

Título / Title:	Journal of Exact Sciences
Título abreviado/ Short title:	J. Ex. Sci.
Sigla/ Acronym:	JES
Editora / Publisher:	Master Editora
Periodicidade / Periodicity:	Trimestral / Quarterly
Indexação / Indexed:	Latindex, Google Acadêmico
Início / Start:	Abril, 2014/ April, 2014

Editor-Chefe / Editor-in-Chief:

Prof. Dr. Mário dos Anjos Neto Filho [MS; Dr; PhD]

Conselho Editorial:

Profa. Dra. Andréia Fátima Zanette- FEITEP – Maringá – PR – Brasil

Prof. Dr. Vinícius Vaulei Gonçalves Mariucci- FEITEP – Maringá – PR – Brasil

Prof. Dr. João Ricardo Nickenig Vissoci- Duke Global Health Inst - NY - EUA; UNINGÁ – Maringá – PR – Brasil

Prof. MS. Lupércio Cascone- FEITEP – Maringá – PR – Brasil

Prof. MS. Odete Bulla Cascone- FEITEP – Maringá – PR – Brasil

Prof. Dr. Julio Cesar Tocacelli Colella- Uningá – Maringá – PR- Brasil

O periódico **Journal of Exact Sciences – JES** é uma publicação da **Master Editora** para divulgação de artigos científicos apenas em mídia eletrônica, indexada à base de dados **Latindex** e **Google Escolar**. Todos os artigos publicados foram formalmente autorizados por seus autores e são de sua exclusiva responsabilidade. As opiniões emitidas pelos autores dos artigos publicados não correspondem necessariamente, às opiniões da Master Editora, do periódico **JES** e /ou de seu conselho editorial.

*The **Journal of Exact Sciences - JES** is an editorial product of **Master Publisher** aimed at disseminating scientific articles only in electronic media, indexed in **Latindex** and **Google Scholar** databases.*

All articles published were formally authorized by the authors and are your sole responsibility.

*The opinions expressed by the authors of the published articles do not necessarily correspond to the opinions of Master Publisher, the **JES** and/or its editorial board.*



Prezado leitor,

*Temos a imensa satisfação de lançar a terceira edição do **Journal of Exact Sciences - JES***

*A **Master Editora** e o periódico **JES** agradecem publicamente aos Autores dos artigos que abrilhantarão esta terceira edição pela colaboração e pela confiança depositada neste projeto. O periódico **JES** é um dos primeiros “open access journal” do Brasil, representando a materialização dos elevados ideais da **Master Editora** acerca da divulgação ampla e irrestrita do conhecimento científico produzido pelas diversas áreas das Ciências Exatas.*

Aos autores de artigos científicos que se enquadram em nosso escopo, envie seus manuscritos para análise de nosso conselho editorial!

Nossa quarta edição estará disponível a partir do mês de Janeiro de 2015!

Boa leitura!

Mário dos Anjos Neto Filho
Editor-Chefe JES

Dear reader,

*We have the great pleasure to launch the third edition of the **Journal of Exact Sciences - JES**.*

*The **Master Publisher** and the **JES** are very grateful to the authors of the articles that brighten this third edition by the trust placed in this project. The **JES** is one of the early open access journal in Brazil, representing the materialization of the lofty ideals of **Master Publisher** about the broad and unrestricted dissemination of scientific knowledge produced by the Exact Sciences.*

*Authors of scientific articles that are interested in the scope of **JES**, send their manuscripts for consideration of our editorial board!*

Our fourth edition will be available in January, 2015!

Happy reading!

Mário dos Anjos Neto Filho
Editor-in-Chief JES

Química

IMOBILIZAÇÃO DE LIPASES REGIOSSELETIVAS EM MATRIZES SOL-GEL E REAÇÃO PARCIAL DEHIDRÓLISE

ANDRÉIA FÁTIMA ZANETTE, JAMAL AWADALLAK, LÚCIO CARDOZO FILHO..... 05

Informática

REDES NEURAIS E SUA CONFIABILIDADE E DESEMPENHO EM PERÍODOS INSTÁVEIS – APLICAÇÃO A AÇÕES ORDINÁRIAS PETR3

AFONSO GENTA PALANDRI, FILIPE ROSEIRO COGO 09

O IMPACTO DO TAMANHO DO BUFFER DE BUSCA DE INSTRUÇÕES EM RELAÇÃO À LARGURA DO ESTÁGIO DE BUSCA EM UMA GPU

JOSÉ CARLOS SANCHES AMENDOA TISSEI, JEFERSON APARECIDO SILVA BELGAMAZZI, LUCAS DE OLIVEIRA TEIXEIRA, CLAUDIA HEIDEMANN DE SANTANA.... 17

A IMPORTÂNCIA DA EXPERIMENTAÇÃO EMPÍRICA NA PESQUISA EM CIÊNCIA DA COMPUTAÇÃO

ANDERSON DA SILVA MARCOLINO, HELIO HENRIQUE LOPES COSTA MONTE-ALTO, LUCAS DE OLIVEIRA TEIXEIRA, JEFERSON APARECIDO SILVA BELGAMAZZI, JOSÉ CARLOS SANCHES AMENDOA TISSEI, CLAUDIA HEIDEMANN DE SANTANA..... 21

Normas para publicação

Tutorial para publicação de artigos no Journal of Exact Sciences..... 28

IMOBILIZAÇÃO DE LIPASES REGIOSSELETIVAS EM MATRIZES SOL-GEL E REAÇÃO PARCIAL DE HIDRÓLISE

IMMOBILIZATION OF LIPASES REGIOSSELETIVAS DIES IN SOL-GEL AND PARTIAL HYDROLYSIS REACTION

ANDRÉIA FÁTIMA ZANETTE^{1*}, JAMAL AWADALLAK², LÚCIO CARDOZO FILHO³

1. Doutora em Engenharia Química pela Universidade Estadual de Maringá – UEM, docente da Faculdade de Engenharia e Inovação Técnico Profissional – FEITEP; 2. Doutorando em Engenharia Química pela Universidade Estadual de Maringá; 3. Doutor em Engenharia de Alimentos pela Universidade Estadual de Campinas – UNICAMP, docente titular do curso de Engenharia Química da Universidade Estadual de Maringá – UEM.

* Rua Iguaçú, 726, Maringá, Paraná, Brasil. CEP 87020-330. andreiazanette@yahoo.com.br

Recebido em 03/11/2014. Aceito para publicação em 06/11/2014

RESUMO

Óleos vegetais enriquecidos com diacilglicerol possuem características organolépticas semelhantes às dos óleos comestíveis convencionais, com a vantagem de possuírem a tendência de não se acumular no organismo, reduzir os níveis de triacilgliceróis e, como consequência, diminuem a massa corporal e gordura visceral nos humanos, contribuindo para a diminuição da obesidade. Visto que os processos enzimáticos são, em geral, preferidos aos métodos químicos, este trabalho visa imobilizar e encapsular enzimas livres comerciais em matrizes hidrofóbicas sol-gel preparadas a partir de dois precursores, o tetraetilortossilicato (TEOS) e o tetrametoxissilano (TMOS) visando sua utilização na produção de diacilgliceróis a partir do óleo de palma e outros óleos vegetais. Os biocatalisadores foram caracterizados morfológicamente por meio da microscopia eletrônica de varredura (MEV), da qual se verificou que o TMOS é o melhor precursor para obtenção de biocatalisadores imobilizados. A eficiência na imobilização das enzimas foi avaliada por meio de uma reação de hidrólise do óleo de palma.

PALAVRAS-CHAVE: Imobilização, lipases, sol-gel.

ABSTRACT

Diacylglycerol enriched with vegetable oils have organoleptic characteristics similar to those of conventional edible oils, with the advantage of not having the tendency to accumulate in the body, reduce triglyceride levels and as a consequence, decrease body weight and visceral fat in humans, helping to reduce obesity. Whereas the enzymatic processes are generally preferred to chemical methods, this work aims to immobilize and encapsulate commercial free enzymes in hydrophobic sol-gel matrices prepared from two precursors, tetraethylorthosilicate (TEOS) and tetramethoxysilane (TMOS) seeking its use in the production of diglycerides from palm oil and other vegetable

oils. The biocatalysts were morphologically characterized by scanning electron microscopy, from which it is found that the best TMOS precursor for obtaining immobilized biocatalysts electron microscopy. The efficiency of the immobilization of the enzymes was evaluated by a hydrolysis reaction of oil palm.

KEYWORDS: Immobilization, lipases, sol-gel.

1. INTRODUÇÃO

Os óleos comestíveis são formados basicamente por triacilgliceróis (TAG), sendo composto também por uma pequena quantidade de monoacilgliceróis (MAG) e diacilgliceróis (DAG)¹.

Estudos recentes sobre as propriedades nutricionais e efeitos dietéticos dos óleos comestíveis sugerem que a presença de DAGs desempenha um importante papel na redução dos níveis de TAG e, em consequência, diminuem a massa corporal e gordura visceral nos humanos, prevenindo a obesidade e outras doenças relacionadas com o estilo de vida². Os óleos enriquecidos com DAGs não apresentam nenhum efeito genotóxico no organismo humano³.

Eom *et al.* (2010)⁴ investigaram o efeito de DAG sintetizado do óleo de atum no peso corporal e marcadores bioquímicos de plasma na obesidade de camundongos C57BL/6J. O consumo da dieta com DAG em camundongos ocasionou um menor ganho de peso corporal e colesterol plasmático total final, triglicerídeos e anemia quando comparado com um grupo de camundongos ingeriu um alto conteúdo de TAG.

Nos processos químicos para a produção de DAG são utilizadas altas temperaturas (200-250 °C) e pressão, juntamente com catalisadores inorgânicos tais como hidróxido de sódio, hidróxido de potássio ou cálcio. Esses

processos, além de exigirem uma alta demanda de energia, geram mudança de coloração e gosto desagradáveis ao produto final⁵.

Devido a essas limitações, os DAGs são preferencialmente obtidos por processos enzimáticos, pois as reações são realizadas em condições brandas, com baixa formação de subprodutos e alta velocidade. Também oferecem vantagens ambientais e redução no consumo de energia, além disso, as lipases apresentam alta seletividade, incluindo seletividade estereoquímica e resultam em produtos de alta pureza e melhor qualidade⁴.

As lipases são enzimas que catalisam reações de esterificação, interesterificação, hidrólise, entre outras. São enzimas altamente seletivas, sendo biocatalisadores aplicáveis à produção de diversos produtos.

Eom *et al.* (2010)⁴ também produziram DAG pela glicerólise com glicerol e óleo de atum utilizando lipases *Rhizomucor miehei* (Lipozyme RM IM) e *Candida Antarctica* (Novozyme 435), sendo que a Lipozyme RM IM foi escolhida para a produção de DAG devido a sua alta especificidade verificada através de análise por HPLC.

Watanabe *et al.* (2005)⁶ estudaram a produção de 1,3-diacilglicerol (1,3-DAG) através da esterificação dos ácidos graxos (FA) e glicerol a partir do óleo de soja em biorreator de leito empacotado usando uma lipase Lipozyme RM IM (*Rhizomucor miehei*) 1,3-regiosseletiva (5% em relação à massa de óleo) imobilizada em resina de troca aniônica microporosa. A esterificação foi realizada através da circulação da mistura reacional entre uma coluna de leito empacotado e uma bomba para remoção da água. Obteve-se um rendimento em diacilglicerol em torno de 70% em 3 horas de reação.

As enzimas podem ser adquiridas comercialmente imobilizadas ou pode-se aplicar técnicas de imobilização, tal como citado por Zarcula *et al.* (2010)⁷ que imobilizaram a lipase de *Pseudomonas fluorescens* (Amano AK) pelo método sol-gel utilizando tetrametoxosilano e trimetoxosilano com grupos alquila ou arila como precursores e líquidos iônicos como aditivos para a imobilização, resultando em biocatalisadores com alta eficiência catalítica. A atividade do biocatalisador foi totalmente recuperada depois da imobilização.

Uma das principais propriedades do método sol-gel é reduzir a velocidade de desnaturação das biomoléculas encapsuladas, provavelmente devido à alta reatividade dos precursores capazes de alterar a superfície da proteína, protegendo-as de efeitos desnaturantes, orientando a acessibilidade do substrato ao sítio ativo da enzima imobilizada no reticulado sol-gel⁸.

O encapsulamento através de sol-gel provou ser uma técnica versátil para a imobilização de uma grande variedade de biomoléculas⁹. Além disso, estudos demonstraram que a atividade da enzima apresenta melhores resultados quando a secagem é realizada em meio super-

crítico, também chamada de secagem por aerogel¹⁰. O material sol-gel oferece baixa resistência difusional, conduzindo a bons resultados em reações de esterificação. A preparação, separação do produto e conservação das propriedades enzimáticas são algumas das vantagens apresentadas por esses sistemas¹¹.

Diante deste contexto, o objetivo deste trabalho foi imobilizar e encapsular enzimas livres comerciais em matrizes hidrofóbicas sol-gel preparadas a partir de dois precursores, o tetraetilortossilicato (TEOS) e o tetrametoxosilano (TMOS) visando sua utilização na produção de diacilgliceróis a partir do óleo de palma.

2. MATERIAL E MÉTODOS

Neste trabalho foram utilizadas as lipases *Rhizopus oryzae* e *Amano M* adquiridas comercialmente da Sigma-Aldrich. Os suportes foram preparados pela técnica sol-gel, empregando os seguintes precursores: tetraetilortossilicato-TEOS (C₈H₂₀O₄Si), metiltrimetoxosilano-MTOS (C₇H₁₈O₃Si) adquiridos da Sigma-Aldrich. Os demais reagentes utilizados foram: heptano PA (Vetec, Brasil), pentano PA (Vetec, Brasil); acetona PA (Merck, Alemanha); etanol comercial; goma arábica em pó, pura (Synth, Brasil). Os reagentes foram utilizados como recebido, sem nenhum tratamento prévio.

O preparo dos biocatalisadores baseou-se nas metodologias descritas por Macario (2009)¹² e Soares (2006)⁸.

Os biocatalisadores imobilizados foram secados utilizando a tecnologia supercrítica (CO₂) a uma temperatura de 40°C para evitar a desnaturação da enzima e a uma pressão de 100 bar por 2 horas.

A caracterização microestrutural dos biocatalisadores imobilizados foi realizada através da microscopia eletrônica de varredura (MEV) em um microscópio Shimadzu SuperScan SS-550. Todas as micrografias foram obtidas das superfícies de fratura recobertas com ouro.

