

INTELIGÊNCIA ARTIFICIAL APLICADA EM ENGENHARIA DE SOFTWARE**ARTIFICIAL INTELLIGENCE APPLIED TO SOFTWARE ENGINEERING** <https://doi.org/10.63330/aurumpub.005-009>**Robson dos Santos Fernandes**

Graduação: Análise e Desenvolvimento de sistemas

Pós-graduação: MBA em Gestão de Vendas

Unifatecie

E-mail: robsonfernandesgym@gmail.com

RESUMO

Este trabalho aborda a aplicação da Inteligência Artificial (IA) na engenharia de software, destacando como diferentes técnicas podem contribuir para melhorar processos de desenvolvimento, reduzir riscos e estimar custos de forma mais precisa. O objetivo principal desta produção foi investigar e apresentar as principais metodologias de IA utilizadas na engenharia de software, com foco especial nas redes neurais, lógica fuzzy e redes bayesianas, analisando seus conceitos, aplicações práticas e resultados obtidos em estudos de caso. A metodologia empregada foi a pesquisa bibliográfica, fundamentada em obras e artigos científicos de diversos autores renomados da área, como Pfleeger, Pressman, Haykin, Barros e Cozman, entre outros. Como resultado, verificou-se que as redes neurais são eficazes na estimativa de custos e esforços de projetos de software, utilizando dados históricos para aprimorar a acurácia das previsões. A lógica fuzzy se mostrou uma ferramenta adequada para avaliar a qualidade de software, considerando fatores subjetivos e imprecisos típicos de ambientes reais. As redes bayesianas, por sua vez, se destacaram pela capacidade de modelar incertezas e estabelecer relações causais entre variáveis, proporcionando simulações úteis na tomada de decisão. Conclui-se que a incorporação dessas técnicas de IA na engenharia de software representa uma evolução significativa, promovendo ganhos em eficiência, qualidade e confiabilidade dos sistemas desenvolvidos. Além disso, observa-se a necessidade de aprofundar pesquisas e aplicações práticas, visando maximizar o potencial dessas ferramentas no contexto do desenvolvimento de software.

Palavras-chave: Engenharia de Software; Inteligência Artificial; Redes Neurais.**ABSTRACT**

This work addresses the application of Artificial Intelligence (AI) in software engineering, highlighting how different techniques can contribute to improving development processes, reducing risks and estimating costs more accurately. The main objective of this production was to investigate and present the main AI methodologies used in software engineering, with a special focus on neural networks, fuzzy logic and Bayesian networks, analyzing their concepts, practical applications and results obtained in case studies. The methodology used was bibliographical research, based on works and scientific articles by various renowned authors in the field, such as Pfleeger, Pressman, Haykin, Barros and Cozman, among others. As a result, it was found that neural networks are effective in estimating software project costs and efforts, using historical data to improve the accuracy of forecasts. Fuzzy logic proved to be a suitable tool for assessing software quality, taking into account subjective and imprecise factors typical of real environments. Bayesian networks, in turn, stood out for their ability to model uncertainties and establish causal relationships between variables, providing useful simulations for decision-making. It can be concluded that the incorporation of these AI techniques into software engineering represents a significant evolution, promoting gains in efficiency, quality and reliability of the systems developed. Furthermore, there is a need for further



research and practical applications in order to maximize the potential of these tools in the context of software development.

Keywords: Software Engineering; Artificial Intelligence; Neural Networks.



1 INTRODUÇÃO

A aplicação da Inteligência Artificial (IA) na engenharia de software constitui um tema de grande relevância no contexto atual, dada a crescente necessidade de aprimorar a eficiência, qualidade e confiabilidade no desenvolvimento de sistemas computacionais. A literatura aponta que a engenharia de software é uma disciplina da engenharia industrial que estuda métodos sistemáticos para garantir que softwares complexos atendam às expectativas de clientes, apresentem confiabilidade e sejam produzidos com qualidade, dentro dos prazos e orçamentos estipulados (Pfleeger, 1998; Pressman, 2010). Paralelamente, a evolução da IA oferece recursos que imitam a capacidade humana de resolver problemas e aprender com eles, promovendo um avanço significativo nas práticas de desenvolvimento de software (Castro, 2009).

