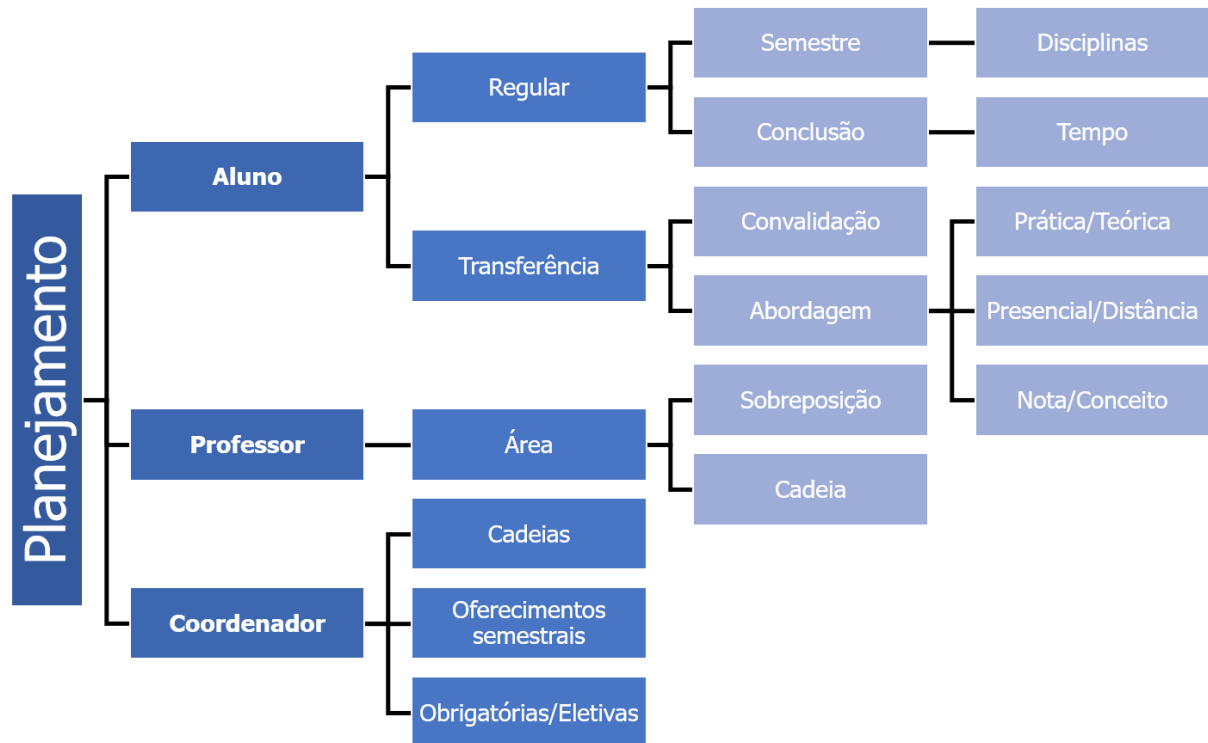


Fig. 3: Planejamento inicial das funcionalidades do sistema.

Criamos um repositório para o projeto na instância do GitLab do IC com o nome MC030-GradView [3] e iniciamos o desenvolvimento.

Após alguns desenvolvimentos notamos que estávamos muito presos no conceito do GDE. O OpenDesign nos foi sugerido para auxiliar a explorar novas ideias.

4.2. Utilizando o OpenDesign

Utilizamos a plataforma OpenDesign [4] para auxiliar com a definição do conceito da ferramenta GradView. Com ela, inicialmente identificamos os potenciais atores envolvidos no uso e desenvolvimento do nosso sistema, conhecidos como *stakeholders*. Na plataforma OpenDesign, o processo de identificação de *stakeholders* é realizada em camadas e vai além dos utilizadores diretos da ferramenta, incluindo um espectro social identificando stakeholders no mercado e comunidade também, como apresentado na Fig. 4.

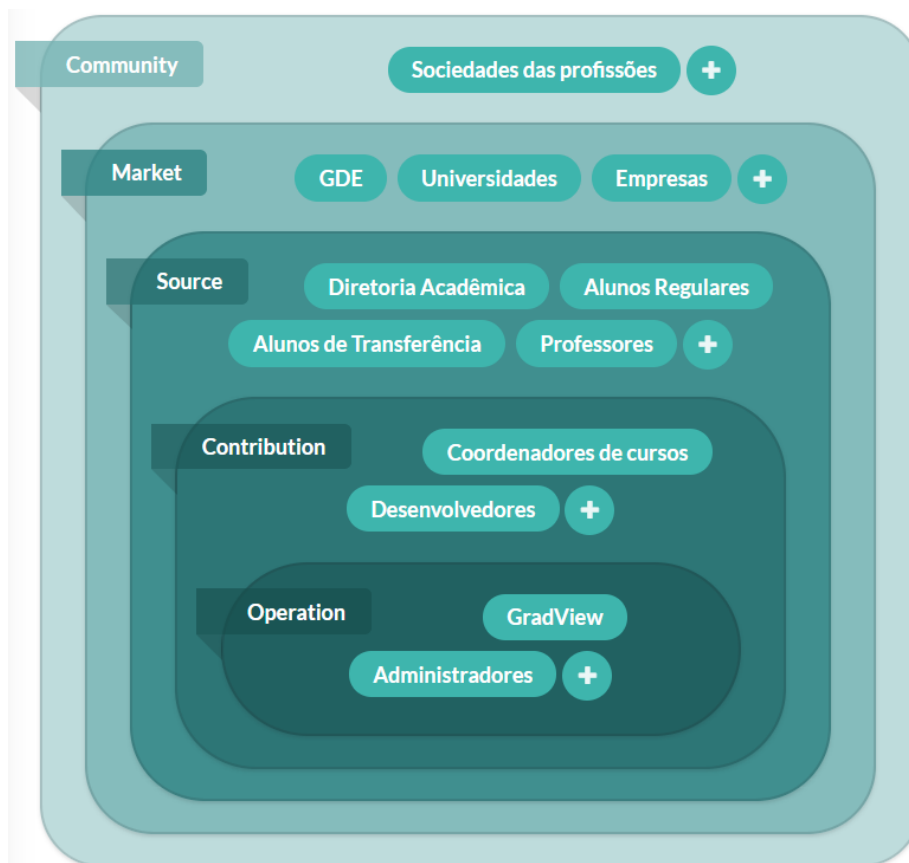


Fig. 4: Diagrama desenvolvido com o OpenDesign, exibindo os *stakeholders* do GradView.

Em seguida, essa plataforma oferece um artefato denominado Quadro de Avaliação, onde é possível identificar para cada *stakeholder* quais seriam os problemas que precisam ser resolvidos e suas potenciais soluções. Todas as entradas de problemas e soluções na plataforma OpenDesign são incluídas automaticamente como *issues* do GitLab marcadas como Problem e Proposal respectivamente.

<p>O vetor das disciplinas possuem informações que são pouco exploradas por serem difíceis de ler</p> <p>#26 · opened 2 months ago by Renato Noronha Maximo</p>	<p>EvaluationFrame Problem SID Source</p>	<p>0 updated 1 month ago</p>
<p>Os alunos muitas vezes desconhecem a existência dos certificados que podem obter focando os estudos em áreas específicas.</p> <p>#67 · opened 1 month ago by Douglas Delgado de Souza</p>	<p>@Alunos Regulares EvaluationFrame Problem SID Source</p>	<p>0 updated 1 month ago</p>
<p>Visualização de certificados de estudos</p> <p>#70 · opened 1 month ago by Douglas Delgado de Souza</p>	<p>@Coordenadores de cursos EvaluationFrame Proposal SID Contribution To Do</p>	<p>0 updated 3 weeks ago</p>
<p>Catálogo gráfico disponível publicamente em sistema oficial</p> <p>#31 · opened 2 months ago by Renato Noronha Maximo</p>	<p>@GDE EvaluationFrame Proposal SID Market To Do</p>	<p>0 updated 1 month ago</p>

Fig. 5: Exemplo de issues geradas no GitLab pelo OpenDesign. A tag Problem é utilizada nos problemas e a tag Proposal para as propostas de solução.

Por fim, preenchemos o Framework Semiótico, obtendo uma visão em escada (diferentes camadas) dos requisitos do GradView.

Com um bom conjunto de *issues* abertas no GitLab seguimos com a implementação do sistema.

