

SUMÁRIO

1	INTRODUÇÃO	3
1.1	OBJETIVOS DO TRABALHO	4
1.1.1	Objetivo Geral	4
1.1.2	Objetivos Específicos	4
1.2	JUSTIFICATIVA DO TRABALHO	4
1.3	ESTRUTURA DO TRABALHO	5
2	FUNDAMENTAÇÃO TEÓRICA	6
2.1	ALGORITMOS	6
2.1.1	Ensino Aprendizagem	6
2.2	PSEUDOLIGUAGEM E PORTUGOL	6
2.3	FERRAMENTAS EXISTENTES	6
2.3.1	UAL e Editor UAL	6
2.3.2	Demais Aplicativos	6
2.3.3	Plataforma Digitais de Ensino	6
2.3.3.1	Udacity	6
2.3.3.2	Codeacademy	6
2.3.3.3	SoloLearn	6
2.4	TECNOLOGIAS WEB	6
2.4.1	Linguagem de Marcação de Texto	6
2.4.2	Liguagem de Programação para Web	6
2.4.2.1	ECMAScript	6
2.4.3	Navegadores Web	6
2.4.4	Aprimoramentos para Uso <i>Offline</i>	6
2.5	COMPILADORES E INTERPRETADORES	6
2.5.1	Análise Léxica	6
2.5.2	Análise Sintática	6
2.5.3	Análise Semântica	6
3	DESENVOLVIMENTO DA APLICAÇÃO	7
3.1	FERRAMENTAS UTILIZADAS	7
3.1.1	Editor de Código ACE	7
3.1.2	<i>Parser</i> Acorn	7
3.1.3	JS-Interpreter	7
3.2	PROCESSO CRIACIONAL	7
3.2.1	Git e Github	7

3.2.2	Node.js e Grunt	7
3.2.3	SASS	7
3.2.4	TDD	7
3.3	INTERFACE GRÁFICA	7
3.3.1	<i>Material Design</i>	7
3.3.1.1	MDL	7
3.3.2	Inspirações	7
	 REFERÊNCIAS	 8

1 INTRODUÇÃO

A base da programação é ensinar máquina a realizar determinada tarefa. Para isso pode-se dizer que independente do paradigma utilizado no processo de desenvolvimento ele envolve algoritmos.

Algoritmos podem não necessariamente serem computacionais, já que são um conjunto de passos finitos. Analogias simplistas são os ingredientes e modo de preparo de uma receita ou interiores para uma tarefa do cotidiano como a troca de um pneu furado em um automóvel.

Porém se tratando de computação deve ter maior exatidão seguindo uma linguagem pois diferente do ser humano uma máquina não é tão interpretativa. Desconsideradas aqui inteligências artificiais que são desenvolvidas por algoritmos para terem comportamento de compreensão humana. Isto pode ser analogamente algo como não entender idioma estrangeiro e as instruções estarem nesta linguagem. Porém em informática isso tem que ser mais especificado com palavras ou caracteres predeterminados de início e fim, de blocos, repetições, etc.

Sendo o berço mais popular da informática os Estados Unidos da América e seus criadores em muitos casos dessa origem ou de países com mesmo idioma as primeiras linguagens de computação e a maioria delas são em inglês.

Isso muda quando falamos do Portugal que possui variações, mas resumidamente sua origem vem como uma tradução de uma linguagem de programação simples e comum com seus termos em nosso idioma nativo no Brasil, pela nossa colonização portuguesa.

Tendo como foco o aprendizado de algoritmos para uma boa fundamentação de futuros programados ou profissionais da área, e muitas vezes a facilidade com o inglês não é algo comum nosso país, a linguagem em Portugal de torna um artifício vantajoso.

O que se propõe aqui em ter ferramenta de aprendizado composta de editor de texto para a escrita dos algoritmos e visualizador do resultado dos mesmo. E que não seja necessária instalação, seja facilmente acessível como qualquer outra página web em computadores ou dispositivos móveis e mesmo assim permaneça acessível mesmo sem conexão disponível com a internet.

Este projeto visa demonstrar que um novo interpretador para a pseudolinguagem Portugal, que tem o propósito de facilitar o estudo da lógica de programação vem ao encontro como solução para muitas dificuldades atuais no âmbito de aprendizagem acadêmica. Para tal, seguiu-se a ideia de implementar este interpretador, onde haverá a preocupação em desenvolvê-lo seu protótipo em dois módulos: Interpretador e Interface Homem-Máquina além de sugerir um Sistema de Informação para o Professor. Essa ferramenta é um interpretador para a pseudolinguagem Portugal, gratuito e com uma interface adequada.

1.1 OBJETIVOS DO TRABALHO

1.1.1 Objetivo Geral

Apresentar o conceito de ferramenta computadorizada para se tornar referência em criação, edição, visualização e interpretação de algoritmos em pseudolinguagem, como alternativa às existentes. Sendo seu diferencial o acesso a partir de navegadores Web por ser em HTML5. Inclusive acessível quando uma conexão com a internet esteja indisponível.

1.1.2 Objetivos Específicos

Visando atingir o objetivo geral em sua totalidade lista-se a seguir objetivos específicos de modo sucinto:

- Desenvolver o protótipo, sendo seu código fonte aberto e disponível para que outros possam manter, ampliar e evoluir a aplicação;
- Fazê-lo compatível com sintaxe UAL e mínimo produto viável para as primeiras lições de ensino aprendizagem;
- Permitir o acesso via navegadores web modernos, sem necessidade de instalações complementares;
- Aplicar aprimoramentos introduzidos na versão 5 do HTML (*HyperText Markup Language*) para uso sem conexão com a internet.

