
Uma Ferramenta para Ensino e
Aprendizado de Exploração de Espaço
de Projeto de Arquiteturas de
Processadores na Era de Dark-Silicon

Erick Rocha Amorim

Uma Ferramenta para Ensino e Aprendizagem de Exploração de Espaço de Projeto de Arquiteturas de Processadores na Era de Dark-Silicon¹

Erick Rocha Amorim

Orientadora: *Prof^a Dr^a Liana D. Duenha*

Dissertação apresentada ao Programa de Pós-Graduação da Faculdade de Computação (FACOM) da Universidade Federal de Mato Grosso do Sul (UFMS), como parte dos requisitos necessários à obtenção para do título de Mestre em Ciência da Computação.

**UFMS - Campo Grande (MS)
Novembro/2023**

*À minha esposa,
Francielly.*

*A meus pais,
Eliane e Edmilson.*

*A minha irmã,
Ellen.*

Agradecimentos

Inicialmente, gostaria de agradecer aos meus pais, a quem devo tanto minha instrução, quanto minha educação. Foram suas mãos que me ampararam e me enveredaram pelo caminho da excelência acadêmica e profissional. Menção honrosa também a minha irmã mais velha. Juntos, eles três me permitiram, pelo emular e pelo ouvir, aprender o que é sacrifício, esforço e dedicação.

Agradeço à minha orientadora, Liana Duenha, pela preocupação e esforço que dedicou a minha pessoa e a este trabalho. Não faltaram percalços, inclusive os de natureza extremamente pessoal, e mesmo assim, em todos os momentos ela foi uma força tranquilizadora e motivadora, um verdadeiro farol de luz que me guiou durante essa árdua travessia.

Agradeço também aos professores do programa, com quem muito aprendi, e em especial aos professores Ricardo e Awdren, que fizeram contribuições diretas para a concepção do trabalho através de sua crítica extremamente construtiva durante a Banca de Qualificação.

Agradeço ainda aos colegas de laboratório, Gregório, Raíssa, Igor, Felipe, Laura, e os demais, todos também envolvidos em empreendimentos de pesquisa relacionados ao MultiExplorer. Também ao Mateus e à Rhayssa, que clarificaram pontos importantes sobre trabalhos anteriores relacionados a ferramenta.

Agradeço também aos meus amigos, que em momentos compartilhados de alegria e descontração me permitiram me ausentar em corpo e mente da dupla jornada a qual fui submetido por quase três anos.

O mais importante deixei para o final, como é de meu feitio. Minha esposa Francielly. Agradecer aqui não seria suficiente, pois não há como. Apenas reitero o compromisso de te apoiar e consolar pelo resto nossas vidas, tanto quanto você o fez por mim durante a realização deste trabalho. Te amo.

Abstract

Design space exploration is an activity useful in many engineering contexts, and considered crucial for the design of MPSoCs (often deployed in mobile devices and embedded systems). Therefore, it is a relevant topic for the future microelectronic technicians of the Brazilian industry. MultiExplorer is a tool initially proposed to support research in the field of processor architectures' design space exploration (DSE) in the dark-silicon era. Many studies were published in international conferences and scientific journals, using MultiExplorer in a range of applications, from mitigating the dark silicon on multicore and many-core architectures and performing heterogeneous computing DSE, to DSE on systems based on general-purpose computing on graphics processing units and cloud computing resource allocation. However, despite the recommendation to include DSE in the curriculum of engineering and computer science programs by the ACM (Association for Computing Machinery), and its relevance for the industrial development of the country, references to this topic in the curricula of those programs in Brazil are scarce. This work proposes the use of an enhanced version of the MultiExplorer DSE tool as an educational resource for teaching and learning dark silicon aware design space exploration on multicore architectures. This work features the enhancement of the tool for better usability, and a case study using the tool on an extracurricular minicourse for undergraduate Computer Science and Computer Engineering programs in Brazil. This has resulted in a more mature software architecture for the tool, transition to an open source code, and the establishment of a graphical user interface as to allow better student interaction with the tool. The prospect from the case study was positive.

Resumo

A exploração de espaço de projeto é uma atividade útil em diversos contextos de engenharia, e especificamente considerada crucial para o projeto de MPSoCs, usualmente empregados em dispositivos eletrônicos portáteis e sistemas embarcados. Portanto, o tópico é relevante para a formação dos futuros profissionais da indústria brasileira de microeletrônicos. O MultiExplorer é uma ferramenta inicialmente proposta para auxiliar a pesquisa no campo de exploração de espaço de projeto de arquiteturas de processadores na era de dark-silicon. Uma série de estudos foram publicados em periódicos científicos e conferências internacionais, utilizando o MultiExplorer em uma gama de aplicações, desde a mitigação de *dark silicon* em arquiteturas multi-núcleo, exploração de espaço de projeto com arquiteturas heterogêneas baseadas em utilização de unidades de processamento gráfico para computação de propósito geral (GPGPU), até alocação de recursos de computação em nuvem. Entretanto, a despeito de sua relevância para o desenvolvimento industrial, e das recomendações curriculares da ACM (*Association for Computing Machinery*), o tópico de exploração de espaço de projeto não parece ser abordado com frequência nos programas de graduação em Engenharia de Computação e Ciência da Computação no Brasil. Dado o contexto, este trabalho tem como objetivo fomentar a abordagem do tópico a nível de graduação a partir da disponibilização de uma versão aberta da ferramenta MultiExplorer que permita a exposição do tópico em um nível de abstração acessível para os alunos. O trabalho resultou no amadurecimento da arquitetura de software da ferramenta, abertura do código-fonte, estabelecimento de uma interface gráfica para melhor interação dos alunos com a ferramenta. Um estudo de caso avaliou o uso da ferramenta para aprendizado na graduação, por meio de um minicurso, e o prospecto foi positivo.

Sumário

Sumário	xii
Lista de Figuras	xiii
Lista de Figuras	xiv
Lista de Tabelas	xv
Lista de Tabelas	xvi
Lista de Abreviaturas	xvii
1 Introdução	1
2 Referencial Teórico	7
2.1 Exploração do Espaço de Projetos	7
2.2 <i>Dark Silicon</i>	9
2.3 <i>MultiExplorer</i>	11
2.4 <i>Design Sprint</i>	14
2.5 <i>Technology Acceptance Model</i>	17
3 Trabalhos Relacionados	18
4 Desenvolvimento do MultiExplorer	24
4.1 Melhoria do Processo de Instalação	24
4.1.1 Docker	25
4.1.2 Script de Construção	25
4.1.3 Snipersim	25
4.2 Prototipação da Interface Gráfica	26
4.3 Implementação da Interface Gráfica	27
4.3.1 Exemplo de Atividade de Exploração	38
4.4 Considerações Finais	43

5	Avaliação do MultiExplorer como Recurso Educacional	47
5.1	Objetivos e Planejamento	47
5.1.1	Introdução Teórica	49
5.1.2	Questionário Teórico	50
5.1.3	Atividade Prática	51
5.1.4	Relatório Técnico	52
5.1.5	Pesquisa de Experiência de Usuário	53
5.2	Execução	54
5.3	Resultados	55
5.3.1	Desempenho dos Alunos no Questionário Teórico	55
5.3.2	Desempenho dos Estudantes no Relatório Técnico	57
5.3.3	Resultados da Pesquisa de Experiência de Usuário	58
5.3.4	Dados Qualitativos	61
5.3.5	Testagem de Correlações entre as Atividades	61
5.4	Conclusões	64
5.5	Limitações e Ameaças à Validade	65
6	Considerações Finais	67
7	Referências Bibliográficas	69
	Referências	76

Lista de Figuras

2.1	Fluxo DSE genérico. Extraído de [1]	8
2.2	Representação gráfica do pipeline da ferramenta MultiExplorer. . .	12
2.3	Representação gráfica do processo de <i>Design Sprint</i> . Imagem extraída do livro “Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days” [2].	15
4.1	Mapa conceitual elaborado durante a sprint.	27
4.2	<i>Mockups</i> da interface elaborados durante a sprint.	28
4.3	Fluxo de execução elaborado para a construção do protótipo. . .	28
4.4	Visão geral da estrutura de diretórios do MultiExplorer (código aberto). Os pacotes selecionados (em azul) foram adicionados durante a implementação da interface gráfica.	30
4.5	Diagrama de classes parcial representando as classes envolvidas na construção da interface de entradas de usuário. A classe <code>InputGUI</code> tem um papel central na orquestração do processo. . . .	32
4.6	Diagrama de sequência ilustrando a comunicação que resulta na criação das abas de entradas de usuário.	33
4.7	Esse diagrama de sequência ilustra a comunicação entre a classe InputTab e a classe InputGUI que resulta na criação dos objetos que irão exibir as entradas de usuário na interface gráfica.	34
4.8	Diagrama de classes das classes envolvidas no controle da execução, apresentação da execução na interface gráfica e apresentação de resultados parciais.	35
4.9	Diagrama ilustrando as classes envolvidas na apresentação de resultados. A tela de resultados (objeto da classe ResultScreen) controla o processo de apresentação realizado pelos apresentadores (objetos de classes derivadas de Presenter	36

4.10	Por meio dos métodos EventFirer::add_handler e Widget::after() um objeto da classe StepDisplay pode monitorar e reagir ao progresso do processo de execução realizado por um objeto da classe Step	37
4.11	O objeto da classe ResultScreen recebe os resultados e apresentadores do fluxo em execução e repassa um <i>frame</i> para cada apresentação.	38
4.12	Na interface inicial, o usuário deve selecionar um fluxo de execução e pressionar o botão “Start”.	39
4.13	Na primeira aba de entradas de usuário do fluxo de exploração de CPUs Heterogêneas o usuário deve informar o número de núcleos, a litografia, e o modelo de núcleo.	40
4.14	Na aba de configurações de DSE o usuário pode modificar a calibragem do algoritmo evolutivo de busca (NSGA-II) e determinar as restrições de número de núcleos, área e densidade de potência máximas.	41
4.15	O fim de cada etapa permite a visualização de resultados parciais (“Preview Results”). Após a conclusão de todas as etapas da execução o botão “See Results” é habilitado.	42
4.16	Após o fim da exploração é possível visualizar um perfil de desempenho e parâmetros físicos da plataforma simulada, bem como um perfil semelhante das plataformas encontradas pelo NSGA-II.	43
4.17	Um das saídas disponibilizadas pela ferramenta é um gráfico ilustrando as plataformas encontradas pelo NSGA-II: as barras azuis representam desempenho, e as laranjas densidade de potência.	44
4.18	Nos casos em que a escala do espaço de busca é razoavelmente pequena, pode-se executar o algoritmo de força bruta para buscar por todas as soluções viáveis.	45
5.1	Distribuição dos alunos que preencheram a pesquisa de experiência de usuário entre os cursos de graduação.	55
5.2	Desempenho dos alunos no questionário teórico na forma de histogramas.	57
5.3	Histograma das notas obtidas pelos estudantes no relatório técnico.	59
5.4	Histogramas da Facilidade de Uso Percebida (PEOU) e Utilidade Percebida (PU).	59
5.5	Gráficos de caixa dos valores de PEOU e PU	61
5.6	Gráficos de caixa das medidas de aprendizagem: questionário teórico x relatório técnico.	63

Lista de Tabelas

1.1	Total de cursos de Engenharia (CE) e Ciência (CC) da Computação que contém os termos chave no projeto pedagógico. Foram encontradas poucas ocorrências relacionadas a DSE.	3
2.1	Dennard x Pós-Dennard.	9
3.1	Comparação de funcionalidades entre as ferramentas.	22
3.2	Comparação entre as ferramentas a respeito da presença de elementos que permitam abordar os tópicos de DSE e <i>Dark Silicon</i>	23
3.3	Comparação de aplicabilidade entre as ferramentas.	23
5.1	Evolução tecnológica (simulada) das arquiteturas baseadas no núcleo <i>Smithfield</i> . D.P.: Densidade de Potência. D.S.: <i>Dark Silicon</i> (área e porcentagem).	51
5.2	Exemplos de arquiteturas apresentadas pelo MultiExplorer após a exploração automática realizada pelo algoritmo NSGA-II. D.P.: Densidade de Potência	52
5.3	Desempenho dos estudantes no questionário teórico, incluindo a nota final ($Q = (q_1 + q_2 + q_3)/3$), as médias (\bar{x}) e desvios padrão (s_2) de cada coluna.	56
5.4	Notas obtidas pelos estudantes no relatório técnico, incluindo a média (\bar{r}) e o desvio padrão (s_2).	58
5.5	Avaliação da Facilidade de Uso Percebida (PEOU) da ferramenta MultiExplorer por parte dos estudantes, incluindo as médias (\bar{x}) e desvios padrão (s_2).	58
5.6	Avaliação da Utilidade Percebida (PU) da ferramenta MultiExplorer por parte dos estudantes, incluindo as médias (\bar{x}) e desvios padrão (s_2).	60

5.7	Sumário do <i>feedback</i> subjetivo dos alunos sobre a ferramenta. A atitude dos alunos parece ter sido neutra em relação a usabilidade, mas positiva no quesito de utilidade.	62
5.8	Essa tabela apresenta um exemplo da maneira que os <i>feedbacks</i> das questões abertas da pesquisa de experiência de usuário foram classificados para apresentação na Tabela 5.7.	62
5.9	Dados utilizados para verificar o efeito entre o desempenho no questionário teórico e o desempenho no relatório técnico.	63
5.10	Dados utilizados para testar o efeito do desempenho no questionário teórico sobre a experiência de usuário.	64
5.11	Dados utilizados para testar os efeitos da experiência de usuário sobre o desempenho no relatório técnico.	65

Lista de Abreviaturas

DSE Design Space Exploration

DS Dark Silicon

MIPS Milhões de Instruções por Segundo

MPSoC [MultiProcessor System-on-Chip]

CT [Centro de Treinamento]

DH [Design House]

CEITEC [Centro Nacional de Tecnologia Eletrônica Avançada]

ACM [Association for Computer Machinery]

CC Ciência da Computação

EC Engenharia da Computação

LSCAD Laboratório de Sistemas Computacionais de Alto Desempenho

TAM Technology Acceptance Model

TRA Theory of Reasonable Action

TPB Theory of Planned Behavior

PU Perceived Usefulness

PEOU Perceived Ease of Use

MVC Model-View-Controller

GUI Graphical User Interface

OO [Orientação a Objetos]

IA *Inteligência Artificial*

GPU *Graphics Processing Unit* %acroSIMT[SIMT]Single Instruction Multiple Thread

CPU *Central Processing Unit*

AM *Aprendizado de Máquina*

DRAM *Dynamic Random-Access Memory*

GP-GPU *General Purpose Graphics Processing Unit*

DS-DSE *Domain-specific Design Space Exploration*

CUDA SDK *Software Development Kit for GPU Computing*

API *Application Programming Interface*

IPC *Instructions per Cycle*

Introdução

A atividade de exploração do espaço de projeto, ou *Design Space Exploration* (DSE), refere-se a uma análise sistemática de diferentes alternativas de projeto. Realizada antes das fases de implementação, é uma atividade orientada por objetivos e restrições. Esses objetivos usualmente consistem na maximização e minimização de indicadores sobre desempenho e custo, como: largura de banda, tempos de atraso, MIPS (Milhões de Instruções por Segundo), consumo energético, etc. As restrições podem corresponder a limitações físicas ou de custo do projeto final, como custo total de implantação, área ou densidade de potência máxima aceitável para um chip, etc. Esta atividade, considerada crucial no projeto de MPSoCs (*multiprocessor system-on-chip*), aplica-se a vários contextos de engenharia, como prototipação rápida, otimização, e integração de sistemas [3, 4]. Portanto, não podemos descartar a importância das técnicas, modelos e ferramentas para exploração de espaço de projeto para o desenvolvimento de circuitos integrados de alta complexidade e sistemas embarcados.

Segundo Silva Filho [5], devido a crescente importância e custos associado aos semicondutores, a ausência de participação na produção de propriedade intelectual e componentes de microeletrônicos é um ônus negativo para o cenário industrial nacional. O governo brasileiro tem empregado esforços para fomentar o desenvolvimento do setor microeletrônicos, como por exemplo o estabelecimento de **CTs** (Centros de Treinamento) e **DHs** (*Design Houses*) desde 2005, e a fundação do CEITEC (Centro Nacional de Tecnologia Eletrônica Avançada), a primeira fábrica de *chips* da América Latina, em 2008, iniciativas integrantes do programa CI Brasil. Apesar disso, o cenário industrial brasileiro de circuitos integrados de alta complexidade ainda não está consoli-

dado. Foi apenas em 2018 que o primeiro microcontrolador considerado como tecnologia 100% brasileira (**ZR16S08**) foi oficialmente reconhecido ¹. Apesar disso, o setor tem um grande potencial de impacto econômico. Segundo estimativas apresentadas por Rogério Nunes, presidente da **ABISEMI** (Associação Brasileira da Indústria de Semicondutores) o mercado consumidor de semicondutores movimentou em 2022 cerca de 11 bilhões de dólares, e 8% dessa demanda teria sido suprida por fabricantes nacionais ². Sendo assim, a capacitação de futuros profissionais que possam atuar de forma efetiva em atividades de integração de larga escala se torna uma preocupação estratégica para o país.

Dado esse contexto, ficam claros a importância da exploração de espaço de projeto em diferentes ramos da indústria e os motivos para a ACM (*Association for Computing Machinery*) ter recomendado a inclusão do tópico no currículo obrigatório dos cursos de Engenharia de Computação e no currículo eletivo dos cursos de Ciência da Computação ³.

Para compreender como esse tópico era abordado nos programas de graduação nas instituições de ensino brasileiras, o autor deste trabalho realizou uma pesquisa nos projetos pedagógicos de 67 cursos de graduação em Ciência da Computação (CC) e 22 cursos de graduação Engenharia da Computação (EC), totalizando 89 cursos. Dentre esses cursos, 81 são oferecidos por instituições públicas (sobretudo universidades federais e estaduais), 8 são oferecidas por instituições privadas. Utilizando os comandos **grep** ⁴ e **pdfgrep** ⁵ e a ferramenta **AstroGrep** ⁶, uma busca por termos chave foi realizada sobre o texto dos projetos pedagógicos dos cursos supracitados. Essa busca foi acompanhada também de uma análise manual dos textos, de forma a expandir o vocabulário de termos e evitar falsos negativos. As buscas com expressão regular ignoraram acentuação, e não diferenciaram entre letras maiúsculas e minúsculas. Combinações parciais e traduções dos termos para o inglês também foram consideradas. A Tabela 1.1 apresenta para cada tópico a contagem de cursos em cujo projeto pedagógico foram encontradas ocorrências dos termos chave relacionados por tópico.

Esses resultados indicam que o tópico de exploração de espaço de projeto não é corriqueiro nos currículos desses programas de graduação quando avalia-se o contexto das instituições de ensino superior brasileiras. Possivel-

¹<https://www.ufsm.br/2018/03/01/desenvolvido-na-ufsm-primeiro-microcontrolador-brasileiro-e-reconhecido-oficialmente>

²<https://www.abiseми.org.br/abiseми/noticia/133/industria-de-semicondutores-devera-encerrar-2021-com-faturamento-recorde-no-pais>

³<https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf>

⁴<https://www.gnu.org/software/grep/manual/grep.html>

⁵<https://pdfgrep.org/>

⁶<http://astrogrep.sourceforge.net/>

Tabela 1.1: Total de cursos de Engenharia (CE) e Ciência (CC) da Computação que contém os termos chave no projeto pedagógico. Foram encontradas poucas ocorrências relacionadas a DSE.

Tópico	Termos Chave	CE	CS	Total
Arquitetura e Organização de Computadores	Arquitetura de Computador(es); Organização de Computador(es); Computer(s) Architecture. Computer(s) Organization.	22	53	75
Circuitos	Circuito(s); Circuito(s) Integrado(s); Integrated Circuit(s); Circuit(s).	22	42	64
Projeto de Circuitos	Projeto de Circuito(s); Desenho de Circuito(s); Projeto de Circuito(s) Integrado(s); Desenho de Circuito(s) Integrado(s); Integrated Circuit(s) Design; Circuit(s) Design; Design of Integrated Circuit(s); Design of Circuit(s).	9	4	13
DSE	EEP; Exploração de Espaço de Projeto; Espaço de Projeto; Automação de Projeto(s); DSE; design space exploration; design automation; VLSI design automation.	0	1	1

mente, o tópico deveria ser abordado mais frequentemente, e de forma mais aprofundada, nos programas de graduação em Ciência da Computação e Engenharia da Computação no Brasil.

Existem várias maneiras de introduzir alunos de graduação a esse tipo de conteúdo, e o uso de simuladores e outros tipos de software como ferramenta de ensino para permitir que os estudantes adquiram experiência prática com tópicos de Organização e Arquitetura de Computadores não é inovador.

Entretanto, Nikolic et al. [6] apresentaram uma análise demonstrando que o processo de avaliação para seleção de uma ferramenta apropriada para o ensino desses tópicos não é trivial.

Se abordarmos práticas modernas de exploração de espaço de projeto, que também incluem exploração automática por meio de técnicas de inteligência artificial [7], algumas ferramentas podem exigir um nível técnico além do que pode ser esperado para estudantes de graduação, tornando a tarefa de selecionar uma ferramenta para ensino um pouco mais complicada. Isto consiste em um obstáculo à aspiração de difundir o tópico de exploração de espaço de projeto em currículos de graduação no Brasil.

Uma das principais motivações para a realização deste trabalho foi a consideração da hipótese de que a disponibilização de uma ferramenta aberta e gratuita para exploração de espaço de projeto que possa ser utilizada para aprendizado incentivaria um aumento na frequência de abordagem desse tópico a nível de graduação no Brasil.

Ao mesmo tempo, questionávamos se o código-fonte e arquitetura de software da ferramenta estavam maduros o suficiente para disponibilizá-la em um repositório público, tornando-a assim um software de código aberto. Portanto, estabelecemos alguns objetivos a serem alcançados por este trabalho:

- Automatizar a configuração de ambiente e instalação da ferramenta MultiExplorer.
- Amadurecer a arquitetura de software e o código-fonte da ferramenta.
- Implantar uma interface gráfica para ferramenta.
- Avaliar a ferramenta como recurso educacional.