5. Implementação

5.1. Arquitetura original

A arquitetura original de implementação do sistema está resumido no diagrama abaixo:

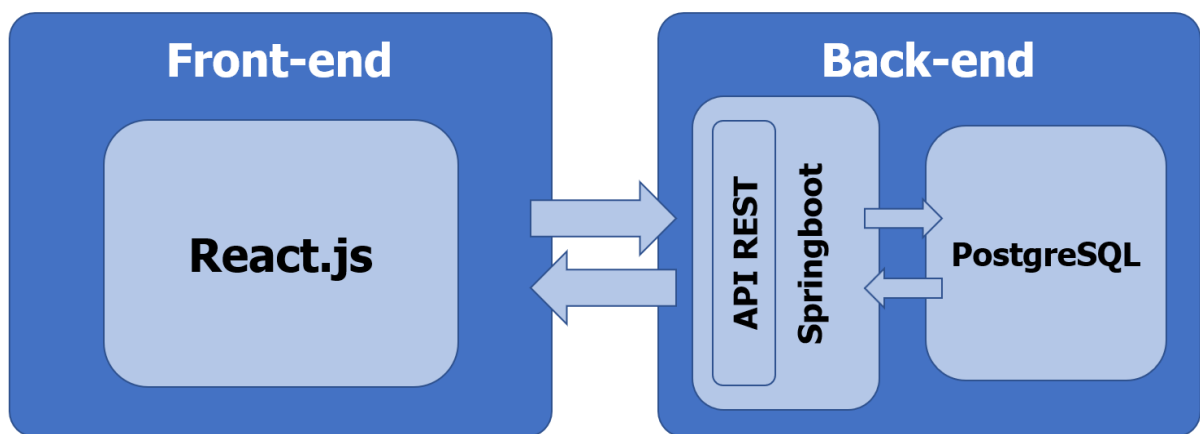


Fig. 6: Diagrama da arquitetura de software e frameworks planejado inicialmente.

A aplicação feita em React.js 16.10.2 é executada no computador do usuário (navegador) e faz chamadas na API REST feita com Springboot, que por sua vez acessa um banco de dados PostgreSQL. Todo o back-end ficaria hospedado em um contêiner do Docker.

A estrutura dos dados foi projetada com base no banco de dados presente no projeto de código aberto do GDE disponível no GitHub [5]. As tabelas foram analisadas e resumidas para conter apenas as informações necessárias para o GradView.

Tab. 1: Tabelas planejadas originalmente para o banco de dados do GradView.

CURSOS		
*	id	int
	numero	smallint
	nome	varchar(255)
	periodo	varchar(50)
*	id_unidade_ensino	int

MODALIDADES		
*	id	int
*	id_curso	int
	nome	varchar(255)

CURRICULOS		
*	id	int
*	id_modalidade	int
	catalogo	int

PROPOSTA_CURRICULO		
*	id	int
*	id_curriculo	int
*	id_disciplina	int
	periodo	smallint

BLOCOS_ELETIVAS		
*	id	int
*	id_curriculo	int
	creditos	int

ELETIVAS		
*	id	int
*	id_bloco_eletivas	int
*	id_disciplina	int

DISCIPLINAS		
*	id	int
	codigo	varchar(5)
	nome	varchar(255)
	vetor	int(8)
	ementa	text
	bibliografia	text
*	id_unidade_ensino	int

PREREQUISITOS		
*	id	int
*	id_disciplina	int
*	id_discip_prereq	int
	catalogo	int

UNIDADE_ENSINO		
*	id	int
	sigla	varchar(10)
	nome	varchar(255)

CERTIFICADOS		
*	id	int
*	id_curso	int
	catalogo	int
	nome	varchar(255)

DISCIPLINAS_CERTIFICADO		
*	id	int
*	id_disciplina	int
*	id_certificado	int

Os dados seriam fornecidos pela DAC e populariam as tabelas do banco de dados do GradView. Para isso, seria necessário que o serviço de consulta ao banco da DAC seja executado no cluster de kubernetes do IC, pois este cluster tem acesso garantido aos dados disponibilizados pela DAC. Com esse fim, foram feitos desenvolvimentos para a criação de uma imagem Docker com o back-end preparado para acessar o banco de dados da DAC.

Infelizmente, não houve tempo hábil para acesso aos dados da DAC, disponibilizado apenas ao final do projeto, e tivemos que buscar outra alternativa.

5.2. Arquitetura final

Após um novo planejamento, optamos por coletar os dados através de alguns *scripts* de coleta de dados web (*scraping*). Esses dados foram então armazenados como objetos JSON e utilizados pelo front-end, podendo ser usados através de requisições web ou através de acesso local.

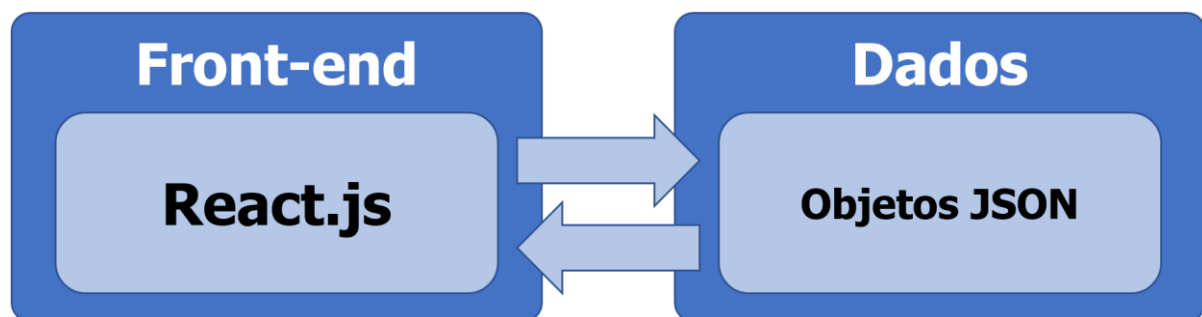


Fig. 7: Diagrama da arquitetura de software simplificada.

Esta estrutura simplificada traz a vantagem de o sistema não exigir muitos conhecimentos de computação para ser implantado, como a configuração de serviços de bancos de dados e a instalação de dependências para o funcionamento de um servidor web. Isso possibilita que o sistema possa ser replicado em outras universidades apenas com a modificação dos dados nos objetos JSON, uma tarefa trabalhosa, mas muito simples. Por outro lado, a fonte de dados é limitada a um formato de visualização utilizado atualmente pelos catálogos da UNICAMP.

5.3. Front-end

Os componentes desenvolvidos foram: App, Catalogue, Subject, PrerequisiteLine, PopupSubjectInfo, SearchInput, CatalogueSelectBar, VectorLegend, Alert, PopupAlert,

Check. A seguir está uma captura de tela do projeto na qual é possível ver onde cada um desses componentes foi utilizado e uma breve descrição de suas responsabilidades e da interface entre eles.

