**Problema de pesquisa**

* Quais ferramentas de estudo que melhores se adequam a compreensão do processo de compilação para estudantes?

**Objetivo Geral**

* O presente trabalho tem como objetivo geral apresentar quais ferramentas de estudo auxiliam no aprendizado do processo de compilação tendo como benefício obter uma absorção rápida do conteúdo.

**Objetivos Específicos**

* Apontar Ferramentas de Estudo
* Explicar Processo de Compilação
* Distinguir Métodos de Aprendizagem
* Pesquisar de que forma o uso de ferramentas de estudo podem influenciar o processo de aprendizagem
* Analisar técnicas de absorção rápida de conteúdo

1. Explique as razões da preferência por este tema.

A principal motivação para sustentar o presente projeto de pesquisa, reside na falta de ferramentas de estudo que facilitem o processo de aprendizagem na matéria de compiladores visto a dificuldade apresentada pelos estudantes.

1. Explique a importância deste tema.

Compiladores se apresenta como uma das matérias mais difíceis nos cursos de computação, tendo grande complexidade, difícil visualização e dependência de outras matérias. Diante dessa premissa, deve-se criar meios que facilitem o aprendizado sem o auxílio direto de um professor, para que assim se obtenha um nível de absorção do conteúdo de forma ampla e prática.

A *internet* se dispõe de forma comunicacional, abrindo portas para o compartilhamento de informações podendo ser na utilização de imagens, vídeos, *gifs*, textos e etc., assim no meio didático pode-se criar ferramentas que auxiliem a construção do conhecimento. Na cibercultura, a lógica comunicacional supõe rede hipertextual, multiplicidade, interatividade, imaterialidade, virtualidade, tempo real, multissensorialidade e multidirecionalidade (Lemos, 2002; Levy, 1999).

* As disposições comunicacionais do computador online estão em sintonia com exigências de qualidade pedagógica em educação online, como dialógica, compartilhamento, colaboração, participação criativa e simulação na construção do conhecimento.
* A internet é um meio de comunicação, compartilhamento de informações, conhecimento, aprendizagem
* Compiladores é uma matéria complexa e de difícil visualização
* Visto a dificuldade apresentada, deve-se criar meios de facilitar
* Falta de auxilio fora das universidades/Estudo em casa

3. Explique porque este tema é relevante.  
4. Identifique se a abordagem proposta tem vantagens ou pontos positivos.

**6.1. Introdução e Justificativa:** A principal motivação para sustentar o presente projeto de pesquisa, reside na falta de ferramentas de estudo que facilitem o processo de aprendizagem na matéria de compiladores visto a dificuldade apresentada pelos estudantes.

Compiladores se apresenta como uma das matérias mais difíceis nos cursos de computação, tendo grande complexidade, difícil visualização e dependência de outras matérias. Diante dessa premissa, devem-se criar meios que facilitem o aprendizado sem o auxílio direto de um professor, para que assim se obtenha um nível de absorção do conteúdo de forma ampla e prática.

A internet se dispõe de forma comunicacional, abrindo portas para o compartilhamento de informações seja na utilização de imagens, vídeos, gifs, textos e etc., assim no meio didático pode-se criar ferramentas que auxiliem a construção do conhecimento. Na cibercultura, a lógica comunicacional supõe rede hipertextual, multiplicidade, interatividade, imaterialidade, virtualidade, tempo real, multissensorialidade e multidirecionalidade (Lemos, 2002; Levy, 1999).

A disciplina de Compiladores no contexto de um curso de graduação, objetiva levar o aluno a perceber concretamente as diversas etapas envolvidas no processo de compilação, elaborando, ao final da disciplina, o projeto de um compilador funcional para uma gramática simples. O Compilador Verto foi escrito na linguagem de programação Java e elaborado

Os compiladores de modo geral, são ferramentas que leem e traduzem um programa mantendo a semântica original. Durante esse processo de tradução, o compilador deve analisar o código escrito em linguagem fonte e sintetiza-lo para uma linguagem objeto.

Especificar a dificuldade que a comunidade encontra no momento (falta de um novo modelo, mais barato, mais prático, mais rápido, amigável e/ou confiável, entre outros), mostrando a importância e relevância da pesquisa. Uma pesquisa científica precisa ser relevante. Não se pode pesquisar, por exemplo, como desenvolver uma roda (tema esgotado), nem projetar um secador de gelo (tema inviável ou inútil). Identificar as razões da preferência pelo tema escolhido em relação a outros temas, ou seja, “vender seu peixe”, especificando a utilidade ou a novidade vantajosa apresentada pela pesquisa.

**6.2. Problema Proposto:** Identificar o problema (algumas questões) que se deseja resolver na pesquisa. Deve ser redigido em forma interrogativa, ou seja, elaborar perguntas que se pretende responder no desenvolvimento do trabalho.

