UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CAROLINE FERREIRA DOS SANTOS

ORGANIZAÇÃO E VISUALIZAÇÃO DE CONCEITOS E EXERCÍCIOS DE ALGORITMOS POR MEIO DE GRAFOS

GUARAPUAVA

2026

CAROLINE FERREIRA DOS SANTOS

ORGANIZAÇÃO E VISUALIZAÇÃO DE CONCEITOS E EXERCÍCIOS DE ALGORITMOS POR MEIO DE GRAFOS

Organization and visualization of algorithm concepts and exercises using graphs

Projeto de Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Profª. Drª Renata Luiza Stange

Coorientador: Prof. Dr. Diego Marczal

GUARAPUAVA

2026

4.0 Internacional

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

RESUMO

A aprendizagem introdutória de programação envolve a compreensão de diferentes conceitos que podem apresentar relações de pré-requisito entre si. Nesse contexto, explicitar essas relações pode contribuir para a organização do ensino de algoritmos e para a identificação dos conhecimentos envolvidos em cada atividade. Este trabalho tem como objetivo implementar, na aplicação web Blade, uma funcionalidade para organizar, relacionar e visualizar conceitos e exercícios relacionados ao ensino introdutório de algoritmos por meio de um grafo direcionado. Como referência, adota-se o grafo genético, formalismo de representação do conhecimento utilizado na modelagem do domínio de Programação de Computadores, adaptado de modo que os nós representem conceitos e as arestas representem relações de pré-requisito. Para isso, será realizado um levantamento de conceitos e exercícios em materiais existentes, como livros e trabalhos acadêmicos, seguido do estabelecimento das relações de pré-requisito entre os conceitos e da associação de cada exercício aos conceitos necessários para sua resolução. A funcionalidade permitirá que professores e administradores cadastrem conceitos, relações e exercícios, e que todos os usuários visualizem o grafo, consultem os exercícios associados a cada conceito e identifiquem os conceitos relacionados a determinado exercício. A implementação será verificada por meio de testes automatizados e manuais baseados nos requisitos estabelecidos. Espera-se que a funcionalidade contribua para a organização e a visualização dos conhecimentos relacionados ao ensino introdutório de algoritmos.

Palavras-chave: ensino de algoritmos; grafo direcionado; grafo genético; pré-requisitos; aplicação web.

ABSTRACT

Learning to program at an introductory level involves understanding various concepts that may have prerequisite relationships with one another. In this context, making these relationships explicit can help organize the teaching of algorithms and identify the knowledge required for specific activities. This project aims to implement a feature within the Blade web application to organize, interlink, and visualize concepts and exercises related to introductory algorithm instruction, using a directed graph. The approach is based on the "genetic graph"—a knowledge representation formalism used in modeling the domain of computer programming—adapted so that nodes represent concepts and edges represent prerequisite relationships. To achieve this, concepts and exercises will be identified from existing materials, such as textbooks and academic literature; subsequently, prerequisite relationships between concepts will be established, and each exercise will be linked to the concepts necessary for its completion. The feature will allow instructors and administrators to register concepts, relationships, and exercises, while enabling all users to view the graph, explore exercises associated with specific concepts, and identify the concepts linked to a given exercise. The implementation will be verified through automated and manual tests based on the established requirements. It is expected that this functionality will contribute to the organization and visualization of knowledge relevant to the introductory teaching of algorithms.

Keywords: algorithm teaching; directed graph; genetic graph; prerequisites; web application.

SUMÁRIO

1	**INTRODUÇÃO**	**4**
1.1	**Objetivos**	**5**
1.1.1	Objetivo geral	5
1.1.2	Objetivos específicos	5
2	**MATERIAIS E MÉTODOS**	**7**
2.1	**Materiais**	**7**
2.2	**Uso de inteligência artificial generativa**	**7**
2.3	**Métodos**	**8**
3	**ANÁLISE E PROJETO**	**10**
3.1	**Escopo do Sistema**	**10**
3.2	**Modelo de Aquisição de Conhecimento**	**10**
3.2.1	Conceitos	11
3.2.2	Relações entre os Conceitos	11
3.2.3	Exercícios e Resultados	12
3.2.4	Estados dos Conceitos	13
3.3	**Levantamento de Requisitos**	**13**
3.3.1	Requisitos Funcionais	13
3.3.2	Requisitos Não Funcionais	14
3.4	**Diagrama de Atividades**	**15**
4	**CONSIDERAÇÕES FINAIS**	**17**
	REFERÊNCIAS	**18**

1 INTRODUÇÃO

A aprendizagem introdutória de programação envolve a compreensão de diferentes conceitos necessários à resolução de problemas. Nesse contexto, os estudantes podem apresentar dificuldades tanto na compreensão desses conceitos quanto em sua aplicação, o que pode comprometer o desenvolvimento da aprendizagem ao longo da formação acadêmica. Essas dificuldades tornam o ensino e a aprendizagem de programação um desafio, especialmente nos primeiros contatos com disciplinas relacionadas a algoritmos e programação (MORAIS; NETO; OSÓRIO, 2020).