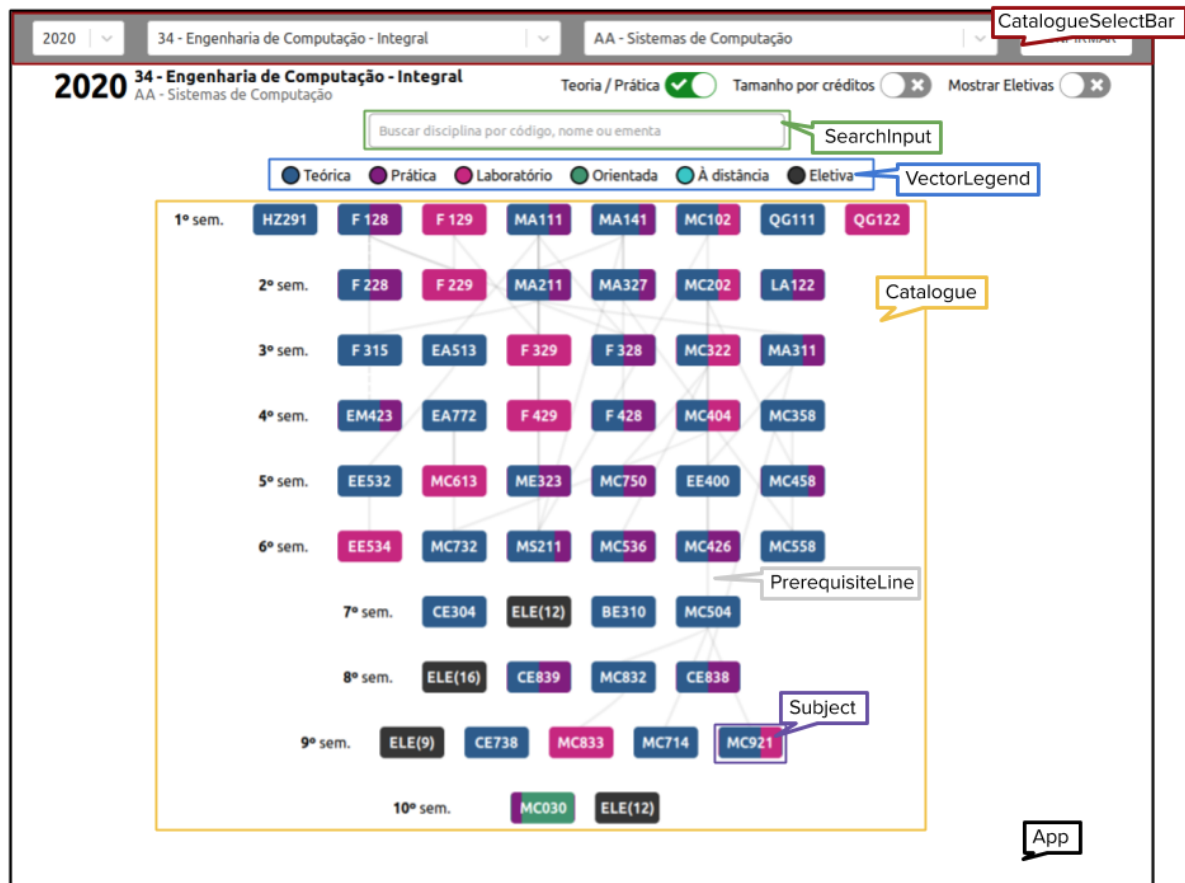


Fig. 8: Captura da tela da aplicação desenvolvida com indicações de seus componentes.

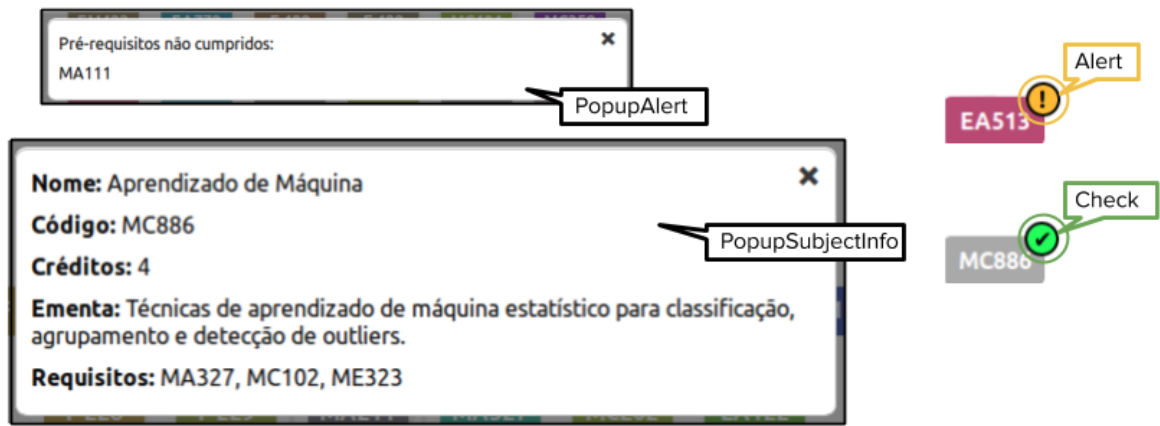


Fig. 9: Capturas de tela de quatro trechos da aplicação desenvolvida com indicações de seus componentes.

O componente **App** é responsável por renderizar o **Catalogue**, **CatalogueSelectBar**, **SearchInput**, **VectorLegend**, os três botões switch para controlar a visualização ('Teoria / Prática', 'Tamanho por créditos' e 'Mostrar Eletivas') e o ano, curso e modalidade da proposta de cumprimento de currículo apresentada na tela. Nele estão as funções que manipulam os dados do catálogo durante as edições e busca na aplicação.

CatalogueSelectBar é responsável por exibir as opções de anos, cursos e catálogos e pela seleção do catálogo a ser exibido. Ele recebe a função a ser executada quando um catálogo é confirmado.

SearchInput é um campo de texto que recebe como props uma função a ser executada toda vez que seu valor é modificado, não sendo necessário clicar em algum botão para que a mesma execute. Essa função verifica para cada disciplina presente no catálogo se a palavra ou frase pesquisada está contida no seu nome, código ou ementa. Em caso negativo diminui a opacidade da disciplina.

VectorLegend é renderizado em App somente na visualização 'Teoria / Prática'. Ele contém a legenda para as cores dessa visualização, isto é, teórica, prática, laboratório, orientada, à distância e eletiva.

Catalogue é o principal componente renderizado em App. Ele renderiza as disciplinas e linhas de pré-requisito para cada semestre e, quando 'Mostrar Eletivas' está selecionado, também renderiza os blocos de eletivas do curso. As principais props que Catalogue recebe são os dados do catálogo selecionado como um objeto, cinco booleans indicando os estados dos três botões de controle, se está em uma busca (existe algo escrito no campo de busca) e se está em uma edição (alguma disciplina está sendo movida no momento). Nele estão as funções a serem executadas quando o mouse entra e sai de uma disciplina e a função para verificar se todos os requisitos de uma disciplina foram cumpridos e em caso negativo quais não foram.

PrerequisiteLine são as linhas que conectam uma disciplina aos seus pré-requisitos. A linha pode ser tracejada, para um pré-requisito parcial, ou contínua, para um pré-requisito pleno, e da cor da disciplina pré-requisito ou preta. Esse componente recebe os códigos da disciplina e de seu requisito e três booleans indicando o tipo de coloração, se está em uma edição e se o pré-requisito é parcial.

Subject são as disciplinas obrigatórias e eletivas. Elas podem ser arrastadas entre os semestres e clicadas para mostrar suas informações detalhadas. Esse componente recebe as informações da disciplina como um objeto, sua opacidade, funções a serem executadas quando o mouse entra e quando sai da disciplina, dois booleans indicando o estado dos botões de controle 'Teoria / Prática' e 'Tamanho por créditos', um boolean indicando se existe um aviso pela falta de cumprimento de pré-requisitos e um vetor com os códigos de pré-requisitos que não são cumpridos até o semestre anterior do qual a disciplina se encontra.

PopupSubjectInfo é o popup com nome, código, ementa e requisitos de uma determinada disciplina. Suas props são as informações da disciplina como um objeto, uma boolean indicando se ele está aberto ou não e uma função a ser executada quando seu botão de fechar é clicado.

Alert é o círculo amarelo com uma exclamação que aparece no canto direito superior de disciplinas que não tem seus requisitos cumpridos até o semestre anterior ao qual estão. Ele recebe uma boolean indicando se deve ou não ser renderizado e uma função que será executada quando ele for clicado.

PopupAlert é o popup aberto ao clicar no **Alert** de uma disciplina e contém os códigos de requisitos não cumpridos. Suas props são um vetor com os códigos de pré-requisitos não cumpridos, uma boolean indicando se ele está aberto ou não e uma função a ser executada quando seu botão de fechar é clicado.

Check é o círculo verde com um check que aparece no canto direito superior de disciplinas eletivas quando já estão planejadas no catálogo, isto é, foram arrastadas para algum semestre. Ele recebe como props uma boolean indicando se deve ou não ser renderizado.

Também foram instalados e utilizados componentes desenvolvidos por terceiros. A seguir está uma lista dos que estão na versão final com seus nomes, referências e em que parte do projeto foram utilizados.

- **@material-ui [6]:** Para botões e ícones de botões
- **react-beautiful-dnd [7]:** utilizada no drag and drop de disciplinas entre semestres e a partir do bloco de eletivas.

- **react-debounce-input [8]:** utilizada no campo de texto de busca de ementa, código ou nome da disciplina. Com esse componente, a função a ser executada após uma mudança no valor do campo só era executada após um período de tempo sem mudanças (50 ms).
- **reactjs-popup [9]:** para os popups de informação da disciplina e de alerta de pré-requisitos não cumpridos.
- **react-switch [10]:** utilizada para os botões switch do controle de tipo de visualizações do catálogo.
- **react-select [11]:** para os campos de seleção do catálogo (ano, curso e modalidade). Foi escolhida por ser pesquisável.
- **randomcolor [12]:** utilizada quando as cores das disciplinas eram geradas no front-end e não recebidas no json do catálogo.
- **react-curveto [13]:** utilizada nas linhas de requisito. Foi alterada para correção de um bug.

