



Evento: XXVI Jornada de Pesquisa

INTELIGÊNCIA ARTIFICIAL EM INTEGRAÇÃO DE APLICAÇÕES EMPRESARIAIS: UMA REVISÃO SISTEMÁTICA¹

ARTIFICIAL INTELLIGENCE IN ENTERPRISE APPLICATION INTEGRATION: A SYSTEMATIC
REVIEW

Matheus H. Rehbein², Rafael Z. Frantz³

¹ Projeto de pesquisa desenvolvido na Unijuí no Programa de Pós-Graduação em Modelagem Matemática e Computacional

² Bolsista PROSUC/Capes; estudante do curso Pós-Graduação em Modelagem Matemática e Computacional

³ Professor do Programa de Pós-Graduação em Modelagem Matemática e Computacional

RESUMO

Geralmente, as empresas possuem softwares heterogêneos em seus ecossistemas. Para compartilhar dados e funcionalidades dos softwares, as empresas costumam integrá-los. A utilização de inteligência artificial na integração de aplicações empresariais pode auxiliar na resolução de problemas durante a integração. Este artigo realizou um levantamento de trabalhos em que utilizou-se diferentes técnicas de inteligência artificial na integração de aplicações empresariais. Observou-se que as técnicas utilizadas foram Redes Neurais Artificiais, Algoritmos Genéticos, e Lógica Fuzzy; os trabalhos encontrados foram classificados em: i) desenvolvimento; ii) segurança; iii) otimização; iv) mapeamento semântico.

Palavras-chave: Integração de Aplicações Empresariais. Revisão Sistemática. Inteligência Artificial.

ABSTRACT

Normally, companies have many heterogeneous applications in their software ecosystems. For sharing data and functionalities, companies usually integrate them. Using artificial intelligence on application integration allows solving problems during the integration. This paper presents a systematic review about the usage of artificial intelligence on enterprise application integration. We observe that the researchers applied the following techniques: artificial neural networks, genetic algorithms, and fuzzy logic; the papers we found were classified in the following categories: i) development; ii) security; iii) optimization; iv) semantic mapping.

Keywords: Enterprise Application Integration. Systematic Review. Artificial Intelligence.

1. INTRODUÇÃO

Geralmente, as empresas possuem aplicações heterogêneas para dar suporte aos seus processos de negócios, formando assim seu ecossistema software heterogêneo (MANIKAS,



2016). Essas aplicações são utilizadas para diferentes finalidades e são concebidas com distintas tecnologias. O compartilhamento de dados e funcionalidades por aplicações heterogêneas, muitas vezes, é considerado um desafio nas empresas, sendo assim a área de integração de aplicações empresariais estuda ferramentas, metodologias e técnicas para criar soluções tecnológicas voltadas para realizar a integração das aplicações a fim de dar suporte aos processos de negócio (HOHPE; WOOLF, 2004).

Algumas ferramentas foram desenvolvidas com o intuito de abstrair os processos de integração, como o Apache Camel (IBSEN; ANSTEY, 2018), Spring Integration (FISHER et al, 2012), Mule ESB (DOSSOT; D'EMIC; ROMERO, 2014) e Guaraná (FRANTZ; CORCHUELO; ROOS-FRANTZ, 2016). Essas ferramentas operam de forma independente às aplicações que serão integradas e utilizam o conceito de troca de mensagens entre as aplicações para realizar as comunicações.

A inteligência artificial tem se tornado um dos principais campos de estudo da ciência da computação, pois permite automatizar diversas tarefas humanas e resolver problemas complexos (LU, 2019). Embora seja um campo antigo, diversas técnicas são apresentadas por pesquisadores, podem ser utilizadas em distintas áreas, como na integração de aplicações empresariais.

