



CITI – Compilador Interativo com Tradutor Integrado: Um ambiente para auxílio ao ensino-aprendizagem de programação para alunos iniciantes

Carlos Henrique Corrêa Tolentino¹, Lucas Augusto Nunes de Barros², Leticia Gomes Sá², Bernard Silva Araujo Wermuth de Carvalho², Pedro Henrique Santana Amaral²

¹Docente na área de informática – IFTO-Palmas. e-mail: chtolentino@ifto.edu.br

²Discentes do EMI - Informática – IFTO-Palmas. Bolsistas do CNPq. E-mail: lucas10797@gmail.com, leticia.gomesdesa@gmail.com, pedrosantana1@gmail.com

Resumo: A programação de computadores é uma área extremamente importante na informática. O ensino de programação tem sido alvo de preocupações entre os docentes do IFTO – Instituto Federal de Educação, Ciência e tecnologia do Tocantins – Campus Palmas, bem como em todo o país. O CITI – Compilador Interativo com Tradutor Integrado é um software para edição de programas em linguagem C que permite a visualização das mensagens de erro em português. A hipótese trabalhada é que a associação entre as mensagens em português e inglês promova um melhor entendimento entre os alunos, auxiliando não só a resolução dos problemas, mas também, contribuindo motivar os alunos, além de auxiliar na aprendizagem da língua inglesa.

Palavras-chave: Compilador, Ensino-aprendizagem, programação, linguagem C, tradução

1. INTRODUÇÃO

A programação de computadores é uma das atividades mais importantes da área de Tecnologia da Informação. Sua relevância se estende desde a compreensão dos processos de desenvolvimento de software e modelagem das regras de negócio até o próprio entendimento sobre o funcionamento do computador como ferramenta de trabalho, bem como é parte essencial na resolução de problemas usando sistemas de informações. O ensino e aprendizagem na área de programação são fundamentais para o bom desenvolvimento de um estudante e futuro profissional da área de Tecnologia da Informação. No Brasil o interesse dos pesquisadores na questão do ensino-aprendizagem de programação de computadores aumentou a partir do ano de 2009. Segundo Aureliano e Tedescu (2012), no decênio 2002-2011 51% das publicações sobre o tema em eventos científicos foram publicadas justamente entre 2009 e 2011. Além disso, 75% dos trabalhos foram realizados com a participação de instituições das regiões Sul e Sudeste, enquanto que poucos estudos foram realizados na região Norte. Outro dado importante é a distribuição dos estudos pelos níveis de escolaridade: 61% restritos à educação superior e apenas 19% com foco no ensino fundamental, médio e técnico. Melhorar os índices de aprovação e, principalmente, a efetividade do aprendizado é uma necessidade detectada tanto pela percepção dos professores, quanto pelos estudantes. Em levantamento realizado no ano de 2014, com 82 estudantes (23 nível superior e 59 nível médio), foram abordados alguns motivos que poderiam explicar as causas dessa dificuldade de aprendizado.

Tabela 1 – Motivo

s que ajudam a explicar as dificuldades na aprendizagem de programação (levantamento feito com 82 alunos)

Dificuldades	Superior	Médio
Entender os problemas	11(17%)	20(15%)
Desenvolver o raciocínio lógico	13(20%)	26(19%)
Passar a lógica para a linguagem de programação	17(26%)	32(24%)
Manusear o compilador	5(8%)	9(7%)
Compreender as mensagens de erro do compilador	10(15%)	21(16%)
Corrigir os erros	9(14%)	27(20%)

Considerando os resultados da Tabela 1, pode-se observar que as causas de dificuldades relacionadas à utilização do compilador correspondem a aproximadamente 40% do total. Diante disso, é importante que esse dado seja considerado nas estratégias de melhoria na qualidade da formação dos alunos. Nesse sentido, outro levantamento recente, realizado no primeiro semestre de 2015, também com os alunos do IFTO – Campus Palmas – permitiu comparar a percepção dos estudantes em relação à usabilidade de dois dos compiladores mais utilizados em introdução à programação: Dev C++ (ambiente Windows) e GCC (ambiente Linux).