5.4. Dados

Todos os dados são armazenados de forma estruturada em dois tipos de objetos JSON:

5.4.1. Dados sobre catálogos, cursos e modalidades

O objeto JSON utilizado inicialmente contém informações sobre todos os catálogos, cursos e modalidades que existem. Essas entradas ainda possuem referência para o arquivo que contém o objeto JSON que carrega informações de um currículo específico, incluindo sua proposta de cumprimento.

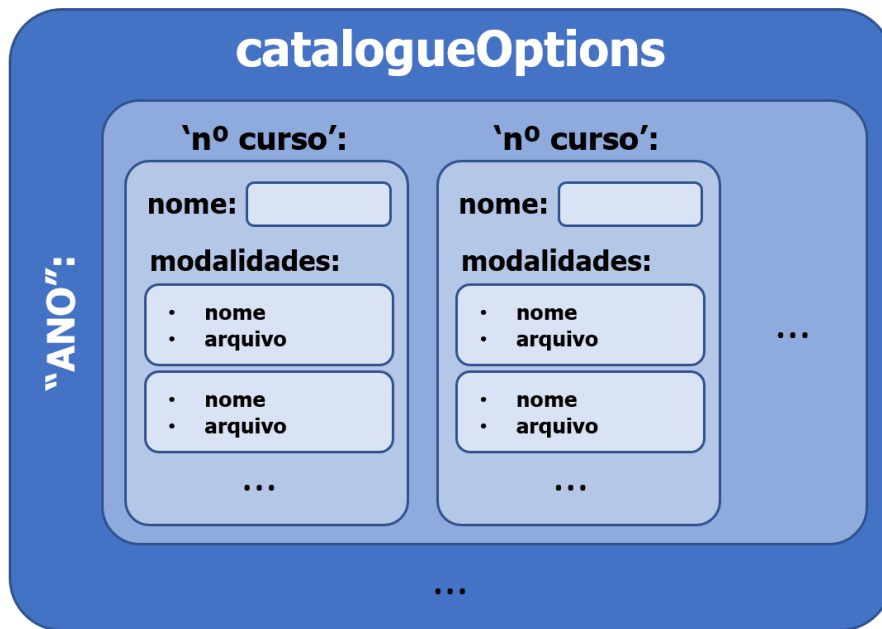


Fig. 10: Estrutura do objeto JSON com informações de catálogos, cursos e modalidades.

Esta estrutura se encontra representada no trecho abaixo:

```
const catalogueOptions = {
  "2012": {
    '1': {
      name: "Matemática - Integral",
      modality: [
        {
          name: 'AA - Bacharelado em Matemática',
          file: '2012/curso1_mod1.js'
        },
        {
          name: 'AB - Licenciatura em Matemática',
          file: '2012/curso1_mod2.js'
        },
        {
          name: 'EF - Ênfase em Física Matemática',
          file: '2012/curso1_mod3.js'
        }
      ]
    },
    '2': {
      name: "Estatística - Integral",
      modality: [
        {
          name: "",
          file: '2012/curso2_mod1.js'
        }
      ]
    }
  },
  ...
},
...
```

```

"2020": {
  '1': {
    name: "Matemática - Integral",
    modality: [
      {
        name: 'AA - Bacharelado em Matemática',
        file: '2020/curso1_mod1.js'
      },
      {
        name: 'EF - Ênfase em Física Matemática',
        file: '2020/curso1_mod2.js'
      },
    ]
  },
  '2': {
    name: "Estatística - Integral",
    modality: [
      {
        name: "",
        file: '2020/curso2_mod1.js'
      },
    ]
  }
  ...
}
};
export default catalogueOptions;

```

Cód. 1: Trechos do objeto JSON com informações de catálogos, cursos e modalidades.

5.4.2. Dados sobre um currículo específico

O conjunto catálogo/curso/modalidade define um currículo único. Este currículo é o conjunto de disciplinas obrigatórias e blocos de créditos em eletivas que precisam ser cumpridos para a obtenção do diploma. A UNICAMP fornece ainda uma proposta para cumprimento do currículo, composto pela distribuição das disciplinas nos semestres, incluindo créditos de eletivas.

O segundo tipo de objeto JSON detalha a proposta de cumprimento de um currículo, incluindo suas disciplinas eletivas:

total créditos:

máximo créditos semestral:

semestres:

'sem-1':

- id
- disciplinas:

'sem-2':

- id
- disciplinas:

...

blocos eletivas:

'bloco-1':

- créditos
- possui restrição
- disciplinas:

'bloco-2':

- créditos
- possui restrição
- disciplinas:

...

disciplinas:

'codigo-1':

- codigo
- nome
- ementa
- semestre
- vetor
- requisitos
- cor
- obrigatória

'codigo-2':

...

Fig. 10: Estrutura do objeto JSON com informações de um currículo específico.

Esta estrutura se encontra representada no trecho abaixo:

```

const catalogue = {
  totalCredits: 240,
  maxCreditsSem: 30,
  semesters: {
    'sem-1': {
      id: '1',
      subjects: ['HZ291', 'F_128', 'F_129', 'MA111', 'MA141', 'MC102', 'QG111',
        'QG122']
    },
    'sem-2': {
      id: '2',
      subjects: ['F_228', 'F_229', 'MA211', 'MA327', 'MC202', 'LA122']
    },
    ...
  },

  eletivas: {
    'elet-1': {
      credits: 13,
      hasRestrictions: false,
      subjects: [],
    },
    'elet-2': {
      credits: 12,
      hasRestrictions: true,
      subjects: ['MC019', 'MC020', 'MC032', 'MC033', 'MC040', 'MC041', 'MC050',
        'MC051'],
    },
    ...
  },

  subjects: {
    'F_128': {
      code: 'F_128',
      name: 'Física Geral I',
      ementa: 'Cinemática do ponto. Leis de Newton. Estática e dinâmica da
        partícula. Trabalho e energia. Conservação da Energia. Momento
        linear e sua conservação. Colisões. Momento angular da partícula

```

```

        e de sistemas de partículas. Rotação de corpos rígidos.',
    semestre: 1,
    vector: {T:2, P:2, L:0, O:0, D:0, HS:4, SL:4, C:4},
    requisitos: '',
    color: '#B9A246',
    obligatory: true
},
'MC102': {
    code: 'MC102',
    name: 'Algoritmos e Programação de Computadores',
    ementa: 'Conceitos básicos de organização de computadores. Construção de
        algoritmos e sua representação em pseudocódigo e linguagens de
        alto nível. Desenvolvimento sistemático e implementação de
        programas. Estruturação, depuração, testes e documentação de
        programas. Resolução de problemas.',
    semestre: 1,
    vector: {T:4, P:0, L:2, O:0, D:0, HS:6, SL:6, C:6},
    requisitos: '',
    color: '#A2B946',
    obligatory: true
},
...
}
};

export default catalogue;

```

Cód. 2: Objeto JSON com informações de um currículo específico, incluindo proposta de cumprimento do currículo e blocos de eletivas.

6. O sistema GradView

6.1. Vista principal

Na figura abaixo apresentamos uma visão do sistema apresentando a proposta para cumprimento do currículo do curso 34, modalidade AA, catálogo de 2020.