Nenhum trabalho foi encontrado na literatura com o intuito de estudar o uso de inteligência artificial no contexto de integração de aplicações empresariais. Portanto, este artigo propõe uma revisão sistemática da literatura sobre integração de aplicações empresariais e o uso de inteligência artificial. Foram identificados trabalhos envolvendo o uso de inteligência artificial para resolver problemas neste contexto, além de classificar os trabalhos a partir da técnica utilizada e o campo em que ela foi utilizada.

Foram encontrados um total de 17 trabalhos, sendo 5 selecionados para a discussão. Observou-se que a inteligência artificial foi utilizada nos campos de: processo de desenvolvimento, segurança, otimização e mapeamento semântico. Além disso, foi verificado que nenhum trabalho está ligado a qualquer ferramenta específica de integração de aplicações empresariais, e apenas um está restrito à otimização do processo de desenvolvimento de integrações. A classificação das técnicas utilizadas se deu por Redes Neurais Artificiais, Lógica Fuzzy e Algoritmos Genéticos.



O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta o protocolo de pesquisa; seguida pela Seção 3, que realiza a implementação do protocolo de pesquisa; por fim a Seção 4 que discute as principais conclusões.

2. PROTOCOLO DE REVISÃO

O Protocolo utilizado neste artigo é baseado no protocolo proposto por Kitchenham et al. (2009), sendo ilustrado na Figura 1. Ele é composto por 6 grandes etapas:

- Definição das questões de pesquisa: nesta etapa é realizada a definição das questões que serão respondidas com a pesquisa realizada.
- Escolha das bases de busca: é escolhida as bases em que serão realizadas as buscas dos trabalhos.
- Definição da string de busca: é montada a string com termos que vão ao encontro do assunto pesquisado.
- Busca dos trabalhos: nesta etapa é realizada a pesquisa dos trabalhos na base de busca.
- Escolha dos trabalhos: constitui-se em realizar uma leitura prévia dos trabalhos encontrados na etapa anterior. Não é realizada a leitura de todo o artigo, apenas do título, resumo, palavras chaves e introdução. Caso o trabalho seja considerado relevante, ele é selecionado. Caso contrário, é descartado. Além disso, também é realizada a inclusão/exclusão de artigos a partir de escolhas do autor.
- Análise dos trabalhos selecionados: são realizadas nesta etapa as questões de pesquisa e descrição dos trabalhos encontrados.
- Questões em aberto: por fim, são verificadas quais foram as questões que ficaram em abertas encontradas com a revisão.

