**UNASP**

**CENTRO UNIVERSITÁRIO ADVENTISTA DE SÃO PAULO**

**CAMPUS SÃO PAULO**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

****

**PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO**

**Simulador de Máquina de Turing: Uma Aplicação Web para o Ensino**

Aluno

**Wellmmer Lucas de Oliveira Pinto**

Orientador

**Prof. Ms. Laercio Martins Carpes**

São Paulo, SP

Setembro de 2016

**SUMÁRIO**

[1 Introdução 1](#_Toc159200186)

[2 Objetivos 2](#_Toc159200193)

[3 Justificativa 2](#_Toc159200196)

[4 Referencial teórico 4](#_Toc159200197)

[5 Metodologia 5](#_Toc159200198)

[6 Recursos e Viabilidade 6](#_Toc159200199)

[7 Cronograma 6](#_Toc159200200)

[8 Referências 7](#_Toc159200201)

[9 Anexos 8](#_Toc159200210)

**Introdução**

De acordo com o *American Heritage Dictionary* (2016), a palavra “algoritmo” é “um processo de solução de um problema passo a passo, especialmente um procedimento computacional recursivo estabelecido para a solução de um problema num número de passos finitos.”. Embora hoje conhecemos muito bem a palavra “algoritmo” e todo seu significado, voltemos para o século XIII onde tudo isso estava no escuro da mente de alguns homens ousados, dos quais chamamos hoje de gênios.

No terceiro capítulo do livro *“O homem que sabia demais”*, de David Leavitt (2007), é esboçada uma introdução muito detalhada da história por trás do *Entscheidungsproblem*, ou melhor dizendo, o problema de decisão. Começando por Raimundus Lullus (1232-1316) e toda sua tese em cima do método geral de solução de problemas, intitulado por ele de *ars magna.* Mais tarde ampliado pelos estudos de Leibniz (1646-1716), resultando no estabelecimento de uma linguagem simbólica que efetivasse a solução do problema, a *characteristica universalis*, e na distinção de duas versões da *ars magna:* a *ars inveniendi*, encontrando as verdadeiras afirmações científicas, e a *ars iudicandi*, permitindo a decisão se uma afirmação científica é verdadeira ou não.

Quase um século depois, o matemático alemão David Hilbert (1862-1943) lança para a comunidade de matemáticos da época o *Entscheidungsproblem,* que caía na perspectiva do *ars iudicandi* deLeibniz, no qual, segundo Hilbert poderia se restringir a uma simples questão de “sim” ou “não” para o caso de haver um algoritmo que decida a validade de uma fórmula de primeira ordem. Isso gerou muita polêmica entre a comunidade de matemáticos, quando de um lado tinham matemáticos prós à uma descoberta para o problema de decisão, e do outro matemáticos contra, como o caso do matemático inglês G. H. Hardy (1877-1947), ao dizer que “isso seria uma infelicidade, pois se houvesse [o algoritmo], precisaríamos ter um conjunto mecânico de regras para a solução de todos os problemas matemáticos, e nossas atividades como matemáticos chagariam a um fim.”.

No entanto, havia um jovem britânico que por seu constante isolamento, possivelmente não estava de nenhum dos lados em relação a solução do *Entscheidungsproblem*, porém encarava o problema de decisão uma simples questão que requisitava-se uma solução. Talvez pelo motivo claro de Turing não considerar o problema esperando um possível resultado negativo ou positivo, ele conseguiu enfrenta-lo de uma maneira totalmente inovadora, e foi desse modo tão literal de pensar que Turing apresentou seus resultados, em 1936, no artigo publicado *“On Computable Numbers, with an Application to the Entscheidungsproblem”*, no qual ele define o conceito e as possíveis funcionalidades da “máquina universal”, e desbanca o problema de decisão. (Leavitt, 2007)

Pensando carinhosamente no legado e todo trabalho deixado por Alan Turing, este trabalho procura por objetivo imputar os conceitos e funcionalidades da *“máquina universal”* de Turing em uma aplicação, desenvolvida em POO (Programação Orientada à Objeto), com recursos visuais que ajudem na aprendizagem dos fundamentos da computação deixados por sua descoberta.

**Objetivos**

Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma aplicação auto-explicativa que torne de uma forma prática e didática, através de recursos visuais e interativos, o aprendizado dos conceitos e funcionamentos da “máquina universal” de Turing.