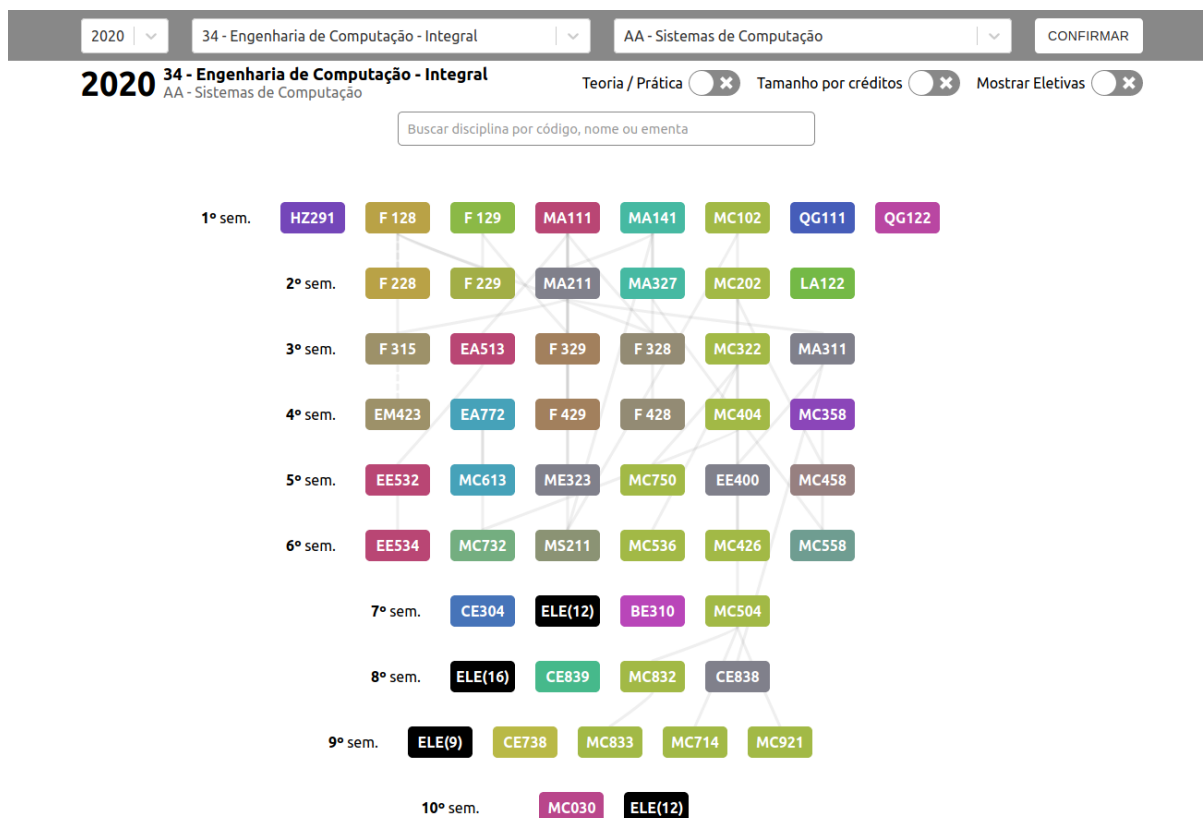


Fig. 11: Captura de tela da aplicação desenvolvida. O catálogo selecionado é o de Engenharia de Computação - Integral, 2020, modalidade AA - Sistemas de Computação .

Na barra superior, em cinza, é feita a escolha da combinação catálogo/curso/modalidade e ao clicar em “CONFIRMAR” o grafo das disciplinas é exibido. Cada retângulo é uma disciplina a ser cursada e as linhas que interligam as disciplinas são as linhas de pré-requisito.

6.2. Efeito quando mouse está sobre uma disciplina

Toda a cadeia de disciplinas pré-requisitos anteriores à disciplina sob o mouse ficam coloridas enquanto as outras disciplinas do catálogo ficam cinza e com baixa opacidade.

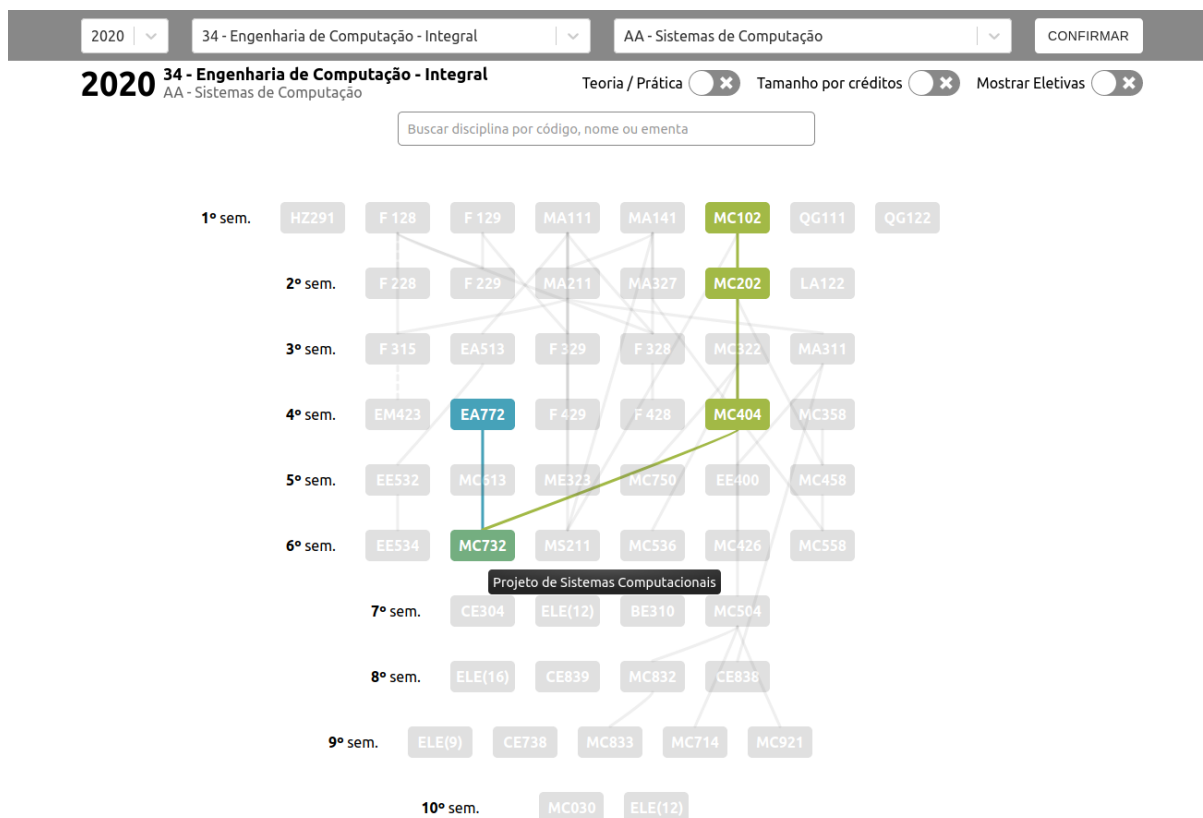


Fig. 13: Captura de tela da aplicação com o mouse sobre a disciplina ‘MC732 - Projetos de Sistemas Computacionais’.

Com o intuito de facilitar a identificação de disciplinas de uma mesma cadeia, a cor de uma disciplina é uma média das cores de seus pré-requisitos. Observamos claramente a cadeia das disciplinas de programação iniciadas com “MC”, todas com quase a mesma cor herdada da disciplina “MC102”.

6.3. Diferentes visualizações

É possível colorir as disciplinas com base na quantidade de créditos de aulas teóricas, práticas, de laboratório, orientadas, à distância que a mesma possui ou se ela é uma sugestão para cursar créditos de eletivas.

Também é possível modificar o tamanho de cada disciplina para que seja proporcional à quantidade total de créditos da mesma.

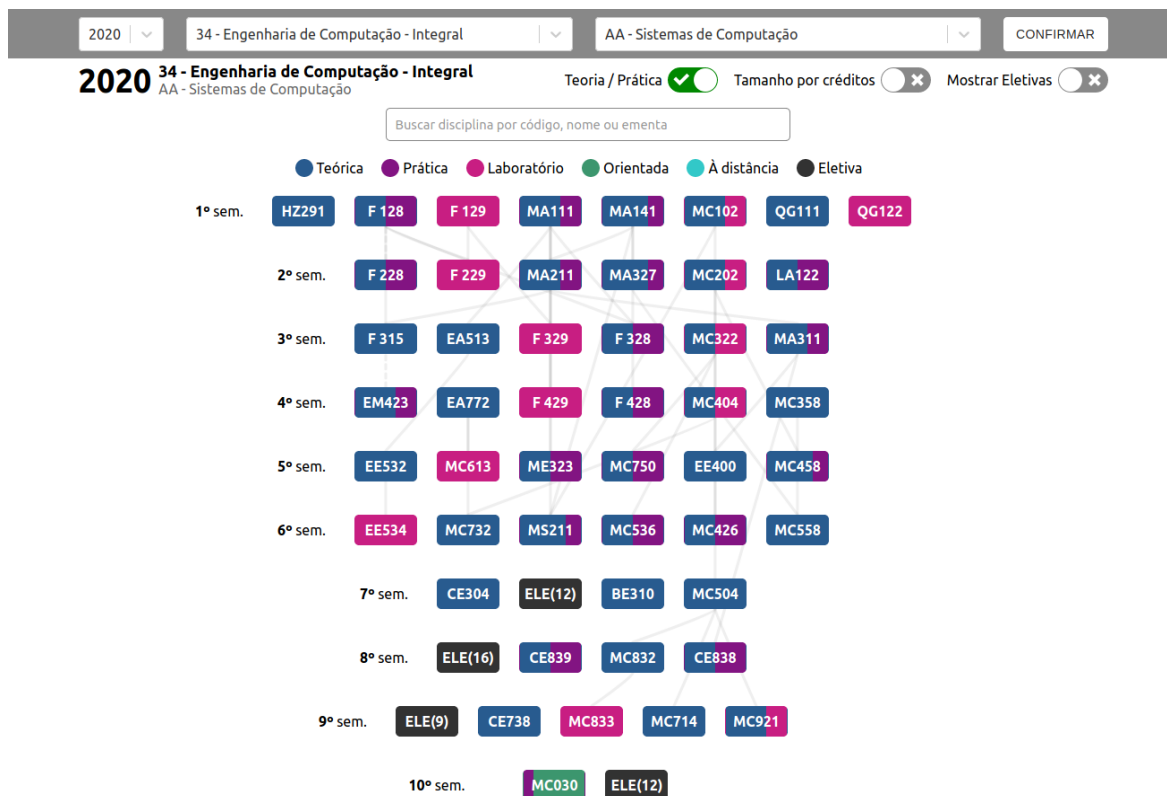


Fig. 14: Captura de tela da aplicação com a visualização ‘Teoria / Prática’ ligada.