As enzimas livres e imobilizadas foram avaliadas por meio da reação de hidrólise parcial dos triglicerídeos. A hidrólise do óleo de palma tem como produtos finais o glicerol e ácidos graxos livres, e como intermediários DAG e MAG, que consistem em TAG sem um ou dois ácidos graxos, respectivamente¹³. As reações foram conduzidas em meio livre de solvente, a 55°C¹⁴, por meio de titulação com solução de hidróxido de sódio. A avaliação das reações nesta etapa foram baseadas na quantificação dos ácidos graxos livres gerados pelo método de titulação, uma vez que a técnica de detecção desse produto é muito mais simples. Os tempos reacionais avaliados foram de 5, 10, 15 e 20 minutos.

3. RESULTADOS E DISCUSSÃO

A Figura 1 mostra as micrografias dos biocatalisadores livres (letras A e D) e imobilizados (letras B, C, E, F).

Pode-se observar que a enzima lipase *Rhizopus oryzae* livre (A) possui uma forma geométrica definida, claramente visível. Ao se imobilizar esta enzima com TMOS (B) verificou-se uma estrutura porosa, com uma área superficial muito menor, diferente do ocorrido ao utilizar o precursor TEOS (C), em que a estrutura pareceu rígida, levando a conclusão que apenas o suporte está presente. Analisando estas microscopias verifica-se uma mudança muito grande na estrutura da enzima antes e depois de imobilizada, e não observando a presença da enzima ao utilizar o TEOS como precursor de sílica.

Ao analisar a lipase Amano M livre (D) verifica-se que ela apresenta uma estrutura esponjosa, bastante porosa, que manteve suas características após a imobilização por TMOS (E). Ao utilizar o TEOS (F), novamente verifica-se apenas a presença de uma estrutura rígida, que aparenta ser apenas do suporte.

A análise da atividade hidrolítica das enzimas confirma o que foi observado nas microscopias: as enzimas acabaram perdendo sua conformação inicial, ocasionando na perda da atividade.

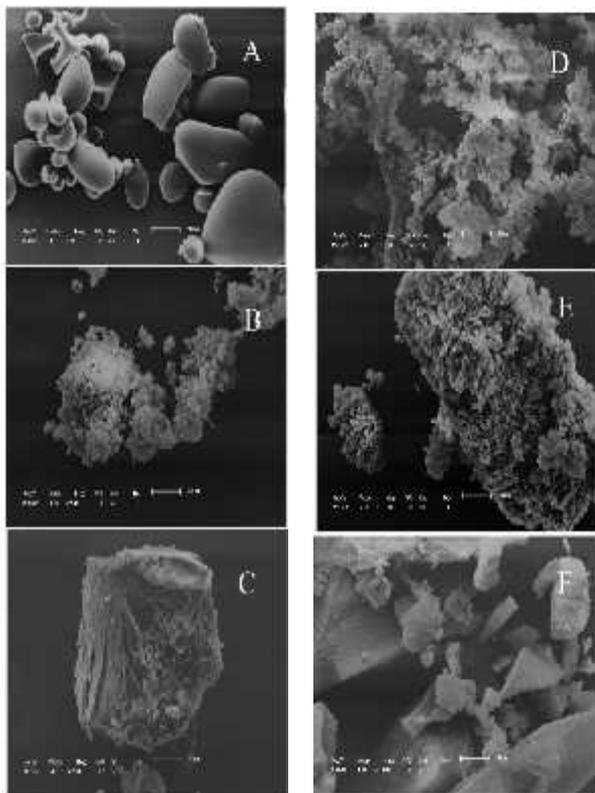


Figura 1. Micrografias eletrônicas dos biocatalisadores: (A) *Rhizopus oryzae* livre (500 X), (B) *Rhizopus oryzae* imobilizada com TMOS (500 X), (C) *Rhizopus oryzae* imobilizada com TEOS (500 X), (D) Amano M livre (1000 X), (E) Amano M imobilizada com TMOS (1000 X), (F) Amano M imobilizada com TEOS (1000 X).

A acidez inicial do óleo de palma foi em torno de 4% e por meio da diminuição da acidez do óleo é possível calcular a atividade das enzimas utilizadas nas reações.

Em geral, as enzimas imobilizadas praticamente não reagiram com 20 minutos de reação, pois a atividade obtida foi próxima a zero, verificado pela não alteração da acidez inicial do óleo de palma.

As enzimas não imobilizadas apresentaram boa atividade catalítica, reduzindo em 15% a acidez inicial do óleo em 5 min de reação.

4. CONCLUSÃO

As técnicas de imobilização de lipases aplicadas neste trabalho apresentaram características distintas. Visivelmente, a técnica que empregou o TEOS não obteve resultados satisfatórios. Já a técnica utilizando o TMOS apresentou uma característica esponjosa, que possivelmente tenha imobilizado parte da enzima. A aplicação da reação de hidrólise parcial do óleo de palma constatou a baixa atividade das enzimas, com o emprego das técnicas utilizadas. Apesar do baixo desempenho nas reações, as técnicas de imobilização de enzimas são importantes aliados na aplicação de enzimas em processos industriais. As vantagens no uso de enzimas imobilizadas incentivam e justificam pesquisas mais detalhadas.

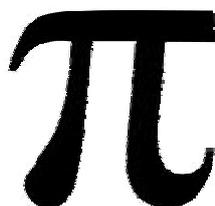
5. FINANCIAMENTO

Agradecemos à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo auxílio financeiro concedido.

REFERÊNCIAS

- [1] Malcata FX, Reyes HR, Garcia HS, Hill Jr, CG, Amundson CH. "Immobilized lipase reactors for modification of fats and oils – A Review", *Journal American Oil Chemist' Society*. 1990; 67:890-910.
- [2] Babicz I, Leite SGF, Souza ROMA, Antunes OAC. Lipase-catalyzed diacylglycerol production under sonochemical irradiation. *Ultrasonics Sonochemistry*. 2010; 17:4-6.
- [3] Kasamatsu T, *et al.* Genotoxicity studies on dietary diacylglycerol (DAG) oil, *Food and Chemical Toxicology*. 2005; 43(2):253-60.
- [4] Eom TK, Kong CS, Byun HG, Jung WK, Kim SK. Lipase catalytic synthesis of diacylglycerol from tuna oil and its anti-obesity effect in C57BL/6J mice. *Process Biochemistry*. 2010; 45:738-43.
- [5] Vu PL, Park RK, Lee YJ, Kim YM, Nam HY, Lee JH, Akoh CC, Lee KT. Two-step production of oil enriched in conjugated linoleic acids and diacylglycerol. *Journal American Oil Chemistry Society*. 2007; 84:123-8.
- [6] Watanabe T, Sugiura M, Sato M, Yamadab N, Nakanishi K. Diacylglycerol production in a packed bed bioreactor. *Process Biochemistry*. 2005; 40:637-43.
- [7] Zarcula C, Corici L, Croitoru R, Ursoiu A, Peter F. Preparation and properties of xerogels obtained by ionic liquid incorporation during the immobilization of lipase by the

- sol-gel method. *Journal of Molecular Catalysis B: Enzymatic*. 2010; 65:79–86.
- [8] Soares CMF, Santos OA, Castro HF, Moraes FF, Zanin GM. Characterization of sol-gel encapsulated lipase using tetraethoxysilane as precursor. *Journal of Molecular Catalysis B: Enzymatic*. 2006; 39:69–76.
- [9] V.B. Kandimalla, V.S. Tripathi, And H.X. Ju, Immobilization of biomolecules in sol-gels: biological and analytical applications. *Crit. Rev. Anal. Chem.* v. 36, p. 73–106, 2006.
- [10] Pierre A, Buisson P. Influence of the porous texture of silica gels on the enzymatic activity of lipases in esterification reactions, *Journal of Molecular Catalysis B Enzymatic*. 2001; 11:639-47.
- [11] Soares CMF, Castro HF, Moraes FF, Zanin GM. “Characterization and utilization of *Candida rugosa* lipase immobilized on controlled pore silica”, *Applied Biochemistry and Biotechnology*. 1999; 77-79:745-58.
- [12] Macario A, Moliner M, Corma A, Giordano G. Increasing stability and productivity of lipase enzyme by encapsulation in a porous organic-inorganic system. *Microporous Mesoporous Mater*. 2009; 118:334-40.
- [13] Rooney D, Weatherley LR. The effect of reactions conditions upon lipase catalysed hydrolysis of high oleate sun flower oil in a stirred liquid-liquid reactor. *Process Biochem*. 2001; 36:947-53.
- [14] Voll FAP. Produção e Separação de Diacilglicerol a partir do Triacilglicerol do Óleo de Palma. Tese de doutorado em Engenharia Química da Universidade Estadual de Maringá – UEM, Maringá. 2011.



REDES NEURAIS E SUA CONFIABILIDADE E DESEMPENHO EM PERÍODOS INSTÁVEIS – APLICAÇÃO A AÇÕES ORDINÁRIAS PETR3

NEURAL NETWORKS AND ITS RELIABILITY AND PERFORMANCE IN UNSTABLE PERIODS – UPPLICATION IN ORDINARY SHARES PETR3

AFONSO GENTA PALANDRI^{1*}, FILIPE ROSEIRO COGO²

1. Mestrando em Tecnologia pelo Instituto LACTEC, docente do curso de Engenharia Elétrica da Faculdade de Engenharia e Inovação Técnico Profissional – FEITEP; 2. Mestre em Ciência da Computação pela Universidade Estadual de Maringá – UEM, docente do curso de Engenharia de Controle e Automação do Centro Universitário de Maringá – CESUMAR.

* Praça Manoel Ribas, 36, Apto 181, Maringá, Paraná, Brasil. CEP 87014-120. afonsopalandri@gmail.com

Recebido em 04/11/2013. Aceito para publicação em 05/11/2013

RESUMO

Este trabalho compara e quantifica o desempenho e a confiança de previsões utilizando Redes Neurais durante períodos instáveis. Como ferramenta de trabalho foi escolhida a linguagem Python por ser de simples programação e possuir abundante material de referência. Foi escolhido como série para previsão os preços das ações PETR3, da Petrobrás, com início em 2006 e término em 2011. Esse período é particularmente interessante por englobar a crise de 2008, onde o mercado apresentou grande instabilidade e insegurança. Comparamos assim o desempenho das Redes Neurais durante períodos pré e pós crise, e verificamos se as previsões correspondem com os valores observados. No ano de 2008, os dados apresentam uma instabilidade muito grande, e a previsão teve uma diferença de R\$3,00 em comparação com o valor obtido. Vale dizer que, neste período, a dispersão dos dados é muito grande também, contribuindo ainda mais na instabilidade da previsão. Durante os outros anos, porém, as previsões se aproximaram bastante dos resultados esperados. Em 2011, por exemplo, a diferença foi de R\$0,16, o que mostra que esta técnica produz resultados significantes e relativamente confiáveis. Mas, como qualquer forma de previsão, esta técnica produz uma estimativa de aonde a série vai estar no futuro próximo. Comparando o erro e o tempo computacional necessário desta técnica, observamos erros significativamente melhores, porém uma redução drástica no tempo de treinamento devido ao algoritmo utilizado.

PALAVRAS-CHAVE: Redes neurais artificiais, bolsa de valores, séries temporais.

ABSTRACT

This paper compares and quantifies the performance and reliability of Artificial Neural Networks (ANN's) during unstable periods in the prediction of time series. The chosen time series is the price of the stock PETR3, belonging to Petrobrás, a Brazilian oil company. The difference between the predicted value and the expected value in 2008 is R\$3.00, when the market is quite unpredictable due the 2008 global crash, and the ANN prediction may be unreliable. During the other years, the difference is smaller, and in 2011 the difference is only R\$0.16, showing that this approach produces significant results with relative reliability. However, as any form of prediction, this technique only produces a rough estimate of where the series is heading in the near future. Comparing the training error of this approach with other techniques (for example, using Back propagation as learning algorithm), shows that the error is sensible smaller using RPROP, but the computational time is reduced dramatically.

KEYWORDS: Artificial neural network, stock market, time series.

1. INTRODUÇÃO

Prever uma série temporal é um problema comum em vários domínios da ciência, como por exemplo, demografia, meteorologia, energia. No entanto, prever uma série temporal financeira de forma acurada é ainda uma tarefa trabalhosa¹.

Uma série temporal consiste de observações em uma ou várias variáveis em função do tempo. Neste tipo de sistema, eventos passados influenciam eventos futuros e atrasos no comportamento são prevalentes (por exemplo, uma ação que ocorre agora só exercerá influência no sistema depois de algum tempo), portanto o tempo é uma

dimensão importante numa série temporal².

Um tipo de série temporal é o mercado de ações. Ações são títulos representativos da menor fração do capital de uma empresa. O acionista, portador do título, passa a ser coproprietário com direito a participação nos lucros³. A previsão desses preços é possível devido ao fato dos valores futuros carregarem todas as informações necessárias para efetuar a previsão, sendo necessário desenvolver métodos analíticos e computacionais para efetuar as previsões dessas séries temporais com o máximo de precisão o possível⁴.

Existem várias ferramentas para previsão de séries temporais como, por exemplo, modelos estatísticos (suavização exponencial ou modelos auto regressivos) ou modelos computacionais (redes neurais, lógica nebulosa ou algoritmos genéticos)⁵. O mercado de ações possui um comportamento não linear e sensibilidade elevada. Portanto, a previsão desse tipo de sistema se mostra um desafio constante, tanto na área de estatística (para desenvolver modelos e métodos mais precisos) quanto na área computacional (para processar a quantidade cada vez maior de dados disponíveis).

Com características como a capacidade de aprender por meio de exemplos, generalizar a informação aprendida, auto-organização e processamento temporal, uma Rede Neural pode ser utilizada para prever os resultados de forma satisfatória⁶.

É possível abordar como uma RNA (Rede Neural Artificial) se comporta na previsão de dados turbulentos no mercado de ações. Para tal análise, foi selecionado como intervalo para testes o histórico das ações PETR3, desde o ano de 2007 ao ano de 2011. Este intervalo é particularmente interessante, pois abrange períodos de grande instabilidade, como por exemplo, a crise mundial de 2008, chamada de Grande Recessão de 2008-2012, e períodos que antecedem e sucede a crise⁷.

Mercado de ações

O mercado financeiro, ou mercado de ações, é um sistema complexo, evolucionário, não linear e dinâmico. O campo da previsão financeira é caracterizado pela intensidade, ruído, natureza desestruturada, alto grau de incerteza e relações ocultas entre os dados⁸.

O mercado de ações brasileiro, comumente chamado de BOVESPA, reúne as ações das principais empresas brasileiras de capital aberto. Uma ação é definida como a menor parcela em que se divide uma empresa, sendo que todos os dias são movimentados milhões entre investidores, seguindo a lei básica da oferta e da procura⁹.