No ensino de algoritmos, os estudantes entram em contato com conceitos como variáveis, estruturas condicionais, estruturas de repetição e funções, que podem ser utilizados de forma combinada na resolução de problemas (SOUZA *et al.*, 2019). Esses conceitos não são necessariamente independentes, pois a compreensão de determinados conhecimentos pode ser necessária para a utilização de outros. Por exemplo, o uso de estruturas condicionais e de repetição pode envolver a compreensão de variáveis. Essas relações evidenciam a importância de organizar os conceitos e suas dependências, de modo a explicitar como os conhecimentos se relacionam no contexto da aprendizagem de algoritmos.

Na tese de Krynski (2013), o grafo genético é utilizado para representar o conhecimento do domínio de Programação de Computadores por meio de vértices e arestas. Os vértices correspondem às perícias, isto é, ao conhecimento procedimental do domínio, relacionado à aplicação dos conhecimentos na realização de tarefas e na resolução de problemas. Os princípios, por sua vez, correspondem ao conhecimento declarativo que explica e justifica essas perícias. Segundo o autor, o conceito de variável consiste em um princípio, enquanto o reconhecimento de variáveis na análise e na resolução do problema, a determinação dos tipos de dados correspondentes e a distinção entre variáveis e constantes constituem perícias. Desse modo, enquanto o princípio expressa o que se sabe sobre um conhecimento, a perícia expressa a capacidade de utilizá-lo, como na troca de valores entre duas variáveis com o uso de uma variável auxiliar.

No grafo genético, as arestas representam relações entre esses conhecimentos, como generalização, analogia, refinamento, desvio e pré-requisito. Na relação de pré-requisito, um conhecimento é pré-requisito de outro quando seu aprendizado anterior é necessário à compreensão e ao desenvolvimento do segundo, o que permite explicitar quais conhecimentos devem ser explorados antes de outros no processo de ensino.

Neste trabalho, esse conjunto de conhecimentos será utilizado como base, revisado e adaptado ao contexto do ensino introdutório de algoritmos com o apoio de Souza *et al.* (2019). Como adaptação, os vértices do grafo serão tratados como conceitos do domínio, como variável, atribuição e estrutura de repetição, e as arestas representarão as relações de pré-requisito entre eles. Além disso, serão selecionados exercícios de fontes bibliográficas relacionadas ao

ensino de algoritmos, e cada exercício será associado a um ou mais conceitos, considerando os conhecimentos necessários para sua resolução.

Este trabalho tem como objetivo implementar uma funcionalidade na aplicação web Blade[1], voltada ao apoio ao ensino introdutório de algoritmos. Enquanto a aplicação contempla recursos para a construção e a simulação da execução de algoritmos, este trabalho se concentra na organização e na visualização dos conceitos e exercícios, bem como na representação das relações entre os conceitos.

Na funcionalidade desenvolvida, os conceitos e suas relações serão apresentados por meio de um grafo direcionado. A interface permitirá que o usuário navegue entre os conceitos, consulte os exercícios associados a cada um deles e identifique os conceitos relacionados a determinado exercício. Dessa maneira, a funcionalidade busca facilitar a visualização das relações entre os conhecimentos e a consulta aos exercícios relacionados ao ensino introdutório de algoritmos.

1.1 Objetivos

Nesta seção são apresentados o objetivo geral e os objetivos específicos deste trabalho, que orientam o desenvolvimento da proposta e delimitam as etapas necessárias para sua realização.

1.1.1 Objetivo geral

Implementar, na aplicação web Blade, uma funcionalidade para organizar, relacionar e visualizar conceitos e exercícios relacionados ao ensino introdutório de algoritmos, a partir do levantamento e da organização de materiais existentes, representando por meio de um grafo direcionado as relações de pré-requisito entre os conceitos.

