

ANÁLISE E PROJETO DE UMA PLATAFORMA WEB PARA ENSINO DE PROGRAMAÇÃO PARALELA UTILIZANDO BLOCKLY

João Victor Rocha Brum¹; Ana Paula Canal²

RESUMO

O ensino de programação paralela é constantemente discutido por sua dificuldade de aprendizado e a necessidade de ferramentas que viabilizem o estudo fora do ambiente institucional. Diante disso, esse trabalho tem como objetivo elaborar análise e projeto de um portal para o ensino dos conceitos de OpenMP e MPI por meio de linguagem visual. Para o desenvolvimento do portal será utilizada a linguagem de programação JavaScript com os frameworks AdonisJS e ReactJS, junto ao banco de dados MySQL e a API Blockly, linguagem visual. Esse projeto visa facilitar o acesso aos conceitos sobre os paradigmas apresentados com o uso de uma ferramenta visual. O trabalho foi norteado pelo método ágil FDD (*Feature Driven Development*).

Palavras-chave: Paralelismo; Programação visual; Computação de alto desempenho; OpenMP; MPI.

Eixo Temático: Tecnologia, Inovação e Desenvolvimento Sustentável (TIDS)

1. INTRODUÇÃO

A programação paralela, desde o surgimento das arquiteturas de múltiplos núcleos (*multicores*), vem constantemente se tornando um conhecimento mais requisitado e presente no dia a dia para os profissionais da área da computação (SOARES; NOBRE; FREITAS, 2019).

O conceito de computação de alto desempenho ou *High-Performance Computing* (HPC) é um dos temas referentes a arquitetura multicore e programação paralela. Este corresponde à supercomputadores ou clusters de servidores construídos para o processamento de grandes volumes de dados e em velocidades altas (ORACLE BRASIL, 2021). Dentro deste tema, a programação paralela é essencial para o devido funcionamento dos supercomputadores, pois, para alcançar

¹ Acadêmico de Ciência da Computação – Universidade Franciscana (UFN) – joaovrb.rs@gmail.com.

² Orientadora – Universidade Franciscana (UFN) – apc@ufn.edu.br.

os níveis de processamento e velocidade desejados se faz necessário a utilização deste paradigma, que é uma forma de executar diversas partes da aplicação simultaneamente, de forma cooperativa, usufruindo da comunicação entre processos e *threads* (BRAGA; BEZERRA; GARCIA, 2015).

Com isso em mente, Soares, Nobre e Freitas (2019) apresentam uma pesquisa referente ao ensino de programação paralela, em cursos de graduação, em diversos locais do mundo, como: Rússia, Brasil, China, Espanha, Estados Unidos da América (EUA), Índia, Suécia, entre outros. Nessa pesquisa, são levantados questionamentos e quantificados artigos relevantes ao assunto. Com essa, foi possível observar que, na maioria das graduações o estudo de paralelismo dá-se ao fim da graduação, por volta do sexto semestre em diante e apenas alguns temas são abordados por serem obrigatórios. Também, que, fora os EUA, a quantia de pesquisas referentes ao ensino de programação paralela é relativamente baixa, variando de uma a cinco pesquisas por nação.

O paralelismo, em termos didáticos, é considerado um tema complexo para o ensino, principalmente para os iniciantes na área da computação, pois diversas novas habilidades necessitam serem adquiridas e treinadas (LIMA *et al.*, 2018).

Um dos desafios do ensino de programação paralela, dá-se por conta de que, ao contrário da programação que é ensinada durante quase todo o decorrer dos cursos de informática, isto é, a programação sequencial, a programação paralela requer diferentes aspectos para serem desenvolvidos, como processos e *threads*, particionamento de dados, comunicação e sincronização entre processos (LIMA *et al.*, 2018).

Segundo Soares, Nobre e Freitas (2019) a apresentação do conceito e programação em paralelo nos períodos iniciais dos cursos juntamente com a programação sequencial é de suma importância para o desenvolvimento profissional do aluno. Isso se dá pelo fato de que o aluno passará a tratar a programação paralela como algo natural na programação, impedindo assim que este conceito seja interpretado como algo avançado e pouco utilizado. Também é apresentado como meio facilitador a possibilidade de portais educacionais que permitem o estudo dos

conteúdos fora do ambiente institucional e é reforçada a necessidade desses estudos e pesquisas, pois poucos trabalhos abordam o tema.