Hipóteses adicionais foram consideradas durante a concepção e realização do estudo de caso:

- A medida de aprendizagem de um aluno sobre exploração de espaço de projeto realizada através de um questionário teórico (primeira atividade) estaria correlacionada com a medida de aprendizagem de um aluno após uma atividade prática com a ferramenta MultiExplorer (segunda atividade)?
- A medida de aprendizagem de um aluno sobre exploração de espaço de projeto realizada através de um questionário teórico (primeira atividade) correlacionada com as medidas de Facilidade de Uso Percebida (**PEOU**) ou de Utilidade Percebida (**PU**) (medidas sobre a experiência de usuário)?
- Existe diferença significativa entre a medida de aprendizagem do aluno antes e depois da atividade prática com a ferramenta MultiExplorer?

Com essas questões em mente, apresentamos por meio desta dissertação, os esforços empregados com o intuito de disponibilizar a ferramenta MultiExplorer [8] como um recurso educacional aberto para aprendizado do tópico de Exploração de Espaço de Projeto. Esses esforços podem ser organizados em três etapas:

- Concepção: conceber uma experiência de usuário acessível e planejar sua implantação.
- Desenvolvimento: Construir e implantar os diversos artefatos de configuração e software necessários.

- Estudo de Caso: Avaliar a ferramenta através de um experimento de aprendizado sobre o tópico de Exploração de Espaço de Projeto.

O primeiro passo para automação da configuração e instalação da ferramenta foi uma análise das pré-condições de uso da ferramenta MultiExplorer. Uma vez com as pré-condições bem estabelecidas, foi possível utilizar o software de containerização **Docker**, de forma a estabelecer um ambiente homogêneo pré-configurado. Isso combinado com um script de construção elaborado em torno do comando unix **Make**, permitiria uma configuração e instalação de forma célere. Mais detalhes sobre esse processo estão na Seção 4.1.

Em seguida utilizamos a metodologia *Design Sprint* para testar idéias e obter um protótipo da interface gráfica de usuário. Uma exposição sobre a aplicação específica da metodologia pode ser encontrada na Seção 4.2.

Seguiu-se um cuidadoso trabalho de concepção arquitetural e codificação, detalhado na Seção 4.3, valendo-se de padrões de projeto pré-existentes e bem documentados.

Ao mesmo tempo, foi concebido um minicurso sobre “Exploração de Espaço de Projeto na Era do *Dark Silicon*”, onde a ferramenta seria utilizada de forma experimental (estudo de caso). Assim que a ferramenta estava munida de uma versão estável da interface gráfica, duas edições do minicurso foram hospedadas pela Faculdade de Computação (Facom) da **UFMS**, e uma análise quantitativa dos dados obtidos foi realizada posteriormente. Esse trabalho foi registrado de maneira mais aprofundada no Capítulo 5.

Dentre os resultados alcançados, podemos destacar:

- A arquitetura de software da ferramenta MultiExplorer amadureceu consideravelmente durante a execução do projeto.
- O código-fonte da ferramenta se tornou aberto.
- O processo de configuração de ambiente e instalação da ferramenta foi em maior parte automatizado, facilitando a adoção.
- Uma interface gráfica para a ferramenta foi implantada, facilitando o uso por alunos de graduação.
- As medidas de aprendizagem, Facilidade de Uso Percebida (**PEOU**) e Utilidade Percebida (**PU**) que foram recolhidas durante o estudo de caso indicam um prospecto positivo a respeito do uso da ferramenta para aprendizado a nível de graduação.
- Não foi possível estabelecer os efeitos descritos nas três hipóteses adicionais consideradas durante o estudo de caso.

Esta dissertação está organizada da seguinte maneira: o Capítulo 2 apresenta o referencial teórico que embasou o trabalho, e o Capítulo 3 apresenta alguns trabalhos relacionados. O Capítulo 4 apresenta com detalhes as atividades e processos empregados durante o projeto de pesquisa, e o Capítulo 5 apresenta de forma mais aprofundada a estrutura, procedimento e resultados do estudo de caso. No Capítulo 6 sintetizamos os resultados da pesquisa, encerramos a discussão, e apresentamos direções de possíveis trabalhos futuros.

Referencial Teórico

Neste capítulo, é apresentado o referencial teórico que embasa a pesquisa. A Seção 2.1 aborda o tópico de exploração do espaço de projetos, a Seção 2.2 descreve o problema conhecido como dark silicon, e a Seção 2.3 descreve a ferramenta MultiExplorer que possui uma solução para mitigar dark silicon durante a exploração do espaço de projetos de sistemas multiprocessadores. A Seção 2.4 aborda a metodologia *Google Design Sprint*, que foi utilizada para a prototipação da interface gráfica para a ferramenta MultiExplorer, e a Seção 2.5 apresenta o Modelo de Aceitação de Tecnologia, ou *Technology Acceptance Model (TAM)*, que foi utilizado para avaliar a experiência de usuário dos alunos com a ferramenta.

2.1 Exploração do Espaço de Projetos

A crescente complexidade dos Chips Multiprocessadores (CMPs) faz com que seja necessária a exploração de diversos parâmetros arquiteturais durante o projeto para se alcançar um equilíbrio entre os diversos indicadores de desempenho e eficiência energética (largura de banda, atrasos, potência, área, etc.) [9].

Essa atividade de busca pela melhor solução arquitetural é frequentemente intitulada **Exploração do Espaço de Projeto**, ou *Design Space Exploration (DSE)*. O tamanho do espaço a ser explorado é proporcional ao produto das cardinalidades dos conjuntos de valores possíveis de cada variável envolvida [1], e por causa dessa magnitude, uma abordagem manual se torna tediosa, propensa a falhas e inadequada para aplicação em larga escala [10].

Segundo Zaccaria et al. [9], esse cenário leva à necessidade de metodolo-

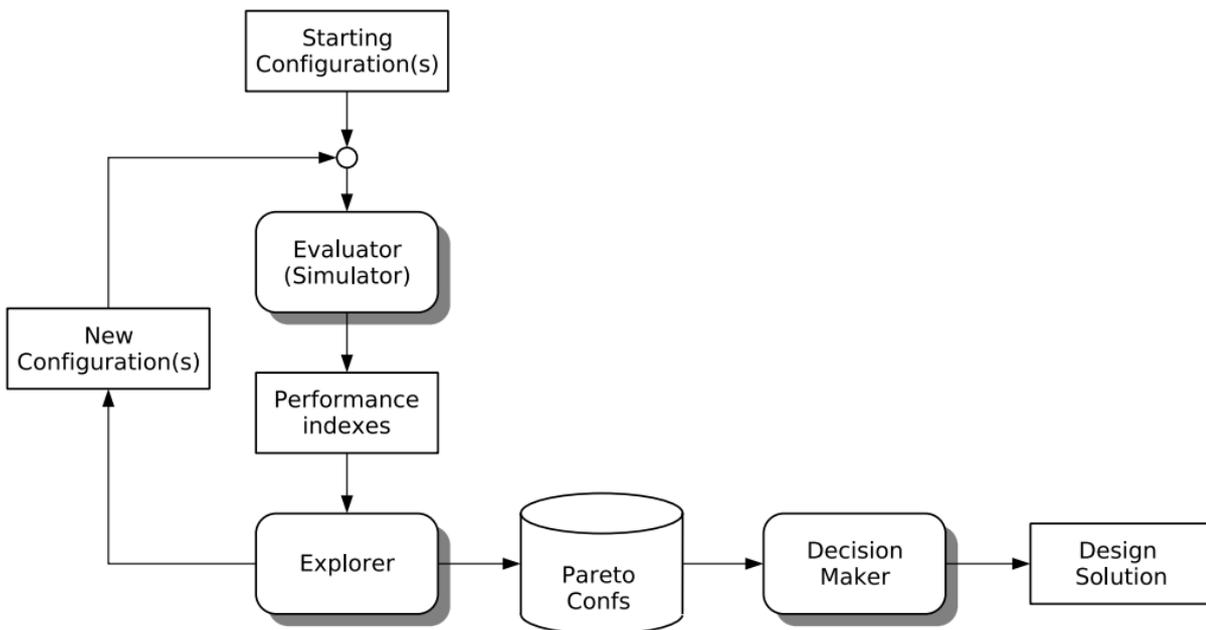


Figura 2.1: Fluxo DSE genérico. Extraído de [1]

gias para automação de DSE, e frequentemente o problema da exploração de espaço de projeto é modelado como um problema de otimização multiobjetivo.

TODO Breve introdução sobre otimização multiobjetivo e análise de Pareto []

Muitas abordagens de solução amparam-se sobre duas componentes: uma responsável pela exploração de espaço de projeto (*Explorer*), frequentemente por meio da geração de novas instâncias de configuração dos parâmetros arquiteturas, e uma responsável pela avaliação das soluções propostas (*Evaluator*). Essas duas componentes interagem de forma cíclica até a convergência em uma solução na Fronteira Pareto [1], conforme indicado no diagrama da Figura 2.1.

Algoritmos de otimização multiobjetivo são frequentemente a escolha para cumprir o papel de *Explorer*. Destaque especial para os algoritmos evolutivos multiobjetivo, ou *Multiobjective Evolutionary Algorithms* (MOEAs), que apresentaram bons resultados experimentais [11].

A avaliação das soluções frequentemente envolve simulação, porém na maioria dos casos o tempo de simulação multiplicado pelo número de simulações necessárias durante o processo de exploração torna essa estratégia custosa demais para ser aplicada. Por isso, é comum que dados de simulação sejam coletados para treinar um *Evaluator* baseado em predição, de precisão aproximada, evitando a execução de muitas simulações [11].

2.2 Dark Silicon

A Lei de Moore [12] e a escalabilidade MOSFET proposta por Dennard [13] guiaram a indústria de desenvolvimento de hardware por 40 anos. Porém, diferentemente do que foi previsto por Dennard, a cada geração de novos transistores fabricados com tecnologia abaixo de 90nm, há uma perda de utilização de área, conhecida como *utilization wall* (barreira de utilização), a qual impede que o *chip* faça uso de sua frequência máxima. A origem desta barreira vem do vazamento da corrente entre os componentes, o que causa a diminuição no limiar da tensão mínima (V_d) de operação dos transistores, causando um aumento exponencial na utilização de potência e a dissipação de calor [14].

A Tabela 2.1 apresenta com mais detalhes como ocorre a barreira de utilização, considerando processo de fabricação a partir de 90nm. Considere S como o fator de redução de escala dos transistores entre duas gerações. Na escala pós-Dennard, não foi possível obter redução no limiar de tensão sem aumentar exponencialmente o vazamento de corrente, mas de acordo com o IRDS (*International Roadmap for Devices and Systems*) [15], só é possível explorar a capacidade computacional de um sistema, se pudermos garantir ganhos equivalentes em eficiência energética [16].

Tabela 2.1: Dennard x Pós-Dennard.

Propriedade do Transistor	Dennard	Pós-Dennard
Quantidade	S^2	S^2
Frequência	S	S
Capacitância	1/S	1/S
V_d^2	$1/S^2$	1
Energia = QFCV ²	1	S^2
Utilização = 1/Energia	1	$1/S^2$

Devido à quebra da escala de Dennard, a área de um chip que pode funcionar com sua frequência máxima vem caindo exponencialmente a cada geração, forçando projetos atuais a inutilizar ou diminuir a frequência em uma parte considerável do chip. A porcentagem do chip que deve ser desligada é denominada *dark silicon*. Há uma estimativa de que o *dark silicon* possa ocupar 75%-85% a mais de espaço em chips com tecnologia de $8nm$, quando comparado a chips de $45nm$ [16, 17]. Vale a pena dizer que o *dark silicon* não significa silício inútil ou não utilizado, significa que é apenas um silício que não é usado o tempo todo, ou em frequência de operação máxima [18].

A mitigação de *dark silicon* tornou-se evidentemente necessária e, então, surgiram algumas abordagens para o manutenção da escala de integração [18]. A primeira abordagem refere-se à redução do tamanho dos chips, a qual foi considerada uma opção fraca pois, futuramente, levaria a *chips*

com baixa lucratividade. Em uma abordagem mais agressiva, o autor supõe o desenvolvimento de outro componente capaz de substituir os transistores ou a recriação do transistor com material diferente do *MOSFET* (*Metal Oxide Semiconductor Field Effect Transistor*). A terceira abordagem consiste em especializar o processamento, tentando ocupar a área de *dark silicon* com alguma unidade funcional de propósito específico que possa executar tarefas a uma frequência reduzida, tornando evidente a dependência entre o hardware heterogêneo e o software que realizará o escalonamento de tarefas.

A última abordagem, descrita por Taylor [18] consiste na utilização adequada da área afetada pelo *dark silicon* e há diversas técnicas baseadas nessa abordagem disponíveis em processadores de última geração. Propõe-se utilizar a área em *dark silicon* em uma frequência de operação mais baixa, a fim de obedecer ao orçamento de potência. A seguir, alguns exemplos de técnicas fundamentadas nesta abordagem:

- utilizarmos processadores que limitam de tensão limiar *Near-Threshold Voltage*(NTV) [19];
- aumentar a área de cache [20];
- empregar arquiteturas baseadas em *arrays* reconfiguráveis de granularidade grossa - *Coarse-Grained Reconfigurable Arrays* (CGRA) [21];
- utilizar técnicas de *sprinting* [22];
- *Turbo boost* [23], um impulso computacional de curta duração, já empregado em processadores do mercado atual [24].
- heterogeneidade multifacetada (funcional, potência-performance, tecnológica, etc.) [25]
- computação aproximada [25, 26]
- cooperação CPU/GPU [27, 28]
- aceleração com uso de FPGAs [29]

Durante o desenvolvimento de um novo projeto, há necessidade de estimar a quantidade de *dark silicon* em um *chip* e, em seguida, buscar técnicas para reduzir ou para mitigar o *dark silicon* encontrado. Existem algumas abordagens para estimativas de *dark silicon* em um sistema, sendo uma destas proposta por Henkel et al. [30], em que relacionam-se os valores de temperatura e dissipação de calor fornecido pela arquitetura com a escala de voltagem e frequência.

A estimativa de área de *dark silicon* proposta por Silva et al. [16], e validada na ferramenta de DSE MultiExplorer, é baseada na densidade de potência do chip antes e depois da evolução tecnológica (os quais foram denominados, nesse contexto, de circuito base e circuito atual, respectivamente).

A densidade de potência (dp) é calculada dividindo-se a potência de pico pela área do chip (a). Em seguida, calcula-se a variação de densidade de potência como a diferença entre a densidade de potência atual (dp_{atual}) e a densidade de potência base (dp_{base}), resultando na potência excedente a cada mm^2 (Equação 2.1):

$$\Delta_{dp} = dp_{atual} - dp_{base}. \quad (2.1)$$

Caso $\Delta_{dp} \leq 0$, então não há *dark silicon* no projeto. Caso contrário, calcula-se o total de potência excedida (t_{pe}):

$$t_{pe} = \begin{cases} \Delta_{dp} * a & \text{se } \Delta_{dp} > 0; \\ 0 & \text{caso contrário.} \end{cases}$$

Por fim, a área estimada de *dark silicon* do projeto (A_{ds}) é calculada com base em valores obtidos de um circuito de referência cuja densidade de potência seja considerável em relação à densidade de potência base, como formulado na Equação 2.2:

$$a_{ds} = \frac{t_{pe}}{p_c} * a_c, \quad (2.2)$$

em que

- a_{ds} : Área de *dark silicon*;
- t_{pe} : Total de potência excedida;
- p_c : Potência do circuito de referência;
- a_c : Área do circuito de referência.

Silva et al. [16] realizaram comparações desta metodologia com estimativas mais conservadoras de *dark silicon*. Os seus experimentos demonstraram que a área de *dark silicon* é quase 30% menor do que o previsto na linha pós-Dennard, sendo mais próximo do que observamos nos chips atuais.

2.3 MultiExplorer

O MultiExplorer [31] é um *framework open source* para exploração de espaço de projeto, concebido e implantado inicialmente em 2015. Desde então, a ferramenta vem sendo mantida e incrementada em projetos de pesquisa do

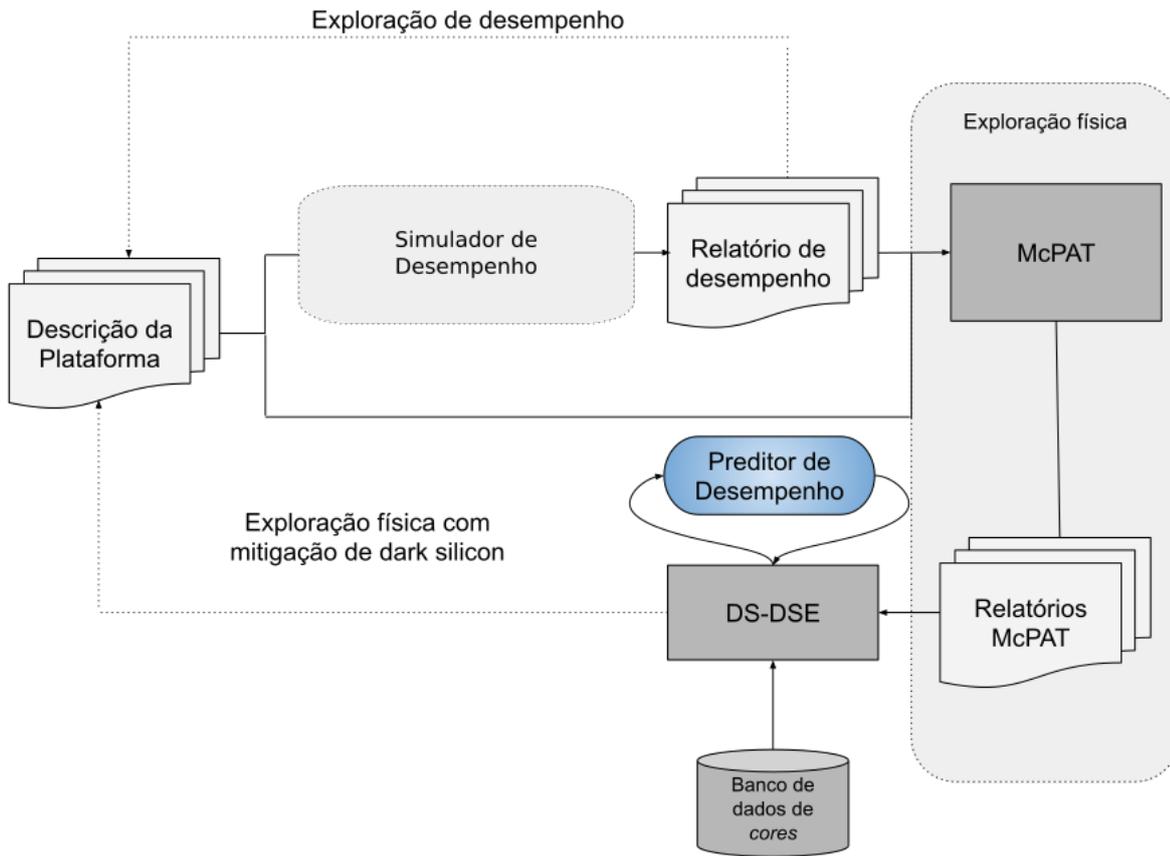


Figura 2.2: Representação gráfica do pipeline da ferramenta MultiExplorer.

Laboratório de Sistemas Computacionais de Alto Desempenho da UFMS (LS-CAD).

Na Figura 2.2 é possível visualizar todo o fluxo de execução percorrido pela ferramenta. O MultiExplorer adota como entrada um arquivo `json` que contém as informações sobre um sistema computacional e as restrições de área e potência impostas. Na etapa de exploração de desempenho é realizada a descrição da plataforma, e a seleção do simulador de desempenho juntamente com a aplicação e o benchmark. Atualmente, pode-se escolher entre o simulador Sniper [32] para sistemas multicore e o simulador GPGPU-Sim [33] para sistemas baseados em GPUs. Os benchmarks que podem ser utilizados são o SPLASH2 [34] e o PARSEC [35] para sistemas multicore e programas do CUDALibrary [36] e Rodinia [37] para sistemas baseados em GPUs.

Após simulação, a ferramenta apresenta informações de desempenho como quantidade de ciclos, tempo de execução, IPC, desempenho das caches e muitos parâmetros físicos de saída do projeto. Logo após o resultado do relatório de desempenho essa saída é adaptada para servir como entrada do processo de exploração física que é efetuado pela ferramenta McPAT [38], que estima parâmetros como potência e área dos componentes do Hardware e é nessa etapa que é calculado o tamanho da área do chip que está em *dark silicon*

a partir de parâmetros como a densidade de potência, conforme metodologia proposta por Santos et al. [16] e apresentada na Seção 2.2.

Posteriormente ao resultado da exploração física ocorre o processo de exploração do espaço de projeto que tem como entrada os resultados do relatório de desempenho, o relatório do McPAT, e o banco de dados de cores. A exploração pode ser realizada de maneira exata por um algoritmo de força-bruta ou por um algoritmo genético denominado NSGA-II que gera novos indivíduos e descartava aqueles que estavam fora dos limitantes impostos pelo usuário. Em ambas as abordagens, é necessário estimar o desempenho de cada solução alternativa heterogênea. Na versão original da ferramenta, o desempenho do sistema heterogêneo era calculado como a soma dos desempenhos dos núcleos da plataforma [39].

Visto que a estimativa de desempenho do sistema heterogêneo era bastante otimista, considerando que o desempenho da plataforma escalava linearmente com a quantidade de núcleos, foi desenvolvido e acoplado à ferramenta um preditor de desempenho de sistemas heterogêneos [40] com base em técnicas de aprendizado de máquina, tornando as estimativas de desempenho mais justas e com baixo custo computacional. O algoritmo de aprendizado de máquina escolhido foi o SVR com kernel RBF e com isso o modelo obteve acurácia de 97%, erro médio de 20,34% para sistemas multicore e o modelo baseado em árvores de decisão para modelos de GPUs, com erro médio 4,97% e desvio padrão de 0,90. Vários outros modelos de preditores de desempenho foram avaliados para sistemas multicore por Santos et al.[40] e para GPUs por Sohonata et al. [41]. Os modelos citados acima foram os mais adequados para serem integrados ao MultiExplorer, por terem alta vazão (quantidade de predições por unidade de tempo) e baixo erro.