1.1.2 Objetivos específicos

- Levantar e organizar conceitos relacionados ao ensino introdutório de algoritmos, a partir de Krynski (2013) e de Souza *et al.* (2019).
- Estabelecer as relações de pré-requisito entre os conceitos identificados.
- Selecionar exercícios de fontes bibliográficas e associá-los aos conceitos necessários para sua resolução.
- Implementar, na aplicação web, o gerenciamento dos conceitos, dos exercícios e das relações estabelecidas entre eles.

[1] Disponível em https://myblade.dev/.

- Implementar a visualização dos conceitos e de suas relações por meio de um grafo direcionado, com consulta aos exercícios associados.

2 MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e os métodos utilizados no desenvolvimento de uma funcionalidade da aplicação web Blade, voltada ao apoio ao ensino introdutório de algoritmos. Os materiais compreendem as tecnologias e ferramentas empregadas no desenvolvimento, enquanto os métodos descrevem as etapas necessárias para alcançar os objetivos propostos.

2.1 Materiais

Os materiais utilizados neste trabalho compreendem as tecnologias, ferramentas, linguagens e bibliotecas empregadas no desenvolvimento da funcionalidade. A seleção desses recursos considera as necessidades de armazenamento e recuperação dos dados, de representação das relações entre os conceitos e de visualização do grafo direcionado na interface da aplicação.

O desenvolvimento utilizará as tecnologias já adotadas na aplicação Blade, desenvolvida com o *framework* Ruby on Rails. A aplicação segue uma arquitetura monolítica, com renderização de páginas no servidor (*Server-Side Rendering* – SSR), utilizando Hotwire, especialmente Turbo e Stimulus, para proporcionar interatividade e atualizações dinâmicas na interface.

Embora React, TypeScript e React Flow já sejam utilizados na página de construção visual (*builder*), essas tecnologias não serão empregadas na funcionalidade proposta. A representação e a interação com o grafo direcionado serão implementadas utilizando Hotwire, mantendo a abordagem tecnológica adotada nas demais páginas da aplicação.

O PostgreSQL é utilizado para a persistência dos dados. O controle de versões do código-fonte é realizado com Git, com o repositório hospedado no GitHub. O Docker é empregado para a execução e a padronização do ambiente de desenvolvimento, enquanto a implantação (*deploy*) da aplicação é realizada por meio do Kamal, ferramenta que automatiza a implantação e o gerenciamento de aplicações em contêineres.

2.2 Uso de inteligência artificial generativa

A definição do tema, do escopo, dos objetivos e do projeto da funcionalidade, bem como a leitura das referências e a incorporação das orientações recebidas, foram realizadas pela autora, em conjunto com seus orientadores. Como apoio a esse processo, foi utilizada a ferramenta de inteligência artificial generativa Claude (ANTHROPIC, 2026), principalmente na revisão da redação e na adequação do texto às normas da ABNT. A ferramenta também auxiliou na verificação da consistência entre objetivos, métodos e requisitos, na localização de trechos da tese de referência e na reelaboração do diagrama de atividades a partir de um esboço da autora.

Todo o conteúdo produzido com apoio da ferramenta foi revisado pela autora, e as informações atribuídas às referências foram conferidas nas fontes originais.

2.3 Métodos

O desenvolvimento será organizado em seis etapas, definidas a partir dos objetivos específicos apresentados na Seção 1.1.2. Os conceitos e exercícios serão definidos com base nas referências bibliográficas selecionadas para o trabalho. Não estão previstas a elaboração de novos exercícios nem a realização de testes com estudantes. As etapas são descritas a seguir.

1. **Levantamento e organização dos conceitos:** serão identificados e organizados os conceitos relacionados ao ensino introdutório de algoritmos, tomando como base o subconjunto do domínio de Programação de Computadores modelado por Krynski (2013, p. 113) e os conteúdos abordados por Souza *et al.* (2019). Essa etapa terá como finalidade definir os conceitos que serão representados como nós no grafo.

2. **Definição das relações de pré-requisito entre os conceitos:** a partir dos conceitos identificados, serão definidas as relações de pré-requisito entre eles, considerando a relação de pré-requisito apresentada no Capítulo 1. Essas relações serão representadas por arestas direcionadas, de modo que seja possível identificar, para cada conceito, quais conhecimentos devem ser explorados antes dele.