Objetivos Específicos

Para um melhor entendimento dos objetivos deste trabalho, dividiram-se em tópicos específicos o que pretende-se ser realizado.

* Desenvolvimento de uma aplicação que simule todo o conceito e funcionamento da *“máquina universal”* de Turing;
* Desenvolvimento de recursos gráficos e visuais para a aplicação que ajudem a tornar o aprendizado da *“máquina universal”* de Turing prático e didático.

**Justificativa**

Em um breve artigo, escrito sob encomenda e destinado à alunos de Ciência da Computação, o renomado Prof. Dr. Eng. Valdemar W. Setzer, do Depto. de Ciência da Computação da USP, enfatizou claramente:

*“Se o leitor está terminando um Bacharelado em Ciência da Computação, e nunca ouviu falar o nome Turing, é melhor começar tudo de novo em uma faculdade de um nível razoável. Isso se deve ao fato de Alan Mathison Turing (1912-1954) ter estabelecido alguns dos fundamentos mais importantes da Ciência da Computação, tanto do ponto de vista prático quanto teórico e, portanto, deve obrigatoriamente ser mencionado e suas descobertas estudadas em qualquer curso dessa área.” (Setzer, V. W. 2003).*

Seguindo este raciocínio, uma de suas maiores descobertas, hoje precursora de vários estudos na área da Ciência da Computação, foi esboçada e modestamente intitulada *“On Computable Numbers, with an Application to the Entscheidungsproblem”*, no qual é abordado detalhadamente o conceito e o funcionamento do que ele chamou de *“máquina universal”*. Turing tinha como objetivo de reunir esses conceitos e provar a insolubilidade do famoso e polêmico *Entscheidungsproblem* de David Hilbert (1862-1943).

Embora o aprendizado dos conceitos da *“máquina universal”* de Turing, contidos em seu artigo citado acima, sejam imprescindíveis tanto para um ingressante quanto para um formando de Bacharelado em Ciência da Computação, como em qualquer outro curso na área da tecnologia, é viável por antemão ser assumido que a compreensão de tais não é algo tão trivial. O estudo sobre a *“máquina universal”* de Turing é algo que pode parecer simples superficialmente, mas é necessário um aprofundamento técnico matemático para compreender certas representações, equações e símbolos usados por Turing ao explanar suas ideais.

De acordo com David Leavitt, autor do livro *“O homem que sabia demais”*:

*“Como muitos trabalhos de Turing, ‘Computable Numbers’ é marcado por uma curiosa mistura de frases cheias de modéstia, especulação meio filosófica e matemática altamente técnica. O resultado, para um leitor comum, é desconcertante, pois invariavelmente essas passagens, cuja importância é fácil de aprender, passam suave e imediatamente em densos pântanos de símbolos não familiares, letras alemãs e gregas, e números binários.” (Leavitt, 2007).*

Por isso, com o intuito de simplificar o entendimento dos conceitos da *“máquina universal”* de Turing, e tornar a aprendizagem de tal, dinâmica e interativa, este trabalho tem como justificativa a disponibilização de todo o material desenvolvido para servir de uso como referencial teórico e prático em estudos futuros, demonstrações, aulas, etc.

**Referencial Teórico**

## **O artigo sobre a *“máquina universal”* e os *“números computáveis”* de Turing**

A *“máquina universal”* foi apresentada por Alan M. Turing, um matemático, lógico e criptoanalista britânico, em 1936 em seu artigo *“On Computable Numbers, with an Application to the Entscheidungsproblem”*.

Na visão de David Leavitt (2007), o artigo é basicamente dividido em três partes: a primeira definindo o conceito dos *“números computáveis”* e da *“máquina computável”*; a segunda abordando o conceito e as funcionalidades de uma possível *“máquina universal”*; e finalmente a terceira aplicando toda a teoria abordada para provar a insolubilidade do *Entscheidungsproblem*. Para este trabalho, não será relevante mostrar a aplicação da *“máquina universal”* de Turing para provar que o *Entscheidungsproblem* é insolúvel.

### **Os *“números computáveis”***

Turing (1936) começa definindo *“números computáveis”* como os “números reias cujas expressões decimais são calculáveis por meios finitos”, porém ainda no final do primeiro capítulo, embora sem nenhuma ostentação ao dizer, os redefine afirmando ousadamente: *“De acordo com minha definição, um número é computável se seu decimal pode ser registrado por uma máquina.” (Turing, 1936)*

Leavitt (2007) comenta que “a importância de tal afirmação não deve ser subestimada, uma vez que falar de uma hipotética *“máquina”* de computação, ainda mais e um artigo sobre matemática nos anos de 1930, era como violar as regras de uma ortodoxia assumidamente rígida, pois nenhuma dessa possíveis *“máquinas”* existia à época, apenas algumas muito cruas e obviamente não programáveis.”