Quais ferramentas de estudo que melhores se adequam a compreensão do processo de compilação para estudantes? Como ferramentas de estudo podem beneficiar o estudante? Quais métodos de aprendizagem facilitam na compreensão de um conteúdo? Como esses métodos podem ser postos para o processo de compilação?

**6.3. Objetivos:** Relatar a intenção pretendida com o tema proposto, sintetizando o que se pretende alcançar com a solução computacional proposta. Objetivos são iniciados com verbos no infinitivo, como: desenvolver (um algoritmo/software/produto, entre outros), comparar (métodos/metodologias/teorias/autores, entre outros), estudar (uma teoria específica, um método famoso/linha de pensamento de um determinado autor, entre outros), fazer um estudo de caso, fazer um levantamento de dados, entre outros.

Desenvolver uma aplicação *web* como uma ferramenta de estudo para auxiliar no aprendizado do processo de compilação, tendo como benefício, para os estudantes, a absorção rápida do conteúdo. Explicar o processo de compilação. Pesquisar de que forma o uso de ferramentas de estudo podem influenciar o processo de aprendizagem.

Observação: Evite resumir excessivamente estes três itens (introdução e justificativa, problema proposto, objetivos), de forma a não permitir que os professores da disciplina avaliem com propriedade a abrangência do trabalho proposto. Procure ser o mais claro e completo na sua construção textual.

1. **Ligação:** Parte introdutória do tema da monografia você deve elaborar uma introdução ao problema;
2. **Problema:** Quais ferramentas de estudo que melhores se adequam a compreensão do processo de compilação para estudantes?
3. **Ligação:** A presença desse elemento de ligação tem como intuito unir o problema e o limite da pesquisa;
4. **Limites:** Restringe o campo de trabalho do estudante para que o mesmo durante a elaboração do TCC.

Veja o exemplo de aplicação dessa fórmula:

Este projeto de pesquisa delimitou-se em propor uma ferramenta de estudo, dentro da cibercultura, que possa influenciar na compreensão do processo de compilação para os estudantes.

**6.4. Trabalhos Relacionados:** Deverão ser selecionados pelo menos 5 (cinco) trabalhos que estejam relacionados ao problema tratado na proposta. Tais trabalhos deverão estar descritos de forma sucinta, mas que permita ao leitor identificar claramente o problema abordado e a forma de solução proposta pelo mesmo. Ao término desse relato deverá ser gerado um quadro apresentando como cada um dos trabalhos descritos aborda o problema e qual a relação com sua pesquisa.

No primeiro paragrafo você deve contextualizar falando que a seção será dedicada a descrição dos trabalhos relacionados. Em seguida, você deve dedicar um paragrafo para cada um dos trabalhos, descrevendo como ele trata aquele tema abordado e por fim mostrando com uma visão crítica seus pontos fortes e fracos. No último paragrafo você pode fazer uma crítica geral aos trabalhos.

A tabela abaixo demonstra resumidamente os problemas, os objetivos, as metodologias e as funcionalidades que de forma geral consistem em apresentar meios alternativos para o processo de aprendizagem na área de informática, podendo ser aplicados para a disciplina de compiladores. Após a tabela, são apresentados os trabalhos relacionados escolhidos.

INTERPRETADOR DA LINGUAGEM D+

O interpretador da linguagem D+ é um software proposto e desenvolvido por Gabriel Ribeiro (2019). Ribeiro enfatiza em seu projeto a dificuldade de aprendizagem dos alunos da disciplina de compiladores no Bacharelado em Ciência da Computação, ele descreve a matéria como muito abrangente e profunda. Em sua pesquisa, o mesmo identificou que mais que a metade dos estudantes de compiladores tem grande dificuldade de aprender a matéria.

Visto o problema da aprendizagem dos estudantes, Ribeiro desenvolveu um software integrado a uma interface com o intuito de amenizar a dificuldade dos alunos no aprendizado e auxilia-los na absorção do conteúdo de compiladores. O software consiste em uma interface de fácil utilização, onde o usuário digita um código em D+ ao lado esquerdo da tela, que é compilado após apertar um botão de executar, e ao lado direito da tela são geradas algumas partes do processo de compilação: a análise léxica, a análise sintática e a tabela de símbolos.

A análise léxica é demonstrada com um autômato finito que contém todos os autômatos possíveis para a linguagem D+, que após a compilação do código digitado pelo usuário são pintados os autômatos que aquele código teria feito. A análise sintática é representada por uma árvore sintática. E a tabela de símbolos mostra ao usuário, os identificadores localizados pelo interpretador e traz informações que auxiliem na compreensão de cada palavra inserida no código.

