

Mapeamento Objeto-Relacional no Contexto de Desenvolvimento Web

Igor Donatti G. da Silva¹, Matheus Anderson L. Gomes¹, Silvana M. Melo¹

¹Faculdade de Ciências Exatas e Tecnológicas (FACET)

Universidade Federal da Grande Dourados (UFGD)

Caixa Postal 364 – 79.804-97015.064 – Dourados – MS – Brasil

{igor.silva069,matheus.gomes498}@academico.ufgd.edu.br

silvanamelo@ufgd.edu.br

Abstract. *With the advancement of technology in development environments, there is an increase in the use of ORMs frameworks, which are used to abstract the database in a way that facilitates development. In this sense, this work seeks to analyze some of the existing tools on the market, measuring their most important aspects and bringing to light their real relevance for increasing productivity, and therefore, cost of time for building applications. The work focuses on the context of web development, using the Prisma tool as the main comparison tool.*

Resumo. *Com o avanço da tecnologia nos ambientes de desenvolvimento, nota-se um aumento da utilização de frameworks ORMs, que são utilizados para abstrair o banco de dados de uma maneira a facilitar o desenvolvimento. Neste sentido, este trabalho busca analisar algumas das ferramentas existentes no mercado, medindo seus aspectos mais importantes e trazendo à tona sua real relevância para o aumento de produtividade, e portanto, custo de tempo para a construção dos aplicativos. O trabalho foca no contexto de desenvolvimento web, utilizando a ferramenta Prisma como ferramenta principal de comparação.*

1. Introdução

No atual contexto globalizado, os sistemas web estão cada vez mais presente no dia a dia da população. Portanto, surgem tecnologias cada dia mais avançadas que buscam melhorar o desempenho de desenvolvimento, facilitando assim a criação das aplicações. Neste contexto de ambiente de desenvolvimento de softwares, a persistência de dados é uma tarefa essencial para grande parte das aplicações. Em geral, isso é feito através de banco de dados relacionais, que armazenam os dados em tabelas e colunas. Porém, a manipulação dos dados não é uma tarefa simples, especialmente em aplicações orientadas a objetos, por possuírem estruturas mais complexas. Os ORMs surgem como proposta de solução para lidar com este problema, fornecendo uma camada de abstração entre o banco de dados e código da aplicação.

Os ORMs têm se tornado muito populares nos últimos anos, com uma grande variedade de ferramentas existentes. Devido a grande variedade de opções de ferramentas torna-se importante uma maneira de avaliá-las a fim de selecionar a mais adequada para um determinado contexto.

A comunidade de pesquisa tem conduzido estudos visando a comparação e avaliação principalmente da eficiência entre os ORMs mais populares, como o framework Hibernate, executando testes com relação a desempenho, velocidade e tempo de execução, curva de aprendizado, entre outras características importantes.

Os estudos relacionados à área de desenvolvimento web e ORMs, trazem uma análise de diferentes ORMs, como por exemplo: Hibernate, Django ORM, Apache, Entity Framework e SQL queries. Apesar do Prisma se mostrar uma ferramenta importante para o Typescript e com crescimento de uso na comunidade de desenvolvedores, não foram encontrados estudos quanto ao seu desempenho e funcionalidade.

O Prisma utiliza o modelo de dados definido pela aplicação para gerar as tabelas e relacionamentos do banco de dados, tornando o processo de desenvolvimento mais simples e intuitivo. O Prisma também oferece suporte para diversos bancos de dados, além de possuir recursos avançados como a geração de esquemas GraphQL automaticamente a partir dos modelos de dados. A área acadêmica carece de estudos sobre ferramentas ORMs, principalmente na área de desenvolvimento web, que só cresce cada dia mais. O Prisma se mostra uma importante ferramenta da atualidade, portanto, sua escolha de objeto de estudo parte desses princípios e visa trazer novos estudos sobre esta área.