Tabela 2 – Percepção dos estudantes sobre a usabilidade dos ambientes de desenvolvimento de programas. (levantamento feito com 42 estudantes)

IDE					
	Negativa	Indiferente	Positiva	Muito Positiva	Não Conhece
Delphi	0	3	1	0	24
Dev C++	1	3	14	4	6
Eclipse	0	3	6	1	18
GCC-Linux	1	6	13	3	5
Lazarus	0	0	0	0	28
NetBeans	0	0	10	7	11
Visual Studio	1	2	3	1	21



Considerando que a linguagem de programação usada como base para o ensino introdutório de programação no IFTO – Campus Palmas – é a linguagem C e observando os resultados, que mostram que os compiladores Dev C++ e GCC-Linux estão entre os mais usados, sugere-se que ambos possuem aspectos que podem ser aproveitados para aumentar o rendimento de alunos iniciantes.

Apesar da percepção em relação à importância da usabilidade ser uma realidade entre os alunos, ao analisar os temas abordados nas últimas edições dos mais importantes eventos nacionais da área de Informática na Educação (SBIE – Simpósio Brasileiro de Informática na Educação e WIE – Workshop de Informática Educativa) percebe-se que existem poucos trabalhos relacionando a usabilidade com a aprendizagem de programação.

Tabela 3 – Quantidade de artigos relacionados à usabilidade no ensino-aprendizagem de programação de computadores.

Ano	2011	2012	2013	2014
Quantidade	2	2	1	1

A justificativa para este trabalho é a necessidade de se estudar e melhorar a usabilidade dos ambientes de desenvolvimento utilizados por alunos iniciantes nas atividades de ensino de programação, especificamente os estudantes do IFTO - Campus Palmas. A hipótese a ser trabalhada é que ao visualizar as mensagens de erro dos compiladores em seu formato original em inglês, juntamente com sua tradução para o português, poderá contribuir com melhorias para o aprendizado de linguagens de programação. Nielsen (1994), considerado o “pai da usabilidade”, apresenta entre as suas 10 heurísticas de usabilidade a consistência entre o sistema e o mundo real, ou seja, as interfaces devem falar a língua do usuário, incluindo idiomas.

Ao cumprir essa heurística, espera-se que os alunos iniciantes tenham melhores condições de corrigir os programas, aumentando a produtividade, diminuindo a perda de motivação e também possam fazer associações entre as mensagens de forma a fixar melhor o seu conteúdo e conseqüentemente levar o conhecimento adquirido quando forem trabalhar com outras linguagens e ambientes de programação. Assim o objetivo deste trabalho é a construção de um ambiente de edição de programas em linguagem C que permita a visualização das mensagens do compilador em dois formatos: original em inglês e traduzidas para a língua portuguesa.