As tecnologias usadas para o desenvolvimento do interpretador de D+ foram a linguagem de programação C++ para a codificação do compilador, QT Creator para criação da interface e JFLAP para a criação dos autômatos finitos.

* a motivação é a dificuldade da matéria de compiladores
* o projeto consiste numa interface gráfica que mostra a análise léxica e sintática do processo de compilação
* o software compila o código e depois o processo de forma visual
  + como o software já é compilado, se o código for muito extenso a análise léxica fica com difícil compreensão
* linguagem utilizada - C++, editor utilizado QT creator, compilador da linguagem - D+
* o projeto é uma ferramenta de compilador D+ que auxilia no aprendizado dos estudantes
* a pesquisa explica o processo de compilação e o IHC

SCC: Um Compilador C como Ferramenta de Ensino de Compiladores

O compilador SCC (SOIS C Compiler), de Foleiss, Assunção, Cruz, Gonçalvez e Feltrim (2009), se destaca entre os compiladores de C por contemplar todo o conjunto de instruções de um compilador, assim proporcionando um serviço de ajuda no ensino de disciplinas que abordam o processo de compilação.

O projeto foi desenvolvido com os objetivos de gerar código que possa ser montado pelo SASM, montador do SOIS, permitindo que os programas da linguagem C sejam executados pelo simulador do ambiente e também de ser uma ferramenta de ensino, tendo funcionalidades que auxiliem a compreensão de áreas específicas do processo de um compilador real.

O SOIS, sistema operacional integrado simulado, foi escolhido justamente por ser um ambiente que permite a escrita, execução e depuração de programas em um ambiente simulado. Por sua vez, o montador SASM possui modos de depuração que podem demonstrar todas as fases do processo de compilação, a inter-relação dos seus artefatos e componentes e também os resultados obtidos.

A pesquisa escolheu o processo de compilação proposto por Louden que possui cinco etapas: análise léxica, análise sintática, análise semântica, geração de código intermediário e geração de código final. No SCC, são geradas a árvore sintática, a árvore de símbolos, a árvore sintática abstrata anotada e um analisador semântico.

* compilador C padrão ANSI que gera código de montagem compatível com o SASM, o montador do ambiente SOIS.
* o SASM possui funcionalidade similar, os modos de depuração podem ser utilizados no ensino, com o intuito demostrar todas as fases do processo de compilação, a inter-relação de seus componentes e artefatos, além de permitir a visualização dos resultados gerados.
* análise léxica, análise sintática, análise semântica, geração de código intermediário e geração de código final.

Compilador Educativo VERTO: ambiente para aprendizagem de compiladores

No Centro Universitário Feevale foi notada a necessidade de desenvolver uma ferramenta, de apoio pedagógico, para a disciplina de compiladores após a mesma enfrentar desmotivação e dificuldades de compreensão dos alunos.

Os compiladores são de forma geral, um tradutor de alguma linguagem para um programa. De acordo com os pesquisadores Schneider, Passerino e Oliveira, a aprendizagem de compiladores consiste em absorver um processo composto por etapas, nas quais se participam técnicas, estratégias e métodos específicos para aprender o processo de compilação, sendo ele: análise léxica, sintática e semântica, tabela de símbolos e geração do código objeto. A pesquisa ressalta também que, a aprendizagem é a construção de uma representação pessoal de um conteúdo que é objeto de aprendizagem (Coll, 1998). Para atingir esse nível de compreensão os estudantes devem ser capacitados a compreender as fases do compilador, sobretudo as fases finais de geração de código intermediário e objeto.

Visto a dificuldade dos alunos e a disciplina ser lecionada em apenas um semestre, foi desenvolvido o Compilador Educativo Verto, que faz parte de um ambiente de aprendizagem para compiladores. O compilador é composto por diversas ferramentas computacionais, um professor mediador, os alunos que o utilizam, as sequencias didáticas que são planejadas no plano de ensino e aprendizagem apoiada pela metodologia de ensino embasada pelo professor.

O processo de compilação do Verto inicia-se com a geração do código intermediário em um formato macro-assembler. O formato dispõe de instruções simplificadas para facilitar a compreensão das estruturas compiladas. Após essa etapa, gera-se o arquivo final que contém as instruções no formato César, uma linguagem objeto criada com fins didáticos, permitindo que o estudante execute e analise o algoritmo. Na figura 1 pode-se observar o esquema de tradução do compilador Verto.

FIGURA 1 - Esquema de Tradução do Compilador Verto.



FONTE: (SCHNEIDER, PASSERINO, OLIVEIRA, 2007)

Para o uso inicial da ferramenta, dispõe-se uma tela para a edição de textos fonte escritos na linguagem Verto. Na figura 2 é possível observar a tela de edição e a tabela de símbolos gerada a partir do código inserido.