Este trabalho, tem por objetivo analisar o desempenho do Prisma, suas características e fornecer uma comparação quanto a outras ferramentas ORMs existentes no mercado, visando contribuir também para a comunidade acadêmica com um estudo aprofundado a cerca das funcionalidades e desempenho do ORM Prisma, servindo de comparação e inspiração para trabalhos futuros.

2. ORM

Por muito tempo, desenvolvedores deveriam mapear e criar manualmente as tabelas relacionais de um banco de dados, confeccionando scripts SQL de maneira direta e escrita, essa criação acaba por muitas vezes ser confusa e complexa, visto que um sistema back-end é construído em meio orientação objeto. Com tal, o surgimento da impedância dos dados foi inevitável, e como forma de resolução foi criado as técnicas de Object-Relational Mapping (ORM).

Object-Relational Mapping (ORM) é uma técnica confeccionada para lidar com a impedância de dados, onde trata-se do mapeamento de objetos de um sistema orientado a objetos para tabelas em um banco de dados relacional. Além de uma solução, tal técnica acaba por fornecer também um maior desempenho quando relacionada à construção de um back-end, visto que tal técnica leva a abstração das complexidades das operações de banco de dados.

Dentre funcionalidades atribuídas aos ORMs podemos citar, a sua interface de programação de alto nível, oferecendo execuções de CRUD (Create, Read, Update, Delete) no banco de dados de forma abstrata e muitas vezes com uma maior simplicidade em operações complexas. Outra funcionalidade é seu auto-mapeamento das classes e objetos da aplicação para tabelas e colunas correspondentes no banco de dados, assim excluindo a necessidade de criação de scripts SQL para tais funcionalidades e do mapeamento de objetos para tabelas, trazendo uma enorme facilidade de acesso e manipulação dos dados envolvidos.

Logo, suas utilizações se tornam inerentes ao desenvolvimento de aplicações web, trazendo inúmeras vantagens, como por exemplo, o maior desempenho, manipulação de dados facilitada e a diminuição de erros. Contudo, não somente vantagens são trazidas à tona por sua utilização, existindo algumas desvantagens sucintas, uma de suas desvantagens se vem por conta da possibilidade de menor eficiência perante scripts SQL manuais, sendo ocasionados especialmente em projetos grandes e complexos.

A partir de uma análise preliminar da literatura [Torres et al. 2017, Prisma 2022] é possível listar alguns dos principais ORMs para Node.js e Java, conforme apresentado na Tabela 1. São apresentadas também as características desses ORMs, como: (i) suporte a bancos de dados, sendo esta, a capacidade do ORM de se conectar e interagir com diferentes bancos de dados; (ii) suporte a cache de segundo nível, sendo esta, a capacidade de armazenar objetos em cache para melhorar o desempenho; (iii) facilidade de uso, sendo esta, a facilidade de configuração e utilização do ORM; (iv) documentação e comunidade ativa, sendo esta, a disponibilidade de documentação e uma comunidade ativa; (v) performance, sendo esta, uma característica importante para garantir a qualidade do sistema.

São listados na Tabela 1, a seguir, um conjunto de ORMs a partir da extração de dados da literatura [Torres et al. 2017, Prisma 2022], como parte dos ORMs mais utilizados, incluindo desde os mais antigos no mercado, quanto os mais novos com características promissoras.

Tabela 1. Principais ORMs para desenvolvimento web

ORMS (Object-relational mapping)	Linguagem de programação	Versão	Suporte a vários bancos de dados	Suporte a cache de segundo nível	Facilidade de uso	Documentação e comunidade ativa
JPA/Hibernate	Java	6.2	Amplo	Sim	Difícil	Excelente
MyBatis	Java	3.5.11	Amplo	Sim	Difícil	Razoável
Sequelize	Javascript	6.28	Amplo	Não	Fácil	Boa
Prisma	Typescript	4.9.4	MySQL, PostgreSQL, SQLite, SQLServer	Não	Fácil	Excelente
TypeORM	Javascript	0.3.11	Amplo	Sim	Fácil	Boa

3. Estudo Empírico

Devido a disponibilidade de uma grande variedade de ORMs hoje no mercado, como apontado na seção anterior, torna-se interessante a comparação entre os mesmos, a fim de identificar quais os mais adequados para determinado contexto, com características distintas. Uma alternativa para realizar esses estudos comparativos é o uso de estudos empíricos.