### **As *“máquinas computáveis”***

Sem tentar justificar muito antes da hora, a razão de sua definição particular para os *“números computáveis”*, Turing (1936) estabelece que “de fato a mente (memória) humana é devidamente limitada” e que podemos “comparar um homem no processo de computação de um número real com uma máquina que é unicamente capaz de um número finito de condições”. Turing chamou a esse número finito de condições de *“configuração-m”*.

De acordo com Turing (1936), a *“máquina”* é alimentada, por assim dizer, com uma “*fita”* percorrendo sua estrutura. A *“fita”* é dividida em seções, como *“células”*, cada uma capaz de receber um *“símbolo”*. A *“célula”* que, no momento, estiver na *“máquina”* é chamada de *“célula registrada”,* quando que o *“símbolo”* da *“célula registrada” é* chamado *“símbolo registrado”*. O possível comportamento da *“máquina”*, para qualquer situação, pode ser determinado pela *“configuração-m”* juntamente com o *“símbolo registrado”*, definindo a *“configuração da máquina”.* Leavitt (2007) resume os possível comportamentos da máquina comentando que:

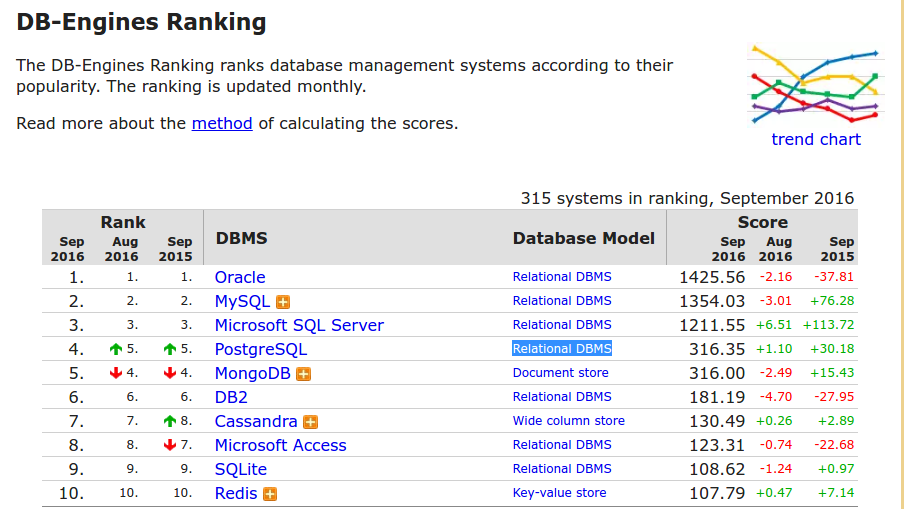
*“Dependendo de sua configuração a máquina vai escrever um símbolo numa célula em branco, apagar um símbolo já escrito lá, mover a fita um espaço para esquerda ou mover a fita um espaço para direita. O que determina como ela irá agir é uma ‘tabela de comportamento’ especificando a sequência das configurações-m de acordo com as quais a máquina pode executar seu algoritmo particular”.*

**Metodologia**

Primeiramente será feito um levantamento das possíveis funcionalidades da aplicação, seguindo o tradicional modelo de engenharia de software, ou mais conhecido como modelo cascata criado por Royce em 1970, justo selecionado pela facilidade de implementação e organização do cronograma de atividades. Depois será realizado um estudo dos casos de uso, apresentado a primeira vez por Jacobson em meados de 1980, para descrição de cenários e funcionalidades de possíveis interações com um usuário ou qualquer possível sistema externo.

Enfim, após toda a parte de análise do projeto, serão desenvolvidos e implementados cada conceito e funcionalidade da *“máquina universal”* de Turing, tendo como aplicação final o *Simulador da Máquina Universal de Turing.* Devido à uma melhor portabilidade do projeto e uma abrangente opção de ferramentas, recursos e plug-ins foi selecionado como IDE (plataforma de desenvolvimento) o Eclipse, em sua versão final *“Neon”*, com plugin Java e JEE*.*

Para o gerenciamento do banco de dados foi selecionada a ferramenta *open source* PostgreSQL. Segundo dados do site DB-Engines Ranking, que podem ser observados na Figura 1 abaixo, o PostgreSQL é atualmente um dos cinco bancos de dados mais populares do mundo, sendo o mais popular entre os *open sources*.