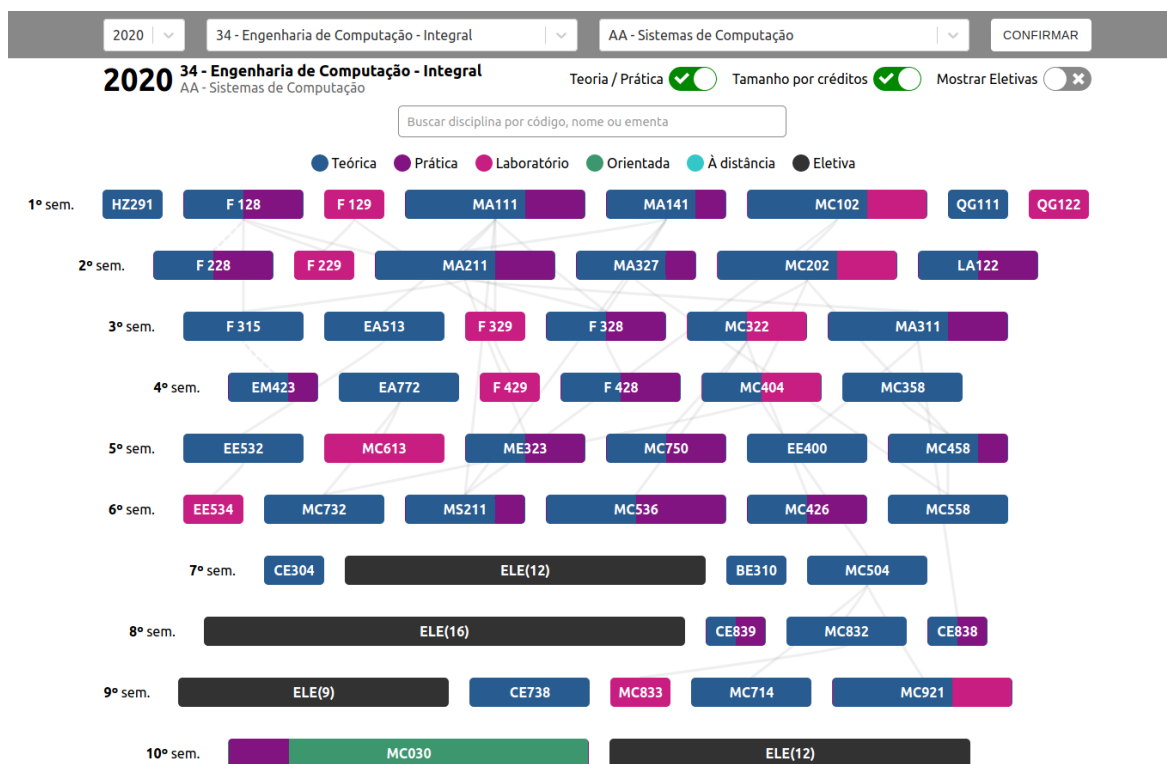


Fig. 15: Captura de tela da aplicação com as visualizações ‘Teoria / Prática’ e ‘Tamanho por créditos’ ligadas.

6.4. Campo de busca por nome, ementa e código da disciplina.

Todas as disciplinas que **não** possuem o texto procurado em seu código, nome ou ementa ficam com baixa opacidade. A busca não diferencia letras maiúsculas de minúsculas, não diferencia letras acentuadas de sua letra sem acento nem ç de c.

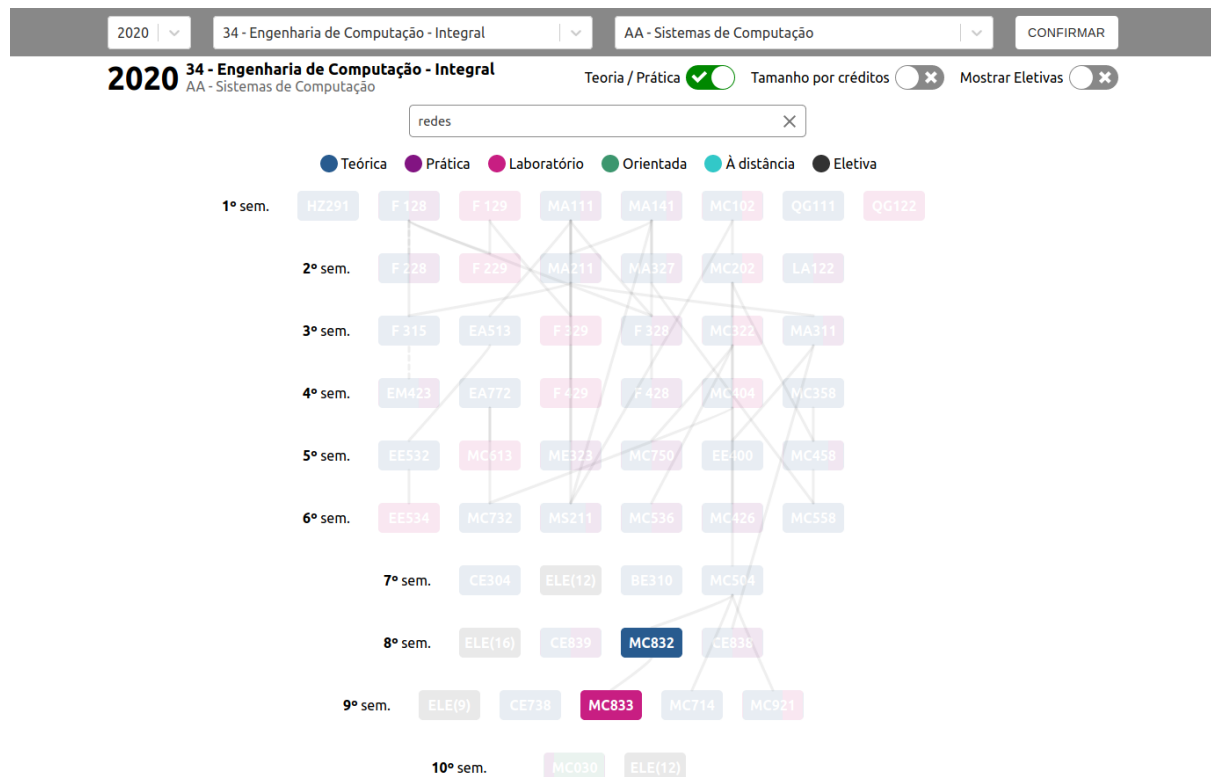


Fig. 16: Captura de tela da aplicação com a visualização ‘Teoria / Prática’ ligada e realizando busca da palavra ‘redes’.

6.5. Arrastar disciplinas dentro do mesmo semestre e para outros.

É possível arrastar as disciplinas e caso ela não tenha todos seus pré-requisitos cumpridos até o semestre em que está posicionada, uma exclamação aparece no canto direito superior da mesma. Ao clicar na exclamação é aberto um popup com a lista de pré-requisitos não cumpridos.