3. **Seleção e associação dos exercícios:** serão selecionados exercícios que envolvam os conceitos definidos na primeira etapa, priorizando enunciados que exercitem poucos conceitos por vez, de modo a facilitar a identificação dos conhecimentos necessários para sua resolução. Para cada exercício, serão analisados o enunciado e a solução esperada, a fim de identificar os conceitos utilizados, e o exercício será associado a esses conceitos. Busca-se que cada conceito representado no grafo possua ao menos um exercício associado.

4. **Modelagem e armazenamento dos dados:** serão definidas as estruturas necessárias para armazenar os conceitos, as relações de pré-requisito entre eles e os exercícios associados, de forma que essas informações possam ser recuperadas pela aplicação para consulta e visualização.

5. **Implementação do gerenciamento e da visualização do grafo:** serão implementadas diretamente na aplicação Blade, utilizando Hotwire, as operações de cadastro e edição de conceitos, relações e exercícios, bem como a visualização do grafo direcionado e a consulta aos exercícios associados a cada conceito e aos conceitos associados a cada exercício, incluindo a implementação dos perfis de professor e administrador, aos quais serão restritas as operações de gerenciamento.

6. **Testes e validação:** as funcionalidades serão testadas por meio de testes automatizados, utilizando o Minitest, *framework* de testes padrão do Ruby on Rails já adotado no Blade, abrangendo os modelos de dados e as regras de acesso, além de testes de sistema com Capybara e Selenium para os fluxos de interface. Para cada requisito funcional especificado no Capítulo 3, será definido ao menos um cenário de teste; por exemplo, para o RF06, será verificado se uma relação de pré-requisito cadastrada é exibida corretamente no grafo e se usuários sem perfil de professor ou administrador são impedidos de cadastrá-la. Os requisitos não funcionais serão verificados por meio de testes manuais nos navegadores e dispositivos definidos. Por fim, a organização dos conceitos, das relações de pré-requisito e das associações com os exercícios será validada junto aos orientadores, a fim de verificar sua coerência com as referências adotadas.

3 ANÁLISE E PROJETO

Este capítulo apresenta a análise e o projeto da funcionalidade a ser desenvolvida na aplicação Blade. Inicialmente, são delimitados o escopo e as funcionalidades previstas para a implementação. Em seguida, são apresentados os elementos do modelo de aquisição de conhecimento de Krynski (2013) que fundamentam a organização dos conceitos, suas relações e os exercícios associados. Por fim, são especificados os requisitos funcionais e não funcionais, além dos fluxos de utilização da funcionalidade por meio de um diagrama de atividades.

3.1 Escopo do Sistema

Este trabalho contempla o desenvolvimento de uma funcionalidade na aplicação web Blade, voltada ao apoio ao ensino introdutório de algoritmos. A proposta consiste em organizar conceitos relacionados a esse domínio e representar suas relações por meio de um grafo direcionado. Para isso, serão identificados conceitos e selecionados exercícios a partir de fontes bibliográficas, considerando sua pertinência ao escopo do trabalho.

A funcionalidade permitirá cadastrar e editar conceitos, estabelecer relações de pré-requisito entre eles e cadastrar exercícios associados aos conceitos abordados. No grafo direcionado, os conceitos serão representados por nós, enquanto as relações entre eles serão representadas por arestas. Além disso, será possível consultar os exercícios associados a determinado conceito e identificar os conceitos relacionados a um exercício.

A consulta ao grafo e aos exercícios estará disponível a todos os usuários da aplicação. As operações de cadastro e edição de conceitos, de estabelecimento de relações e de cadastro e associação de exercícios serão restritas aos perfis de professor e administrador, que serão implementados neste trabalho. Não fazem parte do escopo a resolução de exercícios pelos estudantes na aplicação, a avaliação de respostas e a atualização dos estados de aprendizagem previstos no modelo de Krynski (2013).

3.2 Modelo de Aquisição de Conhecimento

A organização dos conhecimentos abordados neste trabalho terá como referência o modelo de aquisição de conhecimento apresentado por Krynski (2013). Nesse modelo, o grafo genético é utilizado para representar os elementos do conhecimento de um domínio e as relações entre eles. Neste trabalho, o modelo será utilizado como base para selecionar e organizar os conceitos relacionados ao ensino introdutório de algoritmos e representar suas relações por meio de um grafo direcionado.

3.2.1 Conceitos

Conforme apresentado no Capítulo 1, no modelo adotado os vértices do grafo genético representam perícias, enquanto os princípios correspondem ao conhecimento declarativo que as fundamenta. Neste trabalho, como decisão de implementação, os nós do grafo serão denominados conceitos e corresponderão aos conhecimentos do domínio de algoritmos, como variável, atribuição e estrutura de repetição. Essa adaptação busca simplificar a representação e a associação dos conhecimentos com os exercícios na aplicação.