Muitos fatores interferem nas finanças, incluindo eventos políticos, condições econômicas gerais e expectativa dos acionistas. Desta forma, prever o preço de um mercado financeiro é bem difícil. Variações, porém, nos preços não são aleatórias. Estas variações se

comportar de uma forma dinâmica e altamente não linear. A pressuposição de que os preços futuros são aleatórios vem meramente da aleatoriedade que envolve um processo não linear e ruidoso⁸.

Com a tecnologia atual, podemos acompanhar essas movimentações em tempo real, utilizando da Internet. Entre as mais diversas empresas com o capital aberto no mercado de ações, podemos citar algumas significativas, como Vale, Petrobras, Gerdau.

Na Figura 1 se observa, no intervalo de 16/11/2006 a 16/11/2011, os valores diários da ação PETR3. Se tratando de um sistema não linear e bastante sujeito a ruídos externos, podemos utilizar técnicas de representação de sistemas não lineares, como por exemplo, as redes neurais artificiais, para tentar prever o futuro deste sistema.

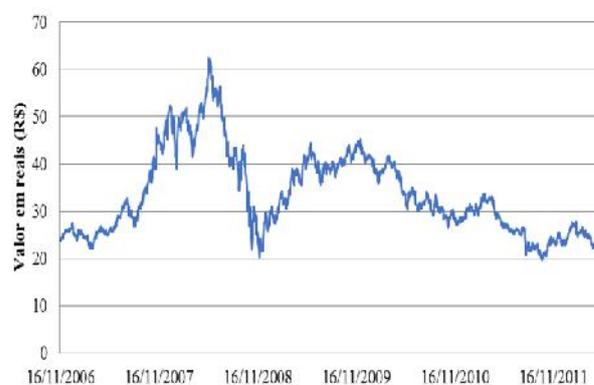


Figura 1. Cotações da ação PETR3 no período de 11/2006 a 05/2012 Redes Neurais Artificiais (Rna).

Redes Neurais Artificiais (RNA) representam uma tecnologia que nasceu de diferentes áreas de pesquisa: neurociência, matemática, estatística, física, ciências da computação e engenharia. As RNAs são empregadas hoje em dia nos mais diversos campos, como modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinal e controle, etc. A vasta aplicabilidade das RNAs deve-se a uma importante característica: a habilidade de aprender a partir dos dados da entrada, com ou sem um supervisor¹⁰.

Uma RNA é constituída de neurônios. Um neurônio é uma unidade de processamento de informação, fundamental para o correto funcionamento de uma rede neural. Como visto na Figura 2, um neurônio consiste de cinco elementos básicos:

- Um conjunto de sinapses ou entradas, representadas em (2) por x_{inp} , caracterizadas pelo peso

ou importância do mesmo.

- Um conjunto de pesos, representados por w_{inp} .

Estes pesos são multiplicados pela sua respectiva entrada.

- Um bloco somador, representado pelo símbolo Σ . Responsável por somar o produto de cada um dos nós ligados a ele.

- Uma função de ativação, ou função de esmagamento, representado pelo símbolo $\varphi(\cdot)$. Este se

encarrega por limitar a amplitude do resultado.

- A saída, y_k , resultante de todas as operações anteriores.

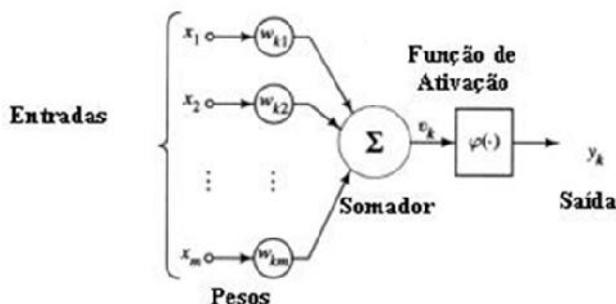


Figura 2. Estrutura Interna de um Neurônio. Adaptado de: http://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/neuronio_artificial.jpg

Um neurônio também pode ser descrito matematicamente, conforme as Equações 1 e 2:

$$u_k = \sum_{j=1}^m w_{kj} * x_j \tag{1}$$

E

$$y_k = \varphi(u_k) \tag{2}$$

Onde:

- x_m é a entrada do nó m.
- w_{km} , o peso da ligação entre o nó m com o nó k.
- u_k representa a soma do produto de todos os nós

ligados ao nó k.

- E y_k , a saída do nó k, dada pela multiplicação de

u_k pela função de esmagamento.

A maneira em que os neurônios de uma rede neural são estruturados está intimamente ligada ao algoritmo de aprendizagem utilizado para treinar a rede¹⁰. Mas antes de falar sobre algoritmos de aprendizagem, falamos sobre o processo de aprendizagem como um todo. No contexto, aprendizagem é definida por:

“Aprendizagem é um processo em que parâmetros livres de uma rede neural se modificam através de um processo de estimulação do ambiente em que a rede está contida. O tipo de aprendizagem é determinado pela maneira em que os parâmetros da rede se modificam¹⁰”.

O processo de aprendizagem se dá em três etapas distintas:

1. A rede é estimulada pelo ambiente.
2. A rede modifica os pesos das conexões entre os nós (w_{km} , os ditos “parâmetros livres”), como

resultado de um estímulo.

3. A rede responde de uma nova forma ao estímulo do ambiente, devido às mudanças que ocorreram em sua estrutura interna.

Um conjunto de regras bem definidas para a solução do problema de aprendizagem é chamado de *algoritmo de aprendizagem*. Existem dois tipos de aprendizagem: supervisionada e não supervisionada.

A aprendizagem supervisionada é caracterizada pela presença de um agente supervisor. Este agente é responsável por comparar o resultado da rede com o resultado esperado. Tanto a rede quanto o agente estão sujeitos ao mesmo ambiente (por exemplo, o ambiente das ações PETR3), porém o ambiente é conhecido para o supervisor. Desta forma, o mundo deve ser totalmente observável. O supervisor sabe que, dado o conjunto de entrada (1,1,1), a resposta a este estímulo deve ser 111. E a partir da comparação entre o resultado esperado e o resultado obtido pela rede, ocorrerá o ajuste dos pesos das conexões, até que a rede seja capaz de responder aos estímulos como esperado, sem o auxílio do agente supervisor.

Já a aprendizagem não supervisionada não possui um agente supervisor. A rede é incapaz de saber se o resultado gerado está correto, ou ainda se está no caminho certo. Desta forma, aprendizagem não supervisionada é utilizada em problemas de classificação, onde a rede tem como objetivo agrupar tipos de dados similares e identificar exceções nos dados. A aprendizagem não supervisionada é utilizada em ambientes parcialmente observáveis, e nas mais diversas tarefas que envolvam classificar dados. Por exemplo, na detecção de dados incorretos ou fraudulentos.

Para o problema da previsão das ações PETR3, por se tratar de um ambiente totalmente observável, foi utilizado aprendizagem supervisionada. Dentre os algoritmos de aprendizagem supervisionada, citamos dois tipos, o algoritmo *Backpropagation* e o algoritmo *Resilient Propagation*.

O algoritmo *Backpropagation* (BPROP) é o algoritmo mais comumente utilizado para aprendizagem supervisionada com redes MLP. A ideia básica deste

algoritmo é aplicar repetidamente a Equação 3 para calcular a influência de cada peso na rede a respeito de uma função erro arbitrária E ¹¹.

$$\frac{\partial E}{\partial w_{km}} = \frac{\partial E}{\partial s_k} \frac{\partial s_k}{\partial net_k} \frac{\partial net_k}{\partial w_{km}} \quad (3)$$

Onde w_{km} é o peso da conexões entre o nó k e o nó m; s_k é o resultado e net_k é a soma das entradas multiplicadas pelos respectivos pesos do nó k. Uma vez que a derivada parcial para cada peso é conhecida, o objetivo de minimizar a função erro é alcançado ao executar uma gradiente descendente, demonstrado na Equação 4¹¹.

$$w_{km}(t + 1) = w_{km}(t) - \epsilon \frac{\partial E}{\partial w_{km}} \quad (4)$$

Onde ϵ é a taxa de aprendizagem. A escolha da taxa de aprendizagem tem um efeito importante no tempo necessário para alcançar a convergência. Se for muito pequeno, muitas épocas serão necessárias para alcançar uma solução aceitável. Ao contrário, uma taxa de aprendizagem grande vai possivelmente levar a oscilação do processo de treinamento, impedindo que o erro diminua depois de certo valor¹¹.

O algoritmo *Resilient Propagation* (RPROP) executa uma adaptação direta na etapa de mudança dos pesos baseado em informações de um gradiente local. Para alcançar isso, foi introduzido para cada peso um valor de atualização individual, Δ_{km} , que determina o tamanho

da mudança do peso. Esta mudança de valores adaptativa evolui durante o processo de treinamento baseado em uma visão local da função erro E , de acordo

com a seguinte regra de aprendizagem:

$$\Delta_{km}^{(t)} = \begin{cases} \eta^+ * \Delta_{km}^{(t-1)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t-1)} * \frac{\partial E}{\partial w_{km}}^{(t)} > 0 \\ \eta^- * \Delta_{km}^{(t-1)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t-1)} * \frac{\partial E}{\partial w_{km}}^{(t)} < 0 \\ \Delta_{km}^{(t-1)}, & \text{senão} \end{cases} \quad (5)$$

$$w_{km}^{(t+1)} = w_{km}^{(t)} + \Delta w_{km}^{(t)} \quad (6)$$

Verbalizando, esta regra funciona da seguinte forma: toda vez que a derivada parcial do peso correspondente

w_{km} muda de sinal indica que a última mudança foi muito grande e que o algoritmo pulou sobre um mínimo local, e seu valor de atualização diminui por um fator η^- . Se a derivada mantém seu sinal, o valor de atualização é aumentado levemente de forma a acelerar a convergência em regiões rasas¹¹.

Uma vez que o valor de atualização foi adaptado para cada peso, o tamanho da mudança de peso executa uma regra simples: se a derivada é positiva (isto é, o erro está aumentando), subtrai-se do peso seu valor de atualização (como visto em 5). Se a derivada é negativa, adiciona-se ao peso seu valor de atualização (regra 6)¹¹.

$$\Delta w_{km}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)}, & \text{se } \frac{\partial E}{\partial w_{km}}^{(t)} < 0 \\ 0, & \text{senão} \end{cases} \quad (7)$$

$$w_{km}^{(t+1)} = w_{km}^{(t)} + \Delta w_{km}^{(t)} \quad (8)$$

Porém há uma exceção: se a derivada parcial mudar de sinal (por exemplo, o passo anterior foi muito grande e o mínimo foi perdido), inverte-se o $\Delta w_{ij}^{(t)}$, como

mostrado na Equação 9.

$$\Delta w_{km}^{(t)} = -\Delta w_{km}^{(t)}, \quad \text{se } \left(\frac{\partial E}{\partial w_{km}}^{(t-1)} * \frac{\partial E}{\partial w_{km}}^{(t)} \right) < 0 \quad (9)$$

O algoritmo RPROP, em comparação ao BPROP, não somente é mais rápido, mas também é mais simples para ser implementado. Em alguns casos, o RPROP chega a ser quatro vezes mais rápido que o BPROP. Sua simplicidade é sua característica mais marcante, pois para um treinamento baseado no algoritmo BPROP ser considerado aceitável, são necessários exaustivos testes com sua taxa de aprendizagem, pois não há forma de determinar analiticamente um valor de ϵ ótimo¹¹.

Existem três tipos principais de topologias de redes, as quais se referem às ligações que são realizadas entre os neurônios:

1. *Feedforward* com uma única camada, ou *Single-Layer Feedforward*, ou ainda conhecida como *Perceptrons*. Embora seja chamada de camada única, possui duas camadas (camada de entrada e camada de

saída), porém como não há processamento na camada de entrada, somente contabiliza a camada de saída¹⁰.

2. *Feedforward* com múltiplas camadas, ou *Multilayer Feedforward*, ou ainda *Multilayer Perceptron*, representada abaixo na figura (3). São conhecidas por conter camadas internas, representadas na figura pela camada *Hidden*, que por sua vez possuem neurônios ocultos. A função desses neurônios ocultos é intervir entre a entrada e a saída de alguma maneira útil. Ao adicionar camadas ocultas, a rede é capaz de extrair estatísticas de maior ordem, e capaz de representar sistemas não lineares. Essa característica é particularmente valiosa quando o tamanho da camada de entrada é grande¹⁰.

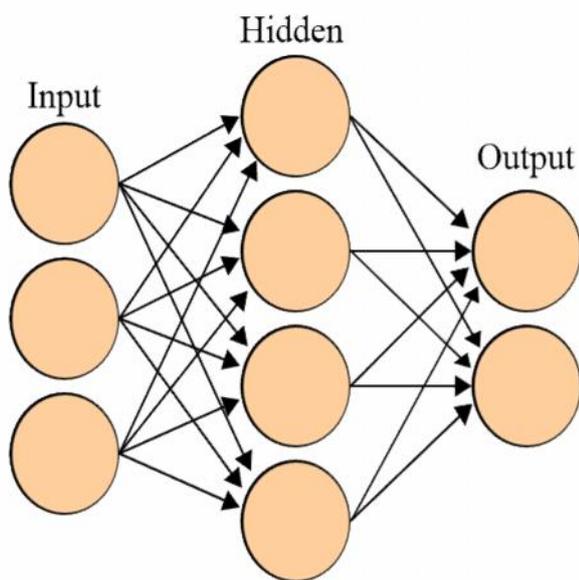


Figura 3. Exemplo de uma rede MLP (3,4,2). Fonte: http://upload.wikimedia.org/wikipedia/commons/e/e4/Artificial_neural_network.svg

3. Redes recorrentes, ou *Recurrent networks*. São redes que possuem pelo menos um loop de *realimentação*. Por exemplo, uma rede recorrente pode consistir de uma única camada de neurônios onde cada neurônio de entrada é alimentado pelo sinal de saída da própria rede.

O objetivo deste trabalho foi observar e comparar a confiabilidade e o desempenho, por meio de métricas como: o tempo de treinamento no caso do desempenho, o erro de treinamento e a diferença entre a previsão e o valor observado de uma RNA na previsão dos valores das ações PETR3, durante o período entre 2007 e 2011.

Foi necessário efetuar um estudo piloto, para delimitar o espectro de testes. No estudo piloto, foi testado mais de 200 redes diferentes, modificando topologia, função de esmagamento e algoritmo de aprendizagem, em busca de uma configuração aceitável.

2. MATERIAL E MÉTODOS

Para definir os parâmetros de configuração da Rede Neural foi feito um estudo piloto cujo objetivo foi comparar diferentes configurações para o mesmo problema. Neste estudo, foram comparados diversos componentes que compõem uma RNA, totalizando 256 redes distintas. Os componentes testados foram:

- A quantidade de camadas ocultas na RNA.
- O número de nós ocultos em cada camada oculta.
- A quantidade de nós de entrada.
- A função de ativação.
- O algoritmo de aprendizagem.
- A normalização ou não dos dados no intervalo [0,1].