O objetivo deste trabalho foi investigar e apresentar as principais metodologias de Inteligência Artificial aplicadas na engenharia de software, com ênfase em três técnicas: redes neurais, lógica fuzzy e redes bayesianas. Como hipótese, considerou-se que a aplicação dessas técnicas pode contribuir significativamente para a melhoria das estimativas de custo, avaliação da qualidade e gestão de riscos em projetos de software. A justificativa da pesquisa fundamenta-se na necessidade de explorar alternativas tecnológicas que superem as limitações das metodologias tradicionais, evidenciadas pelo baixo índice de sucesso em projetos de software, conforme indicam os dados do Standish Group (2009).

A metodologia adotada foi a pesquisa bibliográfica, com base em obras e artigos científicos de referência, como Haykin (2001), Barros e Bassanezi (2006; 2010), Cozman (2000) e outros, que forneceram o embasamento teórico necessário para a análise das técnicas selecionadas. O trabalho foi desenvolvido em quatro partes principais: a primeira parte apresenta os fundamentos da engenharia de software, abordando seu conceito, ciclo de vida e importância para a indústria. A segunda parte fala sobre as Considerações iniciais sobre Inteligência Artificial, destacando seu conceito. A terceira parte trata da aplicação da Inteligência Artificial nesse campo, destacando seus benefícios e desafios. A quarta parte explora de forma detalhada três técnicas específicas: redes neurais, lógica fuzzy e redes bayesianas, exemplificando suas aplicações práticas na estimativa de custos e avaliação da qualidade de software. Por fim, a quinta parte expõe a conclusão, que sintetiza os resultados obtidos e reforça a importância da integração da Inteligência Artificial como estratégia para potencializar o desenvolvimento de software. Assim, este trabalho visa contribuir para o avanço das práticas na área de engenharia de software, incentivando a adoção de abordagens inovadoras e eficientes.



2 DESENVOLVIMENTO

2.1 ENGENHARIA DE SOFTWARE

A engenharia de software é um ramo da engenharia industrial que se dedica a estudar os métodos de trabalho e as práticas ideais adotadas pelos profissionais que desenvolvem sistemas de software (Pfleeger, 1998). Seu foco principal está na aplicação de procedimentos sistemáticos que assegurem que softwares complexos correspondam às expectativas dos clientes, apresentem confiabilidade, sejam fáceis de manter, ofereçam bom desempenho e sejam produzidos dentro dos prazos e orçamentos previstos (Pressman, 2010).

Essa disciplina, também conhecida como engenharia aplicada ao desenvolvimento de programas de computador, envolve a especificação, produção e implantação do código-fonte do software (Rezende, 2002). A engenharia de software abrange todo o ciclo de vida do software (Sommerville, 2007), envolvendo desde a análise de requisitos, elaboração de especificações, concepção do funcionamento interno, escolha de técnicas de programação, desenvolvimento, testes e, por fim, manutenção do sistema (Magela, 2006).

2.2 CONSIDERAÇÕES INICIAIS SOBRE INTELIGÊNCIA ARTIFICIAL

A definição de inteligência artificial (IA) é ampla e complexa, dificultando uma compreensão única e definitiva. Conforme afirmou recentemente Sam Altman, CEO da OpenAI, quando pensamos ter compreendido a IA, na verdade, não a entendemos; somente começamos a captar sua essência quando admitimos que não a compreendemos plenamente¹. A literatura contemporânea distingue três tipos principais de IA: a IA Fraca (ou estreita), a IA Forte (ou Geral) e a IA Superinteligente. A IA Fraca corresponde àquela que encontramos em Chatbots, sistemas de recomendação de conteúdo e em tecnologias de reconhecimento de voz e imagem. Sem possuir autonomia ou "consciência", opera dentro de parâmetros e regras previamente estabelecidos. Por outro lado, a IA Forte representa uma forma hipotética de inteligência capaz de atuar de maneira semelhante ao intelecto humano, dotada, teoricamente, de autoconsciência² e apta a resolver problemas, pensar criticamente e articular experiências como um ser humano. Por fim, a IA Superinteligente, ainda um conceito teórico, superaria as capacidades intelectuais humanas (OPENAI, 2023a).

No âmbito da IA Fraca, distinguem-se duas categorias principais: a IA Generativa e a IA Interativa. A primeira é desenvolvida para realizar tarefas específicas de criação de conteúdo, como textos, imagens, músicas ou códigos de programação. Já a IA Interativa é projetada para interagir com usuários em contextos delimitados, mediante comandos textuais ou de voz, como ocorre com assistentes virtuais, Chatbots ou serviços de atendimento ao cliente. Apesar dessa distinção, ambas podem se sobrepor, como exemplifica o ChatGPT: simultaneamente gera conteúdo e interage em tempo real com os usuários, respondendo perguntas, mantendo diálogos e adaptando-se ao contexto da interação (OPENAI, 2023b).