A seleção dos conceitos será realizada a partir da análise do subconjunto do domínio de Programação de Computadores modelado por Krynski (2013, p. 113) e dos conteúdos apresentados por Souza *et al.* (2019). Entre os conteúdos a serem analisados estão algoritmos, tipos de dados, variáveis, atribuição, expressões, estruturas condicionais, estruturas de repetição, funções e atividades relacionadas à verificação e ao teste de algoritmos. A seleção final dependerá da análise das fontes consultadas e da pertinência dos conteúdos ao escopo definido.

Após a seleção, os conceitos identificados serão representados por nós no grafo direcionado. Essa representação permitirá visualizar os conceitos que compõem a estrutura e as relações estabelecidas entre eles.

3.2.2 Relações entre os Conceitos

As relações entre os conceitos serão estabelecidas a partir da relação de pré-requisito apresentada no Capítulo 1. No grafo direcionado, os nós representarão os conceitos, e cada aresta partirá do conceito que é pré-requisito em direção ao conceito que depende dele.

A Figura 1 apresenta uma representação gráfica dos conhecimentos no protótipo Dandelion, relacionado à abordagem de sequenciamento de enunciados de programação descrita por Oliveira, Pimentel e Maschio (2016). Essa representação serve como referência visual para a organização dos conceitos e de suas relações na aplicação proposta.

Na Figura 1, cada nó representa um conhecimento do domínio de programação, como variáveis, atribuição, entrada e saída, e cada aresta é identificada por uma letra que indica o tipo de relação: P para pré-requisito, G para generalização, A para analogia e R para refinamento. Por exemplo, a aresta P que parte de variáveis em direção a instruções simples indica que a compreensão de variáveis é necessária antes da exploração das instruções simples, uma vez que essas instruções manipulam valores armazenados em variáveis. Já as arestas G que partem de entrada, saída e atribuição indicam que instruções simples generaliza esses três conhecimentos, enquanto as arestas A entre eles indicam que são análogos entre si. Por fim, as arestas R que partem de atribuição indicam que troca de valores e compatibilidade de tipos são refinamentos desse conhecimento.

Embora o grafo genético preveja diferentes tipos de relação, neste trabalho será representada apenas a relação de pré-requisito, por ser aquela que expressa a ordem em que os con-