A configuração apresentada abaixo foi a que apresentou a melhor relação entre tempo de treinamento e erro de previsão, no problema em questão:

- [5] Nós de Entrada
- [5] Camadas Ocultas
- [10] Nós Ocultos em cada camada.

Com relação à função de ativação e ao algoritmo de aprendizagem, foram escolhidas as seguintes opções:

- Função de ativação Sigmóide.
- Algoritmo de aprendizagem *Resilient Propagation*.

A escolha do algoritmo de aprendizagem *Resilient Propagation* (RPROP) foi feita a partir dos resultados obtidos na comparação entre o *Backpropagation* (BPROP) e o *Resilient Propagation*. O tempo de treinamento do RPROP foi aproximadamente 10 vezes mais rápido que o BPROP, obtendo ainda resultados melhores com relação à previsão dos valores estimados.

Durante o estudo piloto não foi possível determinar se é benéfico ou não normalizar os dados antes de treinar a RNA. Por exemplo, ao utilizar um conjunto de dados pequeno (que no caso deste trabalho foi de um mês de valores da BOVESPA, 01/11/2011 a 30/11/2011) foram obtidos resultados melhores com a abordagem de não normalização. Dessa forma, a título de comparação, foi testada esta variável no problema apresentado.

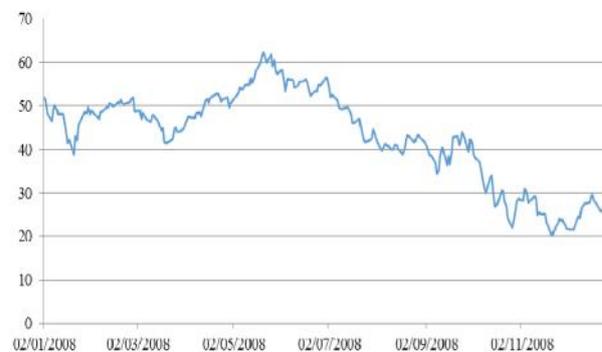


Figura 4. Cotações no ano de 2008.

Uma observação importante é que, durante o estudo, algumas redes apresentaram um tempo de treinamento

bastante curto (em torno de 10 segundos de treinamento), não condizente com o tamanho do conjunto de dados. Isso ocorre, pois durante o treinamento, caso o erro não melhore por um número pré-determinado de épocas, o algoritmo de treinamento entende que chegou ao melhor resultado e termina o treinamento. Depois de observado tal comportamento, foi escolhido arbitrariamente o valor de 1000 épocas, em que, se nessas 1000 épocas o algoritmo não observou nenhuma melhora, ele finaliza o treinamento. Caso haja uma melhora, ele continua por mais 1000 épocas até que essa regra seja completada.

Para o conjunto de treinamento foram escolhidas as ações PETR3 de 2007 a 2011. Foi escolhido um intervalo anual, pois assim podemos observar como a configuração de RNA escolhida se comporta em diferentes períodos.

No conjunto de dados escolhido, temos alguns períodos mais críticos e outros mais calmos, como por exemplo, o ano de 2008, mostrado na Figura 4, primeiro ano da Grande Recessão de 2008-2012⁷. Os valores das ações, que em Junho de 2008 estavam perto de R\$60,00, caíram para menos de R\$30,00 em menos de seis meses. O desvio padrão deste conjunto de dados é de aproximadamente R\$10,51.



Figura 5. Cotações no ano de 2011.

Já no ano de 2011 (Figura 5) a variação no preço é bem menor, com o desvio padrão dos dados cotando aproximadamente R\$3,79. Sendo assim, pode-se esperar uma maior confiabilidade nas previsões geradas por uma RNA neste intervalo de testes.

Os resultados obtidos para os dados normalizados podem ser verificados a partir da observação da Tabela 1.

Tabela 1. Resultado para os dados normalizados.

Período	Tempo de treinamento (s)	Erro de aprendizagem	Valor Previsto (R\$)	Valor Esperado (R\$)
2011	85,533	0,0196	23,16	23,00
2010	81,669	0,0379	29,03	30,45
2009	104,505	0,0401	41,01	41,56
2008	72,630	0,0315	26,24	29,4
2007	108,531	0,0125	47,95	48,01

Como citado anteriormente foram utilizados 5 nós de

entrada e o conjunto de treinamento total de um ano. A primeira coluna da Tabela 1 apresenta o período correspondente. A segunda coluna apresenta o tempo de treinamento que a rede consumiu para alcançar a convergência. A terceira coluna diz respeito ao erro observado durante a aprendizagem. A quarta coluna é o valor previsto pela rede. A quinta coluna é o valor esperado ou valor que a rede deveria prever.

Os resultados para o conjunto de dados não normalizados podem ser observados a partir da Tabela 2.

Tabela 2. Resultado para os dados não normalizados.

Período	Tempo de treinamento (s)	Erro de aprendizagem	Valor Previsto (R\$)	Valor Esperado (R\$)
2011	298,730	1,3783	31,69	23,00
2010	196,692	2,4141	30,66	30,45
2009	281,475	3,5181	39,60	41,56
2008	293,263	9,015	41,70	29,4
2007	121,350	3,6119	39,13	48,01

Assim sendo, podemos observar que para a configuração escolhida foi obtido resultados melhores quando se normaliza o conjunto de dados antes do treinamento. Ao normalizar os dados, ajustamos a escala de cada valor para que os valores fiquem no intervalo [0,1]. Este ajuste é necessário pois caso o conjunto de dados possua valores discrepantes (normalmente possui), estes valores irão influenciar o treinamento da rede de forma errônea.

Outra diferença significativa entre a normalização ou não dos dados foi o tempo de treinamento. Seguindo as regras descritas anteriormente, as redes cujas entradas não foram normalizadas demoraram pelo menos o dobro para terminarem o treino, com resultados longe de satisfatórios.

Se a sua função de ativação está no intervalo [0,1], (por exemplo, função sigmoide), então você deve ter a certeza que o seu conjunto de treino esteja neste intervalo¹².

3. RESULTADOS E DISCUSSÃO

Seguindo a metodologia descrita acima foi possível observar que, para as ações PETR3, é possível utilizar RNA's com significativa confiança. Mas em uma aplicação real a utilização de RNAs serve somente para fornecer subsídios para o processo decisório, o qual deve ser realizado pelo usuário da aplicação. A RNA deve ser vista como mais uma ferramenta e não como a verdade absoluta. O valor obtido pela previsão deve ser visto como uma indicação do comportamento da ação, muitas vezes não importando o valor em si.

Foi observado que, em todos os intervalos, a previsão foi perto do esperado menos em 2008. Embora o resultado obtido pela rede em 2008 esteja relativamente

longe do resultado esperado, a variação neste período é muito grande. No intervalo de uma semana até o dia 07/01/2009 (dia em que a cotação foi de R\$29,40), os valores foram de R\$25,89 a R\$29,40, chegando até a R\$31,02, como apresentado na Figura 6.

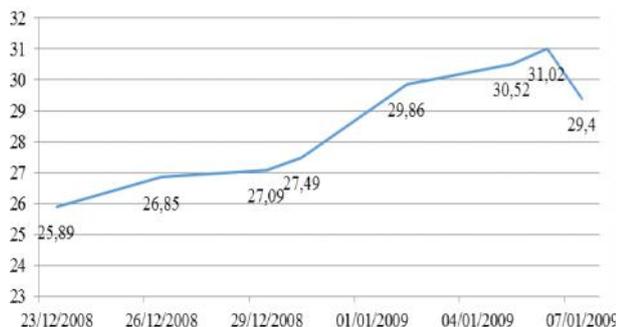


Figura 6. Cotações de 23/12/2008 a 07/01/2009.

Embora o resultado no período crítico (2008) não foi tão perto quanto nos períodos mais amenos, o resultado ainda pode ser considerado satisfatório, porém não ótimo. A previsão para este período (R\$26,24) só vai ser observada no fim de 2011.

Porém em outros períodos, como por exemplo, em 2007 (ano que precedeu a grande crise, caracterizada por uma valorização atípica) ou em 2011 (primeiro ano de relativa estabilidade no mercado de ações) a rede obteve resultados bastante satisfatórios. Em 2009 e 2010, o mercado apresentou um erro similar à rede treinada com o conjunto de 2008, porém devido ao desvio padrão dos dados nestes períodos ser menor, o resultado foi mais perto do esperado.

Utilizando de RNAs também, agora com o algoritmo de aprendizagem *Backpropagation*, conseguiu um erro de 0,0164 e tempo de treinamento de 5000 a 10000 épocas no melhor dos casos, enquanto neste trabalho, utilizando do algoritmo *Resilient Propagation*, foi obtido um erro de 0,0125 e 3000 a 5000 épocas. O trabalho de de Faria foi baseado nas ações PETR4, porém como as ações PETR3 e PETR4 seguem os mesmos índices, foi feita esta comparação.

No trabalho de Koskela *et al.* (1996)¹³ foi comparado como as redes neurais *Multilayer Perceptron*, FIR e Elman se comportam em diferentes cenários. Em um destes cenários foi efetuada a previsão baseada em séries geradas numericamente, cenário que se assemelha ao apresentado durante este trabalho. O erro obtido neste cenário, utilizando as abordagens MLP, Elman e FIR foram 0,0152, 0,0261 e 0,0305, respectivamente. Novamente, o erro obtido, no melhor dos casos, pela abordagem utilizando do algoritmo *Resilient Propagation* foi de 0,0125.

Outro ponto interessante é que as RNA's não são capazes de prever mudanças decorrentes de fatores exteriores (por exemplo, a crise imobiliária de 2008).

Mas com dados suficientes, ela é capaz de prever se as ações continuarão caindo ou subindo.

No problema descrito (e genericamente falando, qualquer problema em que o conjunto de dados seja suficientemente grande), deve-se normalizar antes de alimentar a rede, de acordo com o intervalo de sua função de ativação.

4. CONCLUSÃO.

Tendo em vista estas informações podemos concluir que uma RNA é capaz de gerar dados confiáveis, desde que o conjunto de treinamento seja suficientemente grande e que seja feito um pré-processamento correto antes de treiná-la.

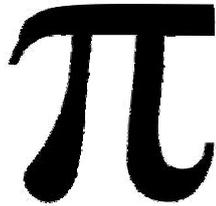
No entanto, como em qualquer técnica preditiva em que valores são o produto desejado, e no caso de mercado de ações, valores de tamanha magnitude, as RNA's devem ser encaradas como uma ferramenta, um auxiliador, e não uma entidade capaz de tomar decisões por conta própria.

REFERÊNCIAS

- [1] Lendasse A, Bodt E, Wertz ., & Verleysen M. Non-linear financial time series forecasting - Application to the Bel 20 stock market index. 2000.
- [2] Woldridge JM. Introductory Econometrics: A Modern Approach. South-Western College. 2008.
- [3] Marques FC, Gomes RM. Análise de Séries Temporais Aplicadas ao Mercado Financeiro com o uso de Algoritmos Genéticos e Lógica Nebulosa. Belo Horizonte: CEFET-MG. 2008.
- [4] Cavalcante F Mercado de Capitais: o que é, como funciona. Rio de Janeiro: Campus / Elsevier. 2005.
- [5] Ribeiro CV, Goldschmidt RR, Choren R. Métodos para Previsão de Séries Temporais e suas Tendências de Desenvolvimento. 2009.
- [6] Braga A, Ludermir BT, & Carvalho AC. Redes Neurais Artificiais: Teoria e aplicações. Rio de Janeiro: LTC. 2000.
- [7] Wessel D. Did 'Great Recession' Live Up to the Name? The Wall Street Journal. 2010.
- [8] Huang W, Nakamori Y, Wang SY. Forecasting stock market movement direction with support vector machine. Elsevier. 2004.
- [9] de Faria EL, Albuquerque MP, González Alfonso JL, Albuquerque MP, Cavalcante JT. Previsão do Mercado de Ações Brasileiro utilizando Redes Neurais Artificiais. Rio de Janeiro. 2008.
- [10] Haykin S. Neural Networks - A Comprehensive Foundation. Pearson Education. 1999.
- [11] Riedmiller M, Braun H. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. University of Karlsruhe. 1993.
- [12] Sarle WS. (11 de 10 de 2002). comp.ai.neural-nets FAQ, Part 2 of 7: Learning. Acesso em 01 de 10 de 2012, disponível em:

<http://www.faqs.org/faqs/ai-faq/neural-nets/part2/>.

- [13]Koskela T, Lehtokangas M, Saarinen J, Kaski K. Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. Tampere University of Technology, Tampere, Finland. 1996.

A large, bold, black Greek letter pi symbol (π) is centered on the page. The symbol is rendered in a classic serif font, with a thick, slightly curved top bar and a vertical stem that tapers slightly towards the bottom.

O IMPACTO DO TAMANHO DO BUFFER DE BUSCA DE INSTRUÇÕES EM RELAÇÃO À LARGURA DO ESTÁGIO DE BUSCA EM UMA GPU

THE IMPACT OF THE INSTRUCTION SEARCH BUFFER SIZE REGARDING THE SEARCH STAGE WIDTH OF A GPU

JOSÉ CARLOS SANCHES AMENDOA **TISSEI**¹, JEFERSON APARECIDO SILVA **BELGAMAZZI**², LUCAS DE OLIVEIRA **TEIXEIRA**³, CLAUDIA HEIDEMANN DE **SANTANA**⁴

1. Aluno do curso de graduação em Ciências da Computação da Faculdade Ingá; 2. Aluno do curso de graduação em Ciências da Computação da Faculdade Ingá; 3. Mestre pela Universidade Estadual de Maringá, docente no curso de Ciências da Computação na Faculdade Ingá; 4. Mestre em Engenharia de Produção pela Universidade Federal de Santa Catarina, docente no curso de Ciências da Computação na Faculdade Ingá.

Rua Lápiz Lazuli, 211, Jardim Santa Helena, Maringá, Paraná, Brasil. CEP: 87083-310 chsantana@gmail.com

Recebido em 04/10/2013. Aceito para publicação em 07/11/2013

RESUMO

Unidades de processamento gráfico (*GPUs*) possuem uma grande capacidade de processamento *multithread* e por esse motivo se tornaram uma plataforma muito popular para a execução de aplicações paralelas. O modelo de programação utilizado permite a criação de centenas de *threads* executando o mesmo *kernel*, isso é possível graças à quantidade de unidades de processamento existentes em uma GPU. Um dos estágios mais importantes para o aumento do desempenho das aplicações é a busca de instruções. Nesse artigo, investigamos o efeito do tamanho do *buffer* do estágio de busca de instruções em relação à largura desse mesmo estágio. Os dados utilizados na análise foram obtidos por meio de simulações heterogêneas CPU-GPU. Através da análise dos resultados é possível notar que esse *buffer* exerce um papel fundamental no desempenho de aplicações em geral, mas quando este é maior ou igual a largura desse estágio não ocorrem grandes variações.