O código do MultiExplorer é escrito na linguagem Python (2.7), e a execução exige um ambiente unix apropriadamente configurado. O código-fonte e instruções para configuração e utilização da ferramenta estão disponíveis em <https://github.com/lscad-facom-ufms/multiexplorer>.

A ferramenta está sendo utilizada em pesquisas por alunos de graduação, mestrado e doutorado da Faculdade de Computação da Universidade Federal de Mato Grosso do Sul. Há uma frente de trabalho atuando em Computação Aproximada, com resultados preliminares já publicados por Catelan [42] e um grupo de trabalho atuando em Computação em Nuvem [43]. Embora o código da ferramenta tenha sido recentemente publicado no github, não houve antes deste trabalho um esforço organizado para oferecer ao público-alvo uma boa experiência de uso nas fases de instalação, configuração e uso da ferramenta, bem como na análise e interpretação dos dados oferecidos como saída. Além disso, é bastante promissor e inovador tornar a ferramenta um apoio

adequado ao ensino de tópicos relacionados à exploração arquitetural, eficiência energética, análise de parâmetros físicos, desempenho e dark silicon em projetos multicore homogêneos ou heterogêneos em disciplinas dos cursos de computação.

2.4 *Design Sprint*

A *Design Sprint* é uma metodologia, um processo com duração pré-fixada de cinco dias que tem o propósito de permitir a obtenção de respostas para perguntas cruciais a partir da prototipação e do teste de ideias junto a membros de um público alvo. A técnica se tornou bastante popular nas áreas de estratégia de negócio, ciência comportamental, projeto de software, etc. [2]. Desenvolvida durante atividades de prospecção de negócios e fomento de novas *startups* da *Google Ventures* ¹ a metodologia foi condensada no formato de um processo passo-a-passo para permitir sua utilização por times de diferentes perfis que tenham como objetivo o desenvolvimento de um produto, mesmo que não necessariamente um produto de software.

O processo é organizado em cinco fases, que teoricamente deveriam ser realizadas cada uma num único dia, inclusive originalmente essas fases recebem os nomes dos cinco dias úteis da semana em inglês (*Monday, Tuesday, Wednesday, Thursday, Friday*). Entretanto, o processo também prevê que antes de iniciar a *sprint* alguns pré-requisitos devem ser cumpridos.

Podemos enxergar o preparo para a *Design Sprint* como uma fase em particular, que é referida no livro [2] como “*Set the Stage*”. Para possibilitar a execução bem sucedida do processo é preciso se atentar aos três principais pré-requisitos:

- **Desafio** (ou **Challenge**).
- **Time** (**Team**).
- **Tempo e Espaço** (**Time and Space**).

O **Desafio** se refere ao problema a ser solucionado, ou ao aspecto do produto que se deseja definir. Por definição, o método *Design Sprint* tem como alvo a chamada **superfície** do problema, i.e., o ponto de encontro entre o produto e seu público alvo.

O **Time** pode ser composto exclusivamente por membros da organização que irá desenvolver o produto final, ou também pode incluir especialistas em *Design Sprint*. A presença desses especialistas é considerada totalmente opcional pelos autores do livro, mas exige a equipe original de um treinamento

¹<https://www.gv.com/>

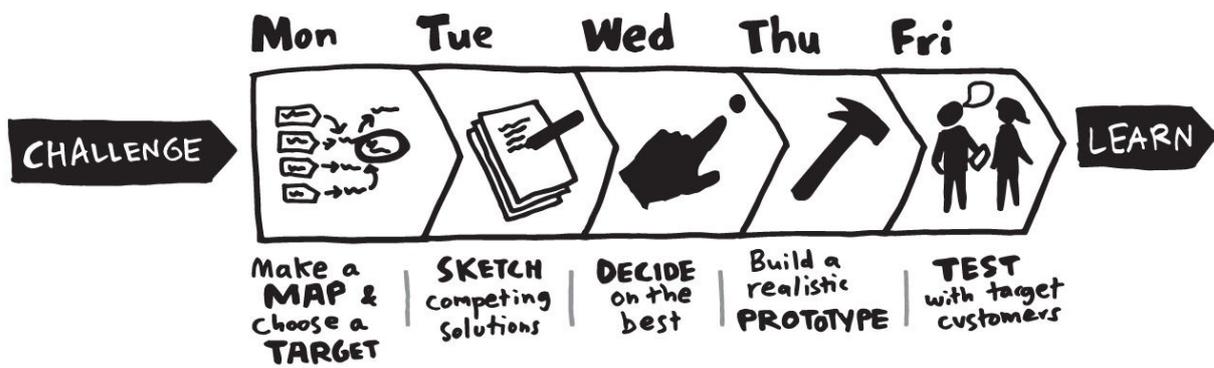


Figura 2.3: Representação gráfica do processo de *Design Sprint*. Imagem extraída do livro “*Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days*” [2].

específico sobre a metodologia. Geralmente é considerado suficiente que um único membro conheça e compreenda todas as etapas do processo, assumindo o papel de **Facilitador**. Também há outros papéis chave no processo, como o de **Decisor**. O processo recomenda o recrutamento de um ou dois decisores para o time da *Design Sprint*. Os decisores, como o nome do papel sugere, são aqueles responsáveis pelas grandes decisões. O processo prevê que eles podem participar parcialmente do processo, comparecendo em algumas atividades chave, ou delegando seu papel para um representante. O processo recomenda um time de sete ou menos membros, e que esse time seja tão diverso quanto possível, recrutando especialistas de diferentes áreas da organização.

O **Tempo e Espaço** se referem a disponibilidade ao time de fatias de tempo pré-definidas e de espaço dedicado para realização das reuniões e demais atividades da *Design Sprint*. O processo não tem requisitos específicos para o espaço, mas é recomendado que todos os membros estejam no mesmo local se possível, e que haja recursos para realizar as atividades, como material para fazer desenhos, diagramas e anotações. O processo recomenda que durante os primeiros quatro dias as atividades sejam realizadas entre 10 horas da manhã e cinco horas da tarde, com um intervalo de uma hora para o almoço entre o turno matutino e o turno vespertino, e um pequeno intervalo de descanso no meio de cada turno. No último dia a agenda é literalmente diferente, com os preparativos para o teste do protótipo se iniciando uma hora mais cedo, às nove horas da manhã. A Figura 2.3 representa todo o processo *Design Sprint* de forma resumida.

O primeiro dia de atividades da *sprint* é focado em planejamento. O processo recomenda uma estratégia de desenvolvimento de soluções chamada “começar pelo fim”. Basicamente o propósito das primeiras atividades é definir um **objetivo de longo prazo**, produzir um **mapa** do fluxo de interação entre o público alvo e o produto, **questionar** os especialistas envolvidos para obter informações sobre o problema, os processos, o produto, e como atingir

os objetivos e definir o **alvo** da *sprint*, o objetivo específico derivado do **objetivo de longo prazo** que a *sprint* tentará atingir.

O segundo dia da *sprint* é focado em soluções. Idéias pré-existentes para a solução são **revisadas, recombina**das e **melhoradas**. Os **esboços** produzidos nesse dia são artefatos importantes para dar continuidade ao trabalho nas etapas seguintes.

No terceiro dia, as atividades são direcionadas a **decisão**. As soluções serão sistematicamente catalogadas, apresentadas, explicadas, criticadas, e por fim as decisões serão tomadas. Dependendo do número de soluções eleitas, e suas características, será possível combiná-las num único protótipo, ou passá-las por uma última rodada de análise e decisão para ou comprometer a equipe a uma proposta única ou, em alguns casos, a proposição de múltiplos protótipos. No último período da terceira fase, um **roteiro**, ou **storyboard**, deverá ser composto. A principal utilidade desse artefato é representar de forma não ambígua o fluxo de interação entre um cliente/testador e o produto, ajudando a estabelecer o cenário de uso do protótipo.

A quarta fase é voltada para a construção do protótipo. O processo recomenda que o protótipo seja apenas uma fachada, ou **facade**, em outras palavras, deve-se prototipar o produto da maneira mais simples e fácil quanto possível, apenas o mínimo para que seja possível simular a interação real a nível de teste. O processo recomenda que as ferramentas adequadas sejam selecionadas pelo time. Dividir a construção em partes que possam ser construídas por membros diferentes acelera a prototipação. Na última fase da *sprint*, é hora de executar o teste: membros do público alvo irão interagir com o protótipo. Quanto ao número de testadores, o processo define que cinco é o número ideal (provavelmente uma conclusão emprestada modelo de Nielsen [44]). Os testes são acompanhados em primeira mão por membros do time, que também irão entrevistar os participantes após a experiência.

O objetivo da *sprint* não é que o protótipo seja necessariamente bem recebido pelos testadores, o verdadeiro objetivo é o aprendizado. O teste com o protótipo permite a identificação de problemas ou defeitos na solução e a descoberta de novas necessidades dos usuários.

O aplicação da *Design Sprint* no desenvolvimento do MultiExplorer forneceu informações valiosas. Mais detalhes de como a metodologia foi contextualizada para a realidade do LSCAD e como ela foi de fato utilizada no desenvolvimento podem ser encontrados na Seção 4.2 do Capítulo 4.

2.5 Technology Acceptance Model

O *Technology Acceptance Model* (TAM), ou Modelo de Aceitação de Tecnologia, foi proposto por Davis [45] em 1986. Derivado da Teoria de Ação Racional (TRA) e da Teoria de Comportamento Planejado (TBP), o TAM se tornou uma ferramenta proeminente para explicar o comportamento de usuários em relação a tecnologia [46].

O modelo propõe que a aceitação de uma tecnologia por um usuário está fortemente correlacionada com duas variáveis:

- Utilidade percebida, ou *Perceived Usefulness* (**PU**)
- Facilidade de uso percebida, ou *Perceived Ease of Use* (**PEOU**)

Segundo Younghwa et al. [47], **PU** pode ser vista tanto como uma variável independente (forte influência sobre a aceitação) quanto como uma variável de resposta (influenciada por **PEOU**). As evidências apontam para um significativo relacionamento entre **PEOU** e **PU**, mas ao mesmo tempo, apontam que **PU** fornece uma melhor predição da aceitação de uma tecnologia do que **PEOU**, e em muitos casos é o fator determinante. Por meio de uma extensa meta-análise, William e Jun [48] concluíram que “as medições realizadas por meio do TAM são altamente confiáveis e podem ser utilizadas em diferentes contexto”.

O modelo não é apenas popular, como também conta com extensões diversas, não tendo simplesmente estagnado em sua proposta inicial: pelo contrário, desde sua inceptação tem sido investigado, testado e incrementado por uma série de trabalhos de pesquisa. Portanto, não é surpresa que o TAM tenha sido utilizado também no contexto de tecnologias voltadas ao ensino. Segundo uma revisão sistemática por Granić e Marangunić [49], é possível inferir do corpo de conhecimento presente que o TAM é uma ferramenta útil para avaliar a aceitação de uma tecnologia para o ensino mesmo quando outros fatores não relacionados a tecnologia também possam ser relevantes.

Mais detalhes de como o TAM foi utilizado neste trabalho de pesquisa podem ser encontradas no Capítulo 5, na Seção 5.1.5.

Trabalhos Relacionados

Há uma variedade de ferramentas de simulação de sistemas computacionais desenvolvidas na última década, tanto para prover infraestrutura de experimentação e validação em pesquisas quanto para apoio ao ensino de arquitetura de computadores, compiladores, sistemas embarcados, projetos de hardware, entre outros tópicos presentes em cursos de computação. Classificar tais ferramentas é uma tarefa difícil, pois há uma grande quantidade de critérios, os quais dependem diretamente do objetivo da ferramenta.

Por exemplo, um simulador de um processador para atender ao objetivo de ensino-aprendizagem em nível de graduação pode ser apenas funcional, com o hardware descrito em alto nível de abstração e sem muitos detalhes microarquiteturais; porém, precisa ter uma interface amigável, ser o mais independente de plataforma possível e adaptado para colocar em prática conceitos claros e fundamentais da área que deseja enfatizar. Já um simulador útil para projetos de pesquisa precisa ter mais alto nível de detalhamento na descrição do hardware simulado para garantir precisão dos resultados da simulação e cobrir uma variedade muito maior de funcionalidades para atender a demandas de potenciais pesquisas.

O objetivo desta seção é apresentar algumas ferramentas de simulação para apoio ao ensino e pesquisa em áreas relativas à Arquitetura e Organização de Computadores (AOC) e exploração arquitetural que se destacaram na revisão bibliográfica do tema. A síntese da comparação entre as ferramentas pode ser encontrada na Tabela 3.1, na Tabela 3.2 e na Tabela 3.3.

O **PerfPred** [50] é uma ferramenta *web* para predição de performance de sistemas de computação. A entrada para cada execução consiste em um conjunto parâmetros de hardware e software, como número de unidades lógicas

aritméticas, esquema de predição de desvio, *benchmarks* a serem executados, entre outros. Apenas um dos parâmetros deve ser selecionado como variável e um intervalo de valores deve ser informado (início e fim). Os demais parâmetros são definidos com um único valor fixo. A predição de performance é realizada por meio de uma rede neural treinada a partir de uma extensa base de dados oriundos de simulações. Os resultados podem ser apresentados na forma de um gráfico, ou em uma forma textual que permite que os dados sejam exportados para outros programas, como o *Excel*, por exemplo. Dados de várias execuções podem ser agrupados num único gráfico. A ferramenta pode ser utilizada para ensinar como configurações de *hardware* e/ou *software* podem afetar o desempenho do sistema. Sua utilização em aulas de Arquitetura de Computadores demonstrou sua eficácia em aprofundar a compreensão dos alunos sobre os tópicos relacionados. Mas ainda assim não é possível utilizá-la para explorar características físicas da arquitetura, como área e potência, nem sistemas heterogêneos (núcleos de processamento diversos entre si).

O **T&D-Bench** (*Teaching and Design Workbench*) é uma ferramenta proposta por Soares & Wagner [51] para o ensino de Arquiteturas de Computadores, e tem como objetivo proporcionar aos estudantes um contato direto com as técnicas contemporâneas de projeto de hardware, como só seria possível com ferramentas CAD profissionais. O **T&D-Bench** permite o projeto de um processador, usando um alto nível de abstração para acelerar o aprendizado. A plataforma projetada pode então ser simulada, programas em linguagem de máquina podem ser executados, e a execução pode ser acompanhada graficamente. Apesar de permitir uma forma manual de DSE, a ferramenta não permite abordar a otimização multiobjetivo de uma plataforma de multiprocessamento, nem explorar de forma mais aprofundada características físicas e de desempenho dos projetos de processador.

Esmeraldo e Lisboa [52] introduziram o **CompSim**, um simulador que proporciona um ambiente visual para o ensino prático de Arquitetura e Organização de Computadores, permitindo programação de baixo nível (código de máquina), cálculo de tempos de acesso, acompanhamento das fases de execução de instruções, visualização da representação dos dados em memória, configuração da plataforma virtual, entre outras funcionalidades. Apesar do grande valor didático dessa ferramenta no ensino de conceitos básicos dessa disciplina, a plataforma virtual é baseado em uma arquitetura teórica específica criada pelos pesquisadores, a “Mandacaru”, e a gama de configurações da plataforma é pequena, de forma que não há como explorar mais a fundo noções sobre o projeto de sistemas digitais no contexto contemporâneo da indústria.

A ferramenta **MARS** [53] permite a programação e simulação de execução

em tempo real da linguagem de montagem da arquitetura MIPS. Sendo implementada em Java, independe de plataforma, e conta com uma interface gráfica de fácil uso, tendo sido engendrada e largamente utilizada para o ensino de Arquitetura e Organização de Computadores. Conjugando as funcionalidades de edição de código, simulação, depuração, visualização de registradores e memória, etc., com uma interface simples, a ferramenta implementa as partes mais educacionalmente importantes do conjunto de instruções MIPS. A avaliação da ferramenta realizada por estudantes, comparando-a ao SPIM *citespin* indicam que os critérios de usabilidade das ferramentas de ensino são importantes na experiência de aprendizado dos estudantes. Apesar do sucesso da **MARS** como ferramenta de ensino, ela não tem muita aplicabilidade ao se abordar o tema de DSE.

Há também um trabalho apresentado por Duenha e Azevedo [54], propondo uma metodologia para uso do **MPSoCBench** [55] como ferramenta para o ensino e aprendizagem de Arquitetura e Organização de Computadores. A ferramenta em questão é um *framework opensource* para simulação de sistemas multiprocessados englobando plataformas para avaliação e exploração de componentes arquiteturais. A metodologia consiste na aplicação de um conjunto de projetos de experimentos nos quais os alunos exercitam e exploram os conceitos teóricos na prática. Os resultados obtidos sugerem que as atividades experimentais aplicadas proporcionaram o amadurecimento da compreensão dos alunos de aspectos conceituais, aumento do interesse pela área, além de um melhor desempenho na disciplina. Duenha et al. [56] também abordaram o uso do **MPSoCBench** para ensino, focando no uso de experimentos de simulação como método avaliativo, de forma a permitir a exploração prática sobre preditores de desvio, apresentando impactos positivos no processo de aprendizado. Sendo assim, podemos dizer que o **MPSoCBench** foi utilizado com sucesso para o ensino de AOC. Entretanto, a flexibilidade de variação de configurações e componentes da arquitetura, bem como o nível de detalhamento das informações para avaliação dos componentes arquiteturais no **MPSoCBench** ainda é limitado para uma atividade de DSE mais realista.

O **QEMU** [57] é um emulador e virtualizador robusto que utiliza um tradutor dinâmico portátil para emular várias ISA - *Instruction Set Architectures* (x86, PowerPC, ARM, SPARC, RISC-V, Spark, x86, etc.), em diferentes *hosts* (Arm, MIPS, PowerPC, RISC-V, s390x, SPARC, x86). Utiliza compilação dinâmica para atingir velocidades de simulação expressivamente superiores às proporcionadas via interpretação pura. Largamente utilizado na implementação de *frameworks* para simulação, emulação, e como base para implementação de outras ferramentas, o **QEMU** é um projeto *open source*, que vem sendo desenvolvido desde meados de 2005, e ainda recebe contribuições e lança

versões novas frequentemente [58]. Apesar de ser uma ferramenta muito poderosa, não conta com nenhum recurso intrinsecamente pedagógico, e sua complexidade é um obstáculo ao desenvolvimento de atividades de ensino e aprendizado.

Há um simulador *open-source* capaz de simular e emular sistemas inteiros baseados na arquitetura x86, o **MARSSx86** [59]. A ferramenta é *open-source*, capaz de precisão ciclo-a-ciclo, em ordem e fora de ordem, para CPUs *single core* e *multicore* que implementem a arquitetura de conjunto de instruções x86. O simulador é capaz de executar sistemas operacionais sem nenhuma alteração, bem como binários de aplicações x86 também inalterados. Segundo o artigo, não foi encontrada à época uma ferramenta *open source* capaz de igualar as funcionalidades do simulador. Atende às características de uma ferramenta profissional para avaliação de novas sistemas, e já foi utilizada em experimentos de otimização de arquitetura [60, 61, 62], mas apesar de ser gratuita há uma curva de aprendizado um tanto quanto íngreme para domínio das ferramentas associadas (PTLsim e QEMU), além de uma complexidade inerente à tarefa que a ferramenta se propõe, que poderia tornar a aplicação da ferramenta num ambiente de ensino tanto quanto restrita, se não inviável.

Heracles [63] é uma poderosa e versátil ferramenta *open source* para pesquisa e ensino de exploração arquitetural de sistemas multicore e para projeto conjugado de hardware-software (como no caso de MPSoCs), baseada em simulação RTL. Permite a simulação, construção e síntese de sistemas inteiros sobre FPGAs, com mínimo esforço. Conta com uma biblioteca de componentes pré-definidos e também permite a reutilização de blocos de hardware definidos pelo usuário em diferentes projetos. Sua interface gráfica permite configuração e implantação rápida e fácil do sistema proposto. Os módulos de hardware são implantados em Verilog e independem de qual plataforma de FGPA está sendo utilizada. Também conta com um encadeamento de compilação que permite mapear aplicações em C ou C++ nos núcleos de processamento. Esta ferramenta permite uma exploração rápida e altamente precisa do espaço de projeto, e sua interface gráfica facilita a adesão. É um exemplo de ferramenta que auxilia em um processo de DSE manual.

A Tabela 3.1 apresenta uma configuração de funcionalidades entre as ferramentas apresentadas neste capítulo. “Configuração de Arquitetura” se refere a possibilidade de variar parâmetros da arquitetura ou sistema que serão simulados ou emulados. A “Análise de Performance” se refere a presença de indicadores gerados pela ferramenta que permitam medir e comparar a performance entre diferentes configurações. “Parâmetros Físicos” diz respeito a presença de informações também sobre características físicas da arquitetura, como área, potência e consumo de energia, por exemplo. “Programação” se re-

fere a possibilidade de introduzir um programa ou código para ser simulado na arquitetura, seja código de máquina ou código de alto nível, de forma que seria possível inclusive realizar atividades de programação com os alunos.

Tabela 3.1: Comparação de funcionalidades entre as ferramentas.

Ferramenta	Configuração de Arquitetura	Análise de Performance	Parâmetros Físicos	Programação
PerfPred	Sim	Sim	Não	Não
T&D Bench	Sim	Sim	Não	Sim
CompSim	Sim ¹	Sim	Não	Sim
MARS	Sim	Sim	Não	Sim
MPSoCBench	Sim	Sim	Sim	Não
QEMU	Sim ¹	Sim	Não	Sim
MARSSx86	Sim	Sim	Não	Sim
Heracles	Sim	Sim	Não	Sim
MultiExplorer	Sim ¹	Sim	Sim	Não

Na Tabela 3.2 vemos uma comparação adicional a respeito de como as ferramentas permitem abordar os tópicos de **DSE** e *Dark Silicon*. “DSE” se refere a possibilidade de se realizar exploração de espaço de projeto com a ferramenta. “DSE Automático” indica se a ferramenta conta com algum tipo de exploração automática do espaço de projeto. “*Dark Silicon*” indica se a ferramenta realiza algum tipo de estimativa de *dark silicon* das plataformas simuladas ou emuladas.