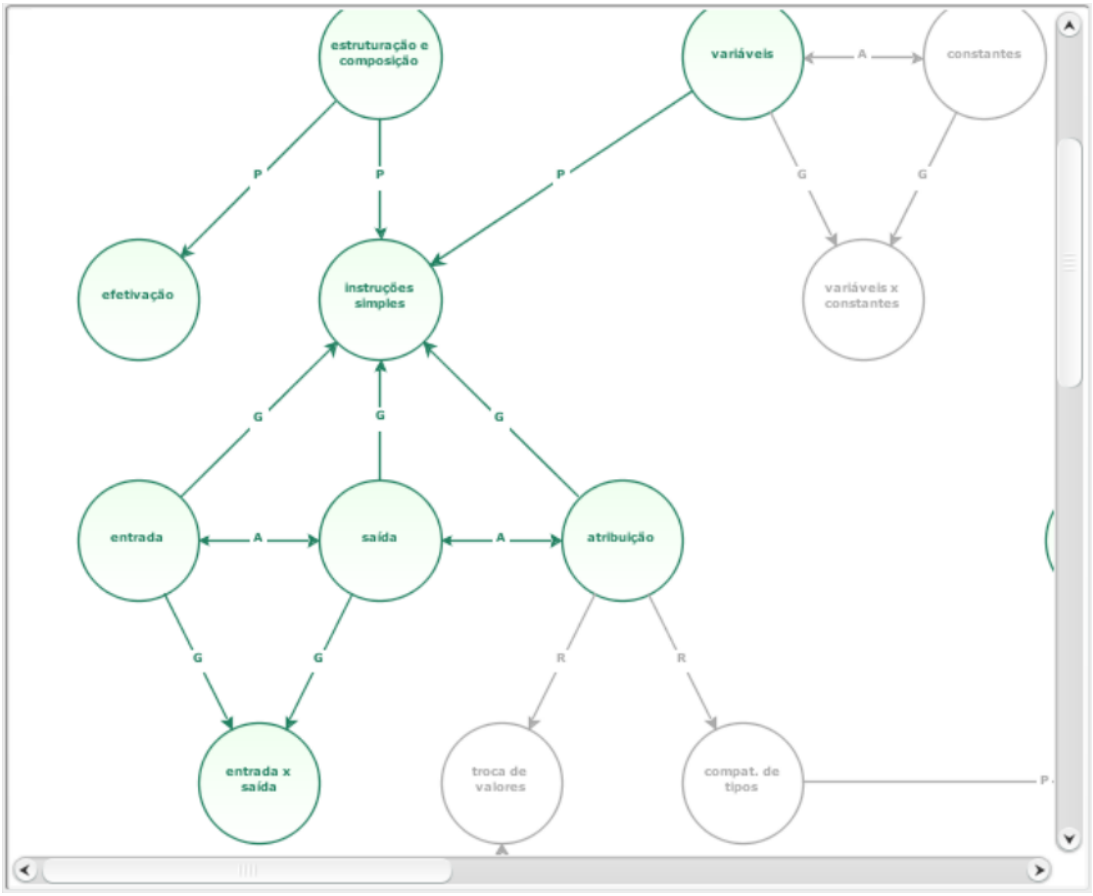


Figura 1 – Representação gráfica dos conhecimentos no protótipo Dandelion
Fonte: Oliveira, Pimentel e Maschio (2016).

ceitos devem ser explorados. Dessa forma, no grafo da aplicação proposta, uma aresta como a que liga variáveis a instruções simples indicará que o primeiro conceito deve ser compreendido antes do segundo, e, ao selecionar um conceito, o usuário poderá consultar os exercícios a ele associados.

3.2.3 Exercícios e Resultados

Os exercícios serão selecionados a partir do catálogo de enunciados apresentado por Krynski (2013, p. 183) e do livro de Souza *et al.* (2019), que servirão como base para a organização das atividades relacionadas aos conceitos de algoritmos. A seleção considerará os conhecimentos necessários para a resolução de cada atividade, de modo que os exercícios possam ser associados aos conceitos correspondentes.

Na aplicação proposta, cada exercício poderá ser associado a um ou mais conceitos representados no grafo. Essa associação permitirá consultar os exercícios relacionados a determinado conceito e identificar os conceitos abordados em cada exercício. Por exemplo, um exercício que solicite a troca dos valores de duas variáveis com o uso de uma variável auxiliar seria associado aos conceitos de variável e atribuição. Dessa forma, os exercícios serão or-

ganizados de acordo com os conhecimentos envolvidos em sua resolução, complementando a representação gráfica dos conceitos e de suas relações.

3.2.4 Estados dos Conceitos

A representação do conhecimento do aprendiz pode ser atualizada a partir das evidências obtidas durante a realização dos exercícios. Os resultados das atividades podem ser utilizados para avaliar as perícias relacionadas aos conhecimentos do domínio e determinar seus estados, conforme os critérios estabelecidos no modelo de aquisição de conhecimento.

Neste trabalho, a avaliação das respostas dos estudantes e a atualização dos estados de aprendizagem não serão implementadas. A aplicação ficará restrita ao cadastro e à apresentação dos conceitos, à representação das relações entre eles e à associação e consulta dos exercícios. Dessa forma, os estados serão abordados apenas para contextualizar os elementos teóricos que não fazem parte do escopo da implementação.

3.3 Levantamento de Requisitos

A definição dos requisitos considera o escopo estabelecido para a aplicação, as funcionalidades necessárias à organização dos conceitos e exercícios e as relações representadas no grafo. Os requisitos funcionais descrevem as operações que a aplicação deverá disponibilizar, enquanto os requisitos não funcionais estabelecem características relacionadas à sua utilização e ao seu funcionamento. A especificação também considera os diferentes perfis de usuários que poderão interagir com a aplicação, conforme os fluxos de utilização previstos.

3.3.1 Requisitos Funcionais

Os requisitos funcionais descrevem os comportamentos que a funcionalidade deverá oferecer. Eles foram organizados de acordo com as permissões de acesso dos usuários, divididos em requisitos de consulta e requisitos de gerenciamento.

Os requisitos de consulta, apresentados na Tabela 1, descrevem as operações disponíveis a todos os usuários da aplicação, independentemente do perfil de acesso.