Dentro desse espectro da IA Fraca, no qual atualmente nos situamos, podemos visualizar uma metáfora útil para compreender parcialmente essa tecnologia: uma grande máquina, onipresente e onisciente, capaz de prever, decidir, criar e agir conforme padrões pré-estabelecidos, sobretudo relacionados ao conhecimento explícito. Trata-se de um sistema que agrega todo o saber técnico e científico humano acumulado, sendo capaz de mobilizá-lo na execução de uma ampla gama de tarefas intelectuais e práticas. O que a IA pode realizar? Em termos gerais, qualquer atividade passível de ser convertida em conhecimento explícito pode ser utilizada pela IA para processar, interrelacionar, analisar e tomar decisões. Assim, funções cognitivas e produtivas que envolvam esse tipo de conhecimento podem ser desempenhadas por sistemas orientados por IA. Entre as principais capacidades destacam-se: reconhecimento de padrões, como leitura facial ou impressão digital, além de identificação de tendências de mercado; processamento de linguagem natural, possibilitando a compreensão e resposta à linguagem humana, inclusive com tradução e análise de sentimentos; aprendizado de máquina, permitindo que o sistema aprenda e aperfeiçoe suas respostas com base em experiências não programadas; resolução de problemas e apoio à tomada de decisões, como na definição de rotas de transporte; visão computacional, com a capacidade de interpretar imagens e vídeos, além de possibilitar a navegação autônoma de veículos; robótica autônoma, que inclui robôs aptos a manusear materiais, realizar cirurgias ou conduzir veículos; criação assistida, com produção de textos, músicas ou imagens; e análise preditiva de dados, aplicada, por exemplo, à previsão do tempo, oscilações do mercado financeiro e planejamento urbano (OPENAI, 2023c).

Sob a perspectiva prática, a IA já desempenha uma série de aplicações relevantes, incluindo: diagnóstico médico, com análise de imagens e dados, bem como acesso a literatura especializada para auxiliar na formulação de diagnósticos e proposição de terapias baseadas em dados atualizados globalmente; previsão meteorológica; assistentes virtuais como Siri e Alexa; recomendações personalizadas em plataformas como Netflix e Spotify; veículos autônomos; automação industrial; análise financeira; Chatbots e serviços de atendimento ao cliente; segurança cibernética; e agricultura de precisão, entre outros (OPENAI, 2023d).

Enquanto a inteligência artificial geral ainda não se concretiza, os Chatbots especializados tendem a ganhar importância significativa no campo das atividades educacionais e profissionais nos próximos anos. Focados em domínios específicos do conhecimento, esses sistemas podem representar ferramentas valiosas tanto para apoiar processos de ensino-aprendizagem quanto para auxiliar profissionais na tomada de decisões. Em determinadas situações, poderão inclusive substituir segmentos inteiros de trabalho humano, uma vez que sua precisão tende a ser altamente confiável.

2.3 INTELIGÊNCIA ARTIFICIAL APLICADA NA ENGENHARIA DE SOFTWARE

Nos tempos recentes, observa-se um contínuo avanço na computação e, paralelamente, no desenvolvimento de software. Embora diversos métodos e ferramentas tenham sido criados, ainda é baixo o número de produtos entregues com confiabilidade e dentro dos prazos estabelecidos. Segundo dados do Standish Group (2009), apenas cerca de 30% dos projetos de software são concluídos com êxito. Essas dificuldades são comuns nas organizações que desenvolvem software, pois analisar riscos e dimensionar o esforço necessário para o desenvolvimento são tarefas complexas (Stahnke; Vahldick, 2013).

Atualmente, existem várias metodologias voltadas à superação desses desafios. Uma alternativa viável é a aplicação da Inteligência Artificial (IA), que, conforme Castro (2009), utiliza informações de projetos anteriores para apoiar na estimativa de novos projetos e na análise de riscos. Nessa linha, Moura, Pinto e Lustosa Filho (2011) indicam que sistemas inteligentes podem complementar ou mesmo substituir outras metodologias tradicionais. Para Castro (2009), a IA busca criar sistemas computacionais que imitem o comportamento humano, baseando-se na capacidade humana de resolver problemas e aprender com suas experiências.