Figura 1 – Raking de DB-Engines

Seguindo para a parte final do trabalho, “o teste de software é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado.” (Dias, 2008) Portanto, serão realizados tanto testes de funcionalidade quanto teste de usabilidade para os recursos visuais.

**Recursos e Viabilidade**

Como recurso necessário para a viabilização deste trabalho será utilizado um computador, ou notebook, no qual será instalado as ferramentas necessárias para o desenvolvimento e implementação dos objetivos propostos. Todas as ferramentas e softwares utilizados no projeto são *open-source* (código aberto) e gratuitos e estão disponíveis nos sites de seus respectivos fabricantes.

**Cronograma**

É de impreterível importância seguir, para um bom andamento do projeto, o seguinte cronograma de atividades, organizado conforme os objetivos propostos, apresentado na Tabela 1 abaixo.

Tabela 1 - Cronograma de Andamento do Projeto

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Atividades** | | | **Meses** | | | |
|  | **Julho** | **Agosto** | | **Setembro** | **Outubro** | **Novembro** |
| **Estrutura do Projeto** |  |  | |  |  |  |
| **Implementação do Projeto** |  |  | |  |  |  |
| **Revisões e Testes** |  |  | |  |  |  |
| **Elaboração do Relatório** |  |  | |  |  |  |
| **Apresentação Final do TCC** |  |  | |  |  |  |

**Referências**

## **Livros**

JACOBSON, I. & Pan-Wei, Ng. **Aspect-Oriented Software Development with Use Cases (Addison-Wesley Object Technology Series)**, Publisher: Addison-Wesley Professional © 2004. ISBN: 0321268881.

LEAVITT, David. **O homem que sabia demais**. Tradução de Samuel Dirceu. 1ª edição. Ribeirão Preto, São Paulo. Editora Novo Conceito Ltda., 2007.

## **Trabalhos e Artigos Publicados**

SETZER, V. W. **“Alan Turing e a Ciência da Computação”.** Depto. de Ciência da Computação da USP.Disponível em: https://www.ime.usp.br/~vwsetzer/Turing-teatro.html. Acesso em 30 de agosto de 2016.

TURING, A. M. **“On Computable Numbers, with an Application to the Entscheidungsproblem”.** Recebido em 28 de Maio de 1936 - Lido em 12 de Novembro de 1936. Proceedings of the London Mathematical Society.

## **Periódicos e Revistas Científicas**

BOEHM, B.l W. **"Software Process Management: Lessons Learned from History"** in ICSE '87 Proceedings of the 9th International Conference on Software Engineering, 1987.

DIAS N., A. C. **Introdução a Teste de Software**. Engenharia de Software Magazine. Nº 1, 2008.

## **Softwares**

**Eclipse Neon**. Version 4.6.0. Copyright © 2000 - 2016 The Eclipse Foundation. All Rights Reserved.

**PostgreSQL.** Version 9.6. RC 1. Copyright © 1996 - 2016 The PostgreSQL Global Development Group.

## **Homepages**

American Heritage Dictionary. **AHDictionary of The English Language.** Copyright 2016 Houghton Mifflin Harcourt. All rights reserved. Disponível em: [https://ahdictionary.com](https://ahdictionary.com/word/search.html?q=algorithm&submit.x=69&submit.y=13) - Acesso em 30 de agosto de 2016.

DB-Engines Ranking. **Knowledge Base of Relational and NoSQL Database Management Systems.** 2016. Disponível em: http://db-engines.com/en/ranking - Acesso em 30 de agosto de 2016.

**Anexos**

Reserva de Recursos do Núcleo de Computação

RESERVA DE RECURSOS DO NÚCLEO DE COMPUTAÇÃO PARA TCC

Informamos que o acadêmico \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ solicitou para desenvolvimento de seu Trabalho de Conclusão de Curso intitulado \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ os recursos computacionais abaixo listados os quais foram reservados e serão disponibilizados oportunamente pelo Núcleo de Computação do UNASP.

Recursos computacionais solicitados

1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

São Paulo, \_\_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_\_\_ de \_\_\_\_\_\_\_.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Núcleo de Computação