1.2 JUSTIFICATIVA DO TRABALHO

Praticar em *software* à abstração em papel auxilia e muito no processo de ensino e aprendizagem, em algoritmos além do pensamento lógico por etapas já pode ser desenvolvido e descrito para o repasse a um computador. “O Aprendizado de algoritmos não é uma tarefa muito fácil, só se consegue através de muitos exercícios” (LOPES; GARCIA, 2002).

Para esse ensino é essencial conhecer dois conceitos principais lógica de programação e algoritmos. Primeiramente é apresentado um paralelo de situações do cotidiano com algoritmos, ao efetuar comparações de instruções passo a passo realizadas diariamente, como fritar um ovo ou trocar um pneu de automóvel. A seguir inicia-se a aplicação de linguagem estruturada em português para descrição dos algoritmos. Em alguns casos, utiliza-se de linguagens em inglês, mo por exemplo, Pascal. Há professores que preferem usar pseudo linguagem em português somente “em papel”, iniciando a codificação diretamente em linguagem C. Tem-se inclusive, também “em papel”, de analisar e interpretar os passos do algoritmo para saber quais serão os valores em memória e a saída na tela.

Simultaneamente, com a prática escrita, sem software específico, pode-se simular os mesmos algoritmos em interpretador ou compilador. Sendo em um interpretador sua construção dividida em partes, cada uma com uma função específica. Geralmente identifica-se: o analisador léxico, o analisador sintático, o analisador semântico, e o resultado de saída. Caso seja um compilador há, ao invés de resultado de saída, o gerador de código, que transforma o programa em linguagem de máquina composta de 0s e 1s (DELAMARO, 2004).

Ao utilizar um programa em navegador via Internet, fazendo com que seja desnecessário instalar e utilizar sempre o mesmo computador, objetiva-se que os exercícios serão resolvidos com maior facilidade. Também é importante que as mensagens de erro sejam claras, e em português já que torna mais acessível a leitura para quem ainda não possui facilidade com a língua inglesa.

A criação do protótipo desse sistema é o ponto principal desse trabalho, pois os objetivo é substituir *softwares* que vêm sendo utilizados atualmente por um prático, atual, acessível e com o código disponível para que possa ser corrigido e melhorado.

Nesse protótipo busca-se eliminar algumas deficiências que existem em ferramentas similares, pois essas podem fazer com que o seu uso seja desgastante e desanimador. Além de propor a inclusão de funcionalidades que poderiam estar disponíveis, as quais são encontradas por exemplo em Ambientes Integrados de Programação, comumente utilizada em outras linguagens.

Inclusive há caso em que a ferramenta inicialmente utilizada não é capaz de executar alguns algoritmos mais avançados. E isto leva a necessidade do uso de outro software na mesma disciplina até com possivelmente uma nova linguagem (por exemplo a linguagem de programação C ou C++), devido a está necessidade alguns educadores optam por esta sendo a única ferramenta utilizada, ignorando linguagem Portugol.

1.3 ESTRUTURA DO TRABALHO

No presente trabalho apresentar-se-á a forma de desenvolvimento do primeiro módulo, o interpretador, a interface com o usuário e diretrizes para a execução do módulo restante que não será executado, e que poderão ser ampliados a partir desta base de estudo e conhecimentos aplicados no presente projeto.

O restante do projeto está organizado da seguinte forma: no capítulo 2 é relatada a forma como foi idealizado o desenvolvimento da ferramenta proposta; no capítulo 3 descreve-se como o módulo interpretador está sendo desenvolvido; nos capítulos 4 e 5 são apresentadas as diretrizes que serão seguidas para os desenvolvimentos dos módulos Interface Homem-Máquina e sugestão de Sistema de Informação para o Professor, respectivamente; no capítulo 5 são apresentados as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ALGORITMOS

2.1.1 Ensino Aprendizagem

2.2 PSEUDOLIGUAGEM E PORTUGOL

2.3 FERRAMENTAS EXISTENTES

2.3.1 UAL e Editor UAL

2.3.2 Demais Aplicativos

2.3.3 Plataforma Digitais de Ensino

Udacity

Codeacademy

SoloLearn

2.4 TECNOLOGIAS WEB

2.4.1 Linguagem de Marcação de Texto

2.4.2 Linguagem de Programação para Web

ECMAScript

2.4.3 Navegadores Web

2.4.4 Aprimoramentos para Uso *Offline*

2.5 COMPILADORES E INTERPRETADORES

2.5.1 Análise Léxica

2.5.2 Análise Sintática

2.5.3 Análise Semantica

3 DESENVOLVIMENTO DA APLICAÇÃO

3.1 FERRAMENTAS UTILIZADAS

3.1.1 Editor de Código ACE

3.1.2 *Parser* Acorn

3.1.3 JS-Interpreter

3.2 PROCESSO CRIACIONAL

3.2.1 Git e Github

3.2.2 Node.js e Grunt

3.2.3 SASS

3.2.4 TDD

3.3 INTERFACE GRÁFICA

3.3.1 *Material Design*

MDL

3.3.2 Inspirações

REFERÊNCIAS

DELAMARO, M. E. **Como Construir um Compilador Utilizando Ferramentas Java.**

São Paulo: Novatec, 2004. Disponível em: <<http://conteudo.icmc.usp.br/pessoas/delamaro/SlidesCompiladores/CompiladoresFinal.pdf>>.

LOPES, A.; GARCIA, G. **Introdução à programação.** Rio de Janeiro: Campus-Elsevier, 2002.