2.4 REDES NEURAIS

Inspiradas na estrutura do cérebro humano, as redes neurais são compostas por múltiplos neurônios artificiais. Essas redes são implementadas por meio de programação ou utilizando componentes eletrônicos específicos (Haykin, 2001).

Os neurônios artificiais, ou nós, constituem as unidades básicas de uma rede neural. Eles recebem valores de entrada, associados a determinados pesos que indicam a força e direção das conexões. Cada neurônio processa a soma ponderada dessas entradas por meio de uma função de ativação, gerando assim uma saída (Russell; Norvig, 2004).

Um exemplo prático do uso de redes neurais foi apresentado por Moura, Pinto e Lustosa Filho (2011), que desenvolveram uma ferramenta para estimar o custo de projetos de software de maneira semelhante ao raciocínio de engenheiros experientes. A ferramenta emprega o modelo de pontos por função, atribuindo pesos a diversos fatores importantes, cujos valores são processados pela rede neural para estimar o custo. A rede foi treinada com dados de 70 projetos reais, e outros 30 projetos foram utilizados para validar as estimativas geradas.

Outro exemplo envolve uma ferramenta também baseada em redes neurais, destinada à estimativa de custos de software, mas utilizando abordagens diferentes. Nesse caso, as entradas são os requisitos a serem implementados e o tempo padrão para sua execução, com o treinamento da rede realizado a partir de tempos indicados por especialistas. A rede ajusta-se gradualmente, através da comparação entre as saídas geradas e os padrões, até alcançar uma estimativa próxima à de um especialista humano (Borsoi *et al.*, 2011).

Ainda utilizando redes neurais, Garcia *et al.* (2011) desenvolveram um modelo para estimar o esforço total de um projeto de software, considerando características comuns entre os diversos projetos. A rede neural foi estruturada com três camadas, onde as entradas representam características relevantes, e a saída corresponde à estimativa final do custo. O treinamento foi realizado com dados de 214 projetos, sendo parte deles usada para validação do modelo.

2.5 A LÓGICA FUZZY

Enquanto os computadores operam com dados precisos, representados por sequências de 1 e 0 e declarações estritamente verdadeiras ou falsas (Barros; Bassanezi, 2006), o cérebro humano é capaz de lidar com informações incertas e julgamentos subjetivos como "o ar está frio" ou "a velocidade é rápida". Além disso, as pessoas utilizam o bom senso para raciocinar em ambientes onde as verdades são parciais (Barros; Bassanezi, 2010).

A lógica fuzzy é uma vertente da Inteligência Artificial que permite ao computador processar informações do mundo real em um espectro contínuo entre o verdadeiro e o falso. Matemáticos lógicos, desde a década de 1920, já haviam estabelecido que muitas questões são graduais, e não absolutas. A lógica fuzzy lida com conceitos vagos como "quente" ou "úmido", permitindo o desenvolvimento de aparelhos como televisores e condicionadores de ar capazes de operar com informações imprecisas. Assim, sistemas fuzzy surgem como uma alternativa às concepções clássicas de pertencimento e lógica formuladas na Grécia Antiga (Benini, 2007).

Quando se traduz a linguagem humana para a lógica clássica, há uma perda de nuances importantes, o que pode ser prejudicial no desenvolvimento de sistemas especialistas. Por exemplo, médicos utilizam medidas exatas, mas o diagnóstico envolve raciocínios imprecisos e subjetivos (Jafelice, 2003). Em situações onde não existem algoritmos claros para determinar a resposta de um sistema, a lógica fuzzy possibilita o controle ou a descrição do sistema com base em regras intuitivas. Esses sistemas geralmente se baseiam em regras fornecidas por especialistas, mas, na ausência deles, podem aprender observando como pessoas operam sistemas reais (Barros; Bassanezi, 2010).

A lógica fuzzy é caracterizada como uma lógica multivalorada, na qual as variáveis podem assumir valores reais entre 0 e 1, ao invés de simplesmente verdadeiro ou falso. Dessa maneira, ela expande a lógica booleana clássica ao incorporar graus parciais de verdade (Barros; Bassanezi, 2010). Segundo Boente (2009), a lógica fuzzy pode ser entendida como uma extensão da teoria de conjuntos, aplicada à modelagem de dados imprecisos. Por essas características, é utilizada como ferramenta para avaliar a qualidade de produtos de software. Na pesquisa de Boente (2009), foram estabelecidos termos e escalas de medição, aplicados questionários para avaliar parâmetros de qualidade e satisfação dos gestores, e então processados os dados com a fórmula da média fuzzy, resultando em um índice de qualidade do software.