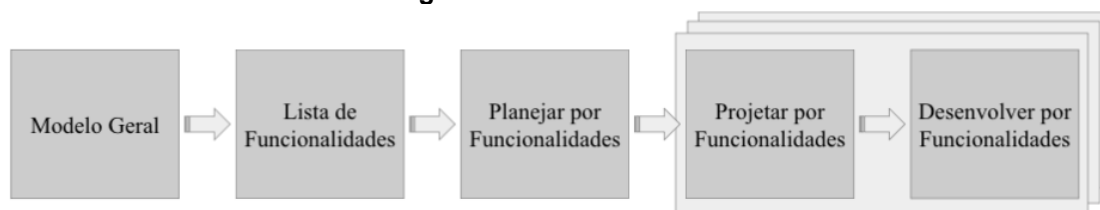
Neste contexto, tem-se por objetivo analisar e projetar uma ferramenta web para o ensino de conceitos sobre programação paralela. Abrangendo, a parte teórica e a parte prática, esta, voltada à programação visual dos temas OpenMP e MPI.

2. METODOLOGIA

A metodologia ágil utilizada para o desenvolvimento do projeto é a FDD (*Feature Driven Development*). O desenvolvimento dirigido a funcionalidades (FDD) utiliza como filosofia as seguintes premissas: comunicação detalhada tanto visualmente quanto verbalmente; gerência das dependências e complexidade via fracionamento embasados nas funcionalidades; e por fim colaboração da equipe. Além disso é salientado, em sua metodologia, a importância de atividades que garantam a qualidade de software, como: inspeções de código e projeto, desenvolvimento incremental e aplicação de auditorias (PRESSMAN, 2010).

Conforme Pressman (2010) existem cinco medidas que compõem o FDD, sendo elas: a definição geral do projeto; definição das funcionalidades, de forma que fiquem agregadas em conjuntos e temáticas semelhantes; planejamento baseado em funcionalidades; projetar por funcionalidades, ou seja, um pacote de projeto ou sequência; e desenvolver por funcionalidades. Como ilustrado na Figura 1.

Figura 1- Processo do FDD



Fonte: Adaptado de Pressman, 2010

Segundo Mazuco (2017), o FDD une as principais vantagens das metodologias disponíveis, possuindo uma única abordagem completa para a Engenharia de Software. Além disso, ela chama a atenção para as suas características, sendo algumas delas: modelagem em objetos de domínio; desenvolvimento orientado por funcionalidades, facilitando a entrega e descrição

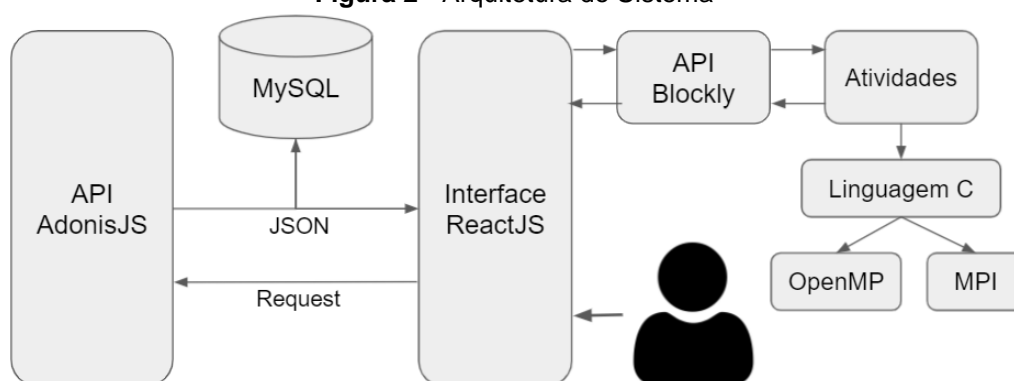
delas; integração regular e hierárquico relacionados aos negócios; resultados úteis a médio prazo; planejamento detalhado; boa visibilidade do projeto.