Também foi realizada uma comparação entre a aplicabilidade das ferramentas nos contextos de ensino, pesquisa e na indústria, bem como se a ferramenta está disponível gratuitamente, ou se é *open source*. Essa comparação pode ser encontrada na Tabela 3.3. A coluna “Ensino” indica se a ferramenta já foi aplicada em contexto de ensino e aprendizado, ou se é considerada apropriada para este fim por seus autores. A coluna “Pesquisa” faz indicação de forma análoga sobre a área de pesquisa, e a coluna “Profissional” sobre o uso da ferramenta por profissionais da indústria. “Download” diz respeito se a ferramenta está disponível para download de graça. “Open Source” indica se a ferramenta é um software de código-aberto.

Pode-se constatar pela revisão bibliográfica apresentada nesse capítulo que apesar de existirem ferramentas para ensino para outros tópicos de Arquitetura e Organização de Computadores, ferramentas para simulação de múltiplas arquiteturas, e outros tipos de ferramentas que podem dar suporte a

¹Funcionalidade limitada. As configurações de arquitetura do CompSim são restritas a algumas políticas de cache e configurações semelhantes. As configurações do QEMU são principalmente sobre o sistema alvo a ser emulado, não sendo especificamente configurações de arquitetura de baixo nível. O MultiExplorer permite manipulação fina das características do *chip*, mas apenas via texto: na interface gráfica é uma configuração limitada de número de cores e modelo de core (especificações no arquivo de texto).

atividade de DSE, uma ferramenta com recursos adequados para ensino e aprendizado de DSE não é fácil de encontrar.

As ferramentas gratuitas adequadas para a atividade de DSE com frequência requerem o uso e compreensão do funcionamento de outras ferramentas e tecnologias para serem utilizadas, o que aumenta muito o tempo e a complexidade de aprendizado de uso da ferramenta, tornando difícil a adoção para ensino e aprendizado. Uma ferramenta que seja fácil de instalar, configurar, e utilizar, e que contasse com recursos visuais para representação dos conceitos teóricos relacionados a DSE, seria muito útil para sanar essa dificuldade.

Tabela 3.2: Comparação entre as ferramentas a respeito da presença de elementos que permitam abordar os tópicos de **DSE** e *Dark Silicon*.

Ferramenta	DSE	DSE Automático	Dark Silicon
PerfPred	Sim ¹	Não	Não
T&D Bench	Sim ¹	Não	Não
CompSim	Não	Não	Não
MARS	Não	Não	Não
MPSoCBench	Não	Não	Não
QEMU	Sim ¹	Não	Não
MARSSx86	Sim	Não	Não
Heracles	Sim	Não	Não
MultiExplorer	Sim	Sim	Sim

Tabela 3.3: Comparação de aplicabilidade entre as ferramentas.

Ferramenta	Ensino	Pesquisa	Profissional	Download	Open Source
PerfPred	Sim	Não	Não	N/A ³	N/A ³
T&D Bench	Sim	Não	Não	N/A ³	N/A ³
CompSim	Sim	Não	Não	Sim	Não ⁴
MARS	Sim	Não	Não	Sim	Sim
MPSoCBench	Sim	Sim	Não	Sim	Sim
QEMU	Não	Sim	Sim	Sim	Sim
MARSSx86	Não	Sim	Sim	Sim	Sim
Heracles	Sim ²	Sim	Sim	Sim	Sim
MultiExplorer	Sim	Sim	Não	Sim	Sim

¹Funcionalidade de DSE limitada, exigindo bastante esforço manual.

²A aplicabilidade para ensino pode ser limitada em programas de graduação, devido aos requisitos de conhecimento técnico.

³Não foi possível determinar.

⁴A página oficial indica que há previsão da adesão do software ao código-aberto quando seu código atingir um certo nível de maturidade.

Desenvolvimento do MultiExplorer

Como foi constatado na pesquisa sobre os projetos pedagógicos dos cursos de graduação em Ciência de Computação e Engenharia da Computação, o tópico de Exploração de Espaço de Projeto está pouco representado no currículo das instituições de ensino superior brasileiras. Como uma estratégia para difundir o tópico, um dos objetivos deste trabalho é habilitar o uso do MultiExplorer como um recurso para ensino e aprendizado de Exploração de Espaço de Projeto em Arquiteturas de Multiprocessamento.

A ferramenta foi utilizado ao longo de muitos anos como aparato e objeto de pesquisa, permitindo que a utilização da ferramenta por alunos de graduação e pós-graduação pudesse ser observada. Alguns dos principais obstáculos relatados pelos alunos foram:

- configuração de ambiente;
- configuração de ferramentas externas (simuladores);
- interações de entrada e saída com a ferramenta via linha de comando.

Neste capítulo iremos apresentar algumas das abordagens utilizadas para mitigar essas dificuldades e proporcionar uma melhor experiência aos usuários da ferramenta.

4.1 *Melhoria do Processo de Instalação*

Inicialmente, a instalação da ferramenta requeria a configuração manual de um ambiente Linux, com versões específicas de bibliotecas Python e outras ferramentas. Uma série de medidas foi empregada para agilizar esse processo.

4.1.1 Docker

O **Docker**¹ é uma tecnologia utilizada para containerização de aplicações, útil não apenas para o desenvolvimento, mas também para a entrega de software. O Docker foi adotado neste projeto devido a facilidade de configuração de um *container*, e sua compatibilidade com múltiplos sistemas operacionais.

É importante destacar que houveram esforços anteriores para estabelecer um ambiente pré-configurado para o MultiExplorer, porém se utilizando da tecnologia de virtualização na forma de um ambiente VirtualBox². Entretanto, o Docker apresenta não só mais facilidades para o processo de desenvolvimento, como também o mecanismo de containerização é em alguns casos e critérios considerado mais eficiente do que o de máquina virtual.

É possível encontrar na literatura análises abrangentes sobre as diferenças entre *containers* e máquinas, como por exemplo a de Sharma [64]. Entre as vantagens dos *containers* mais relevantes para o nosso projeto, destacam-se a possibilidade de versionar as imagens, menor tamanho em disco, menor tempo de construção e a facilidade para entrega/integração contínua.

Com o estabelecimento de um *container* Docker, é possível garantir um ambiente com uma versão fixa do sistema operacional Ubuntu e bibliotecas correspondentes, de forma a tornar a experiência dos usuários mais homogênea, e também dispensar a necessidade do usuário dominar o processo de configuração do ambiente.

4.1.2 Script de Construção

Além da utilização do Docker, também utilizamos o comando **make**³ para elaborar um script de construção da aplicação. Dessa forma, a maior parte das dependências e configuração da ferramenta podem ser resolvidas para o usuário com a execução de uma única linha de comando.

4.1.3 Snipersim

O principal simulador utilizado na versão atual da ferramenta é o Sniper Multi-Core Simulator [65]⁴. Os professores que supervisionam o LSCad relataram que alunos de graduação que participaram de atividades com versões anteriores do MultiExplorer, bem como também alunos de mestrado, frequentemente não conseguiam ou levavam muito tempo para compilar esse simulador e seus *benchmarks* por conta própria. O próprio autor deste trabalho

¹<https://www.docker.com/>

²<https://www.virtualbox.org/>

³<https://www.gnu.org/software/make/manual/make.html>

⁴http://snipersim.org/w/The_Sniper_Multi-Core_Simulator

também teve a oportunidade de experimentar em primeira mão essas dificuldades.

Para sanar a situação, estabelecemos uma versão pré-compilada do simulador ⁵ para ser utilizada junto ao ambiente **Docker**.

4.2 Prototipação da Interface Gráfica

Outra iniciativa considerada importante para melhorar a usabilidade da ferramenta é a introdução de uma interface gráfica de usuário, ou GUI (*Graphical User Interface*). Para evitar retrabalho e melhor orientar o processo de desenvolvimento dessa interface, o desenvolvimento da mesma foi precedido de uma atividade de prototipação seguindo o processo *Design Sprint*.

O *Design Sprint* é um processo de cinco dias utilizado para elucidar questões cruciais sobre um produto ou serviço, e testar novas idéias junto a membros do público alvo. Criado para ser utilizado nas *startups* da Google Venture ⁶, o Sprint se consolidou numa ferramenta poderosa para acelerar o desenvolvimento de novos produtos e serviços, como descrito no livro homônimo [2].

O processo começa com uma série de reuniões e atividades estruturadas para permitir análise, *brainstorming*, e tomada de decisão. Nas últimas etapas um protótipo é implantado e testado. Os testadores são então entrevistados para se obter uma avaliação final.

A equipe que realizou o processo incluiu professores e alunos do LSCAD. Os professores assumiram os papéis de tomada de decisão. Segundo o livro [2], as reuniões deveriam durar cerca de 6 horas por dia, mas abreviamos essa duração devido às restrições de horário dos participantes, com cada atividade durando de uma a duas horas no máximo. O protótipo foi construído utilizando o Figma ⁷.

Alunos que não faziam parte da equipe do laboratório também participaram do processo no papel de testadores. Indo de encontro ao modelo de Nielsen [44], a *Design Sprint* recomenda um número de apenas cinco avaliações. As sessões de testes e as entrevistas que as sucederam foram gravadas, e os resultados também foram registrados na forma de uma planilha ⁸ de forma a serem utilizados como referência para a implementação.

A partir de questões chave e discussão de idéias, foi elaborado um mapa conceitual da ferramenta MultiExplorer, ilustrado na Figura 4.1. Esse mapa

⁵https://drive.google.com/file/d/1GiQGrqf2AhLcd1fX9bfhGLvXP78YsnD3/view?usp=share_link

⁶<http://www.gv.com/>

⁷<https://www.figma.com>

⁸<https://docs.google.com/spreadsheets/d/1wxFrYb-w1kaBkDgjPuSJyXOt-H8HDiqbAsCatXp5pM0/edit?usp=sharing>

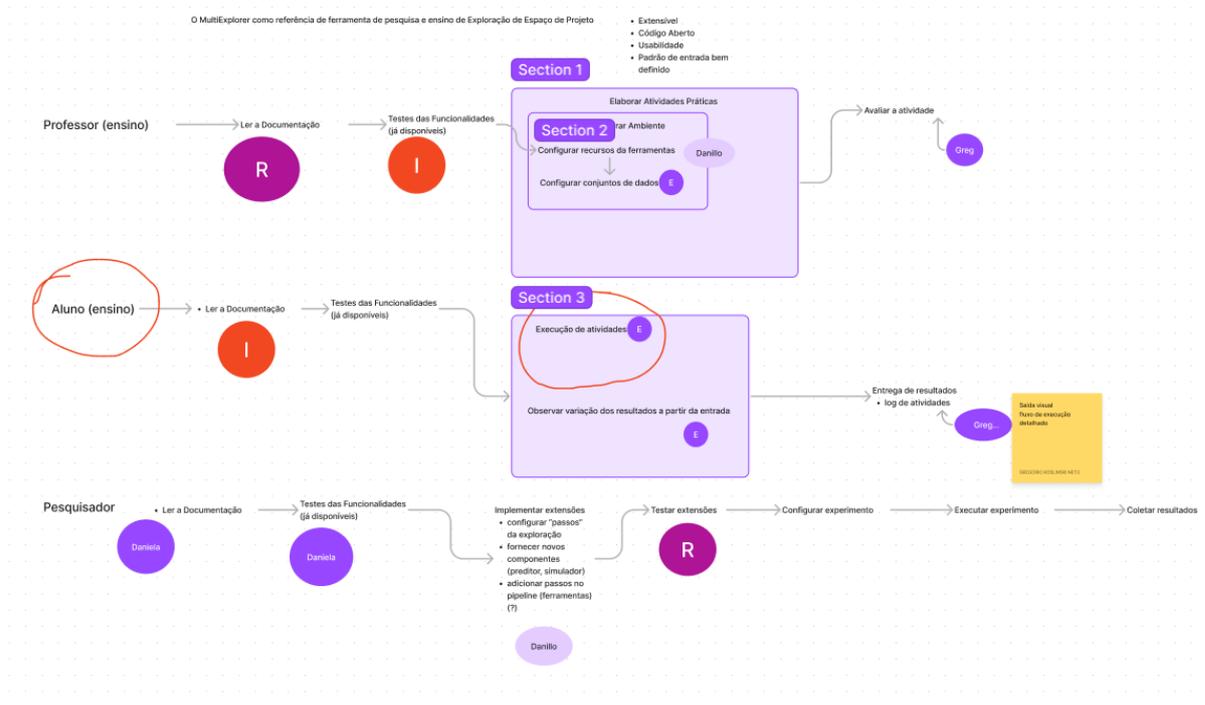


Figura 4.1: Mapa conceitual elaborado durante a sprint.

representa os principais atores e suas interações com a ferramenta.

Também foram elaborados vários diferentes *mockups* da interface gráfica, alguns dos quais podem ser vistos na Figura 4.2. Os *mockups* foram avaliados e as idéias que eles apresentavam sobre a interface gráfica foram refinadas, até o processo convergir em uma proposta final para o protótipo, cujo esboço está presente na Figura 4.3. A versão final do protótipo ainda está disponível em um arquivo do Figma: <https://www.figma.com/file/HBeKu4gBTJ7pv4GPp6RKV4/MultiExplorer?type=design&node-id=0%3A1&mode=design&t=rlcmPoSePyxButpQ-1>.

4.3 Implementação da Interface Gráfica

A implementação da interface gráfica foi realizada com base em um módulo gráfico da linguagem **Python** chamado **Tkinter**⁹. O Tkinter organiza os elementos gráficos na forma de uma estrutura de árvore, na qual a janela principal é a raiz. O nó-pai de um elemento da árvore é frequentemente referenciado como *master* nas propriedades e chamadas de métodos de inicialização e operação das classes desse módulo. Os atores principais nessa árvore são os *frames* (objetos da classe **Tkinter.Frame**), que são utilizados para estabelecer o enquadramento dos elementos na interface, os *widgets* (objetos de classes derivadas de **Tkinter.Widget**), que realizam apresentações gráficas e controlam certas interações entre o usuário e a interface, e os *canvas*

⁹<https://docs.python.org/2.7/library/tkinter.html>

(objetos da classe **Tkinter.Canvas**), que consistem de objetos utilizados para desenhar livremente num espaço determinado da interface. Muitas classes do *framework* implementado derivam de **Tkinter.Frame** ou **Tkinter.Canvas**, ou foram engendradas para controlar e interagir diretamente com objetos dessas classes.

A arquitetura de software implantada foi parcialmente baseada no padrão **MVC** (*Model-View-Controller*) [66], muito utilizado em aplicações *web*. Devido a diferença de contexto e necessidades específicas da aplicação, arquitetura não pode ser classificada como completamente pertinente ao modelo.

Apesar da implementação específica ter sido realizada para permitir utilização via interface gráfica do fluxo de exploração de espaço de projeto ciente de *dark silicon* em arquiteturas de heterogêneas multiprocessamento, a arquitetura proposta tem como objetivo permitir que outros fluxos de exploração sejam conectados com a interface gráfica com baixo esforço de codificação nas camadas de apresentação e de controle. A Figura 4.4 permite visualizar os pacotes que foram adicionados durante a implementação. O pacote **GUI** contém as classes básicas da camada de apresentação, enquanto que o pacote **Infrastructure** contém classes relacionadas a modelagem de dados e controle. Por fim, o pacote **CPUHeterogeneousMulticoreExploration** contém as classes que são específicas do fluxo de execução implementado.

Os pacotes **GUI** e **Infrastructure** tem como propósito providenciar um *framework* para dar suporte a integração de novos fluxos de execução à interface gráfica do MultiExplorer. Um diagrama de classes que proporciona uma visão geral da arquitetura do *framework* pode ser encontrado no repositório da ferramenta no *Github* ¹⁰.

Uma das funcionalidades do *framework* é permitir que as entradas de usuário de um novo fluxo sejam exibidas na interface gráfica, sem que seja necessário modificar diretamente a camada de apresentação: basta pré-definir as entradas do fluxo nas classes de **Step**, na forma de uma lista de objetos das classes **Input** e **InputGroup**. Isso é possível devido a interação da classe **InputScreen** (responsável por apresentar a tela de entrada de dados) com essas entradas pré-definidas via comunicação com as classes **InputTabsController**, **InputTab**, e sobretudo a classe **InputGUI**, que opera como uma *abstract factory* [66] das classes responsáveis por exibir os campos de entrada de usuário. A Figura 4.5 ilustra os relacionamentos entre essas classes, e a Figura 4.6 apresenta um diagrama de sequência contendo a comunicação desde o momento que o usuário clica no botão *Start* para iniciar o fluxo de execução selecionado, até a chamada iterativa do método **InputTabsController::add_step_tab(step)** que irá criar uma nova aba de entradas para cada

¹⁰<https://link.ufms.br/GUnot>

passo do fluxo de execução que exigir ou permitir entradas de usuário. Na Figura 4.7 vemos em detalhes o uso da classe *InputGUI* como a *abstract factory* dos objetos que irão representar os campos de entrada na interface gráfica. A partir das entradas de usuário (objetos das classes **Input** e **InputGroup**) das etapas de execução (objeto da classe **Step**) são criados os componentes gráficos (objetos das classes **InputFrame**, **MultipleInputFrame** e **InputGroupFrame**) que irão exibir e permitir a interação do usuário com esses campos de entrada. A classe **InputGUI** é responsável por mapear o tipo correto de componente gráfico de acordo com os objetos das classes **Input** ou **InputGroup** que são recebidos no método **InputGUI::create_input**.

O controle da execução do fluxo fica a carga da implementação específica das classes **ExecutionFlow**, **Step** e **Adapter**. Por definição, a execução de uma etapa (**Step**) do fluxo de execução é realizada numa *thread* de execução separada para que a interface gráfica continua responsiva durante o período em que a etapa está sendo executada. O diagrama de classes da Figura 4.8 permite a visualização dos relacionamentos entre as classes envolvidas na execução do fluxo e as classes responsáveis por exibir o progresso da execução e quaisquer resultados parciais. A execução de cada etapa de um fluxo (responsabilidade de um objeto da classe **Step**) é realizada em uma *thread* separada, de forma assíncrona, para evitar que a interface gráfica de usuário fique “paralisada” durante o processo. O controle e monitoramento da execução é feito via eventos e *jobs* que verificam periodicamente a situação da *thread* de execução. O diagrama de sequência na Figura 4.10 ilustra o processo por meio do qual a classe responsável por exibir a situação de execução de uma etapa do fluxo (**StepDisplay**) monitora a execução da etapa corresponde (**Step**) utilizando-se de eventos e um *job* iterativo. Devido a esses mecanismos implementados, o funcionamento interno específico de cada fluxo é pouco relevante para o comportamento da interface gráfica, de forma a minimizar a necessidade de mudanças na infraestrutura quando um novo módulo de fluxo de execução é adicionado.

De forma análoga, a implementação específica de cada apresentador (objetos de classes derivadas de **Presenter**) geralmente não requer alterações no fluxo de apresentação, que pode ser visto no diagrama da Figura 4.11. Todos os *frames* (objetos da classe **Tkinter.Frame**) de apresentação são configurados com uma mesma largura, mas a altura é definida pelo apresentador (devolutiva do método **Presenter::present_results**). Esses *frames* são empilhados na ordem que foram construídos dentro de uma área de *scrolling* da tela de apresentação.

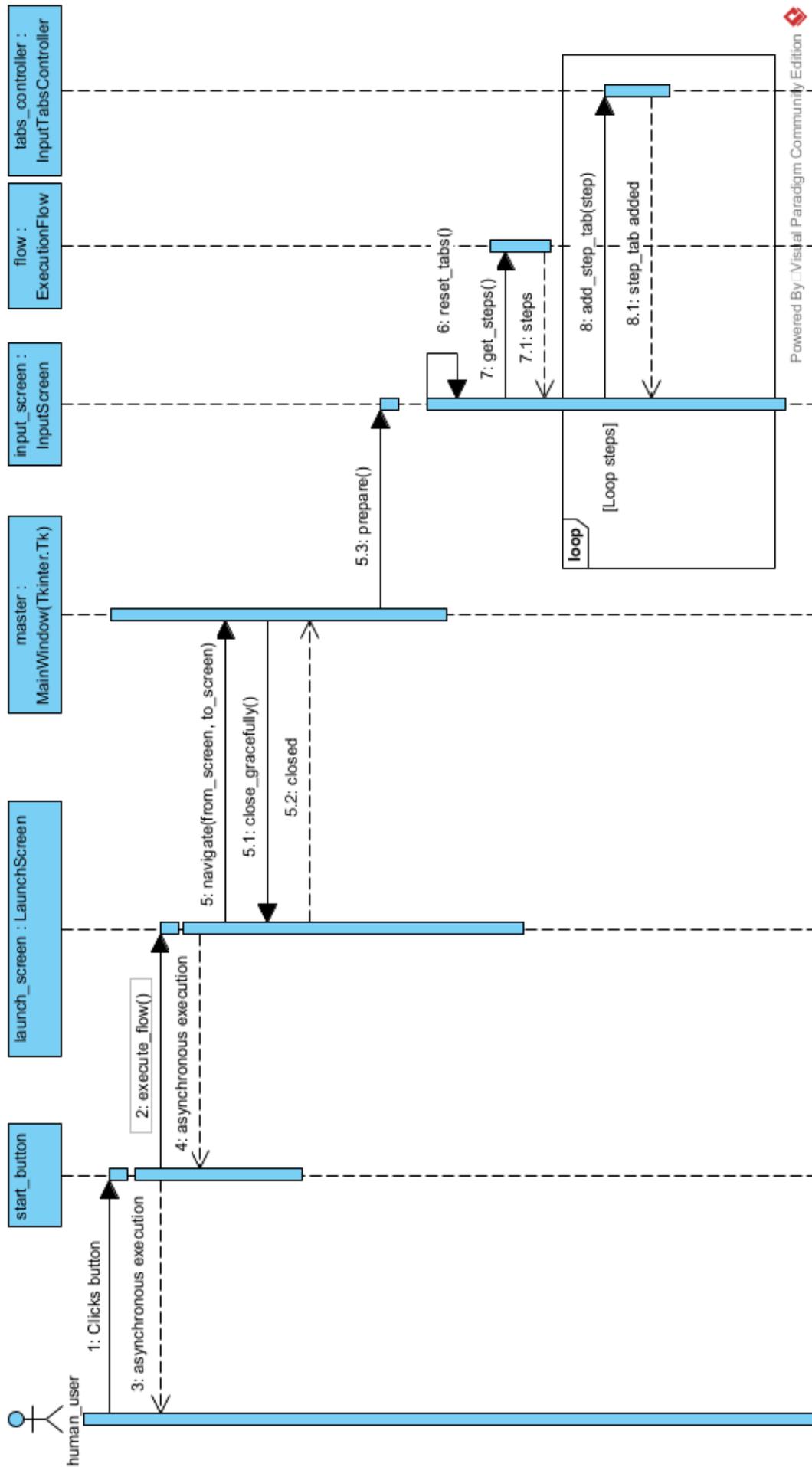


Figura 4.6: Diagrama de sequência ilustrando a comunicação que resulta na criação das abas de entradas de usuário.