O estudo empírico é uma metodologia de pesquisa que procura coletar dados e informações sobre determinado problema a partir da análise de evidências concretas. A realização deste estudo necessita de uma abordagem sistemática e cuidadosa, visando garantir que os resultados obtidos sejam seguros, confiáveis e precisos.

Existem diversos tipos de estudos empíricos, dentre os quais podemos destacar: (i) Experimentos controlados, um estudo que envolve a manipulação de variáveis independentes para observar seu efeito sobre as variáveis dependentes, sendo considerado uma forma de avaliar a eficácia de intervenções e soluções tecnológicas; (ii) Estudos de caso, um tipo de estudo que realiza a análise detalhada de um ou mais casos específicos, com o foco em descobrir as causas e consequências de determinado fenômeno ou problema, servindo de utilidade para gerar novas teorias sobre questões complexas e multifacetadas.

4. Estudos Empíricos sobre ORMs

Um estudo empírico é proposto em [Procaccianti et al. 2016], o estudo conduz uma avaliação empírica sobre a eficiência energética do uso de ORMs, com o objetivo de analisar a eficiência energética em três abordagens diferentes para se acessar bancos de dados em PHP, sendo tais, SQL Queries, TinyQueries e Propel. Esse experimento usa um ambiente controlado, onde três fatores foram considerados, sendo eles, a abordagem em questão, operações de CRUD e o tamanho das tabelas do banco de dados. E teve como resultado uma melhor performance e eficiência energética por meio das SQL Queries, já o Propel trouxe o maior tempo de execução e o maior gasto energético e por fim TinyQueries foi levemente pior que as SQL Queries, porém extremamente melhor que o Propel.

Podemos citar também [van Zyl et al. 2009], outro experimento conduzido, com o intuito de avaliar o desempenho da utilização de ORMs atuando como um intermediário entre uma base de código orientada a objetos e um banco de dados relacional, nesse estudo o ORM analisado foi o Hibernate e suas possibilidades de configurações, sendo este um ORM voltado para a linguagem Java, sendo consolidado e antigo no mercado, estando no topo dos ORMs mais utilizados no contexto web, tabela 1, sua utilização pode ser vista em vários outros experimentos, como mostrado na tabela 4 posteriormente.

Dentre os estudos empíricos na tabela 4, podemos notar a ausência de experimentos envolvendo a linguagem JavaScript ou TypeScript, além disso, podemos notar um novo e promissor ORM, sendo muito utilizado no contexto de desenvolvimento web, para linguagens de JavaScript e TypeScript, o ORM Prisma, é extremamente performático e com muitas funcionalidades e características promissórias [Buzzi 2022]. Contudo, podemos notar que não há estudos empíricos sobre sua eficiência, e seu desempenho como um todo. Sendo então, o foco desse trabalho, onde procuramos trazer sua análise de desempenho por meio de um experimento empírico, e sua comparação perante o ORM Hibernate.

5. Mapeamento Sistemático

Visando encontrar estudos relacionados que propunham a avaliação de ORMs um mapeamento sistemático da literatura foi conduzido, seguindo o processo proposto por [Petersen et al. 2008], sendo dividido em quatro fases distintas, (i) planejamento, (ii) condução, (iii) seleção dos estudos e (iv) análise e discussão dos dados.