Tabela 1 – Requisitos funcionais de consulta

Requisito	Descrição
RF01	O sistema deve permitir que os usuários visualizem o grafo direcionado dos conceitos de algoritmos e das relações de pré-requisito estabelecidas entre eles.
RF02	O sistema deve permitir que os usuários consultem os exercícios associados a um conceito selecionado no grafo.
RF03	O sistema deve permitir que os usuários consultem os conceitos associados a um exercício selecionado.

Fonte: Autoria própria (2026).

Os requisitos de gerenciamento, apresentados na Tabela 2, descrevem as operações restritas aos perfis de professor e administrador.

Tabela 2 – Requisitos funcionais de gerenciamento

Requisito	Descrição
RF04	O sistema deve permitir que professores e administradores cadastrem conceitos de algoritmos, informando nome e descrição.
RF05	O sistema deve permitir que professores e administradores editem o nome e a descrição de conceitos previamente cadastrados.
RF06	O sistema deve permitir que professores e administradores estabeleçam relações de pré-requisito entre conceitos cadastrados.
RF07	O sistema deve permitir que professores e administradores cadastrem exercícios, informando seu enunciado.
RF08	O sistema deve permitir que professores e administradores associem cada exercício a um ou mais conceitos cadastrados.
RF09	O sistema deve permitir que administradores atribuam o perfil de professor ou de administrador a usuários cadastrados.

Fonte: Autoria própria (2026).

3.3.2 Requisitos Não Funcionais

Os requisitos não funcionais estabelecem critérios relacionados à qualidade e às restrições de funcionamento da funcionalidade. Nesta proposta, são considerados aspectos da representação gráfica, da consistência das informações e da compatibilidade com os navegadores utilizados para acessar a aplicação Blade, conforme apresentado na Tabela 3.

Tabela 3 – Requisitos não funcionais

Requisito	Descrição
RNF01	A representação gráfica deverá exibir os nomes dos conceitos e utilizar arestas direcionadas que permitam identificar visualmente a origem e o destino de cada relação.
RNF02	Os conceitos, as relações e os exercícios apresentados nas consultas devem corresponder aos dados cadastrados na aplicação.
RNF03	A funcionalidade deve ser compatível com os navegadores aceitos pela aplicação Blade, que restringe o acesso a versões modernas: Google Chrome 120 ou superior, Mozilla Firefox 121 ou superior, Safari 17.2 ou superior e Opera 106 ou superior.
RNF04	A interface deve ser utilizável em computadores (*desktop* e *notebook*), com resolução mínima de 1366 $\times$ 768 *pixels*, e em *tablets*, com largura mínima de 768 *pixels*.

Fonte: Autoria própria (2026).

3.4 Diagrama de Atividades

O diagrama de atividades apresentado na Figura 2 representa o fluxo de consulta ao grafo de conceitos e aos exercícios relacionados na plataforma Blade. São ilustradas as interações entre o usuário e o sistema, desde o acesso à página do grafo de conceitos até a visualização dos exercícios associados a um conceito e, opcionalmente, dos conceitos relacionados a um exercício. Esse fluxo está disponível a todos os usuários da plataforma, independentemente de seu perfil de acesso.

O diagrama está dividido em duas raias, que separam as ações realizadas pelo usuário daquelas executadas pelo sistema. Ao acessar a página do grafo de conceitos, o sistema apresenta o grafo de conceitos. Em seguida, o usuário seleciona um conceito, e o sistema apresenta os exercícios associados a ele. Nesse ponto, o usuário pode encerrar a consulta ou selecionar um dos exercícios apresentados; nesse último caso, o sistema exibe os conceitos relacionados ao exercício selecionado, encerrando o fluxo.