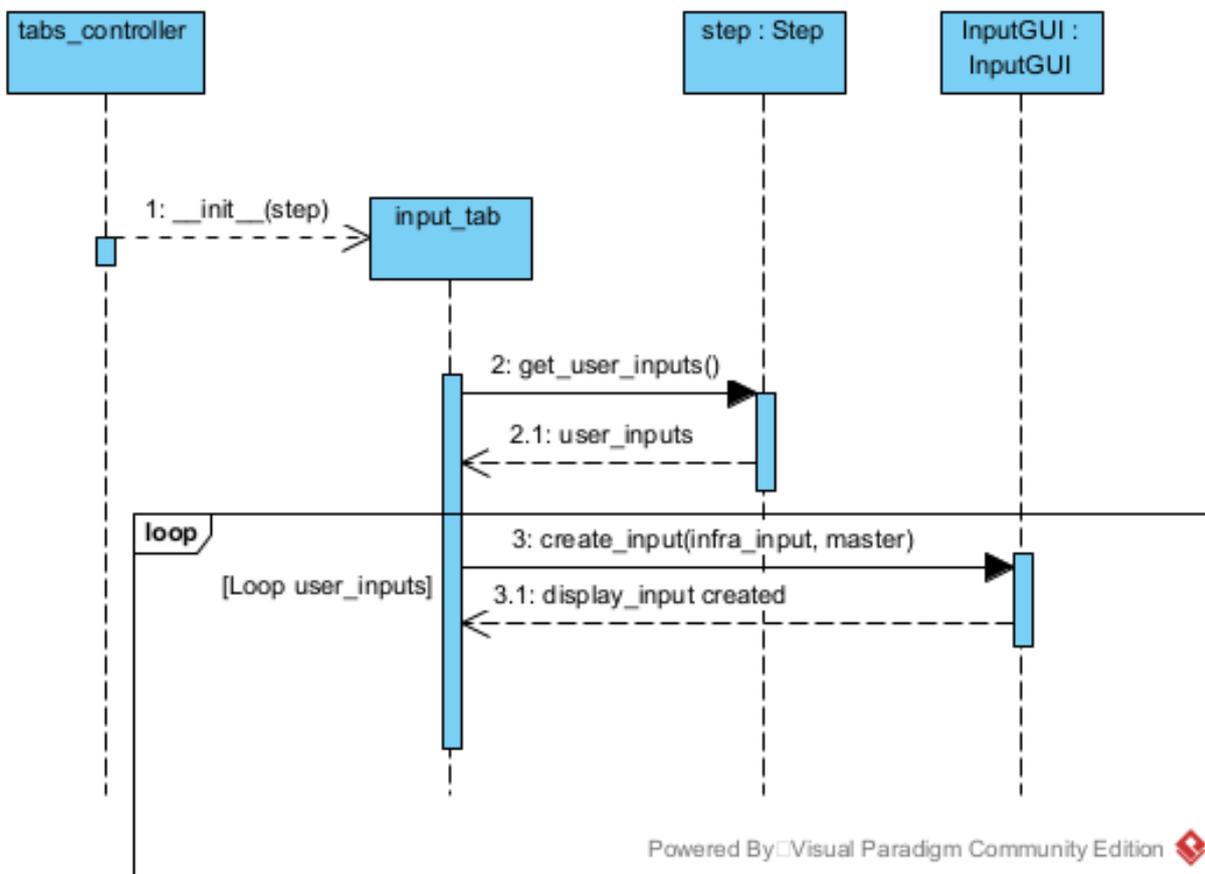


Figura 4.7: Esse diagrama de sequência ilustra a comunicação entre a classe **InputTab** e a classe **InputGUI** que resulta na criação dos objetos que irão exibir as entradas de usuário na interface gráfica.

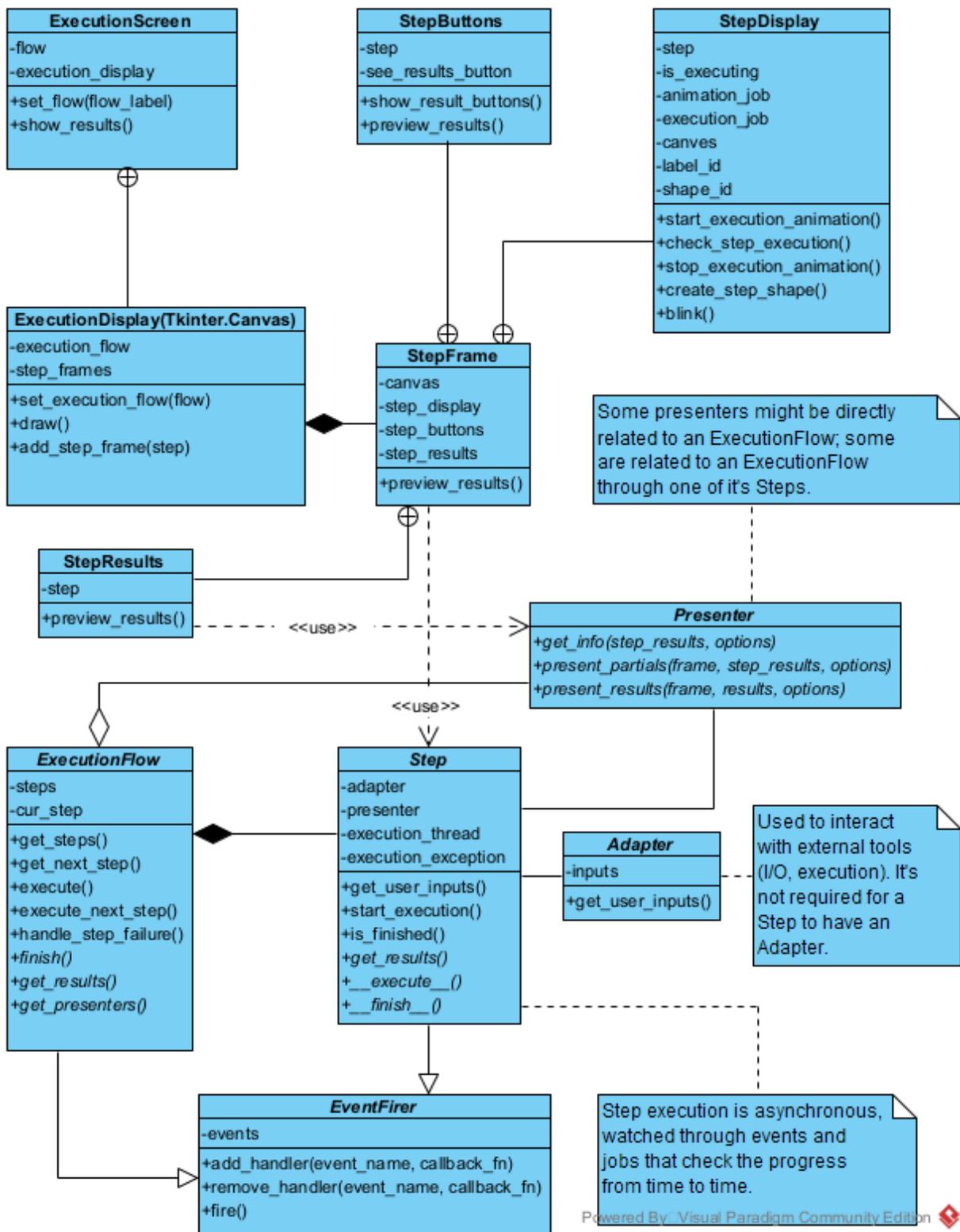


Figura 4.8: Diagrama de classes das classes envolvidas no controle da execução, apresentação da execução na interface gráfica e apresentação de resultados parciais.

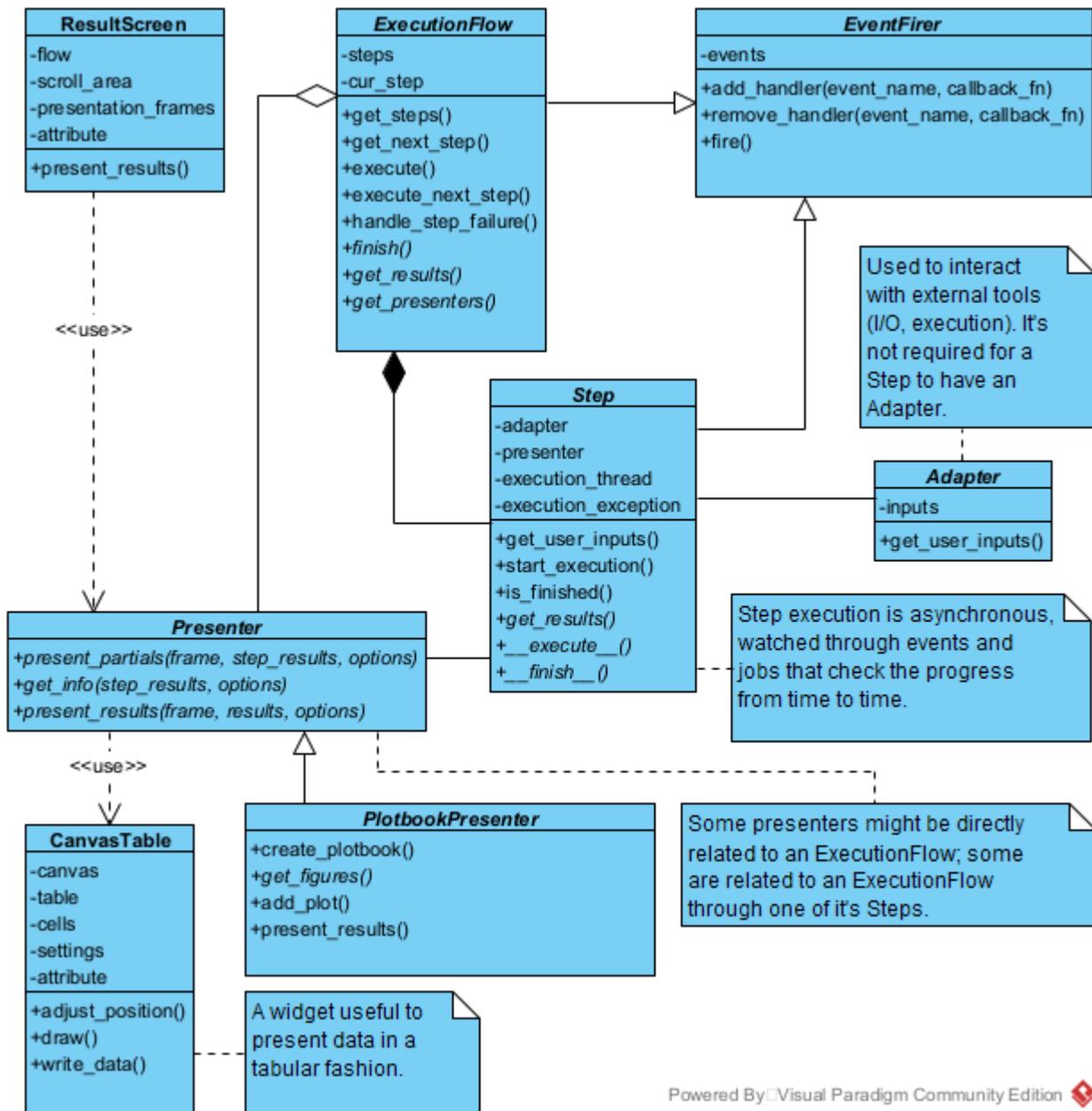


Figura 4.9: Diagrama ilustrando as classes envolvidas na apresentação de resultados. A tela de resultados (objeto da classe **ResultScreen**) controla o processo de apresentação realizado pelos apresentadores (objetos de classes derivadas de **Presenter**).

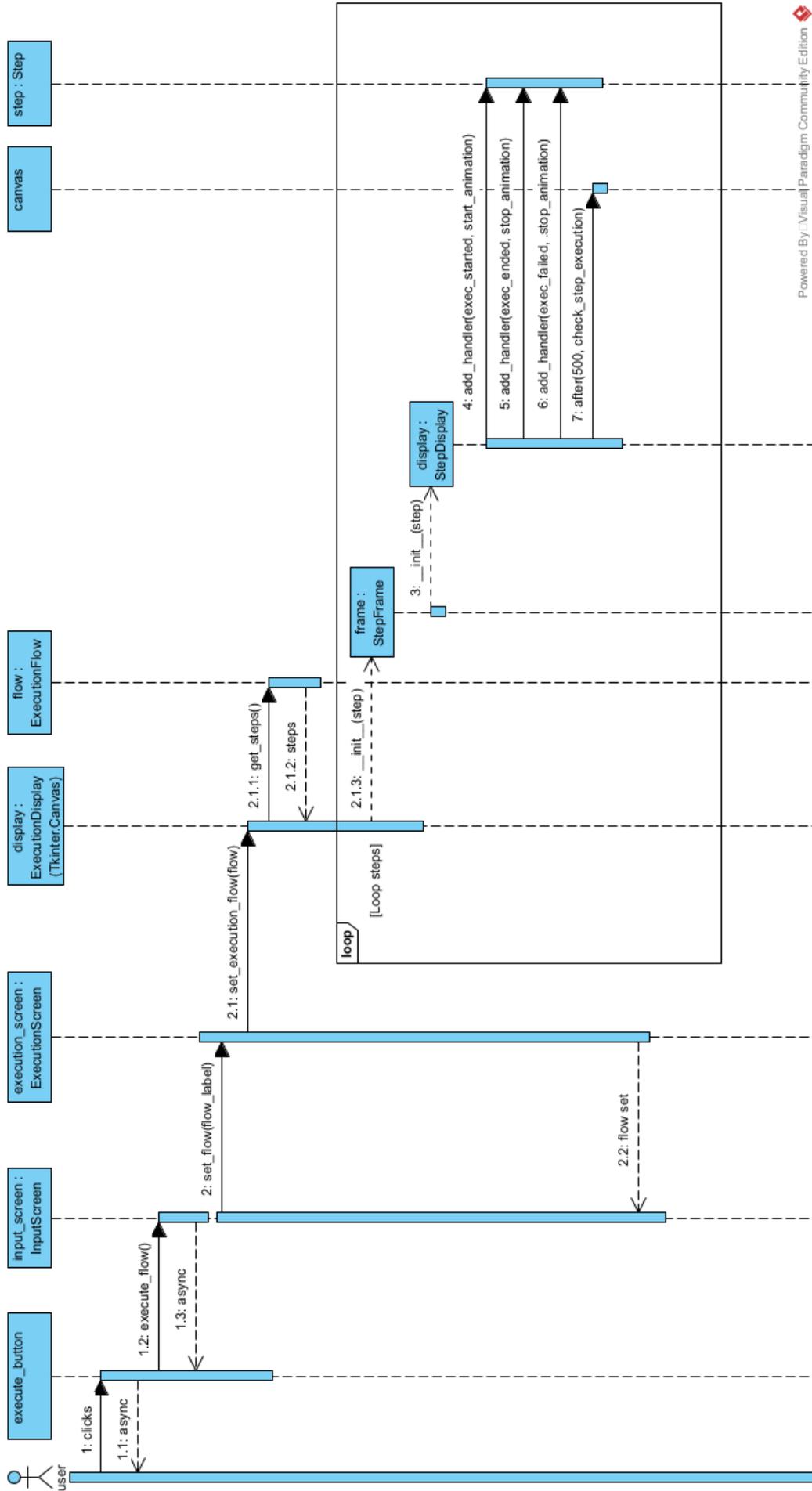


Figura 4.10: Por meio dos métodos **EventFiring::add_handler** e **Widget::after()** um objeto da classe **StepDisplay** pode monitorar e reagir ao progresso do processo de execução realizado por um objeto da classe **Step**.

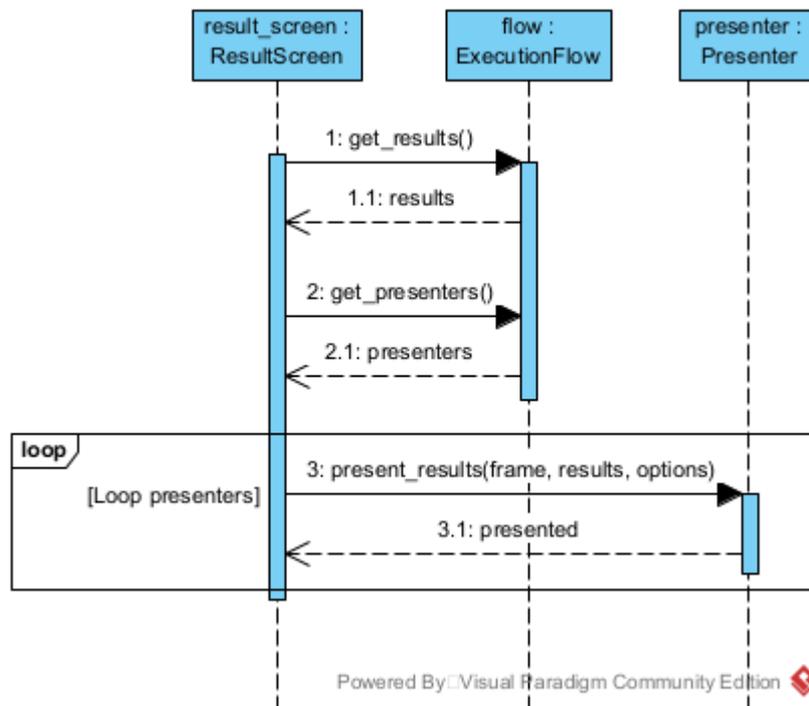


Figura 4.11: O objeto da classe **ResultScreen** recebe os resultados e apresentadores do fluxo em execução e repassa um *frame* para cada apresentação.

4.3.1 Exemplo de Atividade de Exploração

Para ilustrar o fluxo de uso da ferramenta, iremos apresentar aqui uma breve atividade de simulação e exploração de espaço de projeto no MultiExplorer. Vamos considerar que a ferramenta e devidos pré-requisitos já foram instalados e configurados.

O primeiro passo é abrir a interface gráfica. Para isso é necessário acessar o diretório da ferramenta e utilizar o comando:

```
$ python ME.py
```

A partir daí, a operação é feita através da interface gráfica. Na primeira janela, ilustrada na Figura 4.12, é necessário escolher um fluxo de exploração de espaço de projeto. No caso, será selecionado o fluxo de exploração em arquiteturas heterogêneas de processadores.

Na sequência, é necessário configurar a arquitetura que será simulada, caracterizando-a pelo número de núcleos, litografia e modelo do núcleo, como ilustrado na Figura 4.13.

Também é preciso que o usuário especifique as configurações e restrições da busca (Figura 4.14).

A interface de execução permite que o usuário acompanhe o progresso das simulações e da busca (Figura 4.15). Quando a execução estiver concluída o usuário poderá prosseguir para a tela de apresentação de resultados.

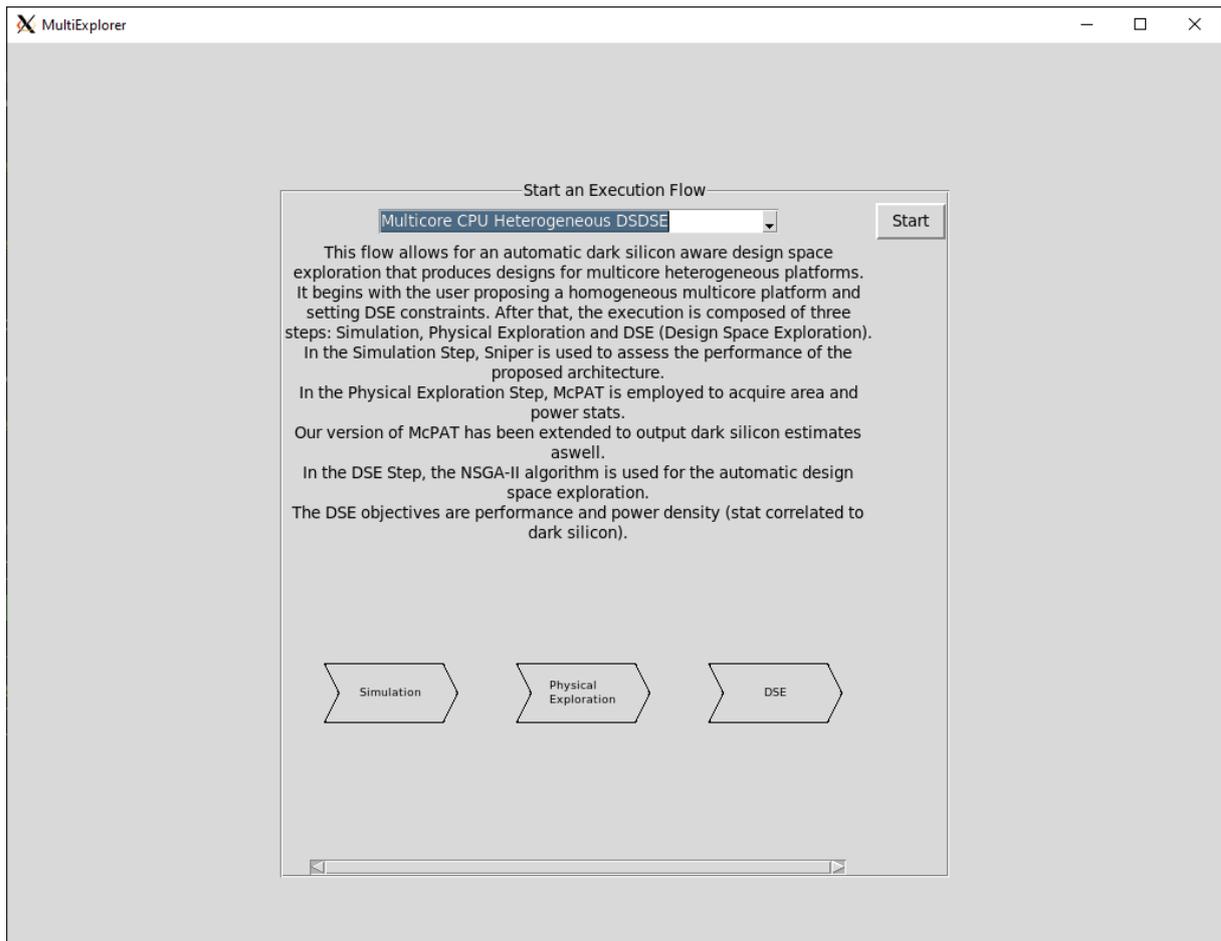


Figura 4.12: Na interface inicial, o usuário deve selecionar um fluxo de execução e pressionar o botão “Start”.