2.6 REDES BAYESIANAS

A representação do conhecimento e a inferência baseada nessas representações originaram diversos modelos (Cozman, 2000). Entre eles, destacam-se os modelos gráficos probabilísticos, especialmente as redes bayesianas, concebidas por Judea Pearl na década de 1980, que se consolidaram como ferramentas valiosas para representar conhecimentos incertos e raciocinar a partir de informações incompletas (Hruschka Jr, 2003).

As redes bayesianas modelam a distribuição de probabilidade conjunta entre variáveis, sendo preferidas em métodos de amostragem estocástica. Elas possibilitam a geração de dados simulados conforme necessário, permitindo que especialistas observem o comportamento do sistema em cenários que, na prática, podem não ser testáveis (Paulino; Turkman; Murteira, 2003).

De acordo com Stahnke e Vahldick (2013), uma rede bayesiana possui duas componentes: qualitativa e quantitativa. A parte qualitativa é representada por um grafo, onde cada nó corresponde a uma variável relevante para a resolução do problema, e as conexões expressam, probabilisticamente, as relações entre essas variáveis. Já a parte quantitativa consiste nas tabelas de probabilidades condicionais associadas a cada nó, indicando as hipóteses envolvidas.

Mendes (2010) aplicou redes bayesianas no desenvolvimento de um software para estimar os custos de aplicações web. Inicialmente, identificaram-se os componentes essenciais para essa estimativa, considerados como variáveis da rede. Posteriormente, construiu-se o grafo, relacionando as variáveis com base nas suas interações causais. Com a estrutura estabelecida, iniciou-se o cálculo das probabilidades. A validação do software foi realizada por meio de simulações com diferentes cenários, todas executadas com sucesso, além de comparações com dados históricos de projetos previamente desenvolvidos.

3 CONCLUSÃO

A pesquisa realizada sobre a aplicação da Inteligência Artificial (IA) na engenharia de software revelou a importância crescente dessas tecnologias no aprimoramento dos processos de desenvolvimento de sistemas. Ao longo do trabalho, foi possível identificar que as metodologias de IA, como redes neurais, lógica fuzzy e redes bayesianas, não apenas oferecem soluções inovadoras, mas também abordam de maneira eficaz as limitações das metodologias tradicionais.

As redes neurais, inspiradas no funcionamento do cérebro humano, demonstraram ser particularmente eficazes na estimativa de custos e esforços em projetos de software. Através da análise de dados históricos, essas redes conseguem aprimorar a acurácia das previsões, permitindo que equipes de desenvolvimento tomem decisões mais informadas. Isso é crucial em um cenário onde apenas 30% dos projetos de software são concluídos com sucesso, conforme indicam dados do Standish Group. A utilização das redes neurais não apenas melhora a precisão das estimativas, mas também contribui para a gestão de



riscos, uma vez que oferece uma visão mais clara das variáveis envolvidas nos projetos.

A lógica fuzzy se destacou pela sua capacidade de lidar com a incerteza e a subjetividade, características intrínsecas ao desenvolvimento de software. Ao permitir que sistemas computacionais processem informações em um espectro contínuo entre o verdadeiro e o falso, a lógica fuzzy se apresenta como uma ferramenta poderosa para avaliar a qualidade de software. Essa metodologia é especialmente valiosa em contextos onde os critérios de avaliação são vagos ou não podem ser quantificados de maneira precisa. Ao incorporar a lógica fuzzy, os desenvolvedores podem obter uma visão mais holística da qualidade do software, considerando fatores que vão além das métricas tradicionais.

Por sua vez, as redes bayesianas se mostraram indispensáveis na modelagem de incertezas e na representação de relações causais entre variáveis. A capacidade de simular cenários e prever comportamentos em condições de incerteza é uma vantagem significativa para a tomada de decisões estratégicas. A aplicação das redes bayesianas na estimativa de custos e na análise de riscos oferece uma nova abordagem que pode auxiliar significativamente na mitigação de falhas em projetos, proporcionando uma compreensão mais profunda das interações envolvidas.