3. DESENVOLVIMENTO

Nesta seção, é descrita a proposta da aplicação como um todo, abrangendo as etapas de desenvolvimento a partir da metodologia FDD.

O sistema proposto será desenvolvido para a plataforma Web, sendo responsivo para adaptar-se aos aparelhos mobile. O portal disponibilizará recursos para serem usados no ensino de programação paralela, possuindo teorias dos conceitos propostos e atividades baseadas em programação visual, em blocos. Será realizada a construção dos blocos de código, na API Blockly, que serão utilizados na aplicação, em linguagem C. A Figura 2 ilustra a arquitetura do projeto.

Figura 2 - Arquitetura do Sistema



Fonte: Autor

Haverá dois tipos de usuários, o usuário aluno e o usuário mentor. Ambos os usuários poderão interagir com a atividade e com as sessões do portal, contudo o usuário mentor poderá realizar a revisão das atividades dos alunos.

Os usuários ao acessar o portal terão acesso ao material de estudos onde serão introduzidos aos conceitos do paralelismo de forma progressiva ou selecionando o tópico específico. Em sequência será apresentado ao usuário as atividades referentes ao assunto estudado que consistirá inicialmente em realizar a construção de código paralelo por meio de blocos na linguagem C.

Assim que a atividade for realizada poderá ser colocada para revisão e realizar o *download*, independentemente de o exercício ter sido revisado, do arquivo

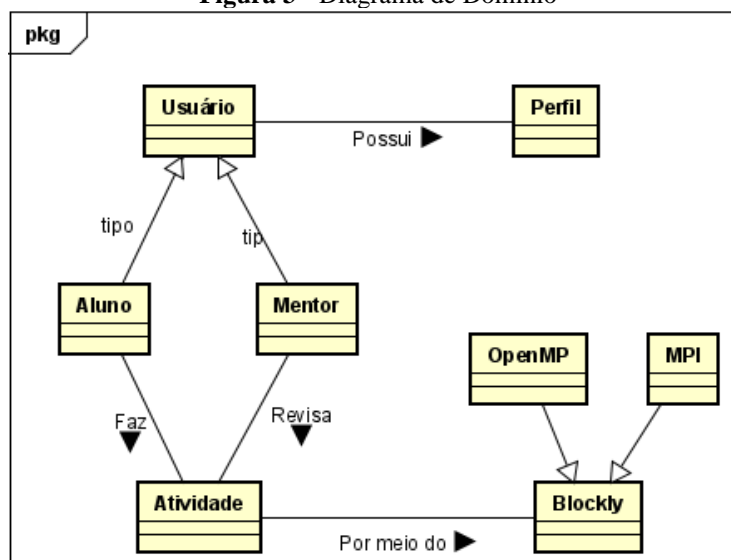
no formato C. Caso o usuário decida salvar o progresso do exercício será necessário realizar o login no portal. Quanto ao usuário mentor, além das permissões já descritas, poderá realizar a avaliação dos exercícios realizados pelo usuário aluno, por meio de pareceres descritos que poderão auxiliar o aluno no seu aprendizado.

- Nesta fase do trabalho foram utilizadas quatro etapas da metodologia FDD, sendo elas:
- Construção do modelo abrangente em que foi elaborado o diagrama de domínio;
- Lista de funcionalidades;
- Planejar por funcionalidades;
- Projetar por funcionalidades que corresponderá ao diagrama de classe.

3.1. MODELO ABRANGENTE

Seguindo as boas práticas do FDD, a primeira etapa do desenvolvimento é a definição do modelo abrangente que compreende o diagrama de domínio. Como ilustra a Figura 3. Nesse modelo é possível visualizar os conceitos e seus relacionamentos.

Figura 3 - Diagrama de Domínio



Fonte: Autor

Após o desenvolvimento do diagrama de domínio, o próximo passo da

metodologia é o desenvolvimento de uma lista de funcionalidades.