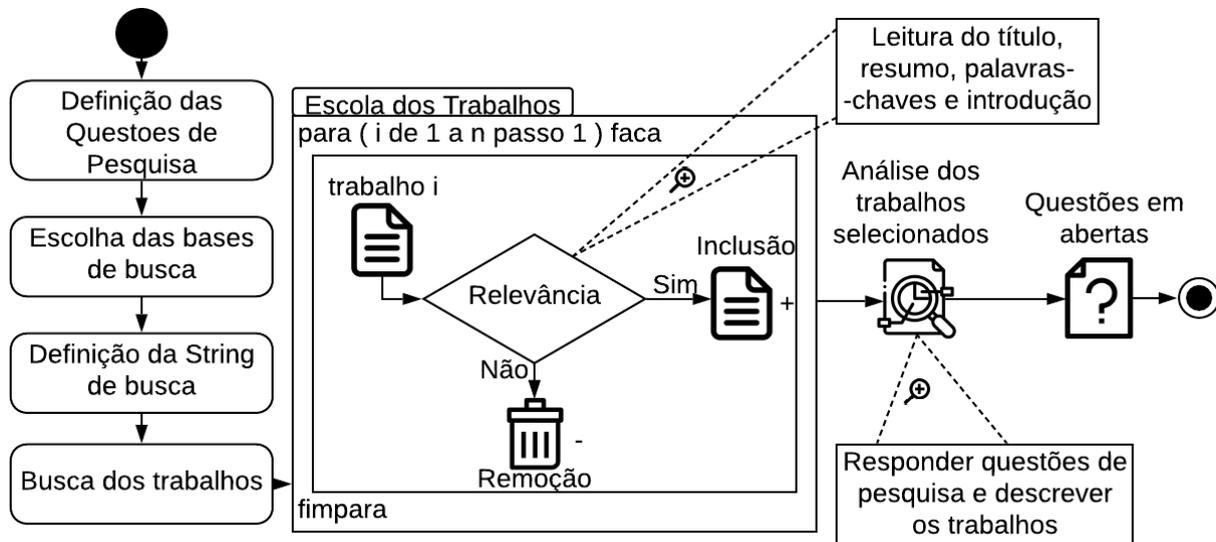


Figura 1. Protocolo de Revisão

3. EXECUÇÃO DO PROTOCOLO DE REVISÃO

Nesta seção, é descrita a implementação do protocolo de revisão proposto. São detalhadas as etapas, apresentados os resultados quantitativos e respondidas as questões de pesquisa.

3.1. DEFINIÇÃO DAS QUESTÕES DE PESQUISA

- QP1: Em quais tópicos da integração de aplicações empresariais foi utilizado inteligência artificial?
- QP2: Quais as técnicas de integração de aplicações empresariais são junto as técnicas de inteligência artificial?

3.2. ESCOLHA DAS BASES DE BUSCA

Existem diversas bases de buscas para encontrar trabalhos. Neste trabalho, optou-se pela utilização apenas da base Scopus, pois consiste em um amplo número de trabalhos disponíveis e de fácil acesso.

3.3. DEFINIÇÃO DA STRING DE BUSCA

Para a string de busca, utilizou-se um termo para realizar a busca de integração de aplicações empresariais e quatro termos ("Artificial Intelligence", "Neural Networks",



"Genetic Algorithm", e "Machine Learning") para buscar trabalhos de inteligência artificial. A string possui a seguinte forma:

```
("ENTERPRISE APPLICATION INTEGRATION") AND  
("ARTIFICIAL INTELLIGENCE" OR  
"NEURAL NETWORKS" OR  
"GENETIC ALGORITHM" OR  
"MACHINE LEARNING")
```

3.4. BUSCA DOS TRABALHOS

Conforme apresentado pela Figura 2, a busca trouxe o total de 17 trabalhos, sendo 10 trabalhos em conferência e 7 trabalhos publicados em periódicos. Entre os 17 trabalhos, 3 foram escritos pelos mesmos autores, Beer e Hassan.