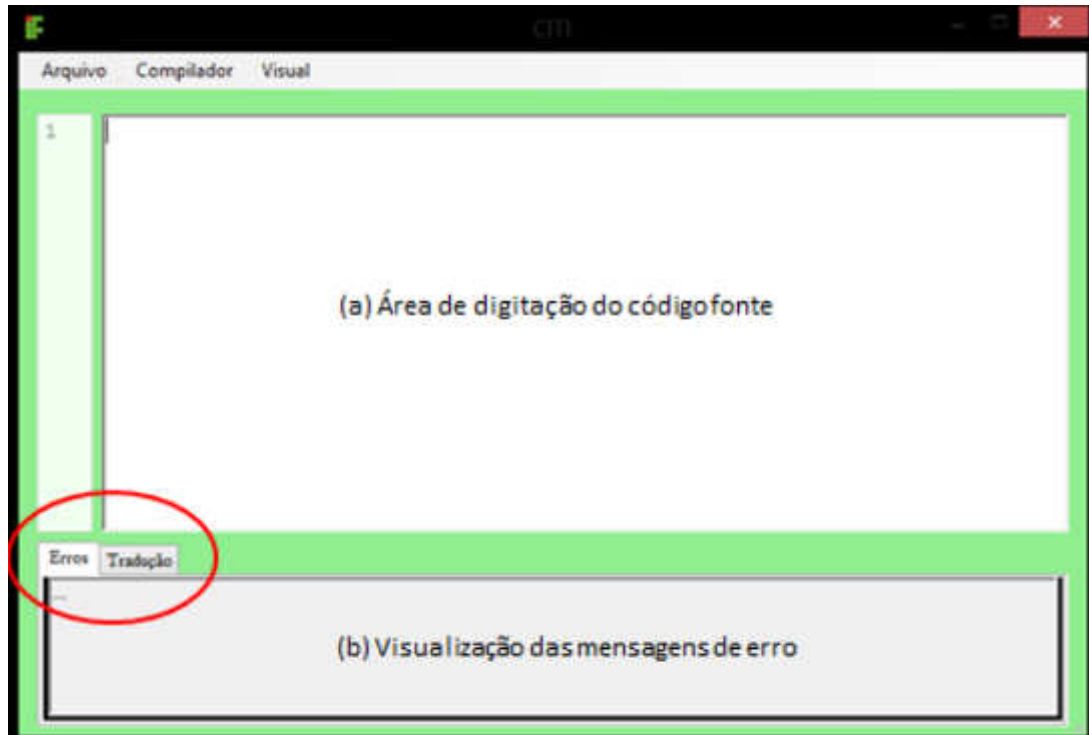
2. MATERIAL E MÉTODOS

O ambiente de edição de programas CITI foi desenvolvido utilizando o IDE Visual Studio 2013 Express (MICROSOFT, 2013). Nessa versão gratuita, foram utilizados os paradigmas de orientação a objetos e orientação a eventos. A interface gráfica segue o formato Windows Form Application e a aplicação executa em modo local e desconectado. A plataforma de hardware utilizada foi um notebook Dell Inspiron 14 3421, Processador Intel I3, 4GB de Memória RAM, 1 TB de HD com o sistema operacional Windows 8.1 Professional. Por não fazer parte do escopo do trabalho, não foi desenvolvido um compilador específico, mas uma versão do compilador GCC distribuída em conjunto com o software DevC++ 4.9.9.2 (BLOODSHED, 2005), em sua versão portátil, foi utilizada por meio de chamadas de sistema. O arquivo temporário que armazena a saída do compilador bem como a base de dados contendo as mensagens de erro traduzidas tem formato de texto. Dessa forma o editor tem maior portabilidade, independendo da utilização de sistemas gerenciadores de bancos de dados ou outros elementos externos à aplicação para armazenamento de dados.

3. RESULTADOS E DISCUSSÃO

O editor CITI foi projetado para ter uma interface simples e intuitiva. A versão inicial inclui apenas os elementos básicos para permitir a escrita e compilação de programas. Porém, algumas especificidades da codificação de programas, tais como utilizar cores diferentes para diferentes elementos da linguagem (por exemplo, a cor verde para diretivas de compilação e vermelho para palavras reservadas) foram incluídas. A Figura 1 a seguir apresenta a interface do CITI.

Figura 1 – Interface principal do CITI



Observe que a área de digitação de programas tem à sua esquerda a contagem das linhas de código, o que facilita a identificação dos erros apontados pelo compilador. Outro detalh importante são as abas do painel de apresentação das mensagens de erro: uma destinada às mensagens originais e outra destinada às mensagens traduzidas. Uma vez digitado o código

fonte a compilação é feita invocando o compilador GCC, como já explicado anteriormente, via menu ou tecla de atalho. A Figura 2 a seguir ilustra essa parte da interface do CITI.



Figura 2 – Exemplo da utilização de menus

Uma vez compilado o código fonte, o resultado é redirecionado para um arquivo temporário e as mensagens de erro são tratadas individualmente: cada uma delas é comparada com uma base de dados que contém identificadores de mensagens e suas respectivas traduções. Caso a tradução para a mensagem seja encontrada ela será exibida juntamente com a mensagem original. Caso contrário, existe ainda a possibilidade de buscar a tradução da mensagem na ferramenta *Google Translator* será apresentada ao usuário. A Figura 3 ilustra a arquitetura do CITI.