3.2. LISTAR E PLANEJAR POR FUNCIONALIDADE

O segundo e terceiro passo do FDD fazem referência a lista de funcionalidades e o planejamento do sistema. Nestas etapas, sob a visão do usuário e desenvolvedor, são analisadas as necessidades de desenvolvimento e a ordem em que estão listadas será a ordem em que serão desenvolvidas, conforme demonstrado os Quadros 1 e 2. Assim são elucidados os requisitos funcionais (RF) e requisitos não funcionais (RNF). Estes requisitos representam, respectivamente: as tarefas e comportamento que o sistema precisar ter; e características do sistema quanto à implementação e interfaces que não constituem as funcionalidades do sistema.

Quadro 1 – Requisitos funcionais do sistema

RF01 – O sistema deverá permitir o administrador gerenciar as funções de cadastro, alteração e exclusão de usuário.	Complexidade: Média	Relevância: Essencial
RF02 – O sistema deverá possibilitar o usuário se cadastrar.	Complexidade: Baixa	Relevância: Essencial
RF03 – O sistema deverá possibilitar o usuário realizar o log in no portal.	Complexidade: Baixa	Relevância: Essencial
RF04 – O sistema deverá permitir o usuário movimentar os blocos e conectar blocos.	Complexidade: Baixa	Relevância: Essencial
RF05 – O sistema deverá permitir o usuário movimentar os blocos do OpenMP.	Complexidade: Média	Relevância: Essencial
RF06 – O sistema deverá permitir o usuário movimentar os blocos do MPI	Complexidade: Média	Relevância: Essencial
RF07 – O sistema deverá possibilitar salvar o progresso atual da atividade e retornar no próximo acesso se for desejo do usuário.	Complexidade: Alta	Relevância: Essencial
RF08 – O sistema deverá permitir o usuário visualizar as atividades desenvolvidas.	Complexidade: Média	Relevância: Essencial
RF09 – O sistema deverá possibilitar um usuário, mentor, visualizar a atividade desenvolvida e realizar comentários.	Complexidade: Média	Relevância: Essencial
RF10 – O sistema deverá possibilitar o usuário	Complexidade:	Relevância:

realizar download do código como um arquivo no formato C.	Média	Essencial
RF11 – O sistema deverá possibilitar o usuário realizar alterações no perfil	Complexidade: Baixa	Relevância: Não - Essencial

Fonte: Autor

Quadro 2 – Requisitos não funcionais do sistema

RNF01 – Linguagem de Programação: O portal será desenvolvido com o uso das linguagens JavaScript, CSS e HTML.
RNF02 – Banco de Dados: Será utilizado para o armazenamento dos dados o banco MySQL.
RNF03 – API Server: O servidor será desenvolvido utilizando NodeJS com o framework AdonisJS.
RNF04 – Layout: O layout do portal será desenvolvido utilizando o framework ReactJS.
RNF05 – Programação em Blocos: Será utilizado a API Blockly do Google.