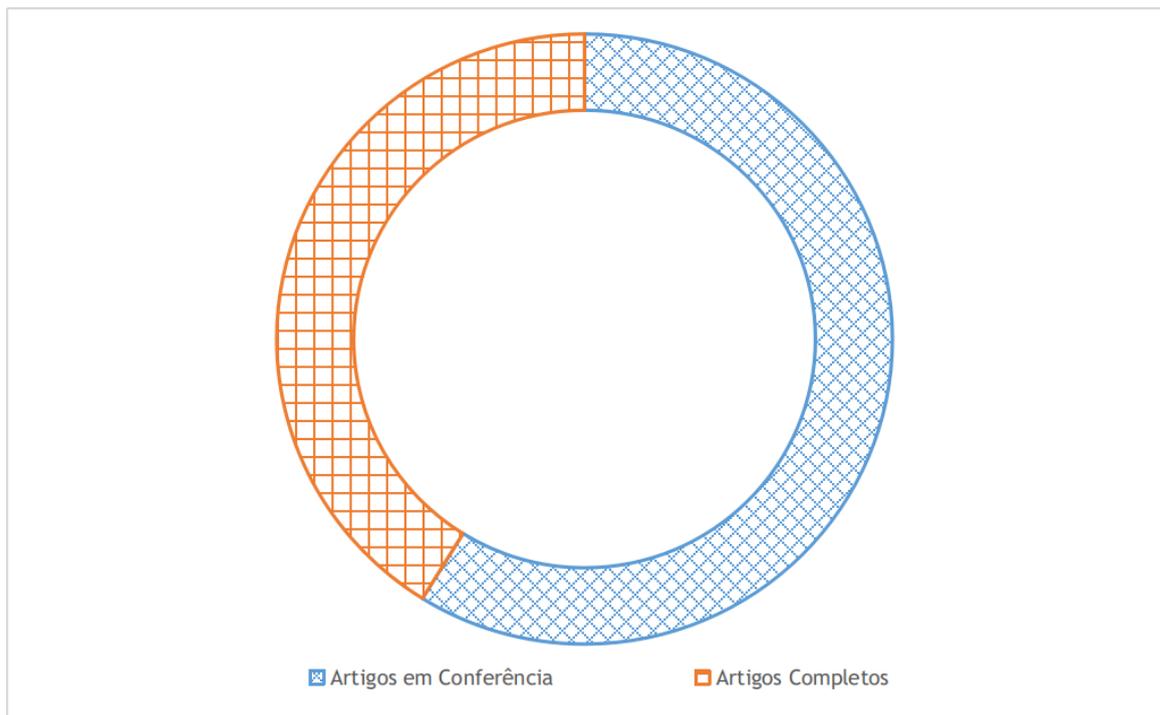


Figura 2: Divisão dos trabalhos por tipo

A Figura 3 apresenta a quantidade de trabalhos encontrados por ano. É possível verificar que a maior quantidade de trabalhos encontrados foi no ano de 2011, com 4 trabalhos. Por outro lado, não foram encontrados trabalhos publicados em 2017.