5.1. Planejamento

Na fase de planejamento, foram seguidas as seguintes etapas, começando pela definição do objetivo de pesquisa, o qual foi definido de forma clara, tivemos por objetivo definido

obter uma visão geral da área de pesquisa sobre mapeamento objeto relacional (ORMs) considerando estudos sobre avaliações experimentais dessas ferramentas. Posteriormente à primeira etapa, foi realizado a definição de protocolo, a qual englobou sub-etapas, dentre tais, a definição das questões de pesquisa, etapa na qual, foi definido de forma clara, as questões envolvendo tal pesquisa. Na pesquisa em questão, uma série de perguntas foram definidas:

- **Q1** - Quais ORMS têm sido avaliados por meio de estudos empíricos?
- **Q2** - Quais os fatores/características avaliados nos estudos sobre ORMs?
- **Q3** - Quais métricas utilizadas?
- **Q4** - Quais Benchmarks ou outra fonte dos dados foram utilizados durante as avaliações?
- **Q5** - Quais os principais resultados obtidos?

As bases de dados que foram utilizadas como fonte de dados para a pesquisa foram, a Association for Computing Machinery (ACM), IEEE Xplore e Scopus. Após a definição da fonte de pesquisa, foi definida a string de busca, sendo tal, a etapa na qual foi selecionado palavras chaves, a fim de se produzir uma string de busca relacionada a pesquisa e seu foco, para cada base de dados, a mesma string de busca avançada foi adaptada; na pesquisa em questão foram geradas três versões para a mesma string de busca, sendo tais, amostradas na tabela 2 a seguir.

Tabela 2. String de busca por base de dados

Base	String utilizada
ACM	((("experiment"or "empirical study"or "controlled study"or "case study"or "evaluation"or "empirical evaluation"or "comparison")) AND Abstract:(and ("ORM"or "ORMS"or "object-relational mapping"or "object relational mapping")))
IEEE Xplore	("experiment"OR "empirical study"OR "controlled study"OR "case study"OR "evaluation"OR "empirical evaluation"OR "comparison") AND ("ORM"OR "ORMS"OR "object-relational mapping"OR "object relational mapping")
Scopus	TITLE-ABS-KEY(object-relational-mapping OR orm) AND TITLE-ABS-KEY(experiment OR empirical-study OR controlled-study OR case-study OR evaluation OR empirical-evaluation OR comparison) AND (LIMIT-TO (SUBJAREA,"COMP"))

Por fim, foram definidos os critérios de inclusão e exclusão, nesta sub-etapa, foram criados critérios para se avaliar se determinado estudo trazido pela string de busca em uma base de dados, seria considerado em nosso mapeamento sistemático ou não. Na pesquisa em questão foram desenvolvidos um total de um critério de inclusão e seis critérios de exclusão, sendo tais critérios:

1. Critério de inclusão:
 - **CI1** - O estudo apresenta uma avaliação empírica de um ou mais ORMs.
2. Critérios de exclusão:
 - **CE1** - O estudo apresenta um ou mais ORMs mas não apresenta uma avaliação empírica.
 - **CE2** - O estudo apresenta uma avaliação empírica, porém não avalia ORMs.

- **CE3** - O estudo está incompleto ou é um resumo.
- **CE4** - O estudo está em uma linguagem diferente do Português e Inglês.
- **CE5** - O artigo é cópia ou resumo de um estudo já considerado.
- **CE6** - A sigla ORM não se refere a Mapeamento Objeto Relacional.

5.2. Condução

Durante o processo de condução colocamos em prática as definições do planejamento. Essa fase é composta por três etapas, sendo elas, a identificação dos estudos primários, sendo a fase responsável pela coleta inicial dos dados, nela foram trazidos 36 artigos pela base de dados ACM, 35 artigos pela IEEE Xplore e 150 pela Scopus.

5.3. Seleção e Análise de Dados

Após a identificação dos estudos primários, foram aplicados os critérios de inclusão e exclusão, sendo nesta parte, realizado a filtragem dos artigos selecionados; ao final dessa fase foram selecionados 14 artigos pela ACM, 4 artigos pela IEEE Xplore e 11 pela Scopus. Após a leitura dos artigos selecionados, realizamos novamente uma análise sobre os critérios de inclusão e exclusão, para sabermos se os artigos selecionados estão realmente de acordo com o intuito da pesquisa, ao final dessa etapa, foram mantidos, 5 artigos pela ACM, 3 pela IEEE Xplore e 4 pela Scopus.