A Figura 3 ilustra a arquitetura do CITI

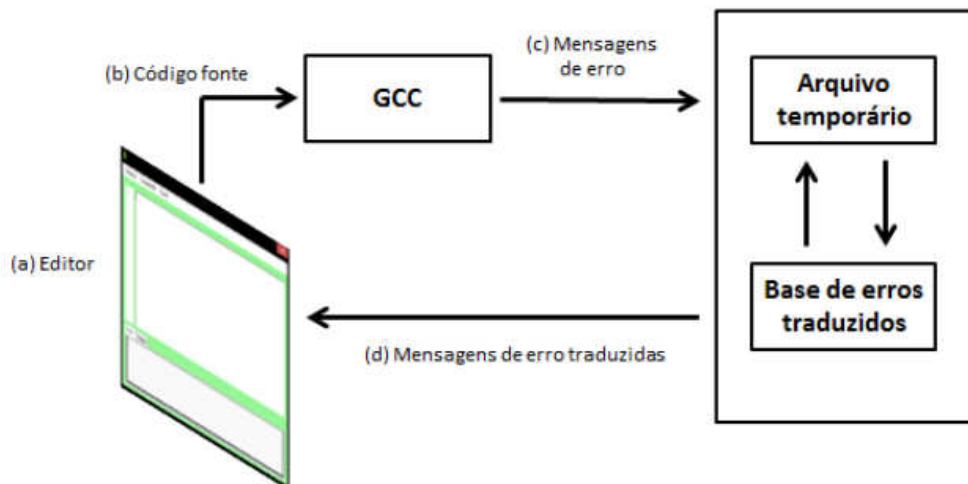


Figura 3 – Arquitetura do CITI



Apesar de não enfatizarem o aspecto da usabilidade, existem outras ferramentas que utilizam a língua portuguesa na programação, porém, os algoritmos são escritos em português estruturado, o que alguns autores costumam chamar de “portugol”, abdicando de uma linguagem de programação real. Mando, Oliveira e Marques (2009) apresentam um IDE para desenvolvimento de algoritmos em portugol. Santiago e Dazzi (2004) desenvolveram um ambiente para interpretação de programas e língua portuguesa e em fluxograma enquanto Scaico et al (2012) apresenta uma abordagem de programação orientada ao design. Se por um lado facilitam a escrita de programas sem a necessidade de aprendizado da língua inglesa, por outro, podem atrasar o desenvolvimento do programador, pois adia uma inevitável familiaridade do mesmo com idioma que predomina na área de informática. Já apresenta um ambiente online para edição e compilação de programas. Pesa contra esse projeto a necessidade de estar em ambiente conectado à Internet.

4. CONCLUSÕES

Segundo as pesquisas apresentadas um dos principais desafios ao estudante na programação e interpretar os erros de código e manusear o compilador. O CITI - compilador interativo vem com o intuito de ajudar o aluno de programação a decifrar os erros de códigos e resolvê-los e neste sentido o programa se mostra eficiente ao realizar a tradução das mensagens de erros e aproveitar as melhores características do Dev-C++ (compilador com a melhor aceitação pelos estudantes do IFTO – Campus Palmas), esse ambiente ajudará muito os estudantes no seu dia a dia de estudos e dessa forma a continuidade deste trabalho deverá seguir duas linhas: a implantação e utilização do CITI nas aulas de programação, para que se possa avaliar o impacto sobre o aproveitamento dos alunos e o desenvolvimento de novas funcionalidades, por exemplo, tornar o ambiente conectado, colaborativo e assistido pelo professor da disciplina.



REFERÊNCIAS

AURELIANO, V. C. O.; TEDESCO, P. C. A. R.; Ensino-aprendizagem para de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE. Anais do 23º Simpósio Brasileiro de Informática na Educação, Rio de Janeiro, 2012.

BLOODSHED; BlodShed Software Dev-C++ 5 Beta 9.2 (4.9.9.2), 2005. Disponível em: <<http://www.bloodshed.net/>>

MANSO, A.; OLIVEIRA, L.; MARQUES, C. G.; Aprendizagem de algoritmos – Portugal IDE. Anais da VI Conferência Internacional de TIC na Educação. Braga, Portugal, pp. 969-983m, 2009.

MICROSOFT; Visual Studio 2013 Express, 2013. Disponível em: <<https://www.visualstudio.com/products/visual-studio-community-vs>>

NIELSEN, J.; Enhancing the explanatory power of usability heuristics. Proc. ACM CHI'94 Conf. Boston, MA, pp 152-158, 1994

SANTIAGO, R.; DAZZI, R. L. S.; Ferramenta de Apoio ao Ensino de Algoritmos. Anais do XIII SEMINCO – Seminário de Computação. Blumenau-SC, 2004.

SCAICO, P. D.; et al. Programação no Ensino Médio: Uma abordagem de ensino orientado ao design. Anais do XVIII WIE – Workshop de Informática Educativa. Rio de Janeiro, 2012.