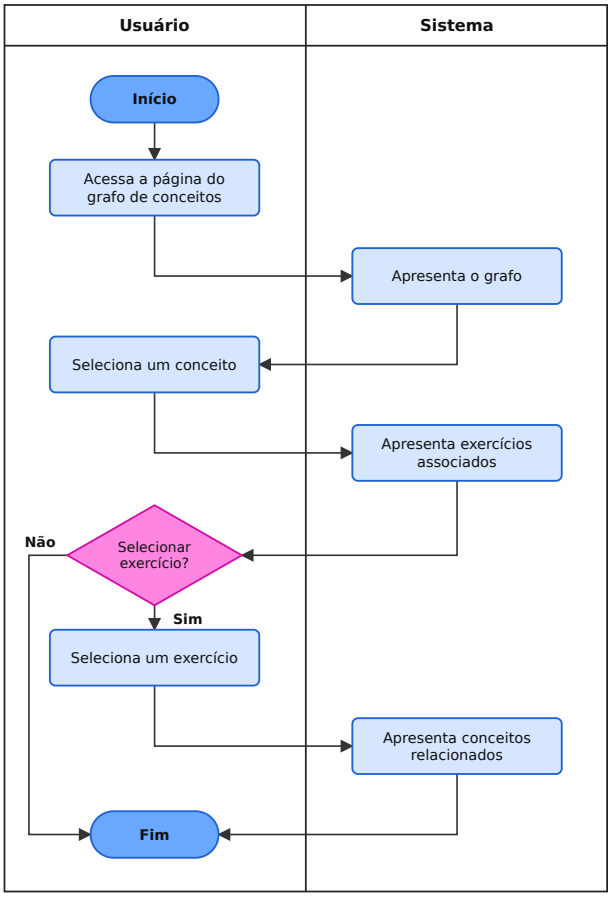


Figura 2 – Diagrama de atividades do fluxo de consulta ao grafo de conceitos e exercícios

Fonte: Autoria própria (2026).

4 CONSIDERAÇÕES FINAIS

Esta proposta apresentou o planejamento de uma funcionalidade para a aplicação web Blade, voltada ao apoio ao ensino introdutório de algoritmos. O objetivo é organizar, relacionar e visualizar conceitos e exercícios desse domínio, representando por meio de um grafo direcionado as relações de pré-requisito entre os conceitos. Para isso, foram definidas etapas que abrangem o levantamento dos conceitos a partir de Krynski (2013) e Souza *et al.* (2019), a definição das relações entre eles, a seleção e a associação dos exercícios, a modelagem dos dados, a implementação do gerenciamento e da visualização do grafo na aplicação e os testes e a validação das funcionalidades.

A relevância do trabalho está em explicitar como os conhecimentos de algoritmos se relacionam. Conforme discutido, conceitos como variáveis, atribuição e estruturas de repetição não são independentes, e a compreensão de uns é necessária para a exploração de outros. Ao tornar essas relações visíveis e associá-las a exercícios, a funcionalidade pode auxiliar professores na organização dos conteúdos e estudantes na identificação dos conhecimentos envolvidos em cada atividade, tomando como referência o modelo de aquisição de conhecimento adotado.

Como resultado, espera-se obter uma funcionalidade integrada à aplicação Blade que permita a professores e administradores cadastrar conceitos, relações e exercícios, e que permita a todos os usuários consultar o grafo, os exercícios associados a cada conceito e os conceitos relacionados a cada exercício.

Entre as limitações, destaca-se que a representação adotada simplifica o modelo de referência: os vértices do grafo correspondem a conceitos, e não a perícias, e apenas a relação de pré-requisito é representada. Além disso, a avaliação das respostas dos estudantes e a atualização dos estados de aprendizagem não fazem parte do escopo, assim como a elaboração de novos exercícios e a realização de testes com estudantes, de modo que a verificação ficará restrita ao atendimento dos requisitos especificados. Entre os desafios esperados estão a definição de critérios consistentes para estabelecer as relações de pré-requisito, uma vez que a modelagem de um domínio é sensível à perspectiva de quem a realiza (KRYNSKI, 2013, p. 114), e a apresentação legível do grafo à medida que o número de conceitos aumenta.

REFERÊNCIAS

ANTHROPIC. **Claude**. 2026. Modelo Claude Opus 5.5. Ferramenta de inteligência artificial generativa. Disponível em: https://claude.ai. Acesso em: 1 nov. 2026.

KRYNSKI, E. M. **Modelagem do processo de aquisição de conhecimento apoiado por ambientes inteligentes**. 2013. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, 2013.

MORAIS, C. G. B.; NETO, F. M. M.; OSÓRIO, A. J. M. Dificuldades e desafios do processo de aprendizagem de algoritmos e programação no ensino superior: uma revisão sistemática de literatura. **Research, Society and Development**, v. 9, n. 10, p. e9429109287, 2020.

OLIVEIRA, C. M.; PIMENTEL, A. R.; MASCHIO, E. **NextStep: um protótipo para o sequenciamento inteligente e adaptativo de enunciados em programação de computadores**. 2016. Apresentação no WAlgProg. Disponível em: https://walgprog.gp.utfpr.edu.br/2016/assets/arquivos/apresentacoes/S3A8.pdf. Acesso em: 28 out. 2026.

SOUZA, M. A. F. *et al.* **Algoritmos e lógica da programação**. 3. ed. São Paulo: Cengage Learning, 2019.