PALAVRAS-CHAVE: Simulação CPU-GPU, Desempenho CPU-GPU, Busca de instruções GPU, Consumo de energia, Processamento paralelo.

ABSTRACT

Graphical processing units (*GPUs*) have a great capacity for *multithread* processing and therefore have become a very popular platform for running parallel applications. The programming model used allows the creation of hundreds of threads running the same kernel, this is made possible thanks to the amount of existing GPU processing units. One of the most important stages in increasing application performance is the instruction fetch. In this paper, we investigate the effect of the

instruction search buffer size of the instruction fetch stage regarding the width of such stage. The data used in the analysis were obtained by heterogeneous CPU-GPU simulations. Analyzing the results it is possible to realize that this buffer plays a key role in the performance of applications in general, but when it is greater than or equal than the width of the instruction fetch stage there is no big variations.

KEYWORDS: CPU-GPU simulation, CPU-GPU performance, GPU instruction fetch, power consumption, parallel processing.

1. INTRODUÇÃO

Prever Usualmente GPUs (*GraphicsProcessing Unit*) são projetadas para um processamento gráfico de alto desempenho. Devido ao alto investimento no mercado de jogos, as GPUs se tornaram um hardware poderoso em relação ao custo monetário. Nas arquiteturas modernas, onde temos GPUs sofisticadas de alto desempenho, a CPU pode preocupar-se mais com os outros tipos de processamento, visto que o processamento gráfico consome muitos recursos de um processador. Além de renderização de imagens, as GPUs vêm sendo usadas também em programas mais genéricos. Para programadores, a GPU tem sido uma boa alternativa em varias aplicações, principalmente as que fazem um grande uso de paralelismo.

Estes tipos de componentes aproveitam o desprendimento dos dados disponível em aplicações gráficas, executando as instruções em vários fluxos paralelos. Devido ao seu grande poder de processamento paralelo,

GPUs também são comumente usadas no processamento de outros tipos de aplicações, e nestes casos são chamadas de GPGPUs (*General-purpose graphics processing unit*).

As GPUs possuem processadores *multithreaded* chamados de *Streaming Multiprocessors* (SMs). Os SMs executam as *threads* de maneira SIMT (*Single Instruction Multiple Thread*). SIMT é uma forma de execução SIMD (*Single Instruction Multiple Data*) onde a mesma instrução é buscada e executada por um grupo de threads ao mesmo tempo, onde cada thread possui seus próprios conjuntos de dados. Em um SM, *threads* são agrupadas em unidades chamadas *Warps* e são executadas simultaneamente. Devido ao fato das *threads* serem executadas em conjunto, a alocação de recursos, principalmente o estágio de busca de instruções, em uma GPU é uma importante questão a ser tratada.

O estágio de busca é responsável por trazer instruções da memória para a sua execução. Inicialmente, as instruções buscadas são mantidas no *buffer* de instruções de busca e, em seguida, enviadas para o estágio de decodificação. Assim, esse estágio age como uma porta de entrada para a execução e possui um grande papel no desempenho do *pipeline*, principalmente em arquiteturas paralelas que necessitam de um constante fluxo de instruções, como as GPUs. Com isso, esse trabalho visa analisar o impacto da variação do tamanho do *buffer* de busca de instruções em relação à largura do estágio de busca em uma GPU através da análise de simulações em um sistema heterogêneo CPU-GPU. Assim, o principal objetivo é tentar realizar a correlação entre o tamanho do *buffer* e a largura do estágio de busca.

2. MATERIAL E MÉTODOS

O presente estudo foi realizado com base na consulta de bibliografia específica da área e da temática em questão. Foram consultados periódicos nacionais e internacionais disponíveis no portal EBSCO host da Faculdade Ingá. Foram selecionados 13 artigos entre os anos de 1996 e 2014, pela relevância de informações em face do objetivo do presente levantamento.

3. RESULTADOS E DISCUSSÃO

A análise dos resultados foi dividida em três pontos: desempenho geral, no qual é discutida a implicação da alteração proposta na variação do número de instruções executadas por ciclo de *clock* (IPC) na GPU e CPU; desempenho das *caches*, no qual é analisada a implicação da alteração proposta na taxa de acerto das caches L1, L2 e L3; e desempenho de energia, em que se analisa a variação no consumo de energia da CPU e GPU implicada pela alteração proposta.

Na maioria das simulações, as alterações no tamanho do *buffer* de busca de instruções da GPU não tiveram grande influência no desempenho da CPU. Esse era um resultado esperado, uma vez que, foram alterados somente configurações da GPU. O único caso especial foi o programa GESUMMV (multiplicação escalar, vetorial e de matrizes) que as alterações do *Baseline-2* e *Baseline-3* proporcionaram um leve *speedup* no IPC da CPU. No geral, as alterações negativas no *Baseline* causaram um leve *slowdown* no IPC da CPU. Com isso, existem indícios de que o tamanho do *buffer* de instruções da GPU impacta o desempenho da CPU.

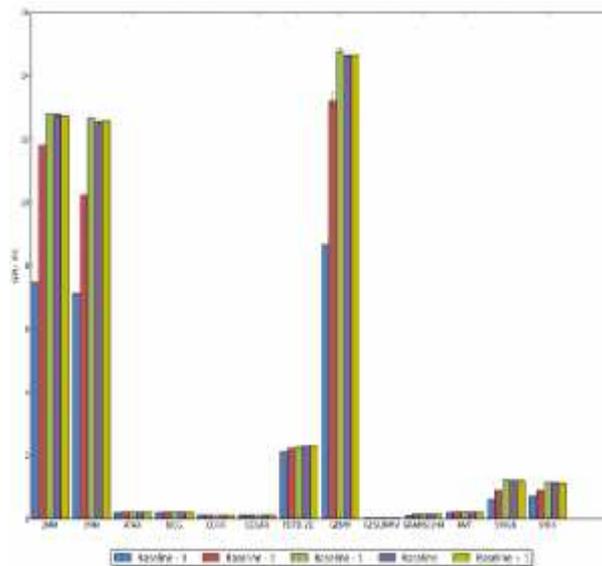


Figura 1. GPU IPC

Analisando o impacto do tamanho do *buffer* de instruções da GPU na quantidade de instruções por ciclo da GPU, apresentada na Figura 1, é possível perceber que a redução do tamanho do *buffer* impacta negativamente no desempenho da GPU. As alterações do *Baseline-3* e *Baseline-2* causaram um grande *slowdown* para grande parte dos programas do Polybench/GPU. No caso dos programas 2MM, 3MM e GEMM, os quais implementam variações de multiplicação de matrizes, o *slowdown* do *Baseline-3* chega a ser de quase metade se compararmos ao *Baseline*. Isto pode indicar que diminuindo o tamanho do *buffer* de busca de instruções da GPU o desempenho é fortemente afetado.

O desempenho das memórias cache L1, L2 e L3 da CPU não foram afetados positivamente ou negativamente pelas alterações no tamanho do *buffer* de busca de instruções da GPU. Novamente, esse era um resultado esperado, uma vez que, as GPU atuais possuem memória própria integrada à placa. Isso significa que os dados utilizados pela GPU ficam na própria memória da GPU. Assim, não existe necessidade de utilizar memórias cache ou RAM para armazenar tais dados.

Finalmente, a última característica analisada é o con-

sumo de energia da CPU e da GPU. Em relação ao consumo de energia da CPU, novamente a alteração do tamanho do *buffer* de busca de instruções da GPU não influenciou bastante o consumo de energia. Da mesma forma, que alguns programas obtiveram *speedupe slowdown*, o consumo de energia se manteve proporcional a essa variação. Isso significa que quanto maior o desempenho da CPU mais energia ela gasta. Em relação ao consumo de energia da GPU, apresentado na Figura 2, é possível observar que o aumento e a diminuição do consumo de energia é proporcional ao aumento e a diminuição do desempenho. Novamente, quanto maior o desempenho da GPU, mais energia ela gasta.

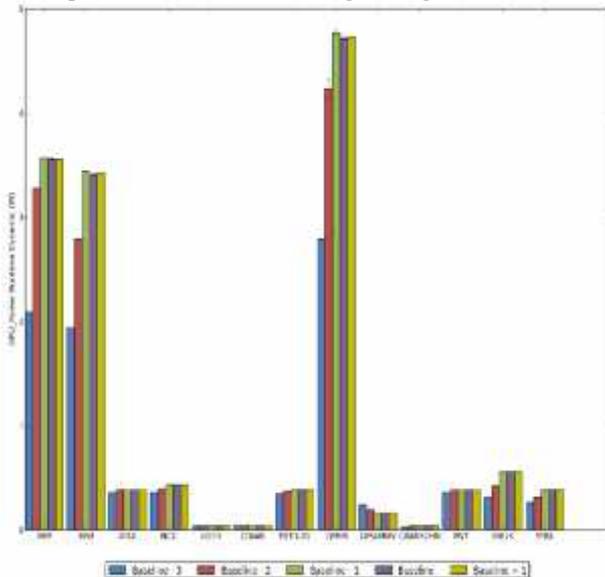


Figura 2. Consumo de energia da GPU.

4. CONCLUSÃO.

De acordo com os resultados apresentados e analisados neste trabalho, é possível concluir que o tamanho do *buffer* de instruções exerce um papel fundamental no desempenho de algumas aplicações executadas na GPU. Por meio de uma análise superficial, é possível notar que existem indícios de que as aplicações com o maior número de instruções vetoriais da ULA são as mais afetadas pela mudança no tamanho do *buffer*, no entanto é necessária uma análise maior e mais aprofundada para se obter tal conclusão. Além disso, o consumo de energia aumenta proporcionalmente com o aumento do desempenho, isso ocorreu em todos os aplicativos analisados.

Em seguida, é possível verificar que não houve grandes variações no desempenho nem no consumo energético da CPU, e isso era um resultado esperado, pois não houve nenhuma alteração em sua arquitetura nessa simulação.

Finalmente, esse trabalho realizou a análise de desempenho e energia através da alteração de apenas uma característica. Nesse caso, apenas o tamanho do

buffer de instruções de busca. No entanto, a alteração de uma característica pode impactar em diversas outras, com isso a análise deve ser feita variando diversas configurações relacionadas que podem alterar o desempenho do sistema como um todo. Assim, um trabalho futuro é realizar a variação da largura do estágio de busca juntamente com o tamanho do *buffer* de instruções de busca na tentativa de encontrar um conjunto de valores ótimo que aumente o desempenho geral do sistema.

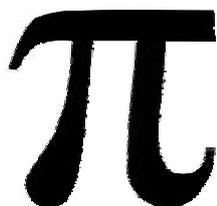
REFERÊNCIAS

- [1] Jiao Y, Lin H, Balaji P, Feng W. Power and performance characterization of computational kernels on the gpu. Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom). IEEE, 2010.
- [2] Lakshminarayana NB, Kim H. Effect of instruction fetch and memory scheduling on gpu performance. Workshop on Language, Compiler, and Architecture Support for GPGPU. 2010.
- [3] Tullsen DM, Eggers SJ, Emer JS, Levy HM, Lo JL, Stamm RL. CM. Exploiting choice: Instructionfetchandissueonanimplementablesimultaneousmultithreading processor. ACM SIGARCH Computer Architecture News. 1996; 24(2).
- [4] Narasiman V, Shebanow M, Lee CJ, Miftakhutdinov R, Mutlu O, Patt YN. Improving gpu performance via large warps and two-level warp scheduling. Proceeding softhe 44th Annual IEEE/ACM International SymposiumonMicroarchitecture. ACM. 2011.
- [5] Xiang P, Yang Y, Mantor M, Rubin N, Hsu LR, Huiyang Z. Exploiting uniform vector instructions for GPGPU performance, energy efficiency, and opportunistic reliability enhancement. Proceeding softhe 27th international ACM conference on International conference on supercomputing. ACM. 2013.
- [6] Amdradeonhd 7990 graphics, 2013. <http://www.amd.com/us/products/desktop/graphics/7000/7990/Pages/radeon-7990.aspx>. Acessado em Outubro de 2014.
- [7] Craig DC. Extensible Hierarchical Object-Oriented Logic Simulation with an Adaptable Graphical User Interface. PhD thesis, Memorial University of Newfoundland, 1996.
- [8] Intel core i7 3820, 2013. <http://ark.intel.com/pt-br/products/63698/Intel-Core-i7-3820-Processor-10M-Cache-up-to-380-GHz>. Acessado em Outubro de 2014.
- [9] Ubal R, Jang B, Mistry P, Schaa D, Kaeli D. Multi2sim: a simulation framework for CPU-GPU computing. Proceedingsofthe 21st international conference on Parallel architectures and compilation techniques. ACM. 2012.
- [10] Grauer-Gray S, Xu L, Searles R, Ayalasomayajula S, Cavazos J. Auto-tuning a high-level language targeted to gpu codes. Innovative Parallel Computing (InPar). IEEE. 2012.
- [11] Li S, Ahn JH, Strong RD, Brockman JB, Tullsen DM, Jouppi NP. Mcpat: an integrated power, area, and timing

modeling framework for multicore and manycore architectures. *Microarchitecture, MICRO-42. 42nd Annual IEEE/ACM International Symposiumon. IEEE. 2009.*

[12]Calborean H, Vintan L. Towardan efficient automatic design space exploration frame for multicore optimization. *ACACES 2010 poster Abstracts. 2010; 135-8.*

[13]Hunter JD. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering. 2007; 9(3):0090-95.*



A IMPORTÂNCIA DA EXPERIMENTAÇÃO EMPÍRICA NA PESQUISA EM CIÊNCIA DA COMPUTAÇÃO

THE IMPORTANCE OF EMPIRICAL EXPERIMENTATION IN RESEARCH ON COMPUTER SCIENCE

ANDERSON DA SILVA **MARCOLINO**¹, HELIO HENRIQUE LOPES COSTA **MONTE-ALTO**², LUCAS DE OLIVEIRA **TEIXEIRA**³, JEFERSON APARECIDO SILVA **BELGAMAZZI**⁴, JOSÉ CARLOS SANCHES AMENDOA **TISSEI**⁵, CLAUDIA HEIDEMANN DE **SANTANA**⁶

1. Doutorando em Ciências da Computação e Matemática Computacional do Instituto de Ciências Matemáticas e de Computação na Universidade Estadual de São Paulo; 2. Mestre em Ciência da Computação pela Universidade Estadual de Maringá, docente no curso de Ciência da Computação da Universidade Tecnológica Federal do Paraná; 3. Mestre em Ciência da Computação pela Universidade Estadual de Maringá, docente no curso de Ciências da Computação da Faculdade Ingá; 4. Aluno do curso de graduação em Ciências da Computação da Faculdade Ingá; 5. Aluno do curso de graduação em Ciências da Computação da Faculdade Ingá; 6. Mestre em Engenharia de Produção pela Universidade Estadual de Santa Catarina, docente no curso de Ciências da Computação da Faculdade Ingá.