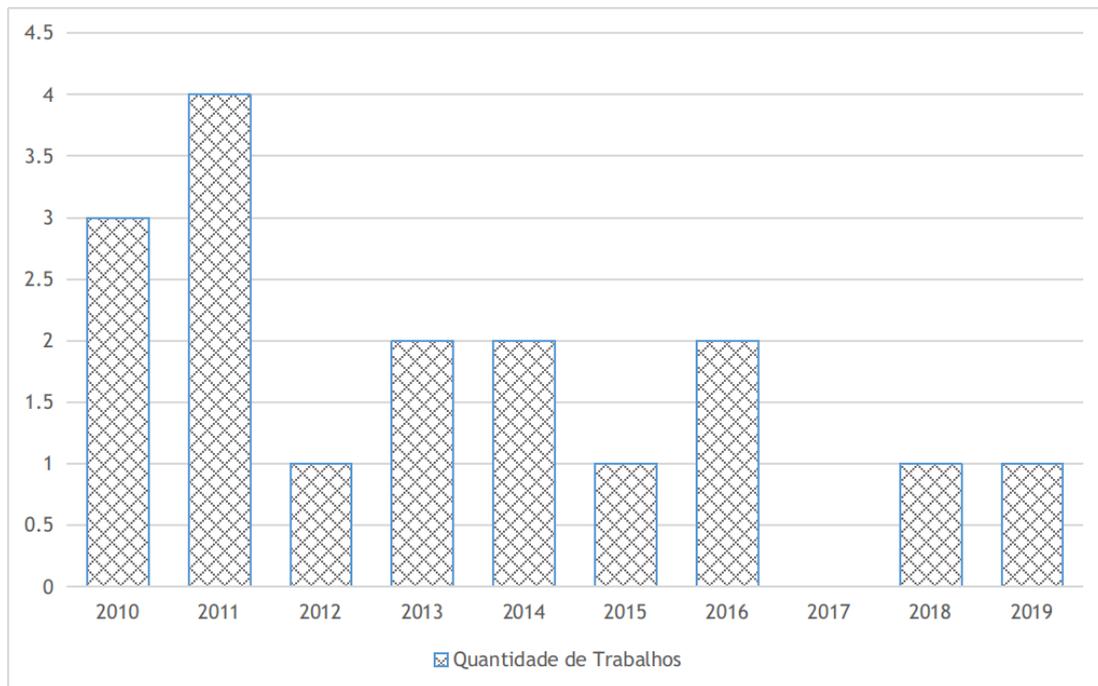


Figura 3: Divisão dos trabalhos por ano

3.5. ESCOLA DOS TRABALHOS

A escolha dos trabalhos foi realizada a partir das leituras do título, resumo, palavras-chaves e introdução. Caso a leitura desses tópicos apresentasse relação com a pesquisa, o trabalho era selecionado. Muitos trabalhos foram encontrados pela string de busca devido a termos controlados e não controlados, entretanto no momento da leitura eles foram removidos. Além disso, foram adicionados critérios de exclusão: trabalhos anteriores ao ano de 2010, revisões de conferências e trabalhos que não foram escritos em inglês.

3.6. DISCUSSÃO

Nesta Seção, são discutidas as questões de pesquisas propostas na Seção 3.1. São abordados as questões 1 e 2, realizando também uma discussão sobre os trabalhos.

3.6.1 QP1

A triagem de defeitos é uma etapa do desenvolvimento de software que ocorre durante a fase de testes com complexo processo que busca unir diferentes times, como desenvolvimento, negócio, e teste (KAUSHIK et al, 2013). Ela é subdividida em três



processos: análise, priorização, e atribuição de defeitos (KAUSHIK et al, 2013). A priorização dos defeitos impacta diretamente na qualidade, custo, e lançamento do produto. Segundo GUPTA, KUMAR e KAPUR (2019), a priorização é realizada de forma subjetiva e manual pelos coordenadores dos times de negócio. Em um cenário de integração de aplicações empresariais, que recebe em média 80 a 100 defeitos diários, os times acabam acumulando mais de 1000 defeitos, causando assim uma grande necessidade de priorização dos defeitos. O trabalho de (GUPTA; KUMAR; KAPUR, 2019) propôs um *framework*, com o uso de inteligência artificial, para realizar a escolha dos defeitos que serão priorizados durante o desenvolvimento de integração de aplicações. Este trabalho é classificado no campo de desenvolvimento de integração de aplicações.

Uma das principais tecnologias presentes nas ferramentas de integração de aplicações empresariais, é a Arquitetura Orientada a Serviço (SOA). Sua arquitetura é composta por: *Requester* que procura pelo serviço necessários que são publicados no *Registry* e consome as funcionalidades necessárias oferecidas pelo *Provider*. A comunicação na SOA é realizada a partir de WSDL, que é um documento XML que descreve a interface da comunicação (IBRAHIM; HASSAN, 2015). O trabalho de IBRAHIM e HASSAN (2015) propôs um *framework*, chamado de Motor de Segurança Inteligente (ISE), que atua como forma de segurança de aplicações que utilizam SOA, neste caso as ferramentas de integração de integração de aplicações empresariais. Ele pode ser colocado diretamente em um servidor web em que os *web services* estão rodando, ou em um servidor *proxy*. Este trabalho é classificado no campo de segurança de integração de aplicações empresariais.

Uma extensão do trabalho proposto por IBRAHIM e HASSAN (2015) foi realizada no trabalho de BEER e HASSAN (2018). Neste novo trabalho, os pesquisadores introduziram uma nova filosofia para o *framework* ISE: "*Learn, Predict, and Prevent*". O conceito de *Learn* está ligado ao aprendizado de possíveis ameaças à segurança na SOA utilizando inteligência artificial; *Predict* visa prever possíveis ataques na SOA a partir de resultados obtidos por modelos de segurança teóricos; *Prevent* visa criar algoritmos para realizar a prevenção de ataques na SOA. Neste trabalho, eles também realizaram testes comparativos com Seguranças Web Padrões, Produtos de Fornecedores (solução baseada em hardware que opera no nível do roteador IP) e o ISE. A comparação na prevenção de ataques de DoS/DDoS mostrou que o ISE permaneceu estável, o Produto de Fornecedores não conseguiu se manter estável quando