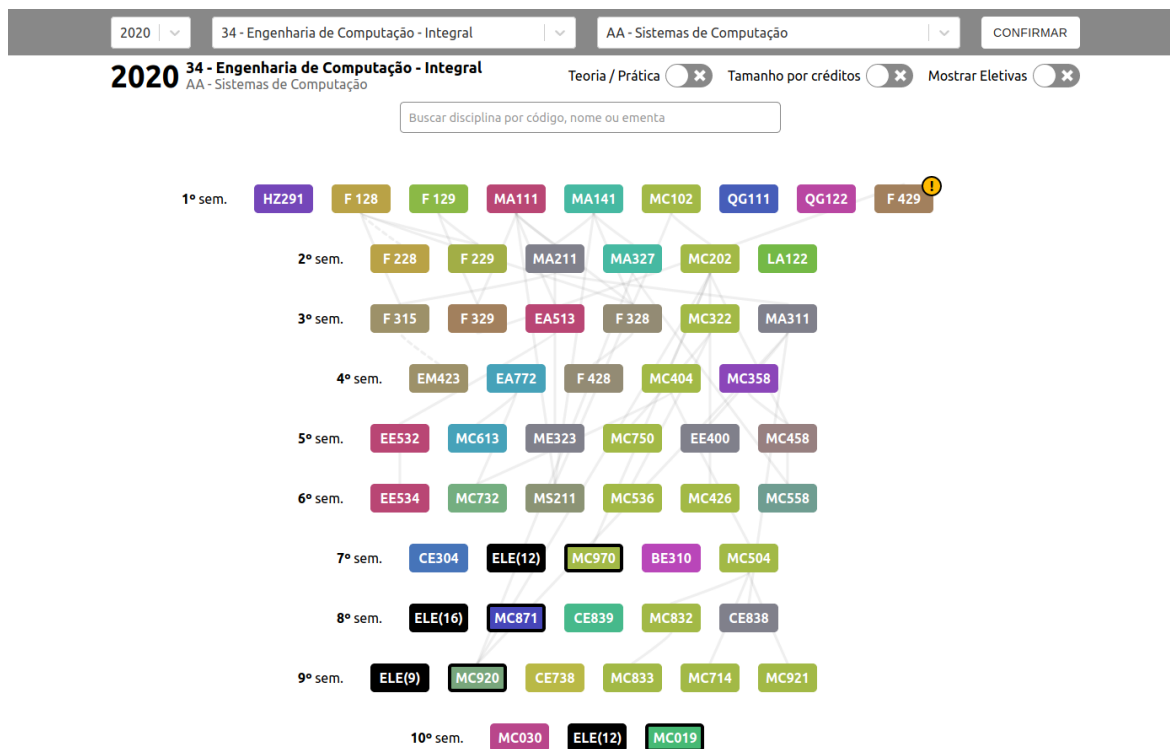


Fig. 17: Captura de tela da aplicação com catálogo de 2020, curso 34 e modalidade AA com disciplina 'F 429' movida para o semestre 1.

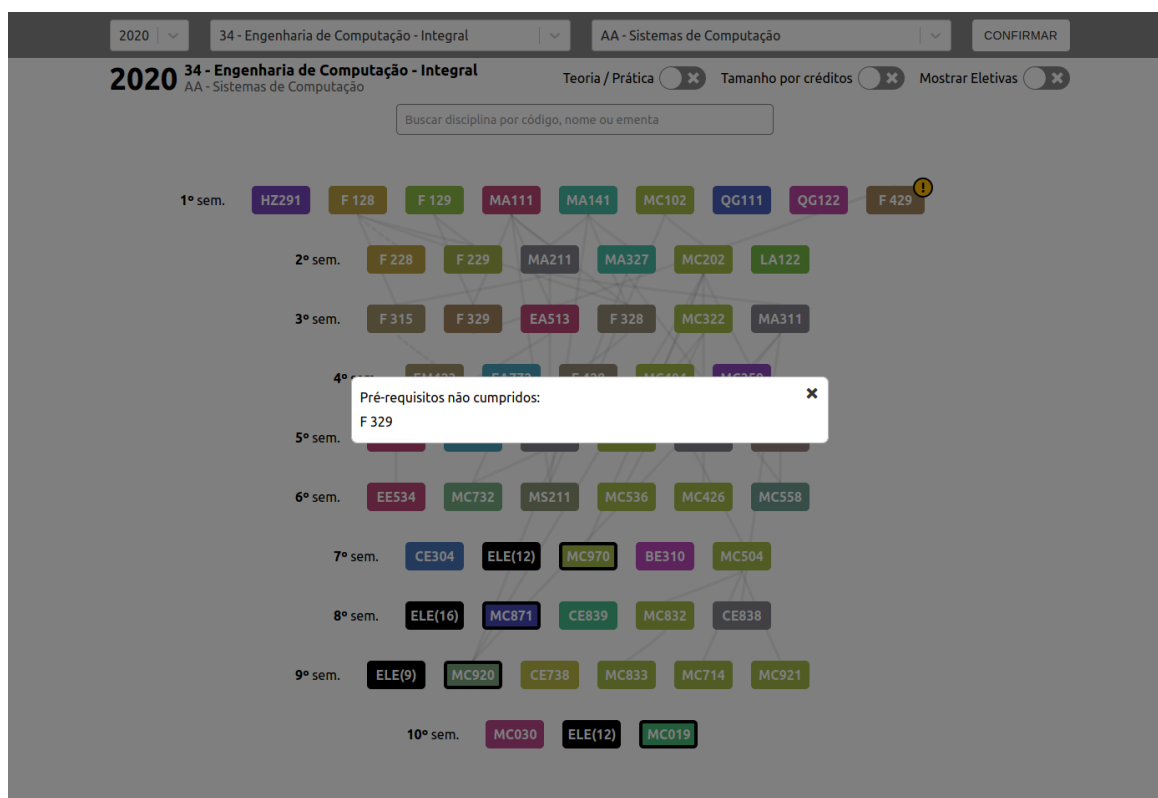


Fig. 18: Captura de tela da aplicação. Popup dos pré-requisitos não cumpridos pela disciplina 'F 429' após ser movida para o semestre 1.

6.6. Planejamento de eletivas

É possível visualizar os blocos de eletivas do curso ao clicar em 'Mostrar Eletivas' e arrastá-las para os semestres. As eletivas movidas para um semestre terão uma borda preta e podem ser removidas arrastando para o retângulo com texto 'REMOVER'.

2020 34 - Engenharia de Computação - Integral AA - Sistemas de Computação Teoria / Prática Tamanho por créditos Mostrar Eletivas

13 créditos dentre: Qualquer disciplina da UNICAMP

0/12 créditos dentre: MC019 MC020 MC032 MC033 MC040 MC041 MC050 MC051

0/12 créditos dentre: MC851 MC853 MC855 MC857 MC859 MC861 MC871 MC881

0/12 créditos dentre: MC346 MC521 MC621 MC646 MC658 MC721 MC821 MC886
MC896 MC906 MC907 MC919 MC920 MC937 MC940 MC949
MC950 MC959 MC970 MC971

REMOVER

1º sem. HZ291 F 128 F 129 MA111 MA141 MC102 QG111 QG122

2º sem. F 228 F 229 MA211 MA327 MC202 LA122

3º sem. F 315 EA513 F 329 F 328 MC322 MA311

4º sem. EM423 EA772 F 429 F 428 MC404 MC358

5º sem. EE532 MC613 ME323 MC750 EE400 MC458

6º sem. EE534 MC732 MS211 MC536 MC426 MC558

7º sem. CE304 ELE(12) BE310 MC504

8º sem. ELE(16) CE839 MC832 CE838

9º sem. ELE(9) CE738 MC833 MC714 MC921

10º sem. MC030 ELE(12)

Fig. 19: Captura de tela da aplicação. 'Mostrar Eletivas' ligado, tornando os blocos de eletivas do curso visíveis.

2020 34 - Engenharia de Computação - Integral AA - Sistemas de Computação Teoria / Prática ☒ Tamanho por créditos ☐ Mostrar Eletivas ☒

13 créditos dentre: Qualquer disciplina da UNICAMP

12/12 créditos dentre: MC019 MC020 MC032 MC033 MC040 MC041 MC050 MC051

12/12 créditos dentre: MC851 MC853 MC855 MC857 MC859 MC861 MC871 MC881

8/12 créditos dentre: MC346 MC521 MC621 MC646 MC658 MC721 MC821 MC886
MC896 MC906 MC907 MC919 MC920 MC937 MC940 MC949
MC950 MC959 MC970 MC971

REMOVER

1º sem. HZ291 F 128 F 129 MA111 MA141 MC102 QG111 QG122

2º sem. F 228 F 229 MA211 MA327 MC202 LA122

3º sem. F 315 EA513 F 329 F 328 MC322 MA311

4º sem. EM423 EA772 F 429 F 428 MC404 MC358

5º sem. EE532 MC613 ME323 MC750 EE400 MC458

6º sem. EE534 MC732 MS211 MC536 MC426 MC558

7º sem. CE304 ELE(12) MC970 BE310 MC504

8º sem. ELE(16) MC871 CE839 MC832 CE838

9º sem. ELE(9) MC920 CE738 MC833 MC714 MC921

10º sem. MC030 ELE(12) MC019

Fig. 20: Captura de tela da aplicação com ‘Teoria / Prática’ e ‘Mostrar Eletivas’ ligados e algumas eletivas movidas para os semestres.