FIGURA 2 - Tela de Edição do Compilador Verto.

FONTE: (SCHNEIDER, PASSERINO, OLIVEIRA, 2007)

A ferramenta focou as etapas finais da compilação, utilizando uma técnica de análise léxica simples e um método de análise sintática. A fase de análise léxica, que verifica os caracteres encontrados no programa fonte como lexemas válidos, pode ser visualizada pelo estudante na figura 2, onde os tokens de identificação e lexemas são listados. A análise sintática, onde é feita a geração de erros, análise semântica e geração do código-objeto, é feita de forma recursiva, onde cada regra sintática reconhecida pode levar o compilador a disparar uma rotina de ação semântica ou de geração de código. A ampliação da saída do semântico (Figura 2), pode-se ser observada na figura 3, onde é registrada a sequência de regras reconhecidas pelo compilador.

FIGURA 3 - Aba: Saída do Sintático



FONTE: (SCHNEIDER, PASSERINO, OLIVEIRA, 2007)

AUXÍLIO NO ENSINO EM COMPILADORES: SOFTWARE SIMULADOR COMO FERRAMENTA DE APOIO NA ÁREA DE COMPILADORES

Os pesquisadores Costa, Silva e Britto declararam em seu projeto a existência da necessidade de se elaborar uma técnica que pudesse facilitar a compreensão dos alunos dos cursos que envolvem computação, mais especificamente compiladores. Com essa premissa, eles definiram a elaboração e desenvolvimento de um software capaz de simular claramente o funcionamento interno das fases de um compilador.

A ferramenta CompilerSim foi criada para executar um processo de análise de código, baseado na linguagem de programação Pascal, e posteriormente para converter, o código, para a linguagem de baixo nível Assembly. O desenvolvimento da ferramenta se baseou na série de artigos “Let’s Build a Compiler”, escritos pelo cientista Ph.D. Jack W. Crenshaw na década de 80. A partir dos artigos foi extraído o código original e o mesmo passou por modificações e adaptações com o intuito de que as etapas do processo de compilação fossem demonstradas através de uma interface gráfica.

A interface gráfica do software apresenta algumas funcionalidades para o usuário, o passo inicial da aplicação é dado na inserção do código fonte, que pode ser feito manualmente pelo usuário (figura 1) ou pode ser selecionado e carregado a partir da aba “Exemplos”, que dispõe o código de exemplos prontos (figura 2).

Após a execução do código, a ferramenta também dispõe os seguintes resultados: os tokens gerados a partir da análise léxica do código, a interpretação e geração de código intermediário na linguagem Assembly e quando cabível os erros gerados na compilação. A visualização desses itens pode ser observada nas figuras 3 e 4 respectivamente.

C-gen – Ambiente Educacional para Geração de Compiladores

O C-gen é uma ferramenta, criada pelos pesquisadores Backes e Dahmer, com o intuito de sanar a carência de ferramentas com uma interface gráfica capaz de ser utilizada para orientar um aluno de compiladores, exibindo o funcionamento de todo o processo de compilação.

A ferramenta desenvolvida permite que o aluno a utilize enquanto aprende, possibilitando que a teoria de compiladores seja visualizada, assim facilitando a compreensão do conteúdo. Para isso, a ferramenta atende vários requisitos, dentre eles: possibilitar a definição e disponibilizar uma interface adequada para os analisadores léxicos, sintáticos e semânticos; permitir o acompanhamento do processo de reconhecimento das fases da compilação, passo a passo.

A arquitetura do sistema é organizada para que cada passo do processo de compilação se comunique com o passo posterior através de uma representação intermediária, permitindo ser executado independentemente (figura 1). Os passos da compilação são implementados com plug-ins que reconhecem as suas respectivas entradas e produzem as saídas correspondentes.

O plug-in responsável pelo analisador léxico foi implementado de forma que seja produzido um analisador através da definição de autômatos. A definição é informada pelo usuário, que é responsável pela criação de estados e transições do autômato e os caracteres que o mesmo deve reconhecer em cada transição. Após a edição do autômato, quando iniciado o processo de reconhecimento, a ferramenta converte os dados em um autômato finito determinístico e cria a tabela de transições, conforme a figura 2.

A análise sintática é feita através da especificação da gramática, considerando como terminais os tokens gerados pelo analisador léxico. A gramática é gerada pelo usuário no editor, que deve criar os símbolos não terminais. O propósito do usuário gerar sua própria gramática é justamente para que o mesmo não precise aprender uma nova notação para utilizar a ferramenta. A tela para essa confecção e de resultado da análise é exibida na figura 3.

1. introduzir qual o intuito do projeto
2. quais os problemas q o projeto visa solucionar?
3. qual a solução proposta?
4. como foi construído o projeto?
5. quais os pontos negativos do projeto?