o número de requisições DoS é aumentado, e a Segurança Padrão Web mostrou-se totalmente inadequada. Por outro lado, o ISE apresentou maior consumo de processamento do CPU em relação ao Produto de Fornecedores, e menor consumo em relação a Segurança Padrão Web. No teste de nível de proteção das mensagens, o ISE obteve melhores resultados com o incremento do número de requisições em relação aos dois outros competidores. Por fim, para a ocorrência de ataques de segurança para as solicitações de amostra fornecidas o ISE obteve melhor resultado que as demais. Com esse trabalho, os pesquisadores puderam observar que o ISE obteve melhores resultados que as outras técnicas de segurança disponíveis. Este trabalho é classificado no campo de segurança de integração de aplicações empresariais.

Integração de dados é constituída pela troca de dados entre as aplicações a partir de *queries*. Ao utilizar integração de dados, um desafio encontrado é a eficiência das *queries*. O trabalho proposto por CORNIȚĂ, STRUNGARU e PAȘCA (2013) apresentou um Módulo de Integração de Software (SIM), que faz parte do servidor de kernel do sistema de banco de dados, que cria um considerável número de *queries* SQL por segundo para um banco de dados relacional, com a utilização de tabelas temporárias e *buffers*. A proposta deles inicia-se com a adição de um campo *timestamp* nas tabelas dos bancos de dados, após a integração de um conjunto de dados, o campo T_s correspondente será atualizado com o momento da integração desses dados. A condição $T_s > T_1 \& T_s \leq T_2$, onde T_1 e T_2 serão representados pelo início e fim da integração, será levada em consideração para realização da modificação do SQL. A proposta visa que todo o SQL de integração seja executado em um período máximo de 12 segundos, usando tempos de execução SQL anteriores, os tempos limites da integração (T_1 e T_2), e tempos limites a fim de determinar novos tempos limites e tempos limites de integração. Este trabalho é classificado no campo de otimização de integração de aplicações empresariais.

O mapeamento semântico ocupa uma importante função quando diferentes aplicações devem interagir entre si, como é o caso de aplicações presentes em uma integração, pois possui o objetivo de identificar e mapear entidades semanticamente relacionadas (VILLÁNYI; MARTINEK, 2011). O trabalho de VILLÁNYI e MARTINEK (2011) provou que o desempenho do mapeamento de semântica pode variar em larga escala se não estiverem configurados adequadamente para o cenário específico e verificou que uma fonte potencial de



imprecisão pode ser a classificação falsa de pares de entidades com distâncias semânticas próximas ao valor limite. Com isso, eles propuseram uma nova forma de avaliar a precisão dos mapeamentos de semântica (chamada *weighted f-measure*), além de realizarem uma otimização na técnica de maximização da medida da precisão. Este trabalho é classificado no campo de mapeamento de semântica de integração de aplicações empresariais.

3.6.2 QP2

O trabalho de GUPTA, KUMAR e KAPUR (2019) utilizou Algoritmos Genéticos para realização de priorização de defeitos a serem corrigidos em ambientes de integração de aplicações empresariais. A técnica dos autores utiliza um sistema de pontuação para inferir informações no modelo. A pontuação foi criada a partir da identificação de atributos na bibliografia, e então são ranqueados por *experts*. A Equação 1 apresenta a equação final encontrada pelos autores, onde:

- SS: gravidade. Os valores são: 7 crítica, 5 alta, 3 média, e 1 baixa.
- PS: prioridade. Os valores são: 7 muito alta, 5 alta, média, 1 baixa.
- BTC: número de casos de teste bloqueados para cada defeito.
- R: classificação proposta para corrigir o *n*ésimo defeito
- TF: tempo necessário para corrigir o defeito
- EAI: pontuação numérica dos valores de EAI para cada defeito. Os valores dos defeitos são: 7 orquestração, 5 mapas/esquemas, 3 *pipelines*, e 1 adaptadores.

$$FS_n = (SS_n * PS_n) + \frac{BTC_n}{\sum_{k=0}^n BTC_k} + \frac{11 - R_n}{TF_n} + EAI_n$$

Equação 1: Equação de Pontuação

A proposta dos autores apresentou uma melhora de 46% no número de defeitos fechados fechados por um desenvolvedor para a próxima *release* da aplicação, 56% de melhora no fechamento das duas principais gravidades/prioridades de defeitos, e 21% de redução no bloqueio de casos de teste.



Nos trabalhos de IBRAHIM e HASSAN (2015), e BEER e HASSAN (2018). a estratégia utilizada no algoritmo de segurança é composta por: Chaves Simetricamente e Assimetricamente Criptografadas, Assinatura Digital, Certificados e Autoridades de Certificação, Infraestrutura de Chave Pública, Especificação de Gerenciamento de Chave XML, e Redes Neurais Artificiais. Sendo que as Redes Neurais Artificiais buscam aprender possíveis ameaças à segurança na SOA.

O trabalho de CORNIȚĂ, STRUNGARU e PAȘCA (2013) os autores utilizam Redes Neurais Artificiais para ajustar os tempos limites das *queries* eT_1 e T_2 baseado nos tempos de execução dos SQLs anteriores. A Rede Neural, é utilizada para prever novos T_1 e T_2 , que serão utilizados para modificar as cláusulas *where* do SQL a fim de diminuir o tempo de execução. Com este trabalho, os pesquisadores verificaram que o tempo de execução dos SQLs estava entre 3-10 segundo, muito menos se comparado com integração de SQL estática, segundo os autores.

Os autores VILLÁNYI e MARTINEK (2011) utilizaram a Lógica Fuzzy para otimizar a técnica de maximização da medida de precisão. Com os resultados, eles verificaram que a técnica de maximização da medida da precisão revisada apresentou, na maioria dos testes, aumento da medida de precisão escolhida. Também verificaram que os valores da maximização da medida da precisão revisada acabaram sendo mais altos em relação à aproximação de referência na maioria dos casos.

3.7 QUESTÕES EM ABERTO

A partir dos trabalhos encontrados, observou-se que diversas áreas da integração de aplicações empresariais podem ser exploradas. Entre elas:

- Gerenciador de Recursos: algumas integrações utilizam parametrizações de recursos (memória máxima, *threads*, quantidade máxima de mensagens de entrada) sendo assim poderia ser utilizado técnicas capazes de prever picos na integração e então alocar mais recursos computacionais. Além de reduzir os recursos em momentos de estabilidade. Isso permitiria que, quando necessário, houvesse um incremento de desempenho, além de uma redução dos mesmos



em momentos estáveis, ou seja, economizando recursos para momentos críticos.

- **Monitoramento:** identificação de possíveis falhas durante a integração e então corrigi-las em tempo de execução. Isso permite uma maior confiabilidade para as integrações, uma vez que as falhas podem gerar dados errôneos e/ou interromper as integrações.
- **Gerenciamento de Integração:** priorizações de processamento de determinadas atividades realizadas pela integração. Essa utilização permite que mensagens de maior prioridade sejam processadas primeiramente, sendo a prioridade determinada por técnicas de inteligência artificial acopladas nas integrações.
- **Ferramentas de Integração:** as ferramentas de integração auxiliam no desenvolvimento de integrações, por abstrair diversos processos. Com a utilização de inteligência artificial, mais abstrações poderiam ser realizadas para facilitar ainda mais o desenvolvimento de integrações.