Em suma, a integração de técnicas de IA na engenharia de software não apenas representa uma evolução necessária, mas também um caminho promissor para enfrentar os desafios contemporâneos da área. O estudo conclui que a adoção dessas metodologias pode levar a ganhos substanciais em eficiência, qualidade e confiabilidade dos sistemas desenvolvidos. Contudo, é fundamental que as pesquisas continuem a explorar novas aplicações práticas e a maximizar o potencial das ferramentas de IA, sempre visando à inclusão e à melhoria contínua dos processos de desenvolvimento de software.

Portanto, recomenda-se que as organizações invistam na capacitação de suas equipes em relação a essas tecnologias inovadoras, promovendo uma cultura de aprendizado e adaptação às novas ferramentas disponíveis. Somente assim será possível elevar o padrão da engenharia de software, garantindo que os sistemas não apenas atendam às expectativas dos clientes, mas também sejam sustentáveis e adaptáveis às constantes mudanças do mercado.



REFERÊNCIAS

- BARROS, L. C.; BASSANEZI, R. C. Tópicos de lógica fuzzy e biomatemática. Campinas: UNICAMP/IMECC, 2006.
- BENINI, L. C. Estimação da densidade de solos utilizando sistemas de inferência fuzzy. Tese (Doutorado), Universidade Estadual Paulista, Botucatu, 2007.
- BOENTE, A. N. P. Um modelo fuzzy para avaliação da qualidade de produtos de software e da satisfação dos gerentes de projetos numa fundação pública estadual. Dissertação (Mestrado), Universidade Estácio de Sá, Rio de Janeiro, 2009.
- CASTRO, R. D. Avaliação de riscos em projetos de software a partir do uso de técnicas de inteligência computacional. Monografia, Escola Politécnica de Pernambuco – Universidade de Pernambuco, Recife, 2009.
- COZMAN, F. G. Generalizing variable elimination in bayesian networks. Workshop on Probabilistic Reasoning in Artificial Intelligence, 2000.
- HAYKIN, S. Redes Neurais: Princípios e Práticas. 2. ed. Porto Alegre: Bookman, 2001.
- HRUSCHKA JR, E. R. Imputação Bayesiana no contexto da Mineração de Dados. Tese (Doutorado), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.
- MAGELA, R. Engenharia de Software Aplicada: Princípios. Rio de Janeiro: Alta Books, 2006.
- MENDES, E. Construindo um modelo especialista de estimativa de esforço de aplicações web usando redes bayesianas. WAMPS, 2010.
- MOURA, Í. B.; PINTO, J. P.; LUSTOSA FILHO, J. Neural Estimate: Uma ferramenta para a estimativa de custo de software. Escola Regional de Computação dos Estados do Ceará, Maranhão e Piauí, Teresina, 2011.
- OPENAI. ChatGPT-4. Em qual dessas categorias entraria a IA generativa e a IA interativa? Acesso em: 26 Mai. 2025.
- OPENAI. ChatGPT-4. Enumere as principais atividades cognitivas que a inteligência artificial é capaz de realizar. Acesso em: 26 Mai. 2025.
- OPENAI. ChatGPT-4. Enumere as principais atividades práticas que a inteligência artificial é capaz de realizar. Acesso em: 26 Mai. 2025.
- OPENAI. ChatGPT-4. Quais são as diferentes atividades cognitivas que a inteligência artificial é capaz de realizar? Acesso em: 26 Mai. 2025.
- OPENAI. ChatGPT-4. Quais são os diferentes tipos de inteligência artificial? Acesso em: 26 Mai. 2025.
- PAULINO, C. D.; TURKMAN, M. A.; MURTEIRA, B. Estatística Bayesiana. Lisboa: Fundação Calouste Gulbenkian, 2003.



PFLEEGER, S. L. Software Engineering: theory and practice. Nova Jersey: Prentice-Hall, 1998.

PRESSMAN, R. S. Engenharia de Software. 6ª ed. Porto Alegre: AMGH Editora, 2010.

REZENDE, D. A. Engenharia de software e sistemas de informação. 2ª ed. Rio de Janeiro: Brasport, 2002.

RUSSELL, S.; NORVIG, P. Inteligência Artificial. Rio de Janeiro: Elsevier, 2004.

SOMMERVILLE, I. Engenharia de Software. 8ª ed. Boston: Addison Wesley, 2007.

STAHNKE, E.; VAHLICK, A. Inteligência artificial aplicada na engenharia de software. Resumos Internos, vol. 2, no. 1, 2013.