Na tabela 3 são apresentados os 12 estudos primários incluídos ao final do mapeamento, incluindo o título e o ano de publicação.

Tabela 3. Artigos incluídos

Id	Título	Ano
1	Energy Efficiency of ORM Approaches: an Empirical Evaluation	2016
2	The influence of optimisations on the performance of an object relational mapping tool	2009
3	An Evaluation of the Hibernate Object-Relational Mapping for Processing Interactive Social Networking Actions	2014
4	Comparing the performance of object databases and ORM tools	2006
5	Enhanced segment trees in object-relational mapping	2013
6	Performance Evaluation of Transparent Persistence Layer in Java Applications	2010
7	Performance Analysis of .NET Based Object-Relational Mapping Frameworks	2014
8	Object Oriented Application Cooperation Methods with Relational Database (ORM) based on J2EE Technology	2007
9	Understanding the Energy, Performance, and Programming Effort Trade-offs of Android Persistence Frameworks	2016
10	Investigating the Effects of Object-Relational Impedance Mismatch on the Efficiency of Object-Relational Mapping Frameworks	2020
11	Performance Comparison of CRUD Methods using .NET Object Relational Mappers: A Case Study	2020
12	A Comparative Study of the Features and Performance of ORM Tools in a .NET Environment	2010

Por fim, após a realização da filtragem da base de dados, foi realizada a última etapa do processo de condução, sendo tal a extração dos dados, tal etapa, é a etapa onde

serão respondidas as questões de pesquisa desenvolvidas. Podemos notar nos resultados apresentados na tabela 4, o ORM Hibernate sendo o mais avaliado por meio de estudos empíricos (citado em 5 estudos), também podemos notar a característica mais comumente analisada, como sendo o tempo de execução, característica a qual tende a ter sua métrica em segundos, no final, podemos perceber, 4 dentre 12 experimentos foram feitos utilizando Database pessoal, e 2 dentre 8 restantes se utilizaram do Benchmarck OO7.

Além dessas características durante o processo de extração foram avaliados também os fatores de análise, este parâmetro, trata-se de quais ações foram feitas para realizar os experimentos, mas acaba por ser omitido por sua composição no geral ser formada por ações de CRUD, sendo essas ações básicas que podem ser realizadas em um banco de dados, tais ações são denominadas respectivamente Create (Adicionar dados), Read (Recuperar dados), Update (Modificar dados) e Delete (Remover dados), sendo a base para qualquer sistema de gerenciamento de dados.

Os resultados dos estudos levam a crer que, em geral, o uso de SQL Queries, tendem a ter um desempenho melhor quando comparados ao uso de ORMs. Também trazem conclusões sobre as configurações do ORMs, as mesmas são sensíveis e devem ser feitas de acordo ao projeto, sendo determinísticas para um melhor desempenho. Contudo, de forma geral, ORMs não somente trazem um desempenho em tempo de execução, mas também, no fluxo como um todo do desenvolvimento.

Tabela 4. Estudos empíricos que avaliam ORMs

Id	ORM	Características	Métricas	Benchmark/ Fonte dos dados
1	SQL queries, Propel, Tiny-Queries	Tempo de execução e Gasto energético	Segundos e Watts	https://www.favoroute.com
2	Hibernate	Desempenho (Tempo de execução e velocidade de execução)	Segundos e Porcentagem de melhoria na velocidade de execução	OO7
3	Hibernate	Tempo de resposta e Número de consultas SQL geradas	Tempo Médio de Resposta (RT) e Ação Social Avaliação (SoAR)	BG
4	Hibernate e db4o	Velocidade de execução	Segundos	OO7
5	Hibernate	Tempo de execução	Milissegundos	Esquema de banco de dados para empresas que fazem marketing multinível
6	Hibernate e Apache	Tempo de execução	Milissegundos	Database pessoal
7	Entity Framework e NHibernate	Tempo de execução	Segundos	Dellstore database