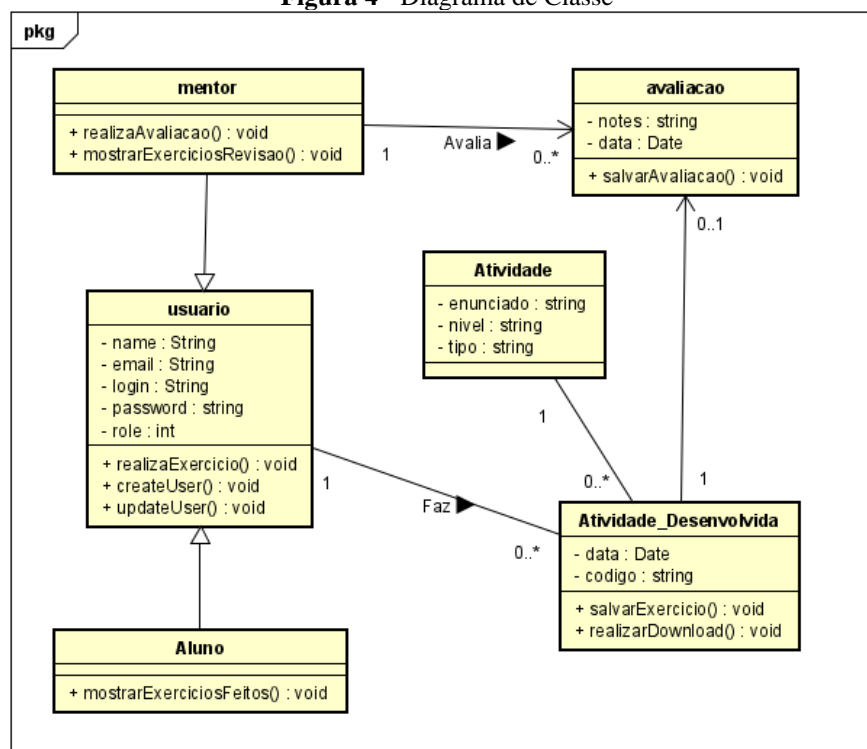
Fonte: Autor

Com a conclusão da lista de funcionalidades e o planejamento, é possível realizar o desenvolvimento do quarto passo da metodologia que é representada por planejar por funcionalidades.

3.3. PROJETAR

O quarto passo da metodologia corresponde à etapa de projetar por funcionalidades que possibilita uma visão estática do sistema que é representada pelo diagrama de classes, conforme a Figura 3. O diagrama apresenta as classes que irão fazer parte da implementação do sistema, e seus respectivos relacionamentos com outras classes. Cada classe possui alguns atributos e métodos que serão utilizados.

Figura 4 - Diagrama de Classe



Fonte: Autor

4. CONSIDERAÇÕES FINAIS

Neste trabalho, foi realizada a apresentação da análise e projeto para o desenvolvimento de um portal web, que poderá ser usado como recurso para o ensino de conceitos de programação paralela com OpenMP e MPI, com a linguagem visual com o auxílio da API Blockly, cujo objetivo é tornar o estudo do tema mais acessível e de fácil entendimento. Com os estudos realizados foi possível observar diferentes formas que a programação paralela era abordada, porém era notória a escassez de trabalhos referentes ao ensino.

A utilização da metodologia ágil FDD foi de grande relevância para apontar os passos a serem seguidos, de forma clara e direta, durante o desenvolvimento e elaboração do projeto. Com o desenvolvimento, até o momento, das quatro etapas da metodologia sendo elas o modelo geral como diagrama de domínio, lista de funcionalidades, planejamento por funcionalidades e projetar por funcionalidades com o diagrama de classes, foi possível ter uma noção maior quanto ao escopo e a complexidade do projeto.

Dessa forma, com os conceitos de programação paralela e suas bibliotecas OpenMP e MPI, as tecnologias para o desenvolvimento e etapas, referentes ao FDD, apresentadas neste artigo contribuirão para o desenvolvimento futuro da plataforma web para ensino de programação paralela utilizando Blockly.

REFERÊNCIAS

BRAGA, S.; BEZERRA, H.; GARCIA, F. “Crianças brincando”: Uma ferramenta para auxílio na aprendizagem de programação concorrente. In: XLIII CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 2015, Santo André. **Anais...** Santo André: [s.n.], 2015.

LIMA, A. C. *et al.* Uma Oficina para Ensino de Algoritmos Paralelos por Meio de Computação Desplugada. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2018, Fortaleza. **Anais...** Fortaleza: Sociedade Brasileira de Computação, 2018, p. 619–628.

MAZUCO, A. S. C. **Percepções de práticas ágeis em desenvolvimento de software**: benefícios e desafios. 2017. Dissertação (Mestrado em Computação Aplicada) – Universidade de Brasília, Brasília, 2017.

ORACLE BRASIL. **O que é computação de alto desempenho (HPC)?**. Disponível em: <https://www.oracle.com/br/cloud/hpc/what-is-high-performance-computing>. Acesso em: 28 set. 2021.

PRESSMAN, R. S. **Engenharia de Software**: Uma Abordagem Profissional. 7 ed. Porto Alegre: Mc Graw Hill, 2010.

SOARES, F. A. L.; NOBRE, C. N.; FREITAS, H. C. Parallel Programming in Computing Undergraduate Courses: a Systematic Mapping of the Literature. **IEEE Latin America Transactions**, v. 17, n. 08, p. 1371–1381, ago. 2019. Disponível em: <https://ieeexplore.ieee.org/document/8932371>. Acesso em: 28 set. 2021.