Rua Lápiz Lazuli, 211, Jardim Santa Helena, Maringá, Paraná, Brasil. CEP: 87083-310. chsantana@gmail.com

Recebido em 05/10/2013. Aceito para publicação em 08/11/2013

RESUMO

A experimentação é um processo de validação de técnicas, abordagens, ferramentas e teses. Com isso, é importante em todas as áreas de conhecimento. Especificamente, na ciência da computação e na engenharia de software em que os estudos atuais quase sempre propõem abordagens, processos e produtos novos, o processo de experimentação deve ser parte fundamental do processo de criação de modelos. No entanto, nem sempre isso acontece na prática. Existem diversos motivos para isso, como: a intangibilidade do produto, muitas variáveis a ser controladas, custo elevado da experimentação, entre outros. Diante disso, o objetivo deste trabalho é apresentar os princípios benéficos provenientes da experimentação da ciência da computação. Além disso, mostrar quais as principais dificuldades e motivos da não realização de experimentos, bem como um conjunto de ferramentas especializadas de suporte ao processo de experimentação.

PALAVRAS-CHAVE: Engenharia de software, Engenharia de software experimental, Pesquisa em ciência da computação, Experimentação.

ABSTRACT

Experimentation is a process of validation of techniques, approaches, tools and theses. Thus, it is important in all areas of knowledge. Specifically in computer science and software engineering which the current studies almost always propose new approaches, processes and products, the process of experimentation must be a fundamental part of the model creation

process. However, this does not always happen in practice. There are many reasons for that, such as: the intangibility of the product, many variables to be controlled, high cost of experimentation, among others. Therefore, the aim of this paper is to present the main benefits from experimentation in computer science. Also, show the main difficulties and reasons for not carrying out experiments, as well as a set specialized tools for supporting the process of experimentation.

KEYWORDS: Software engineering, Experimental software engineering, Research on computer science, Experimentation.

1. INTRODUÇÃO

A ciência da computação, muitas vezes considerada uma ciência ainda imatura perante as mais tradicionais¹, possui fundamental importância tanto na vida acadêmica, quanto corporativa, incluindo também as demais áreas de tecnologia e afins. Os resultados obtidos na área trazem facilidades que vão de encontro com as tarefas do dia-a-dia e ainda, trazem lucros e aumento de produtividade.

Quando traçamos um paralelo desta ciência com as demais, em termos acadêmicos, nos deparamos com problemas, principalmente relacionados aos estudos promovidos pelos cientistas e pesquisadores desta área: faltam resultados comprovadamente testados e experimentados. E assim, acumulasse estudos incompletos e muitas vezes duvidosos, devido aos critérios de avaliação utilizados.

As técnicas propostas pela ciência na área de expe-

rimentação não são práticas comuns, para a grande maioria dos estudiosos na área tecnológica¹, porém estas são fundamentais para comprovar as pesquisas realizadas, mesmo se tratando de uma área em que os experimentos e variáveis, em sua grande maioria, são intangíveis².

Existem ainda ferramentas de software que permitem reproduzir simulações com uma grande carga de iterações, com as propostas de pesquisa, gerando resultados confiáveis e que muitas vezes demandaria uma vasta equipe e demasiado tempo. Tais ferramentas, como os *benchmarks*, propiciam a comprovação de hipóteses e assim, permitem refutar ou aceitar estudos. No entanto, ao focarmos a área de engenharia de software, temos um fator que, muitas vezes não pode ser medido por *benchmarks*, o fator humano.

O objetivo deste trabalho é realizar uma revisão de literatura sobre a importância da experimentação empírica na ciência da computação. Além disso, discutir os possíveis motivos para a não realização de experimentos formais em computação, os principais benefícios e ferramentas de suporte ao processo de experimentação.

2. MATERIAL E MÉTODOS

O presente estudo foi realizado com base na consulta de bibliografia específica da área e da temática em questão. Foram consultados periódicos nacionais e internacionais disponíveis no portal EBSCO host da Faculdade Ingá. Foram selecionados 22 artigos entre os anos de 1964 e 2012, pela relevância de informações em face do objetivo do presente levantamento.

3. RESULTADOS E DISCUSSÃO

A experimentação é o centro de todo processo científico, ou seja, novos métodos, técnicas, linguagens e ferramentas não deveriam ser apenas sugeridos, publicados ou apresentados sem uma experimentação e validação minuciosa³. Porém, especificamente, no caso de engenharia de software, em que geralmente os pesquisadores buscam o desenvolvimento de novas técnicas, métodos ou ferramentas, a experimentação é pouco utilizada.

O experimento é uma etapa de pesquisa científica no qual o pesquisador manipula e controla uma ou mais variáveis independentes e observa a variação nas variáveis dependentes concomitantemente à manipulação das variáveis independentes⁴.

O objetivo de controlar e manipular as variáveis é capturar a relação entre as causas e seus respectivos efeitos, assim um experimento é um método que investiga relações causais entre as variáveis para comprovar ou refutar pressupostos teóricos (Figura 1).

Um pretexto bastante utilizado pelos pesquisadores para evitar a experimentação é também uma das principais diferenças entre a ciência da computação e as ciên-

cias tradicionais: o fato de que informação não é nem energia nem matéria. No entanto, isso não faz diferença na aplicabilidade do método científico tradicional. Para entender a natureza da informação, deve-se observar fenômenos, formular explicações e teorias, e testá-las.



Figura 1. Conceitos de um Experimento⁵

Além disso, experimentação ajuda a observar novos fenômenos que podem ser usados para induzir novas teorias. Sintetizando os principais benefícios da experimentação, podemos citar⁶:

- A experimentação pode ajudar a construir uma base de conhecimento confiável e assim reduzir a incerteza sobre quais teorias, métodos e ferramentas são adequados;
- A observação e a experimentação pode conduzir a novos conhecimentos úteis e não esperados e, dessa forma, abrir novas áreas de investigação;
- A experimentação pode acelerar o processo de eliminação de abordagens infrutíferas, orientando a engenharia e a teoria às direções certas.

Quando focamos o processo de experimentação em engenharia de software, nos deparamos com diversas barreiras e algumas incógnitas geradas pela forma como esta é muitas vezes tratada, seja como engenharia ou como ciência. Desta forma, a experimentação para tais estudos requerem formalismos e técnicas das ciências humanas, quando nos referenciamos ao fator humano, diretamente presente e ativo em tais estudos.

Experimentação em engenharia de software

A engenharia de software, devido o caráter duplo do software, é muitas vezes considerada ciência e/ou engenharia. Este caráter duplo é identificado quando se considera, a princípio, o processo de criação do produto de software - sendo assim, apresenta características explícitas de produção ou engenharia. Quando são apresentados aspectos relacionados a *time-to-market* e competição, há a necessidade de melhoria contínua e sequencial da qualidade do processo e do produto - neste contexto, apresentasse características científicas⁷.

Deste modo, metodologias específicas são necessárias

para ajudar a estabelecer uma base de engenharia e de ciência para a engenharia de software. Existem quatro métodos relevantes para a condução de experimentos nesta área: científico, de engenharia, experimental e analítico⁸.

O método científico é um paradigma indutivo, que consiste na observação do mundo, sugestão de modelos ou teorias de comportamento, medições, análises e verificação de hipóteses do modelo ou da teoria. Este método pode ser utilizado quando é necessário entender o produto de software, o processo ou o ambiente. Assim, o objetivo é extrair do mundo algum modelo que possa explicar um fenômeno, e avaliar se o modelo é realmente representativo para tal fenômeno, que é foco da observação. Assim, se trata de uma abordagem para construção de modelos.

O método da engenharia mantém as observações nas soluções existentes, sugerindo as soluções mais adequadas, mede e analisa, e repete até que nenhuma melhoria adicional seja possível. Desta forma é uma abordagem orientada à melhoria evolutiva, assumindo sempre a existência de algum modelo do processo ou produto de software, modificando este com o propósito de melhorar os objetos de estudo.

O método analítico, ou também denominado matemático, sugere uma teoria formal, desenvolve a teoria, deriva os resultados e comparados tais resultados com as observações empíricas. Assim, é um método dedutivo que não necessita de um projeto experimental no sentido estatístico, entretanto, fornece uma base analítica para o desenvolvimento de modelos.

E por último, o método experimental desenvolve o método qualitativo e/ou quantitativo, aplica um experimento, mede e analisa, avalia o modelo e repete o processo. Com isso, pode ser caracterizado como uma abordagem orientada à melhoria evolucionária. O processo tem início com o levantamento de um modelo novo, sendo este não obrigatoriamente baseado no já existente, e tenta estudar o efeito do processo ou produto sugerido pelo novo modelo.

Estudos experimentais, atualmente, são considerados a melhor abordagem para o melhor entendimento e avaliação das tecnologias que estão sendo desenvolvidas⁹.

Mesmo tendenciados ao uso do método experimental, os demais podem ser necessários, como por exemplo: o método científico pode ser utilizado para compreender a maneira como o software está sendo construído por uma organização para a verificação de tal ferramenta, validando se esta pode ser utilizada para automatizar o processo; o método de engenharia ajuda a demonstrar que uma ferramenta possui um desempenho melhor que outra; e por último, o método analítico pode provar modelos matemáticos para conceitos, como o crescimento da confiabilidade, a complexidade do software, o projeto ou código propenso a erros, etc⁹.

Ainda em relação à pesquisa experimental, existem

vários métodos, podendo ser classificados em¹⁰:

- Pré-experimentais: São estudos exploratórios e não se tem quase nenhum controle sobre as variáveis, a seleção das unidades não é aleatória, portanto são experimentos simples e pouco confiáveis;

- Experimentais verdadeiros: Possuem grupos de controle além dos grupos experimentais e a seleção de unidades é totalmente aleatória;

- Quase experimentais: Estudos que realizam a medição ao longo de um determinado período de tempo, porém a seleção de unidades não é aleatória;

- Estatísticos: Utilizam controles estatísticos e análise de variáveis externas, são mais complexos, porém apresentam mais informações e permitem a avaliação de mais de uma variável ao mesmo tempo.

A escolha de qual método de pesquisa utilizar depende de vários fatores, como: os requisitos da investigação, disponibilidade das unidades e recursos, propósito da pesquisa e da análise dos dados obtidos⁸.

Entretanto, mesmo tendo conhecimento do processo de experimentação em si e até mesmo nos demais métodos, ainda há uma grande lacuna a ser preenchida quando observamos os trabalhos na área de ciência da computação¹¹. Existem evidências de que mais de 30% de artigos publicados em diversos periódicos relacionados à software não possuem qualquer validação experimental e apenas 10% dos artigos demonstram algum tipo de abordagem formal de experimentação¹¹.

Quando focamos em engenharia de software, uma pesquisa comandada por Tichy *et al.* (1995)¹² apresentou 400 artigos desta área, dos quais 40% não apresentaram nenhum tipo de experimentação, sendo um número alto, visto que em outras disciplinas o percentual é mantido em 15%.

Tais estudos nos levam à conclusão de que a comunidade de engenharia de software deve melhorar seus trabalhos, reportando resultados avaliados corretamente. Com isso, facilitando a adoção de tais resultados na indústria e em outros ramos². Deste modo, passamos a analisar as várias desculpas dos cientistas da computação e pesquisadores da área para a não realização de experimentos.

No estudo desenvolvido por Juristo & Moreno (2010)² são apresentados algumas dificuldades para o uso da experimentação na engenharia de software:

- Desenvolvedores de software não estão a par da importância e do significado do método científico, que atuam validando teorias e convertendo-as em fatos;

- Desenvolvedores de software não são capazes de entender facilmente como analisar os dados de um experimento ou como eles serão analisados por outros, devido à falta de treinamento;

- A ausência de livros sobre projeto e análise experimental na área de engenharia de software dificulta o processo de aprendizagem e utilização desta em estudos;

- Estudos empíricos, conduzidos para verificar propostas já existentes, apresentadas por outros pesquisadores, não são muito publicados;

- A grande quantidade de variáveis que influenciam no desenvolvimento de software;

- Dificuldades em obter resultados globais, pois é difícil determinar sobre quais circunstâncias uma técnica deve ser selecionada. Em outras palavras, é difícil provar que uma determinada alternativa é sempre a melhor alternativa;

- Outra restrição importante sobre a execução de experimentos em engenharia de software é o efeito do fator humano, pois esta é uma disciplina cujo resultado é dependente dos praticantes;

- Por último, um fator já apresentado anteriormente: o fato da grande quantidade de dinheiro circulando no mercado de software hoje, o que implica em uma gama de novas abordagens sendo injetadas no mercado sem uma real análise dos benefícios de seu uso.

Estas e outras dificuldades poderiam ser combatidas, se os clientes - usuários finais de tais objetos de estudo - solicitassem resultados que fossem validados experimentalmente.

No estudo apresentado por Feitelson (2006)¹³, este conclui, após apresentar diversos fatores, que as falhas no processo de realização de experimentação em ciência da computação é na verdade um problema cultural e não técnico. Ainda há muitas barreiras a serem derrubadas, para isso, basta analisarmos as muitas argumentações infundadas e utilizadas como desculpas para não avaliar as pesquisas.

Deste modo, é necessário explicitar o quão fundamental são os resultados validados com o método científico e quais os reais benefícios para a comunidade da ciência da computação, e principalmente, para a engenharia de software, objetivando assim atingir adeptos à prática da experimentação.

Argumentos utilizados para não realizar experimentação

É uma tendência na área de ciência da computação, principalmente por pesquisadores de engenharia de software, ignorar ou dar pouca importância à validação das ideias por meio de experimentação. Apesar de ser uma tendência geral, questiona-se o fato dos pesquisadores brasileiros não realizarem experimentos de software pela forma como o assunto foi abordado em sala de aula durante sua vida escolar. É também famosa a cultura do “*jeitinho brasileiro*”, em que geralmente prefere-se executar ao planejar, optando pelo caminho mais fácil.

É notável que em outros países exista mais foco na experimentação como parte fundamental do ensino. É comum em filmes estrangeiros, que apresentam o contexto colegial, professores e alunos realizarem experi-

mentos. Essa cultura está presente desde os níveis mais básicos da educação, tornando a experimentação presente na maioria dos trabalhos científicos.