8	Hibernate, JPOX e JPA	Curva de aprendizado, tempo de construção, desempenho, portabilidade, implantação	Segundos	Database pessoal
9	Android SQLite, ActiveAndroid, greenDAO, OrmLite, Sugar ORM, e Java Realm	Desempenho	Segundos	DaCapo H2
10	Django ORM	Desempenho (Tempo de compilação, memória usada)	Milissegundos e Kb	Database pessoal
11	Entity Framework Core 2.2, nHibernate 5.2.3 and Dapper 1.50.5	Velocidade de execução	Milissegundos	Database pessoal
12	Microsoft Entity Framework, NHibernate	Tempo de execução	Milissegundos	banco de dados AdventureWorks

6. Metodologia e discussão proposta

Em busca de melhor compreender o assunto e prover um estudo empírico sobre experimentos para avaliar ORMs, seguindo fielmente os cenários existentes sobre a utilização de ORMs o desenvolvimento da pesquisa foi dividido nas seguintes etapas:

1. Analisar os cenários de experimentos recolhidos no mapeamento sistemático confeccionado anteriormente.
 - a. Método: revisão da literatura.
 - b. Artefatos gerados: cenários de experimentos definidos.
2. Definir características/métricas de experimentos
 - a. Método: planejamento e modelagem dos experimentos.
 - b. Artefatos gerados: características definidas.
3. Definir Benchmarck/Fonte de dados
 - a. Método: revisão da literatura e experimentos por meio da utilização.
 - b. Artefatos gerados: Benchmarck/Fonte de dados definidas.
4. Desenvolver cenário final de experimentos para análise de ORMs
 - a. Método: planejamento, modelagem dos experimentos, implementação e teste dos experimentos.
 - b. Artefatos gerados: avaliação experimental de ORMs.

7. Avaliação Experimental de ORMs

Avaliação experimental de ORM trata-se de um tema comum na comunidade de desenvolvedores. Essa avaliação é utilizada como forma de comparação entre a eficácia de diferentes ferramentas ORM em relação a um conjunto de critérios pré-definidos. Essa é avaliação da qual será submetida a ferramenta Prisma, comparando ela com ferramentas que são fortemente difundidas no mercado e na literatura acadêmica. Assim, as próximas etapas da pesquisa são descritas a seguir:

1. **Estudar o sistema e definir os objetivos:** nesta etapa, busca-se estudar as funcionalidades e possibilidades com a ferramenta que é alvo das comparações, neste caso, o ORM Prisma.
2. **Determinar os serviços oferecidos pelo sistema:** esclarecer quais são os reais objetivos e o intuito da ferramenta, visando encontrar o melhor cenário de teste para a mesma.
3. **Selecionar métricas de avaliação:** com os dados obtidos anteriores, determinar quais serão os pontos a serem analisados, juntamente com o padrão de testes.
4. **Determinar os parâmetros que afetam o desempenho do sistema:** nesta etapa, são definidas as variáveis que afetam a ferramenta.
5. **Determinar o nível de detalhamento da análise:** aqui define-se, através das métricas, até onde a pesquisa irá abordar da ferramenta, definindo limites de pesquisa.
6. **Determinar a Técnica de Avaliação apropriada:** através das análises, é possível definir quais são as melhores técnicas que irão gerar os melhores resultados.
7. **Determinar a carga de trabalho característica:** visa estimar o tempo e o esforço necessários para completar um projeto de software.
8. **Realizar a avaliação e obter os resultados:** etapa na qual é realizado os testes na ferramenta, juntamente com todas as informações acumuladas.
9. **Analisar e interpretar os resultados:** comparação com as métricas estabelecidas e com os testes encontrados no mapeamento sistemático.
10. **Apresentar os resultados:** apresentação para a comunidade acadêmica dos resultados encontrados. Nesta etapa pretende-se submeter o artigo que contém os resultados do estudo em um evento científico ou periódico na área de pesquisa.