4. CONCLUSÃO

A inteligência artificial é um dos principais campos de pesquisa da ciência da computação, e seu uso está se tornando comum em diversos campos de desenvolvimento de software. A integração de aplicações empresariais, permite com que diferentes aplicações possam comunicar-se e compartilhar dados e funcionalidades. Este artigo realizou uma revisão sistemática a fim de encontrar trabalhos que utilizam inteligência artificial durante a integração de aplicações empresariais.

Entre os trabalhos encontrados, 12 foram descartados pois estavam de acordo com os critérios de exclusão. Por outro lado, foram analisados 5 trabalhos dos quais a principal técnica utilizada foi Redes Neurais Artificiais, seguido por Algoritmos Genéticos e Lógica Fuzzy, sendo a classificação do uso de inteligência artificial representada por: i) desenvolvimento; ii) segurança; iii) otimização; iv) mapeamento semântico. Também observou-se que nenhum trabalho abordou o uso de ferramentas de integração de aplicações empresariais e também nenhum trabalho abordou explicitamente o uso de troca de mensagens para a realização de integrações.



AGRADECIMENTOS

Este trabalho é financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes).

REFERÊNCIAS BIBLIOGRÁFICAS

IBRAHIM, B. Mohamed; HASSAN, Mohd Fadzil. A new customizable security framework for preventing WSDL attacks. In: **2015 International Symposium on Mathematical Sciences and Computing Research (iSMSC)**. IEEE, 2015. p. 24-29.

BEER, Mohamed Ibrahim; HASSAN, Mohd Fadzil. Adaptive security architecture for protecting RESTful web services in enterprise computing environment. **Service Oriented Computing and Applications**, v. 12, n. 2, p. 111-121, 2018.

CORNIȚĂ, Vasile; STRUNGARU, Rodica; PAȘCA, Sever. Adaptive software integration module using neural networks. **UPB Scientific Bulletin, Series C: Electrical Engineering**, v. 75, n.1, p. 101-110, 2013.

DOSSOT, David; D'EMIC, John; ROMERO, Victor. **Mule in action**. Greenwich: Manning, 2014.

FISHER, Mark et al. **Spring integration in action**. Manning Publications Co., 2012.

FRANTZ, Rafael Z.; CORCHUELO, Rafael; ROOS-FRANTZ, Fabricia. On the design of a maintainable software development kit to implement integration solutions. **Journal of Systems and Software**, v. 111, p. 89-104, 2016.



GUPTA, Viral; KUMAR, Deepak; KAPUR, P. K. Optimizing the Defect Prioritization in Enterprise Application Integration. In: **Software Engineering**. Springer, Singapore, 2019. p. 585-597.

HOHPE, Gregor; WOOLF, Bobby. **Enterprise integration patterns: Designing, building, and deploying messaging solutions**. Addison-Wesley Professional, 2004.

IBSEN, Claus; ANSTEY, Jonathan. **Camel in action**. Simon and Schuster, 2018.

KAUSHIK, Nilam et al. Defect prioritization in the software industry: challenges and opportunities. In: **2013 IEEE Sixth International Conference on Software Testing, Verification and Validation**. IEEE, 2013. p. 70-73.

KITCHENHAM, Barbara et al. Systematic literature reviews in software engineering—a systematic literature review. **Information and software technology**, v. 51, n. 1, p. 7-15, 2009.

LU, Yang. Artificial intelligence: a survey on evolution, models, applications and future trends. **Journal of Management Analytics**, v. 6, n. 1, p. 1-29, 2019.

MANIKAS, Konstantinos. Revisiting software ecosystems research: A longitudinal literature study. **Journal of Systems and Software**, v. 117, p. 84-103, 2016.

VILLÁNYI, Balázs; MARTINEK, Péter. Justified performance measurement of schema matching solutions. In: **2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI)**. IEEE, 2011. p. 115-120.