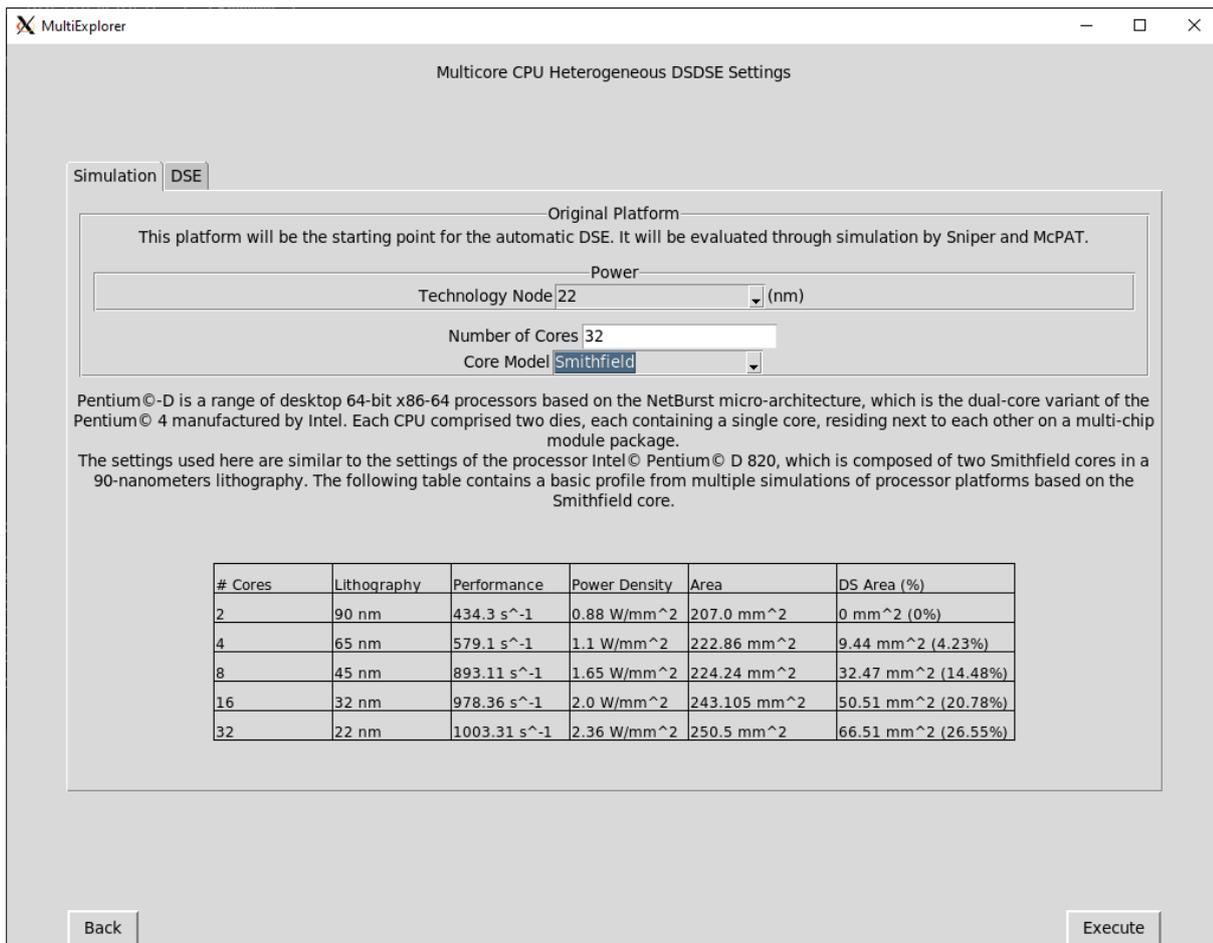


Figura 4.13: Na primeira aba de entradas de usuário do fluxo de exploração de CPUs Heterogêneas o usuário deve informar o número de núcleos, a litografia, e o modelo de núcleo.

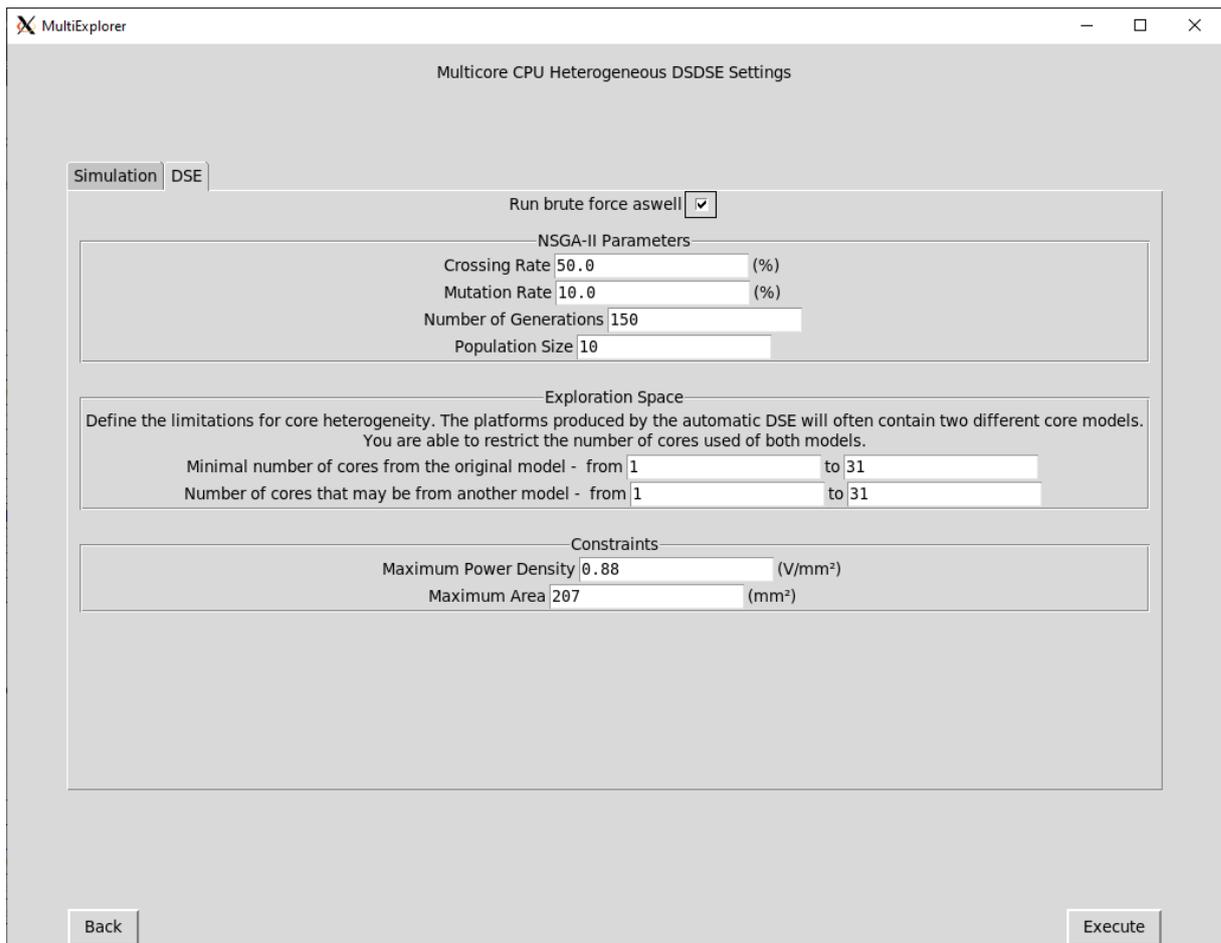


Figura 4.14: Na aba de configurações de DSE o usuário pode modificar a calibragem do algoritmo evolutivo de busca (NSGA-II) e determinar as restrições de número de núcleos, área e densidade de potência máximas.

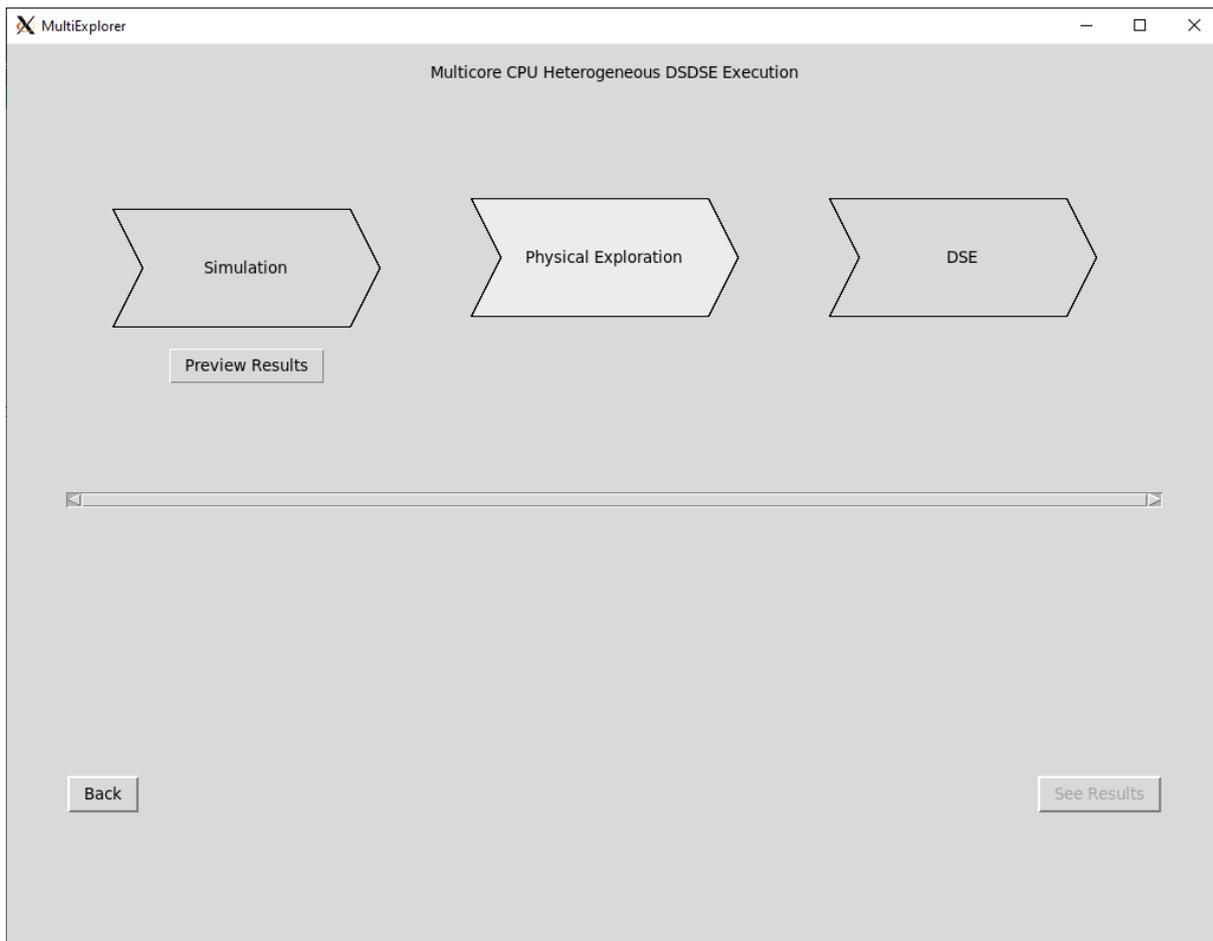


Figura 4.15: O fim de cada etapa permite a visualização de resultados parciais (“Preview Results”). Após a conclusão de todas as etapas da execução o botão “See Results” é habilitado.

Os resultados da ferramenta incluem um perfil da arquitetura simulada (Figura 4.16), o perfil das arquiteturas encontradas pelo NSGA-II (Figura 4.16), um gráfico ilustrando a comparação dessas arquiteturas (Figura 4.17) e também um perfil das arquiteturas encontradas pelo algoritmo de força bruta (Figura 4.18).

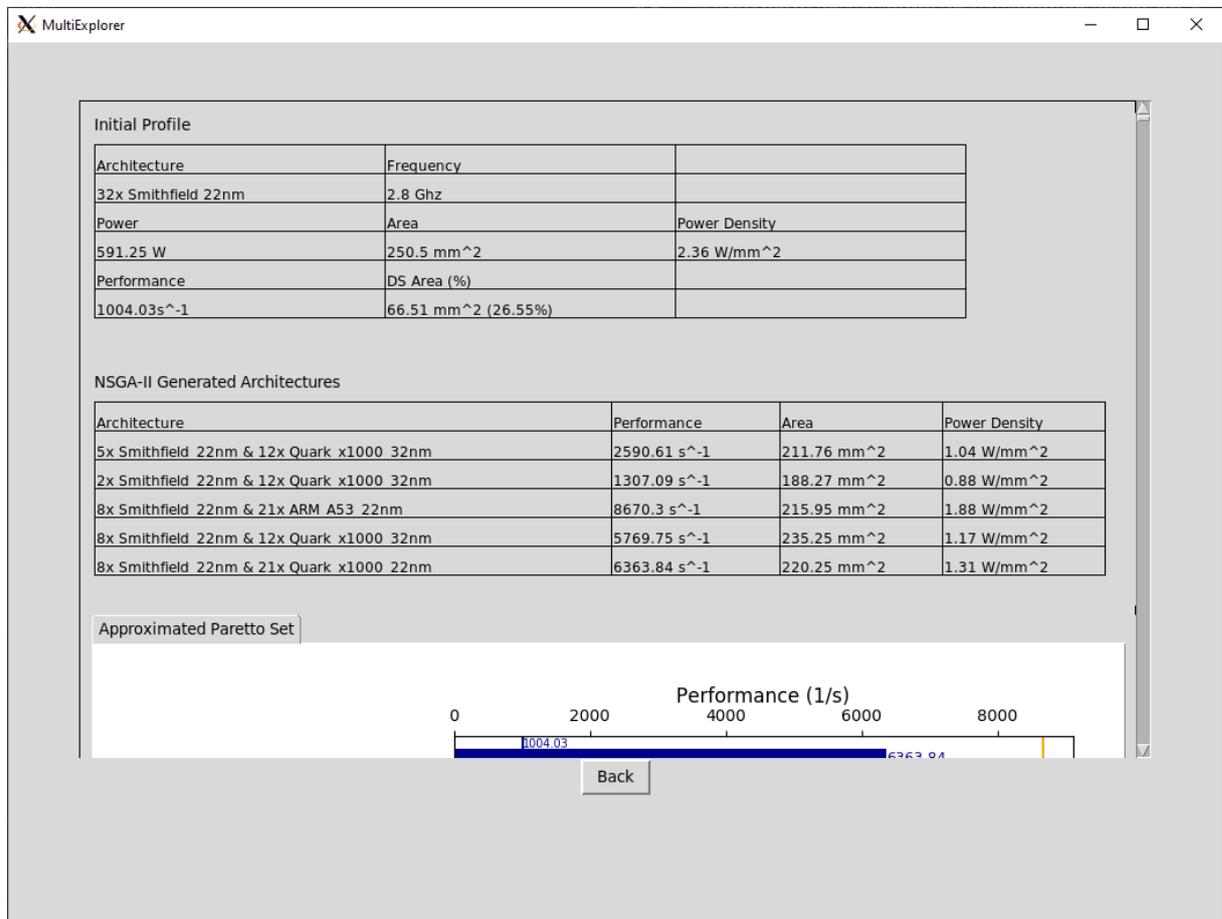


Figura 4.16: Após o fim da exploração é possível visualizar um perfil de desempenho e parâmetros físicos da plataforma simulada, bem como um perfil semelhante das plataformas encontradas pelo NSGA-II.

4.4 Considerações Finais

Todo o processo de desenvolvimento da ferramenta MultiExplorer, desde a concepção e prototipação, até a implementação em **Python** foi realizado ao longo de cerca de dois anos. O trabalho foi fundamentado em metodologias e padrões reconhecidos e bem documentados. O processo de implementação foi realizado por um único desenvolvedor: o autor deste trabalho. Devido a dificuldades técnicas e restrições de tempo não foi possível estabelecer políticas e métodos para teste unitário ou de integração do software. Este é um ponto que tem potencial para ser explorado no futuro. Após a realização do estudo de caso que é descrito no Capítulo 5, houve uma etapa adicional de

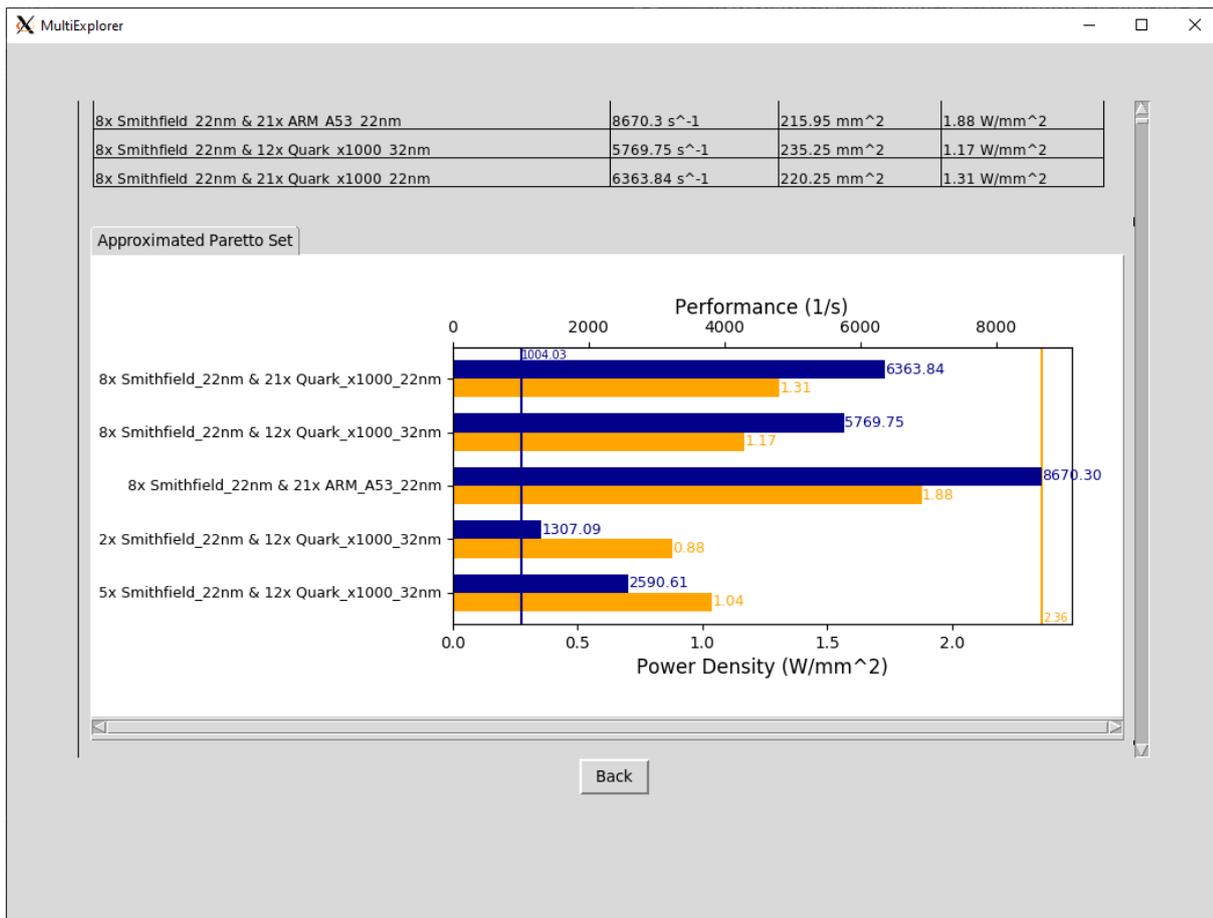


Figura 4.17: Um das saídas disponibilizadas pela ferramenta é um gráfico ilustrando as plataformas encontradas pelo NSGA-II: as barras azuis representam desempenho, e as laranjas densidade de potência.