8. Resultados Esperados

Ao final do trabalho, espera-se obter uma análise completa do framework Prisma para desenvolvimento web, focado em obter resultados relacionado ao seu desempenho comparado a outros frameworks existentes no mercado e na literatura. Os resultados do trabalho serão publicados no formato de artigo científico, buscando divulgar os resultados da pesquisa para a comunidade de pesquisa em banco de dados e também como forma de contribuição aos estudos e avanço da tecnologia na parte de desenvolvimento web.

9. Conclusões

O artigo discute o problema de impedância de dados existentes na atualidade, no contexto de desenvolvimento web e como as Ferramentas ORMs propõem uma solução para este problema através de uma camada de abstração entre o código e o banco de dados. O uso de ORMs vem sendo cada vez maior no mundo de desenvolvimento, com estudos mostrando seu impacto na produtividade dos desenvolvedores que o utilizam.

A ferramenta Prisma é um exemplo de ORM que tem recebido atenção na comunidade de desenvolvedores e chamando atenção pela sua fácil e rápida curva de aprendizagem. O presente trabalho propõe uma análise empírica do ORM Prisma, comparando os seus resultados com outras ferramentas já existentes no mercado e com outros estudos já realizados na academia.

Este trabalho propõe um estudo experimental utilizando a ferramenta Prisma, realizando uma avaliação com outras ferramentas já existentes no mercado e com outros estudos presentes na academia. Para tal, foi realizado um mapeamento sistemático com o objetivo de buscar e considerar os estudos já feitos na área de desenvolvimento web. O trabalho encontra-se na fase de planejamento do experimento, levando em conta as teorias e práticas aprendidas através do mapeamento sistemático dos artigos encontrados.

Os resultados do mapeamento sistemático conduzido levam a crer que embora existam estudos de avaliação de desempenho para alguns dos principais frameworks de ORMs utilizados na atualidade, nenhum estudo teve como foco a ferramenta Prisma, sendo os estudos, em sua grande maioria, sobre o ORM Hybernate, utilizado através da linguagem Java, demonstrando estudos sobre desempenho e aumento de produtividade.

O trabalho busca contribuir para a comunidade de desenvolvimento web e banco de dados demonstrando o ganho de desempenho real com a utilização de frameworks ORMs na construção de aplicações. Os resultados dessa pesquisa podem servir como guia para pesquisadores e desenvolvedores que visam estudar e analisar as ferramentas ORMs, impactando também na área de pesquisa com resultados que podem ser comparados e replicados em estudos futuros.

Referências

- [Buzzi 2022] Buzzi, F. (2022). Prisma: uma das melhores coisas que já aconteceu no ecossistema?
- [Petersen et al. 2008] Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, page 68–77, Swindon, GBR. BCS Learning Development Ltd.
- [Prisma 2022] Prisma (2022). Best 11 orms for node.js, query builders & database libraries in 2022.
- [Procaccianti et al. 2016] Procaccianti, G., Lago, P., and Diesveld, W. (2016). Energy efficiency of orm approaches: An empirical evaluation. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '16, New York, NY, USA. Association for Computing Machinery.
- [Torres et al. 2017] Torres, A., Galante, R., Pimenta, M. S., and Martins, A. J. B. (2017). Twenty years of object-relational mapping: A survey on patterns, solutions, and their implications on application design. *Information and Software Technology*, 82:1–18.
- [van Zyl et al. 2009] van Zyl, P., Kourie, D. G., Coetzee, L., and Boake, A. (2009). The influence of optimisations on the performance of an object relational mapping tool. In *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, SAICSIT '09, page 150–159, New York, NY, USA. Association for Computing Machinery.