7. Avaliações e/ou Experimentos Realizados

7.1. Conversa com a coordenadora do curso

Uma versão preliminar do GradView foi apresentada para a coordenadora do curso de Engenharia de Computação e novas funcionalidades foram sugeridas para

desenvolvimento. O sistema foi considerado visualmente agradável e não houveram críticas à sua usabilidade geral.

7.2. Conversa com a DAC

Uma versão mais avançada foi apresentada para os mantenedores do banco de dados da DAC. No geral eles gostaram do sistema e demonstraram interesse em disponibilizá-lo no próprio site da DAC. Também se dispuseram a ajudar com o fornecimento dos dados públicos da DAC de forma mais estruturada.

7.3. Apresentação no WTD 2019

Apresentamos o GradView quase finalizado como uma das *demos* do XIV Workshop de Teses, Dissertações e Trabalhos de Iniciação Científica do Instituto de Computação da Unicamp. Para tal evento, preparamos um formulário pedindo sugestões e outro formulário para nosso preenchimento, acompanhando o uso do sistema.

Diversas sugestões criativas foram recebidas durante o evento. Algumas delas já foram implementadas para esta versão do projeto. Outras coincidiram com features que já estavam previstas de serem implementadas, mas se mostraram inviáveis no tempo disponível. E o restante das sugestões foram avaliadas e adicionadas ao backlog do projeto, podendo ser implementadas futuramente.

7.3.1. Respostas à pergunta “O que poderia ser melhorado ou adicionado?”

Usuário 1

- Fazer aparecer um “hint” quando passa o mouse sobre disciplina, exibindo informações da disciplina (pelo menos o nome).
- Guardar estado e permitir voltar/desfazer alterações
- Poder Trocar ano/curso/modalidade sem ter que clicar em buscar
- Colocar legenda na exibição das horas teóricas/práticas

Usuário 2

- Alinhar itens da lista (cursos, modalidades) à esquerda
- Atualizar exibição sem necessidade de clicar em buscar

- Melhorar aparência do botão buscar
- Ícone na janela do site
- Botão para limpar campo de busca
- Colocar legenda na exibição das horas teóricas/práticas
- Reduzir saturação das cores

Usuário 3

- Ao invés de colocar linhas ligando as disciplinas aos pré-requisitos, usar bolinhas ou traços sobre as disciplinas com cores indicando de onde vêm
- Ver a árvore de dependência para baixo (o que depende de uma disciplina)
- Exibir uma mãozinha para evidenciar que as disciplinas podem ser arrastadas
- Tomar cuidado pois pode retirar a intuição de que as disciplinas podem ser clicadas
- Colocar as informações das disciplinas em um painel separado ou em um balão de informações
- A exclamação não está chamando atenção o suficiente
- Quando houver pré-requisito não cumprido deixar bem mais evidente, inclusive reforçando o traço que liga ao pré-requisito não atendido
- Na primeira vez que houver um warning, dar um warning mais chamativo pra ensinar o usuário que a exclamação é importante
- Colocar legenda na exibição das horas teóricas/práticas
- Colocar uma escala na visualização “tamanho por créditos” parecida com escala de mapa
- Criar uma Tag Cloud próxima à barra de busca
- Permitir usuário desfazer alterações
- Permitir visualização paralela do currículo original

Usuário 4

- Permitir que o usuário possa salvar o planejamento

Usuário 5

- Reduzir cruzamento das linhas
- Ver progresso em outros cursos

7.3.2. Uso observado do sistema, sem instruções

O usuário percebeu que:	1	2	3	4
Os campos ano/curso/modalidade são pesquisáveis				
Passar o mouse sobre disciplinas revela a cadeia de pré-reqs				
Existem linhas contínuas e tracejadas indicando tipo de pré-req				
As cores indicam relação entre as disciplinas				
Uma disciplina pode ser “clificada”				
Uma disciplina pode ser arrastada				
Uma exclamação eventualmente aparece ao arrastar disciplinas				
A exclamação pode ser “clificada”				

O usuário realizou as ações:	1	2	3	4
Trocou de ano / curso / modalidade				
Fez uma busca no campo de buscas				
Testou e compreendeu a opção “Teoria / Prática”				
Testou e compreendeu a opção “Tamanho por créditos”				

*O usuário 5 apenas observou o GradView.

Quanto ao uso do sistema, observamos que poucos usuários conhecem a fundo as normas da Unicamp com relação ao catálogo. Coisas como “pré-requisito parcial” e “vetor da disciplina” eram desconhecidas pela maioria dos usuários. A tradução visual

de tais elementos dos catálogos da DAC devem melhorar os conhecimentos dos usuários sobre os próprios cursos.

Poucos usuários perceberam que as disciplinas podem ser arrastadas e reorganizadas. Uma sugestão que recebemos foi transformar o cursor em uma mão quando posicionado sobre uma disciplina, fornecendo um *feedback* de que ela pode ser arrastada.

8. Conclusão

O sistema desenvolvido pode se tornar mais uma ferramenta para a comunidade interna e externa à Unicamp, modernizando o acesso à informação sobre os cursos.

O GradView foi projetado para ter acesso irrestrito e sem login, podendo ser consultado por qualquer pessoa que conheça seu endereço web. Ele torna explícitas algumas informações que antes eram desconhecidas pelos usuários dos catálogos em texto da DAC e essa visibilidade de informações irá melhorar a compreensão de todos os usuários dos catálogos.

Os dados armazenados pelo sistema estão no formato JSON, um formato textual simples. Isso deve permitir que qualquer pessoa possa alterar o conjunto de informações por simples comparação com os JSON originais do sistema. Através de uma cópia local dos arquivos, informações erradas podem ser corrigidas, atualizadas ou até mesmo uma nova re-implementação do sistema pode ser feita para outras universidades.

Uma aplicação muito interessante para a ferramenta é ser integrada ao próprio GDE, fazendo uso dos dados do próprio sistema e servindo com uma boa atualização para as árvores de integralização.

Um dos intuitos do projeto era tornar os catálogos da dac o mais acessível possível, e para isso é essencial que a aplicação também seja acessível para pessoas com deficiência. Algumas ideias para melhorar o projeto nesse aspecto foram: permitir a mudança das cores escolhidas para representar o vetor da disciplina por meio de um color picker aberto ao clicar no círculo colorido na legenda e permitir configurar o tamanho das disciplinas.

Algumas ideias que visam melhorar a usabilidade e poderiam ser desenvolvidas em trabalhos futuros são:

- Tornar o catálogo mais acessível para pessoas com deficiência.
- Contabilizar os créditos por semestre e verificar se estão dentro do limite do curso.
- Quando existir algo no campo de busca, ao abrir as informações de uma disciplina 'grifar' o trecho encontrado na busca.
- Tratar casos em que os requisitos possuem mais de uma possibilidade de cumprimento (A ou B ou ..)
- Abrir popup avisando que os dados alterados serão perdidos ao trocar de catálogo ou fechar o site.
- Salvar um histórico de edição das disciplinas, permitindo desfazer uma alteração.
- Criar um tutorial guiado para ensinar como usar a aplicação.
- Permitir importar as matérias cursadas ao copiar a integralização ou marcar disciplinas como cursadas.
- Tornar a aplicação mais responsiva, principalmente em celulares.

Referências

- [1] GDE - <https://grade.daconline.unicamp.br/>
- [2] DAC - Informações Gerais sobre Disciplinas dos Cursos de Graduação
<https://www.dac.unicamp.br/sistemas/catalogos/grad/catalogo2020/legenda.html>
- [3] GitLab do Instituto de Computação da Unicamp - Projeto GradView. Acesso restrito para usuários do sistema.
<https://gitlab.ic.unicamp.br/ra176675/mc030-gradview>
- [4] OpenDesign - A Collaborative Platform for Community-driven Design
<http://demo.opendesign.ml:3000/>
- [5] GDE - Repositório do GitHub
<https://github.com/guaycuru/gde>
- [6] Material-UI
<https://material-ui.com/pt/>
- [7] react-beautiful-dnd
<https://www.npmjs.com/package/react-beautiful-dnd>
- [8] react-debounce-input
<https://www.npmjs.com/package/react-debounce-input>
- [9] reactjs-popup
<https://www.npmjs.com/package/reactjs-popup>
- [10] react-switch
<https://www.npmjs.com/package/react-switch>
- [11] react-select
<https://www.npmjs.com/package/react-select>
- [12] randomcolor
<https://www.npmjs.com/package/randomcolor>
- [13] react-curveto
<https://www.npmjs.com/package/react-curveto>