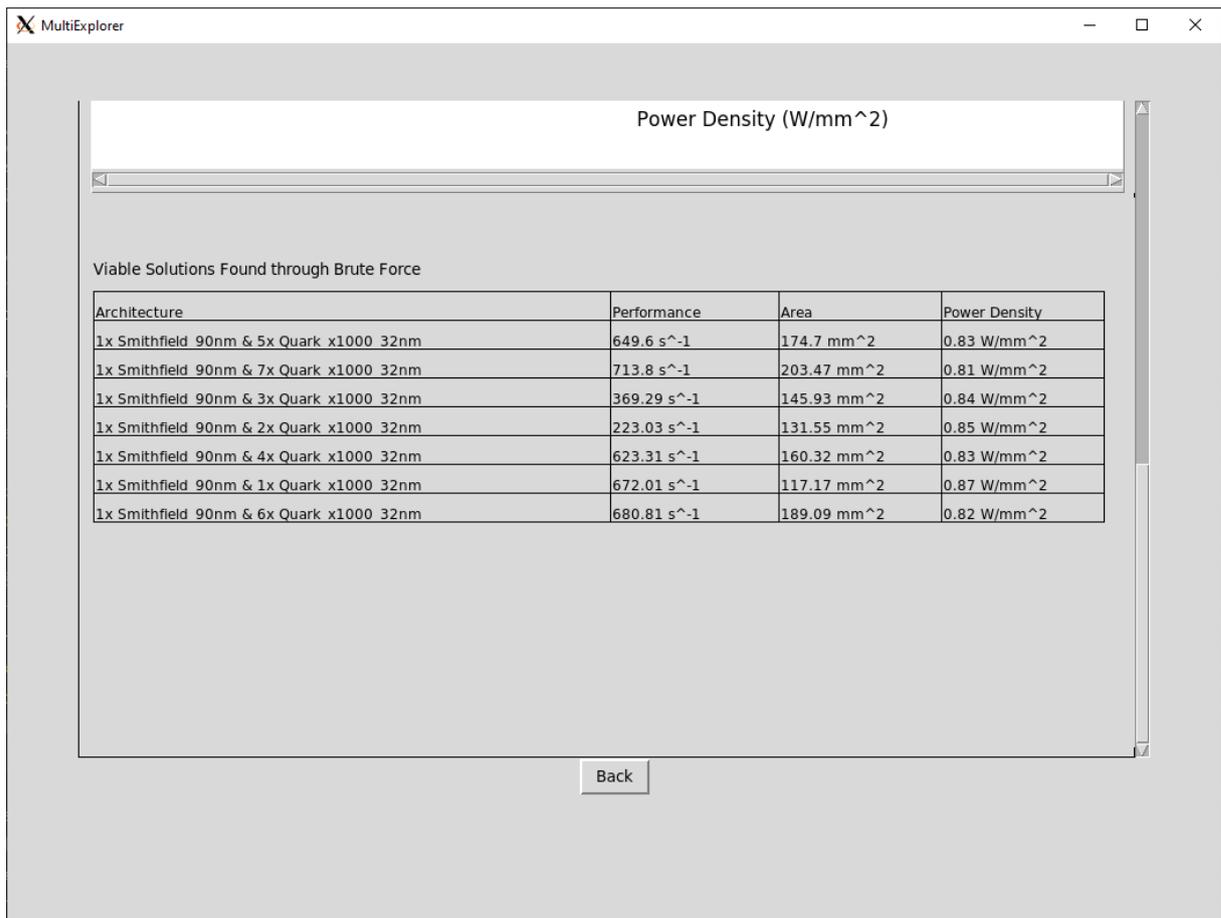


Figura 4.18: Nos casos em que a escala do espaço de busca é razoavelmente pequena, pode-se executar o algoritmo de força bruta para buscar por todas as soluções viáveis.

implementação para refinamento da interface gráfica que levou em consideração o *feedback* dos alunos sobre a Facilidade de Uso (**PEOU**) e Utilidade (**PU**) da ferramenta. Mais informações sobre esse *feedback* também está disponível no Capítulo 5.

Avaliação do MultiExplorer como Recurso Educacional

Este capítulo apresenta um estudo de caso cujo principal objetivo era avaliar a ferramenta MultiExplorer no contexto do ensino e aprendizado do tópico de Exploração de Espaço de Projeto em Arquiteturas de Computadores na Ciente de *Dark Silicon*. Para realização do estudo, foi proposto um minicurso extra-curricular organizado em duas componentes: introdução teórica e atividade prática, cada uma das componentes sendo apresentada aos alunos em uma única aula com duração entre duas e três horas, totalizando um minicurso de duas aulas com um total de carga horária entre quatro e seis horas. A aprendizagem dos alunos foi medida por meio de um questionário teórico e um relatório experimental. A experiência de usuário proporcionada pela ferramenta foi aferida através de uma pesquisa modelada a partir do **TAM**.

5.1 *Objetivos e Planejamento*

A realização deste estudo de caso foi motivada por duas questões:

- O MultiExplorer pode ser aplicado como recurso educacional no aprendizado do tópico “Exploração de Espaço de Projeto em Arquiteturas de Computadores Ciente de *Dark Silicon*” a nível de graduação?
- Os conceitos relacionados ao tópico podem ser compreendidos e articulados por alunos de graduação?

Para clarificar a primeira questão, era necessário observar a experiência de uso de alunos de graduação com a ferramenta no contexto de aprendizado, e

para clarificar a segunda era necessário também medir a aprendizagem desses alunos. Seria então necessário proporcionar aos alunos oportunidade para utilizar a ferramenta em um ambiente controlado, segundo um roteiro específico, e então medir de alguma forma a sua experiência. Como apresentado no Capítulo 2, Seção 2.5, o **TAM** pode ser utilizado para explicar o comportamento de usuários em relação a uma nova tecnologia que lhes é introduzida, e de acordo com uma revisão sistemática da literatura realizada por Granić & Nikola [49], também tem sido utilizado com frequência para estudos na área educacional e é considerado um modelo confiável para investigar o uso de novas tecnologias no campo do ensino e aprendizado. Sendo assim, foi decidido que os alunos deveriam utilizar a ferramenta para execução de uma atividade de exploração de espaço de projeto pré-definida, então submetidos a uma pesquisa de opinião sobre sua experiência de usuário para investigar as variáveis de Facilidade de Uso Percebida (**PEOU**) e Utilidade Percebida (**PU**) descritas no modelo **TAM**.

Entretanto, a própria execução de uma atividade de exploração de espaço de projeto requer dos alunos um mínimo de contextualização sobre o assunto, e assim sendo, eles deveriam primeiro receber uma introdução teórica sobre os conceitos, criando a oportunidade para medir sua aprendizagem antes e depois do uso da ferramenta. Houveram algumas considerações *a priori*: pareceu razoável assumir que alunos que compreenderem melhor os conceitos teóricos irão ter melhor desempenho na atividade prática e uma experiência melhor com a ferramenta, bem como alunos que tiverem mais facilidade de utilizar a ferramenta terão melhor desempenho na atividade prática. Também é esperado que o uso da ferramenta melhore a aprendizagem dos alunos. Para verificar essas considerações, é preciso utilizar os dados para o teste de algumas hipóteses:

- A medida de aprendizagem realizada antes do uso da ferramenta está correlacionada com a medida realizada após o uso da ferramenta?
- Existe diferença significativa entre as medidas de aprendizagem antes e depois do uso da ferramenta?
- As medidas de aprendizagem tem correlação com as medidas de facilidade de uso percebida e utilidade percebida?

Para responder as questões e testar as hipóteses levantadas, foi proposta a seguinte estrutura e sequência para o experimento: primeiro os alunos são introduzidos a conceitos relacionados com a Exploração de Espaço de Projeto Ciente de *Dark Silicon* e a ferramenta MultiExplorer; na sequência, a aprendizagem dos alunos é medida através de um questionário composto por três questões abertas; a seguir, os alunos realizam uma atividade de exploração de

espaço de projeto, seguindo um roteiro pré-definido; os alunos devem produzir um relatório técnico sobre essa atividade, com o registro dos dados fornecidos pela ferramenta, e uma discussão sobre os resultados da exploração, habilitando a segunda medição de aprendizagem; por fim os alunos preenchem a pesquisa de opinião sobre a experiência de usuário com a ferramenta. Mais detalhes sobre cada uma dessas etapas e seus instrumentos podem ser encontrados nas subseções a seguir.

5.1.1 Introdução Teórica

A introdução aos conceitos relacionados a Exploração de Espaço de Projeto Ciente de *Dark Silicon* e a ferramenta MultiExplorer foi estruturada na forma de uma aula expositiva com duração de até duas horas e meia, dividida em duas partes. A primeira parte teve duração entre uma hora e meia e duas horas, tempo utilizado para apresentar e contextualizar conceitos teóricos relacionados à Exploração de Espaço de Projeto, *Dark Silicon*, e à ferramenta MultiExplorer, sendo que os dois primeiros conceitos ocuparam mais tempo, principalmente o tópico de exploração. Na segunda parte, com duração em torno de meia hora, a ferramenta MultiExplorer é apresentada aos alunos em uma breve apresentação, consistindo da explicação e demonstração das funcionalidades da ferramenta e um experimento simples de simulação e exploração. A ementa da introdução teórica consta a seguir:

- **Conceitos e Contextualização**

- Exploração de Espaço de Projeto: definição, conceitos básicos, história, contexto e aplicações; VLSI; otimização multiobjetivo: dominância de Pareto, conjunto Pareto-ótimo, fronteira Pareto-ótima.
- *Dark Silicon*: definição, história e contexto; relação com Lei de Moore e Escala de Dennard; barreira de utilização; impactos na exploração de espaço de projeto em arquiteturas de multiprocessadores; escalabilidade Dennard x Pós-Dennard;
- MultiExplorer: introdução da ferramenta; visão geral de funcionalidades.

- **Apresentação da Ferramenta**

- Apresentação da ferramenta; pré-requisitos de software; comandos; introdução a interface gráfica.
- Funcionalidades de simulação: configuração de arquitetura, índices de performance e parâmetros físicos.

- Funcionalidades de exploração: parâmetros, restrições e resultados da exploração automática.

Uma cópia dos slides que foram apresentados aos alunos está disponível em: <https://link.ufms.br/DsOao>.

5.1.2 Questionário Teórico

Após a introdução teórica, um questionário foi enviado aos alunos através da plataforma Google Forms ¹. Os alunos deveriam entregar as respostas para o questionário antes de realizar a atividade de exploração com a ferramenta MultiExplorer.

O questionário teórico é composto por três questões discursivas:

- (q_1) O que é Exploração de Espaço de Projeto? Qual a sua relevância para o campo da computação?
- (q_2) O que é *Dark Silicon*?
- (q_3) O que é MultiExplorer? Qual a relação com os dois conceitos anteriores?

As respostas dos estudantes para cada questão receberam uma nota de um a cinco de acordo com a corretude e cobertura:

1. **Inválida:** O estudante foi vago e não abordou os conceitos correlatos;
2. **Incorreta:** O estudante tentou abordar os conceitos relacionados, mas a elaboração foi incorreta;
3. **Parcial:** O estudante apresentou alguns conceitos de forma correta e outros de forma incorreta;
4. **Incompleta:** O estudante apresentou apenas conceitos de forma correta, porém não apresentou todos os conceitos esperados;
5. **Completa:** O estudante apresentou todos os conceitos esperados de forma correta.

A nota final do aluno neste questionário, composta da média das notas em cada questão ($Q = (q_1 + q_2 + q_3)/3$), é a primeira medida de aprendizado.

¹<https://www.google.com/forms/about/>

5.1.3 Atividade Prática

A atividade prática consiste em uma exploração de espaço de projeto baseada nas simulações de evolução tecnológica do circuito de referência, baseado no modelo de núcleo *Smithfield*. Os objetivos da exploração são maximizar o desempenho e minimizar o *dark silicon*, respeitando a restrição de área do circuito de referência. Os alunos teriam no máximo três horas para utilizar a ferramenta e realizar as atividades, mas poderiam entregar o relatório resultante dentro do prazo de uma semana.

Roteiro da atividade:

- Apresentação do circuito de referência (*Pentium®-D*, dois núcleos *Smithfield* na litografia de 90nm) e as simulações de evolução tecnológica
- Exploração de Espaço de Projeto: parâmetros de busca baseados no circuito de referência; registro de resultados e comparação de arquiteturas.

O circuito de referência utilizado na atividade prática foi baseado na série **Pentium®-D**^{2 3}, utilizando especificações semelhantes às do processador **Intel® Pentium® D 820**⁴. A Tabela 5.1 apresenta os resultados das simulações das arquiteturas baseadas no núcleo *Smithfield*.

Tabela 5.1: Evolução tecnológica (simulada) das arquiteturas baseadas no núcleo *Smithfield*. **D.P.:** Densidade de Potência. **D.S.:** *Dark Silicon* (área e porcentagem).

# Núcleos	Litografia (nm)	Performance	D.P. (W/mm ²)	Área (mm ²)	D.S.(mm ² (%))
2	90	434,3s ⁻¹	0,88	207,0	0 (0%)
4	65	579,1s ⁻¹	1,1	222,86	9,44 (4,23%)
8	45	893,11s ⁻¹	1,65	224,24	32,47 (14,48%)
16	32	978,36s ⁻¹	2,0	243,105	50,51 (20,78%)
32	22	1003,31s ⁻¹	2,36	250,5	66,51 (26,55%)

Baseando-se na nas restrições de área e densidade de potência da arquitetura caracterizada por dois núcleos *Smithfield* de 90 nm, os alunos devem realizar cinco explorações de espaço de projeto com a ferramenta MultiExplorer, utilizando o núcleo *Smithfield* em qualquer litografia como base, tentando encontrar arquiteturas heterogêneas que sejam vantajosas nos quesitos de performance e *dark silicon*.

A calibragem do algoritmo NSGA-II não precisa ser alterada entre esses experimentos, já que uma experimentação para observar os efeitos da calibragem não pode ser comportada na duração do minicurso. Portanto os alunos

²<https://ark.intel.com/content/www/br/pt/ark/products/codename/5788/products-formerly-smithfield.html>

³https://en.wikipedia.org/wiki/Pentium_D

⁴<https://ark.intel.com/content/www/br/pt/ark/products/27512/intel-pentium-d-processor-820-2m-cache-2-80-ghz-800-mhz-fsb.html>

precisam apenas determinar as restrições de número de núcleos, área e densidade de potência. Detalhes da operação da ferramenta podem ser vistas no Capítulo 4, Seção 4.3.1.

5.1.4 Relatório Técnico

O relatório técnico (*r*) consiste basicamente dos resultados das simulações e explorações realizadas durante a atividade prática, registrados pelo aluno em uma planilha, junto com uma comparação discursiva entre as arquiteturas de acordo com os conceitos, objetivos e restrições apresentados previamente. Um modelo foi disponibilizado para os alunos via Google Sheets ⁵. Os alunos tem um máximo de três horas para realizar a atividade prática, mas recebem um prazo de uma semana para entregar o relatório. Um exemplo dos resultados de exploração pode ser visto na Tabela 5.2.

Tabela 5.2: Exemplos de arquiteturas apresentadas pelo MultiExplorer após a exploração automática realizada pelo algoritmo NSGA-II. **D.P.:** Densidade de Potência

Núcleos	Performance	Área	D.P.
5x Smithfield 22 nm & 12x Quark x1000 32 nm	2590.61 s^{-1}	211.76 mm^2	1.04 W/mm^2
2x Smithfield 22 nm & 12x Quark x1000 32 nm	1307.09 s^{-1}	188.27 mm^2	0.88 W/mm^2
8x Smithfield 22 nm & 21x ARM A53 x1000 22 nm	8670.3 s^{-1}	215.95 mm^2	1.88 W/mm^2
8x Smithfield 22 nm & 12x Quark x1000 32 nm	5769.75 s^{-1}	235.25 mm^2	1.17 W/mm^2
8x Smithfield 22 nm & 12x Quark x1000 22 nm	6363.84 s^{-1}	220.25 mm^2	1.31 W/mm^2

Assim como o questionário teórico, o relatório técnico também recebe uma nota numa escala de 1 a 5, de acordo com a aderência ao formato, correteude e completude:

1. **Inválido:** O relatório não contém as informações requeridas no modelo, ou não contém a análise discursiva sobre as plataformas simuladas e/ou encontradas na busca.
2. **Incorreto:** Apesar do relatório conter os valores especificados, a discussão analítica sobre as plataformas fez afirmações incorretas ou tirou conclusões indevidas.
3. **Parcial:** O relatório contém todas as informações requeridas no modelo. A análise discursiva contém exposições corretas e também exposições incorretas sobre os conceitos analisados.
4. **Incompleto:** O relatório contém todas as informações requeridas no modelo, e a análise discursiva não contém incorretudes, porém, alguns conceitos foram ignorados na análise e portanto o relatório não pode ser considerado excelente.

⁵<https://link.ufms.br/zZ83m>

5. **Completo:** O relatório contém todas as informações requeridas no modelo, e a análise discursiva foi completa e coerente.

5.1.5 *Pesquisa de Experiência de Usuário*

Um dos objetivos desse estudo era avaliar a experiência de usuário da ferramenta MultiExplorer, bem como investigar o impacto da ferramenta no aprendizado dos alunos.

Para realização da pesquisa, foi disponibilizado um formulário online na plataforma do Google Forms ⁶. A pesquisa de experiência de usuário foi organizada em quatro seções:

- **Introdução e consentimento:** Nessa seção é apresentada uma breve exposição a respeito da pesquisa, seus objetivos, métodos, e também é aqui que participante deve fornecer seu consentimento expresso para utilização dos dados obtidos na pesquisa. O experimento foi dispensado de passagem pelo comitê de ética por regulamento específico.
- **Perfil:** Nessa seção o participante responde a uma série de questionamentos sobre seu perfil demográfico e acadêmico, como: idade, gênero, curso de graduação, grau de instrução formal prévia sobre arquitetura de computadores, grau de instrução formal prévia sobre exploração de espaço de projeto, etc.
- **Facilidade de Uso:** Essa seção contém uma série de afirmações positivas a respeito da experiência de usuário no que diz respeito a facilidade de uso da ferramenta, e o participante deve expressar seu grau de concordância com essas afirmações. Ao fim da seção há uma questão aberta para que o participante expresse livremente sua percepção sobre a facilidade de uso da ferramenta.
- **Utilidade:** Essa seção contém uma série de afirmações positivas a respeito da experiência de usuário no que diz respeito a utilidade da ferramenta e seu impacto na experiência de aprendizado, e o participante deve expressar seu grau de concordância com essas afirmações. Ao fim da seção há duas questões abertas para que o participante expresse livremente sua percepção sobre a utilidade da ferramenta e sobre o impacto da ferramenta na experiência de aprendizado, respectivamente.

Nas seções sobre Facilidade de Uso e Utilidade da pesquisa, o participante expressava seu grau de concordância com as afirmações numa escala de cinco graus: 1 - Discordo Completamente; 2 - Discordo Parcialmente; 3 - Neutro; 4 - Concordo Parcialmente; 5 - Concordo Completamente.

⁶<https://link.ufms.br/PJnZL>

Nessas seções também haviam questões abertas sobre usabilidade, utilidade e aprendizado. As respostas foram analisadas de forma a classificar o teor do *feedback* numa escala de cinco graus:

1. Negativo: a análise discursiva do usuário mencionava apenas defeitos, dificuldades e possíveis melhorias, sem elogios ou pontos positivos.
2. Principalmente Negativo: a análise discurso do usuário incluía pontos positivos e elogios, mas era focada em defeitos, dificuldades e possíveis melhorias.
3. Neutro: a análise discursiva do usuário abordou defeitos, dificuldades, possíveis melhorias, pontos positivos e elogios de forma equilibrada.
4. Principalmente Positivo: a análise discursiva do usuário abordou principalmente pontos positivos e elogios.
5. Positivo: a análise discursiva do usuário abordou apenas pontos positivos e elogios.

5.2 Execução

A execução do experimento foi realizada através da oferta de um minicurso extra-curricular aos alunos da Faculdade de Computação (**FACOM**) da Universidade Federal do Mato Grosso do Sul (**UFMS**). Devido a duração das atividades, o minicurso foi dividido em dois encontros de três horas. No primeiro encontro os alunos assistiram à aula expositiva que tinha como propósito introduzi-los aos conceitos teóricos e à ferramenta MultiExplorer. Ao fim desse encontro o questionário teórico foi enviado para eles. Antes do segundo encontro um lembrete para responder o questionário também foi enviado. No segundo encontro eles realizaram a atividade prática de exploração com a ferramenta. O modelo do relatório técnico foi disponibilizado no início da atividade. Os alunos receberam um prazo de uma semana para entregar o relatório. Durante esse período, dois lembretes foram enviados.