Tichy (1998) [6] defende que o principal assunto de pesquisa em ciência da computação não é meramente o computador, mas sim estruturas e processos de informação. Desta forma, deve-se pensar sobre computadores e programas como modelos. De fato, modelagem é uma das principais tradições utilizadas na ciência, uma vez que ajuda a estudar fenômenos mais detalhadamente.

Muitos pesquisadores alegam que o nível de experimentação nos trabalhos em geral é bom o suficiente. Tal argumento mostra-se falho quando se observa a quantidade de trabalhos que requerem suporte empírico, mas não o fazem. Cerca de 40 a 50% dos artigos publicados na área de computação carecem desse tipo de validação [6].

Outro argumento usado pelos pesquisadores da área para eliminar a necessidade de experimentação é o custo adicional acarretado pela realização dos experimentos, uma vez que requerem muitos recursos. No entanto, ao compara com outras áreas, é possível perceber que tal custo é aceitável, pelo fato de que muitas vezes, os benefícios compensam o custo.

Alguns pesquisadores também alegam que demonstrações são suficientes, de modo que substituem a necessidade de experimentos. De fato, demonstrações podem prover provas de conceito ou incentivos para estudos mais aprofundados, de modo que apenas ilustram o potencial da abordagem proposta. Assim, demonstrações não produzem evidências sólidas, uma vez que dependem da capacidade do observador de vislumbrar potenciais. É necessário que haja uma análise cuidadosa envolvendo experimentos para se obter evidências sólidas.

Outra linha de defesa contra a experimentação é a alegação de que há muito ruído na coleta dos dados. Muitas vezes, por haver muitas variáveis para controlar, os pesquisadores imaginam que resultados não terão significância por causa do ruído. No entanto, tal problema pode ser minimizado pelo uso de *benchmarks*. Um *benchmark* é uma amostra de um conjunto de trabalho que deve ser executado pelo computador, com ou sem acompanhamento humano. Tais programas devem ser projetados de forma que se possa coletar os dados desejados com o mínimo de ruído e ainda facilitar a repetição de experimentos.

A construção de *benchmarks* costuma ser custosa e trabalhosa. No entanto, é possível que vários laboratórios cooperem entre si para sua elaboração, uma vez que determinados tipos de experimento podem ser de interesse de vários grupos de pesquisa. Além disso, uma vez definido, um *benchmark* pode ser executado repetidamente a um custo moderado. *Benchmarks* vêm sendo utilizados com sucesso em diversas áreas, incluindo recuperação de informações, reconhecimento de padrões, reuso de software, arquitetura de computadores, avaliação de de-

sempenho, etc.

De fato, em algumas subáreas da ciência da computação muitas vezes não é possível utilizar *benchmarks*, uma vez que envolvem a interação humana. No entanto, isto não é desculpa para não realizar experimentos. Áreas como medicina e psicologia, que lidam com o fator humano, encontraram várias técnicas para lidar com a variabilidade humana - como: grupos de controle, distribuição aleatória, placebos, pré e pós-teste, balanceamento, baterias de testes estatísticos, etc.

É comum também, por parte dos pesquisadores, pensar que experimentações irão retardar o progresso da pesquisa. No entanto, o aumento da proporção de artigos com boa validação pode acabar acelerando o progresso, uma vez que ideias questionáveis seriam descartadas mais rapidamente, fazendo com que haja mais concentração nas ideias mais promissoras.

Segundo Mudge (1996)¹⁴, o desenvolvimento de novas tecnologias ocorre tão rapidamente que quando os resultados são confirmados por meio de experimentos eles podem não ser mais relevantes. No entanto, se uma questão se torna irrelevante tão rapidamente, talvez seja pelo fato de tal ser efêmera e não valer a pena utilizar. Os pesquisadores deveriam focar nas abordagens fundamentais e promissoras. Além disso, deve-se antecipar tais mudanças e, proativamente, realizar experimentos que permitam explorar as consequências de tais mudanças⁶.

Outro problema que surge ao realizar experimentação na área de computação é que muitos editores e revisores de publicações têm dificuldade em avaliar trabalhos empíricos. No entanto, há muitos periódicos e conferências que os valorizam. É interessante também dividir o trabalho em pequenos passos que possam ser publicados aos poucos, pois melhoram nosso entendimento e levantam novas questões.

Desta forma, podemos sumarizar os principais argumentos utilizados pelos cientistas da computação e profissionais da área para a não realização de experimentos⁶:

- O método científico tradicional não é aplicável;
- O nível atual de experimentação é suficiente;
- Experimentos são caros;
- Demonstrações são suficientes;
- Experimentos são trabalhosos;
- Experimentação atrasa o progresso;
- Tecnologias mudam muito rápido;
- Dificuldade de publicação.

Benefícios da experimentação em engenharia de software

O processo de desenvolvimento de software atual se baseia em boas práticas, experiências pessoais e observações gerais, porém apenas esses fatores não podem assegurar a qualidade e conformidade do artefato resultante. Assim, o processo de desenvolvimento deve ser

seguido imediatamente de uma avaliação para verificar a conformidade com os requisitos iniciais com o objetivo de identificar pontos falhos¹⁵.

Com isso, estudos experimentais possuem como objetivo identificar se o artefato resultante do processo de desenvolvimento obtém os resultados esperados, além de identificar os problemas e dificuldades que as pessoas tiveram ao utilizá-lo na prática¹⁶.

Além de satisfazer seu objetivo principal, a experimentação possui os seguintes benefícios^{6, 17}:

- Construção de uma base confiável de conhecimento reduzindo a incerteza sobre quais teorias, métodos e ferramentas são adequados;
- Propicia novas compreensões sobre áreas que estão sendo pesquisadas;
- Exploração de áreas desconhecidas, estendendo a fronteira do conhecimento, possibilitando o descobrimento de novas áreas de pesquisa;
- Eliminação de abordagens mal fundamentadas, errôneas e modismos, orientando a teoria em direções mais promissoras;
- Demonstração da real utilidade e benefícios de novas técnicas, métodos e tecnologias;
- Finalmente, no processo de formação da engenharia de software como ciência, por meio do aumento de trabalhos científicos com experimentos significativos.

Atualmente, os argumentos para não realizar experimentação não são válidos, uma vez que existem muitos benefícios em sua realização. Notada a importância da avaliação de pesquisa, observa-se também a adesão de ferramentas de software e técnicas para a realização das mesmas, visando à minimização das não ocorrências destas ou a redução do número de falhas, bem como auxiliar e acelerar tal processo.

Ferramentas de software e técnicas para a avaliação de pesquisa

Dentre as ferramentas existentes para auxiliar etapas no processo de experimentação, com foco em diversas áreas, não somente na área da Engenharia de Software, temos:

- *GQM (Goal Question Metric)*: Ferramenta que visa facilitar a adoção de princípios sistemáticos e possui importância na obtenção de métricas para o experimento [19]. Baseia-se nos princípios da avaliação orientada a objetivos que, segundo Gladcheff, Sanches e Da Silva (2012)¹⁸, visa servir como uma metodologia genérica para orientar a elaboração e execução de programas de avaliação da qualidade de processos na área de engenharia de software.

- *SGWfC (Sistemas de Gerência de Workflows Científicos)*: Ferramenta que apóia a execução de *workflows* de modo controlado e documentado. Entretanto, tal sistema não oferece recurso para acompanhar as três etapas

do experimento científico como um todo (composição, execução e análise). Esta ferramenta não tem o foco na engenharia de software, visando dar suporte ao processo de experimentação científico em laboratórios, entretanto, pode ser usada na área tecnológica. Como exemplos de tais sistemas temos Taverna, Kepler e VisTrails¹⁹.

É possível citar ainda outras ferramentas de software que auxiliam no processo de tratamento dos resultados obtidos com a experimentação e na aplicação de avaliações estatísticas, tais como:

- Microsoft Excel e o Calc (BrOffice): Ambos programas para a geração de planilhas eletrônicas;

- Statistica, Minitab, MathLab: Programas que lidam com informações estatísticas, permitindo trabalhar com uma grande quantidade de dados e tratá-los com as mais diversas fórmulas de análise e comparação de dados numéricos;

- AVAs (Ambientes Virtuais de Aprendizagem): Ferramentas disponíveis na Internet, objetivando compartilhamento de conhecimento, podendo servir como modo de compartilhamento de resultados e explanação sobre os mais diversos assuntos. Servindo assim, para que equipes possam desenvolver seus trabalhos, mesmo estando geograficamente distribuídas. Como exemplos temos o Moodle, Trello, edX, etc.

Estas ferramentas auxiliam o processo, mas ainda necessitam de certo formalismo, para serem utilizadas, ou seja, a aplicação de técnicas que permitam um controle sistematizado do experimento, desde seu projeto até a apresentação concisa dos resultados obtidos. Assim, podemos mencionar algumas das técnicas existentes hoje, para a realização deste processo:

- Estudo de Caso: É uma técnica poderosa e flexível. Usada principalmente para investigações exploratórias, tanto prospectiva e retrospectiva, que tentam compreender e explicar fenômenos ou construir teorias. Caracteriza-se por ser, geralmente, observacional ou de natureza descritiva, embora possa ser relacional. É aplicada também na avaliação de resultados de investigação²⁰.

- Experimento formal: É mais rigoroso que estudos de caso, por considerar um estado variável - um fator que pode caracterizar a influência do projeto e da avaliação de resultados. Experimentos formais podem ser usados para prover o contexto no qual certos padrões, métodos e ferramentas são recomendadas para uso²¹.

- *Surveys*: Experimentos executados em retrospecto, quando uma técnica ou método já está em uso por alguma empresa e deseja avaliá-la em algum aspecto formal³. As maneiras mais conhecidas e bastante utilizadas são questionários e entrevistas.

- *Quasi-experiments*: Experiências em que as unidades de estudo são aplicadas a grupos experimentais não randomizados. Eles permitem que as investigações de relações causa-efeito em ambientes em que a randomização é inadequada, impraticável, ou muito cara²².

Finalmente, temos que estudos empíricos são compostos por experimentos formais, estudos de caso e *surveys*²¹. Experimentos empíricos são aqueles comprovados pela prática, ou seja, é o estudo de um objeto através de tentativa ou erro, sem nenhum parâmetro estabelecido, apenas utilizando a observação.

4. CONCLUSÃO.

Enfim, é possível perceber que a experimentação é central no processo científico, pois apenas experimentos testam teorias, permitem explorar fatores críticos e trazem à tona novas questões, de modo que teorias possam ser formuladas e corrigidas. De acordo com Tichy (1998)⁶, a ciência da computação está em perigo se os pesquisadores não usarem experimentação para validar suas teorias.

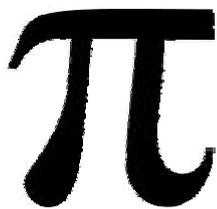
Algumas teorias da ciência da computação, como programação funcional, orientada a objetos e métodos formais, nunca foram avaliadas para se verificar sua real veracidade. Muitos destas acabam sendo, de certa forma, experimentadas diretamente pelo usuário final, como em empresas e também no meio acadêmico. Isto pode agregar muitas vezes gastos e riscos desnecessários, pois o uso de técnicas e métodos não comprovados podem mais prejudicar do que colaborar com o desenvolvimento de um projeto.

Ainda falta visão quanto ao impacto de um estudo carente de avaliação, principalmente quando falamos de tecnologia. Com o advento da Internet, em poucos minutos, novos conhecimentos são difundidos de um lado do globo a outro, e muitas empresas se baseiam no conhecimento divulgados sem se preocupar em identificar a veracidade dos resultados e estudos disponibilizados. Assim, acabam muitas vezes investindo em algo que, muitas vezes, não gerará o retorno esperado.

REFERÊNCIAS

- [1] Denning PJ. Is computer science science? Communications of the ACM. 2005; 48(4):27-31.
- [2] Juristo N, Moreno AM. Basics of software engineering experimentation. Springer Publishing Company, Incorporated. 2010.
- [3] Amaral EAGG. Empacotamento de experimentos em engenharia de software. Dissertação (Mestrado), Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ, Rio de Janeiro. 2003.
- [4] Kerlinger FN, Lee HB, Foundations of behavioral research: Educational and psychological inquiry. New York: Holt, Rinehart and Winston. 1964.
- [5] Basili VR, Selby RW, Hutchens DH. Experimentation in software engineering. Software Engineering, IEEE Transactions on. 1986; 7:733-43.
- [6] Tichy WF. Should computer scientists experiment more? Computer. 1998; 31(5):32-40.
- [7] Travassos GH, Gurov D, Amaral EAGG. Introdução à en-

- genharia de software experimental. UFRJ. 2002.
- [8] Wohlin C, et al. Experimentation in software engineering. Springer. 2012.
- [9] Travassos GM, Shull F, Fredericks M, Basili VR. Detecting defects in object-oriented designs: using reading techniques to increase software quality. *ACM SigplanNotices*. 1999; 34(10).
- [10] Malhotra NK. Pesquisa de marketing: uma orientação aplicada. Bookman. 2012.
- [11] Zelkowitz MV, Wallace DR. Experimental models for validating technology. *Computer*. 1998; 31(5):23-31.
- [12] Tichy WF, Lukowicz P, Prechelt L, Heinz EA. Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software*. 1995; 28(1):9-18.
- [13] Feitelson DG. Experimental computer science: The need for a cultural change. 2006.
- [14] Mudge T. Report on the panel: how can computer architecture researchers avoid becoming the society for irreproducible results. *Computer Architecture News*. 1996; 24(1):1-5.
- [15] Babar MA, Gorton I. Comparison of scenario-based software architecture evaluation methods. *Software Engineering Conference, 2004. 11th Asia-Pacific. IEEE*. 2004.
- [16] Ciolkowski M, Shull F, Biffl S. A family of experiments to investigate the influence of context on the effect of inspection techniques. NA. 2002.
- [17] Travassos GH, Shull F, Carver J. Working with uml: A software design process based on inspections for the unified modeling language. *Advances in Computers*. 2002; 54:35-98.
- [18] Gladcheff AP, Sanches R, Silva DM. Um instrumento de avaliação de qualidade de software educacional: Como elaborá-lo. *Pensamento & Realidade. Revista do Programa de Estudos Pós-Graduados em Administração-FEA*. 2002; 11.
- [19] Mattoso M, *et al.* Desafios no apoio à composição de experimentos científicos em larga escala. *Seminário Integrado de Software e Hardware, SEMISH*. 2009; 9:36.
- [20] Perry DE, Sim SE, Easterbrook SM. Case studies for software engineers. *Software Engineering. ICSE 2004. Proceedings. 26th International Conference on. IEEE*. 2004.
- [21] Fenton NE, Pfleeger SL. *Software metrics: a rigorous and practical approach*. PWS Publishing Co. 1998.
- [22] Kampenes VB, Dyba T, Hannay TE, Sjoberg DIK. A systematic review of quasi-experiments in software engineering. *Information and Software Technology*. 2009; 51(1):71-82.



NORMAS PARA PUBLICAÇÃO

JOURNAL OF EXACT SCIENCES

ESCOPO EDITORIAL

Journal of Exact Sciences - JES é um periódico voltado a publicações da área das Ciências Exatas, com periodicidade trimestral, exclusivamente *online*, no formato *Open Access Journal**, publicado em Língua Portuguesa pela **Master Editora**.

Journal of Exact Sciences - JES destina-se à publicação de artigos originais, revisões/ atualizações literárias e relatos de casos/ técnicas profissionais, os quais uma vez encaminhados, serão submetidos à análise *ad hoc*.

Journal of Exact Sciences - JES tem como **público-alvo**: estudantes de graduação e pós-graduação (*Lato sensu* e *Stricto sensu*), além de docentes e pesquisadores em áreas das Ciências Exatas e afins, estando aberta à comunidade científica nacional e internacional.

Os manuscritos submetidos para publicação no periódico **JES** devem ser originais e não divulgados previamente. Serão aceitos para submissão: manuscritos originais, de revisão/ atualização e relatos de casos ou técnicas profissionais.

O manuscrito será submetido inicialmente ao Editor-Chefe do periódico **JES** para uma análise preliminar de mérito, relevância e contribuição para expansão da fronteira do conhecimento científico. Eventualmente, os artigos poderão ser publicados a convite do Editor-Chefe do periódico **JES**.

Uma vez que o manuscrito submetido seja aceito para publicação, a Master Editora e o periódico **JES** passam deter os direitos autorais exclusivos sobre o seu conteúdo, podendo autorizar ou desautorizar a sua veiculação, total ou parcial, em qualquer outro meio de comunicação, resguardando-se a divulgação de sua autoria original. Para tanto, deverá ser encaminhado junto com o manuscrito uma “**Carta de Transferência de Direitos Autorais**” (disponível na página de **JES**), contendo a assinatura de cada um dos autores.

Com o parecer preliminar favorável do Editor-Chefe, o manuscrito seguirá para dois pareceristas *ad hoc*. Cada um deles receberá os textos sem a identificação do(s) autor(es). Após o recebimento do parecer das duas análises *ad hoc*, o Editor-Chefe encaminhará o parecer final ao(s) autor(es).

Dois pareceres *ad hoc* desfavoráveis à publicação do manuscrito implica na recusa automática do periódico **JES** em publicar o manuscrito sob a forma de artigo científico, sendo as considerações motivadoras da deci-

são editorial devolvidas ao(s) autor(es).

O parecer favorável para publicação de apenas um dos pareceristas *ad hoc*, leva o manuscrito novamente para análise do Editor-Chefe, que poderá decidir pela publicação (como um terceiro parecerista), ou efetivamente encaminhá-lo para uma terceira análise *ad hoc*, após o que, havendo dois pareceres favoráveis o manuscrito poderá ser publicado.

Quando e se necessário, serão solicitadas alterações e revisões aos autores. Ao Conselho Editorial reserva-se o direito de aceitar, sugerir alterações ou recusar os trabalhos encaminhados para publicação. Ao periódico **JES** se reserva ainda o direito de realizar alterações textuais de caráter formal, ortográfico ou gramatical antes de encaminhá-lo para publicação.

Frisa-se que os conceitos emitidos nos textos são de responsabilidade exclusiva dos autores, não refletindo obrigatoriamente a opinião do Corpo Editorial do periódico **JES**. Finalmente, a Editora Master e o periódico **JES**, ao receber os manuscritos, não assume tacitamente o compromisso de publicá-los.

* O formato *Open Access Journal* (Revista Científica de Acesso Aberto) garante que o acesso aos artigos publicados seja irrestrito e gratuito. Os autores não terão nenhum custo financeiro para submissão e a subsequente análise do manuscrito pelo corpo editorial de **JES**. Entretanto, caso um manuscrito seja **aceito para publicação**, o autor responsável (autor de correspondência) poderá confirmar o interesse pela publicação realizando o **pagamento da taxa de publicação, no valor de R\$ 150,00 (cento e cinquenta reais)**, em função dos custos relativos aos procedimentos editoriais.

Em caso de **dúvidas, críticas ou sugestões**, entre em contato com a Editor-Chefe do periódico **Journal of Exact Sciences - JES** pelo e-mail:

jes@mastereditora.com.br

1. SUBMISSÃO do MANUSCRITO

Os manuscritos redigidos conforme as normas editoriais de **Journal of Exact Sciences - JES** deverão ser submetidos online para análise editorial. No menu do website de **Journal of Exact Sciences - JES**, será possível escolher a opção “**Submissão de Artigos**”. Se este for seu primeiro acesso escolha a seguir a opção

“Cadastre-se”. Complete os campos solicitados com seus dados, e escolha uma senha, para ao final de seu cadastro acionar o botão “Confirmar”.

Novamente na página “Submissão de Artigos”, entre com seu e-mail e senha cadastrada.

Será solicitado que você complete os campos “Título do artigo”, “Subtítulo do artigo” (se houver), “Autores”, na mesma ordem do manuscrito e seleção do periódico de seu interesse; no caso, **Journal of Exact Sciences - JES**.

Finalmente, faça o upload de seu arquivo (word.doc) e acione o botão “Enviar” e encerre seu ambiente online acionando o botão “Sair”.

2. NORMAS PARA PUBLICAÇÃO

Estrutura do manuscrito

Original (experimental clássico): identificação do(s) autor(es), resumo, palavras-chave, abstract, keywords, introdução, material e métodos, resultados, discussão, conclusões, agradecimentos (se houver), financiamento (se houver) e referências.

Revisão ou Atualização: identificação do(s) autor(es), resumo, palavras-chave, abstract, keywords, introdução, material e métodos (descrevendo os parâmetros utilizados para a seleção das referências bem como de outros parâmetros que o(s) autor(es) julgar(em) necessários), discussão -exclusivamente textual ou ilustrada com elementos gráficos como figuras e tabelas, para melhor visualização dos dados sob análise do(s) autor(es), conclusões, agradecimentos (se houver), financiamento (se houver) e referências.

Relato de Caso Profissional: identificação do(s) autor(es), resumo, palavras-chave, abstract, keywords, introdução (breve), descrição do caso, discussão (contemporizando o caso apresentado com a literatura científica especializada), conclusões, agradecimentos (se houver), financiamento (se houver) e referências.

Relato de Técnica: identificação do(s) autor(es), resumo, palavras-chave, abstract, keywords, introdução (breve), descrição da técnica relatada, discussão (contemporizando a técnica em questão e suas inovações, potencialidades e/ou limitações com aquelas comumente descritas na literatura científica especializada), conclusões, agradecimentos (se houver), financiamento (se houver) e referências.

Preparação do manuscrito, segundo as normas editoriais do periódico

1- **TEXTO:** deve ser redigido em no máximo **12 páginas**, em formato eletrônico utilizando como editor de texto o Microsoft **Word**, com **espaçamento simples**, sem espaços ociosos entre os parágrafos, fonte **Times New Roman** e fonte de **tamanho 11**. As margens devem

ter **3 cm à esquerda e à direita e 2 cm acima e abaixo**. O texto deverá estar **justificado à página**. Para a redação, utilize-se da **terceira pessoa do singular e do verbo na voz ativa**.

2- **TÍTULO:** em **português**, deverá estar em **negrito e centralizado** no topo da primeira página, utilizando-se **fonte de tamanho 12, em caixa alta**. O título em **inglês**, logo abaixo, deverá ser redigido em **fonte de tamanho 11**.

3- **IDENTIFICAÇÃO DO(S) AUTOR(ES):** o(s) autor(es) deverá(ão) se identificar logo **abaixo dos títulos** em português e inglês, com o nome digitado em **CAIXA ALTA e justificado à página**. Depois do nome do(s) autor(es), deve constar respectivamente a **titulação e instituição a que pertence/representa**. Exemplos:

JOÃO DA SILVA. Aluno do curso de graduação em Engenharia Civil da Faculdade Nonono.

JOSIANE MEDEIROS DE MELLO. Doutora pela Faculdade de Agronomia da Universidade Nonono, docente do curso de Agronomia da Universidade Momomo.

4- **ENDEREÇO PARA CORRESPONDÊNCIA:** abaixo da identificação do(s) autor(es), deve conter os dados do autor responsável pela correspondência: **rua, bairro, cidade, estado, país, CEP e e-mail**. Preferencialmente, o orientador do estudo deverá ser o contato com o Corpo Editorial do periódico **JES**, fornecendo preferencialmente seus contatos profissionais.

5- **RESUMO/ ABSTRACT:** logo abaixo do endereço para correspondência, deverá ser digitada a palavra **RESUMO**, alinhado à esquerda, em **negrito e em caixa alta**. Na linha seguinte, deverá ser apresentado um breve resumo do manuscrito, com no máximo 200 palavras, seguido de 3 a 5 **PALAVRAS-CHAVE***. O resumo deve ressaltar o **fator motivador** para a redação do trabalho, sendo composto por **frases simplificadas** (concisas), **afirmativas**, sem apresentação de itens enumerados com tópicos. Deverá ser redigido em **parágrafo único**. Símbolos que não sejam comumente utilizados, fórmulas, equações, diagramas, entre outros, devem ser evitados. O **ABSTRACT**, de mesmo teor do resumo deverá ser apresentado abaixo do resumo e seguido pelas **KEYWORDS***, com significado equivalente às palavras-chave.

6- **INTRODUÇÃO:** abaixo do Abstract, escreva a palavra **INTRODUÇÃO**, centralizada e em **negrito**. Neste item, deve ser abordado o **referencial teórico pesquisado** para a elaboração do artigo. Se necessário, o texto poderá ser subdividido por subtítulo(s) sugestivo(s), grafados com alinhamento à esquerda e em **negrito**. A introdução deverá ser finalizada com a hipótese e o(s) objetivo(s) do estudo realizado, sem a necessidade de evidenciá-los em subtítulos.

7- **MATERIAL E MÉTODOS:** depois da introdução, deverá constar o item **MATERIAL E MÉTODOS**, centralizado e em **negrito**. Neste item, os autores devem

detalhar os recursos materiais e metodológicos utilizados para realização do trabalho.

- **Abreviaturas:** para unidades de medida, utilize somente as unidades do Sistema Internacional de Unidades (SI). Utilize apenas abreviaturas e símbolos já padronizados, evitando incluí-las no título do manuscrito e no resumo. O **termo completo deve preceder a abreviatura** quando ela for empregada pela **primeira vez**, salvo no caso de unidades comuns de medida.

8 – **RESULTADOS:** a seguir, constar o item RESULTADOS, centralizado e em negrito nos manuscritos chamados de originais (experimentação clássica), com resultados inéditos. **Revisões/ Atualizações da Literatura ficam dispensadas deste item de formatação.** Figuras e Tabelas (se houver) deverão ser inseridas pelos autores no corpo do texto em local onde sua visualização facilite a compreensão dos resultados apresentados. No Relato de Caso Profissional e no Relato de Técnica os resultados fazem parte da discussão, não sendo especificados separadamente.

- Se houver **Figuras**, recomenda-se que sejam **coloridas**, com numeração arábica progressiva. O **título** da figura deverá aparecer **abaixo** desta, seguido pela sua respectiva legendas (se houver), em **fonte de tamanho 10**. As figuras devem possuir pelo menos **300 dpi, no formato .JPG**. Não serão aceitas imagens fora de foco;
- Se apresentar **Tabelas**, o **título** desta deverá ser inserido sobre (**acima**) da tabela, com numeração arábica progressiva, indicando, logo abaixo, a **fonte da pesquisa** (se houver), ou algum item de observação relevante para interpretação de seu conteúdo. Os resultados apresentados em tabelas não devem ser repetidos em gráficos, e vice-versa;
- Note que não deverá ser feita inserção os elementos denominando-os como: esquema, diagrama, gráfico etc. Os **elementos gráficos** do artigo necessariamente **deverão ser chamados de Figura ou de Tabela**.
- Recomenda-se que o total de Figuras e Tabelas **não seja superior a oito**.
- No texto, a referência às Tabelas ou Figuras deverá ser feita por algarismos arábicos.

9- **DISCUSSÃO:** após a apresentação dos resultados, deve constar o item DISCUSSÃO, centralizado e em negrito. Os autores deverão comentar sobre seus achados experimentais, contextualizando-os com os registros prévios na literatura científica especializada.

10- **CONCLUSÕES:** Após a discussão, deve constar o item CONCLUSÕES, centralizado e em negrito. O(s) autor(es) deverá(ão) responder de modo afirmativo ou negativo sobre a hipótese que motivou a realização do estudo, por meio do alcance dos objetivos propostos. No último parágrafo, o(s) autor(es) poderá(ão) expressar sua

contribuição reflexiva (de cunho pessoal), e/ou versar sobre as perspectivas acerca do estudo realizado.

11- **FINACIAMENTO:** o(s) autor(es) deve(m) indicar ainda a(s) fonte(s) de financiamento da pesquisa (agências de fomento, empresas, etc.).

12- **REFERÊNCIAS:** É o último item de formatação do manuscrito, sendo identificada pela palavra REFERÊNCIAS, centralizada e em negrito. As referências devem ser numeradas de forma consecutiva de acordo com a ordem em que forem mencionadas pela primeira vez no texto e normalizadas no estilo Vancouver. Utilize **fonte Times New Roman de tamanho 9**. Listar todos os autores quando até seis; quando forem sete ou mais, listar os seis primeiros, seguidos de et al. As referências são de responsabilidade dos autores e devem estar de acordo com os originais.

Exemplos de referências:

1. Coutinho KD. Método de otimização topológica em estruturas tridimensionais. Dissertação (Mestrado), Universidade Federal do Rio Grande do Norte. Natal, Rio Grande do Norte, 2006
2. Díaz A, Sigmund O. Checkerboard Patterns in Layout Optimization. Structural Optimization. 1995; 10:40-45.
3. Olukanni DO, Ducoste JJ. Optimization of waste stabilization pond design for developing nations using computational fluid dynamics. Ecological Engineering. 2011; 37(11):1878-1888.
4. EPA – Environmental Protection Agency. Test methods for polynuclear aromatic hydrocarbons 8310. Disponível em: <<http://www.epa.gov/osw/hazard/testmethods/sw846/pdfs/8310.pdf>>. Acesso em: 28 jul. 2008.
5. Baird C. Química ambiental. 2. ed., Bookman, Porto Alegre, 2002.
6. Silva ECN. Técnicas de otimização aplicadas no projeto de peças mecânicas. São Paulo: Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, Escola Politécnica da USP, 2001. (Apostila)