O minicurso foi hospedado em um laboratório com 22 computadores, espaço cedido pela **FACOM**. A instalação e configuração da ferramenta MultiExplorer e seus pré-requisitos nas máquinas desse laboratório foi realizada previamente pelo autor deste trabalho, com assistência da equipe de tecnologia de informação da **FACOM**. Os alunos se inscreveram no minicurso através de um formulário disponibilizado via Google Forms ⁷. Duas edições do minicurso foram realizadas, ambas no mesmo laboratório supracitado. Houve um total

⁷<https://link.ufms.br/WoCZ1>

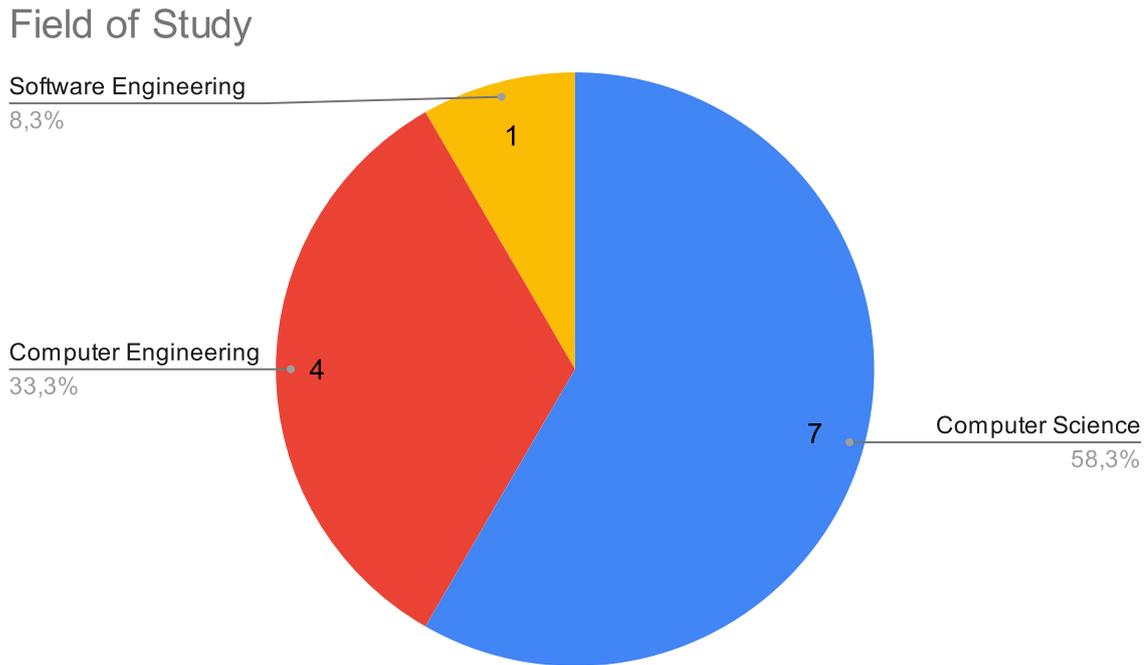


Figura 5.1: Distribuição dos alunos que preencheram a pesquisa de experiência de usuário entre os cursos de graduação.

de 40 inscrições, mas nem todos os alunos inscritos compareceram aos encontros, e dentre os presentes nem todos entregaram as atividades propostas. 18 alunos entregaram as respostas para o questionário teórico, 14 entregaram o relatório técnico, e 12 responderam a pesquisa sobre a experiência de usuário com a ferramenta. A maioria dos estudantes que preencheram a pesquisa de experiência de usuário eram de Ciências ou Engenharia da Computação, conforme pode ser visto no gráfico da Figura 5.1.

5.3 Resultados

Nas subseções a seguir são apresentados as medidas de aprendizagem dos alunos (questionário teórico e relatório técnico) e as medidas de Facilidade de Uso percebida (**PEOU**) e Utilidade Percebida (**PU**), bem como a análise desses dados, primeiro de forma isolada, e depois de forma conjunta, de forma a testar as hipóteses de correlação entre essas medidas.

5.3.1 Desempenho dos Alunos no Questionário Teórico

A Tabela 5.3 apresenta as notas obtidas pelos estudantes no questionário teórico. A nota final do questionário é calculada fazendo uma média entre as três questões ($Q = (q_1 + q_2 + q_3)/3$). As médias e os desvios padrão da amostra também foram incluídos. A Figura 5.2 apresenta os mesmos dados na forma

Tabela 5.3: Desempenho dos estudantes no questionário teórico, incluindo a nota final ($Q = (q_1 + q_2 + q_3)/3$), as médias (\bar{x}) e desvios padrão (s_2) de cada coluna.

Aluno	q_1	q_2	q_3	Q
1	4	3	4	3.67
2	3	3	3	3.00
3	2	4	4	3.33
4	3	3	3	3.00
5	4	3	5	4.00
6	3	2	4	3.00
7	4	4	3	3.67
8	5	5	5	5.00
9	1	5	5	3.67
10	3	4	5	4.00
11	4	4	4	4.00
12	4	4	4	4.00
13	4	4	5	4.33
14	5	4	5	4.67
15	3	3	4	3.33
16	4	3	4	3.67
17	4	3	4	3.67
18	2	3	3	2.67
\bar{x}	3.44	3.56	4.11	3.70
s_2	1.04	0.78	0.76	0.57

de histogramas.

Utilizando o teste de Shapiro, não foi possível afirmar que as notas em cada questão seguem uma distribuição normal, mas a nota final sim ($\alpha = 0.05$; p-value: $q_1 = 0.04122$; $q_2 = 0.0139$; $q_3 = 0.00243$; $Q = 0.5844$). Uma transformação exponencial sobre as notas da primeira questão (q_1) permitiu ao teste atingir um *p-value* que permite aceitar a hipótese nula de normalidade (*p-value* = 0.0603), e apesar de nenhuma transformação ter funcionado de forma análoga para as notas das questões dois e três (q_2 , q_3), isso pode indicar que uma amostra maior revelaria que a distribuição das notas em cada questão segue de fato uma distribuição normal.

Também foram calculados intervalos de confiança ($\alpha = 0.05$; $Pr = 95\%$) para as médias de q_1 ($CI = [2.93; 3.96]$), q_2 ($CI = [3.17; 3.95]$), q_3 ($CI = [3.73; 4.49]$) e Q ($CI = [3.4; 4]$). Usando testes de Wilcoxon, verificamos a hipótese de que as médias esperadas dessas questões para a população em geral de estudantes fossem iguais a três ($\bar{q}_1 = \bar{q}_2 = \bar{q}_3 = \bar{Q} = 3.0$) (q_1 : $V = 109$, p-value = 0.1045; q_2 : $V = 17$, p-value = 0.01503; q_3 : $V = 69$, p-value = 0.0007669). Como os valores de Q seguem uma distribuição normal, utilizamos um teste T ($t = 0$, $df = 17$, p-value = 0.0001216). Os resultados sugerem que as médias de q_2 , q_3 e Q estão acima de três. Isso poderia se dever a uma assimetria entre a

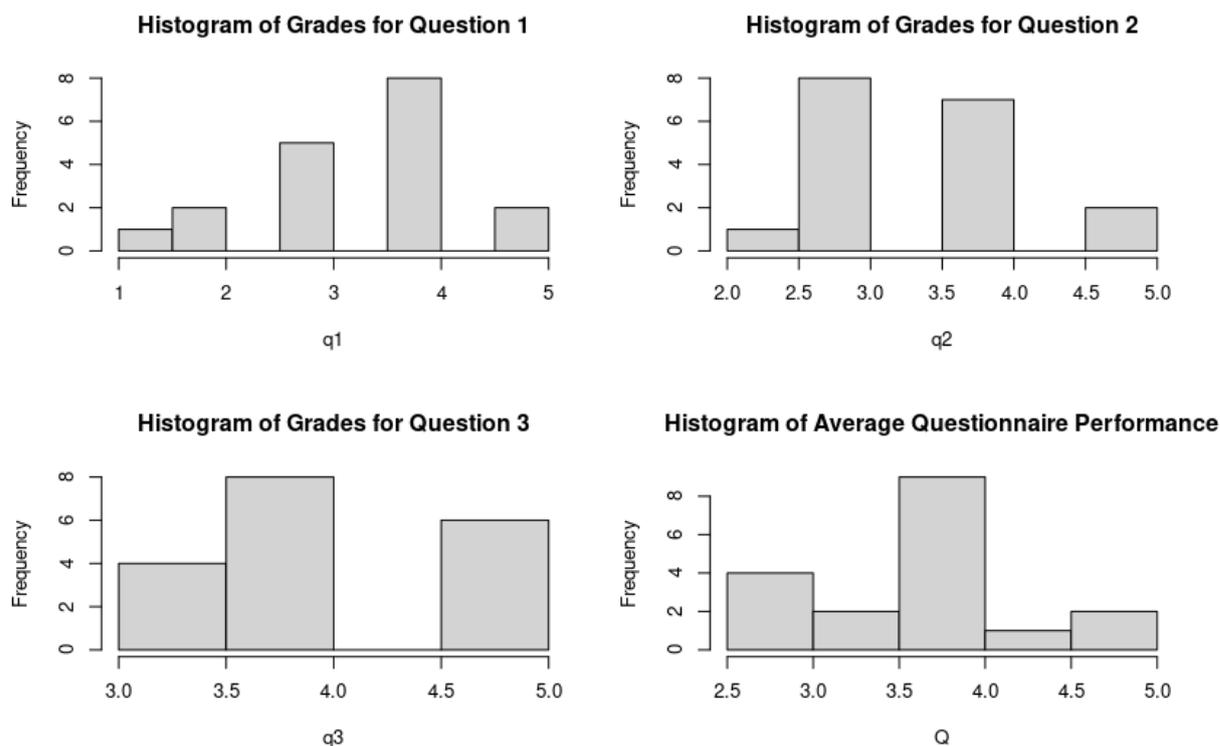


Figura 5.2: Desempenho dos alunos no questionário teórico na forma de histogramas.

complexidade das questões (q_2 e q_3 são mais fáceis que q_1), ou simplesmente a um desempenho diferenciado dos alunos da amostra. A média de Q poderia estar mais próxima de 3.5 do que 3.0 (T-test $\bar{Q} = 3.5, p - value = 0.169$).

Dado que as médias se encaixam na escala ordinal de avaliação entre “3 - **Parcial**” e “4 - **Incompleta**” é possível afirmar que os conceitos apresentados sobre os tópicos em questão (**DSE** e *dark silicon*) não estão fora do alcance de alunos de graduação, principalmente quando levamos em consideração a curta duração da intervenção de aprendizado. É possível que num período mais longo, a aprendizagem da maioria dos estudantes poderia ser classificada na escala num grau “4 - **Incompleta**” ou superior.

5.3.2 Desempenho dos Estudantes no Relatório Técnico

O desempenho dos alunos no relatório técnico pode ser encontrado na Tabela 5.4. A maior parte das notas está entre os graus três e quatro, conforme pode ser visto no histograma da Figura 5.3.

As notas do relatório técnico não parecem seguir uma distribuição normal (Shapiro Test $p - value = 0.0001478$). A média e moda obtidas ($\bar{r} = 3.71; Mo(r) = 4$) parecem indicar que o grau de aprendizado esperado da população de estudantes em geral pode ser maior do que três (Wilcoxon $\mu = 3$ p-value = 0.02334). O intervalo de confiança também aponta para a mesma direção ($\alpha = 0.05$; Pr

Tabela 5.4: Notas obtidas pelos estudantes no relatório técnico, incluindo a média (\bar{r}) e o desvio padrão (s_2).

Aluno	r
1	4
2	4
5	3
6	4
7	4
9	4
10	4
12	1
13	4
14	5
15	4
16	4
17	4
18	3
\bar{r}	3.71
s_2	0.91

= 95%; $CI = [3.19; 4.24]$). A média indica um grau de compreensão acima de “3 - Parcial” e próximo a “4 - Incompleto”, então podemos assumir que essa atividade também não está fora do alcance de alunos de graduação.

5.3.3 Resultados da Pesquisa de Experiência de Usuário

Tabela 5.5: Avaliação da Facilidade de Uso Percebida (**PEOU**) da ferramenta MultiExplorer por parte dos estudantes, incluindo as médias (\bar{x}) e desvios padrão (s_2).

Aluno	$peou_1$	$peou_2$	$peou_3$	$peou_4$	PEOU
1	5	5	5	5	5
2	5	4	4	5	4.5
3	4	4	4	3	3.75
5	4	4	4	5	4.25
9	4	4	5	5	4.5
10	1	2	5	3	2.75
11	4	4	4	4	4
12	4	4	5	5	4.5
13	4	4	4	4	4
15	5	5	5	5	5
16	3	4	5	4	4
18	4	4	4	4	4
\bar{x}	3.92	4	4.5	4.33	4.19
s_2	1.08	0.74	0.52	0.78	0.6

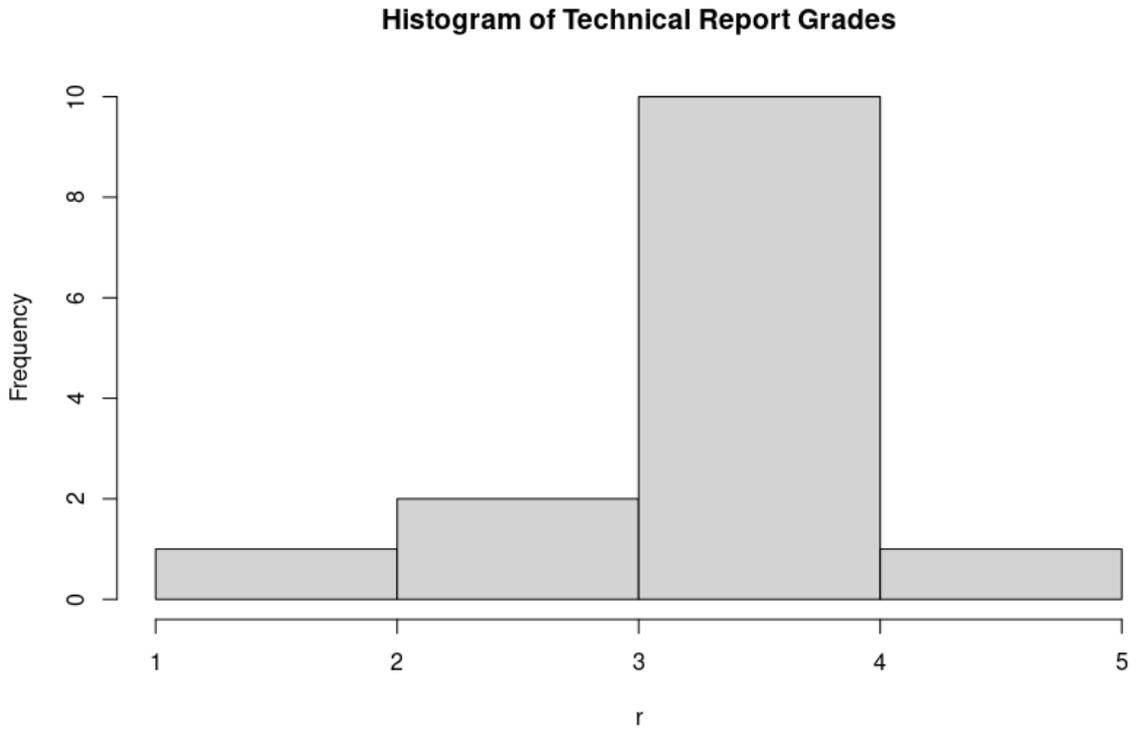


Figura 5.3: Histograma das notas obtidas pelos estudantes no relatório técnico.

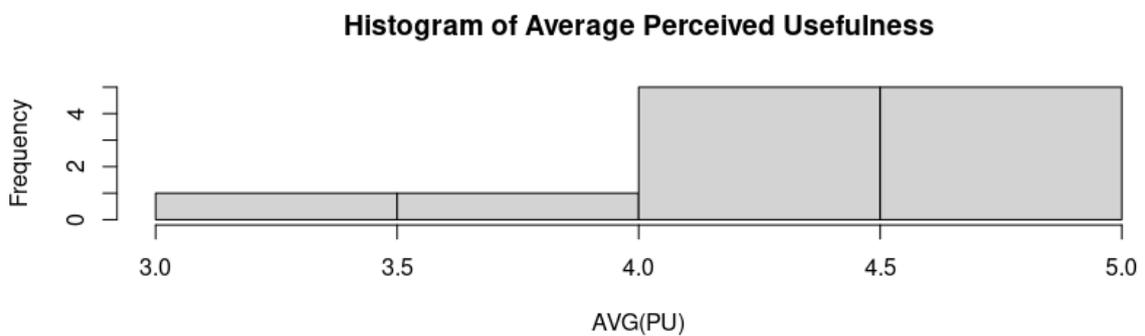
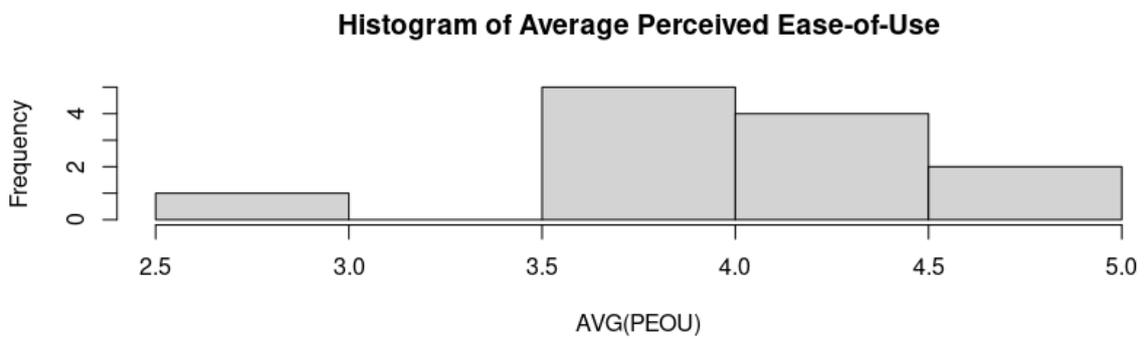


Figura 5.4: Histogramas da Facilidade de Uso Percebida (**PEOU**) e Utilidade Percebida (**PU**).

Tabela 5.6: Avaliação da Utilidade Percebida (**PU**) da ferramenta MultiExplorer por parte dos estudantes, incluindo as médias (\bar{x}) e desvios padrão (s_2).

Aluno	pu_1	pu_2	pu_3	pu_4	pu_5	PU
1	5	5	5	5	5	5
2	5	5	5	5	5	5
3	5	4	4	4	4	4.2
5	5	5	4	5	5	4.8
9	5	5	5	5	5	5
10	3	3	4	3	4	3.4
11	5	4	4	4	5	4.4
12	5	5	3	4	5	4.4
13	4	4	4	4	4	4
15	5	5	5	5	5	5
16	4	4	4	4	5	4.2
18	5	4	4	4	4	4.2
\bar{x}	4.67	4.42	4.25	4.33	4.67	4.47
s_2	0.65	0.67	0.62	0.65	0.49	0.51

De forma a mensurar quão acessível eram o uso e compreensão da ferramenta MultiExplorer, bem como o impacto da ferramenta em sua aprendizagem, convidamos os alunos a participar de uma pesquisa modelada a partir do **TAM**. A Tabela 5.5 contém os dados sobre Facilidade de Uso Percebida (**PEOU**) e a Tabela 5.6 os dados sobre Utilidade Percebida (**PU**).

Apesar dos valores dos *feedbacks* para cada item individual não seguirem uma distribuição normal, os valores finais de **PEOU** ($PEOU = (peou_1 + peou_2 + peou_3 + peou_4)/4$) e **PU** ($PEOU = (pu_1 + pu_2 + pu_3 + pu_4 + pu_5)/5$) estavam próximos da normalidade ($\alpha = 0.05$; Shapiro Test *p*-values: **PEOU** = 0.1453; **PU** = 0.09199). O *feedback* dos estudantes foi semelhante em ambos os quesitos, como indicado no gráfico de caixa da Figura 5.5. Isso foi confirmado por um teste de Wilcoxon (*p* - value = 0.2787). Os valores obtidos nesta amostras indicam que a média populacional para essa variáveis tem valor próximo ao grau quatro ou superior: **PEOU** 95%, $CI = [3.8; 4.57]$, Wilcoxon test $\mu = 4.0$ *p* - value = 0.2591; **PU** 95%, $CI = [4.14; 4.79]$, Wilcoxon test $\mu > 4.0$ *p* - value = 0.008803. Uma Análise de Variância (**Anova**) também revelou uma forte correlação entre **PEOU** e **PU** ($Pr(> F) = 5.111e - 05$). Esse resultado está alinhado com a literatura [47]. O *feedback* dos estudantes sobre Facilidade de Uso foi ligeiramente acima do grau “4 - Concordo Parcialmente”, indicando uma experiência de uso de regular a boa, e no quesito de Utilidade foi um pouco acima disso, indicando uma boa utilidade, talvez não distante de muito boa.

É de interesse avaliar individualmente o item pu_5 , pois se trata de um item cujo objeto específico é a percepção do estudante sobre o impacto da ferramenta na sua aprendizagem. Para realizar a análise, foi calculado o intervalo

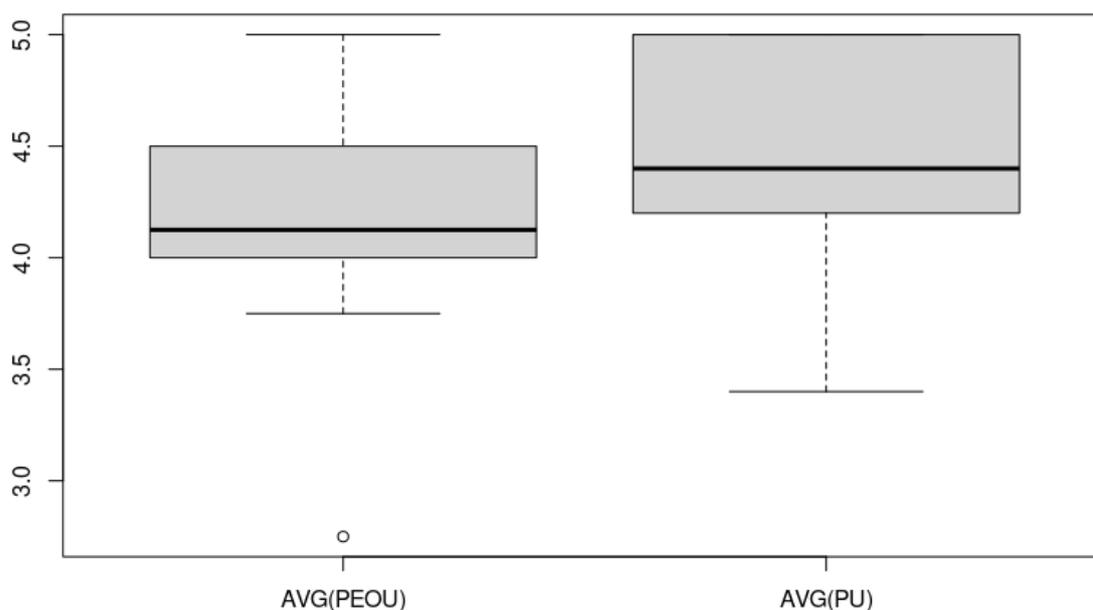


Figura 5.5: Gráficos de caixa dos valores de **PEOU** e **PU**.

de confiança ($\alpha = 0.05$; $Pr = 95\%$; $CI = [4.35; 4.98]$), e também foi realizado um teste de Wilcoxon ($\alpha = 0.05$; $\mu \leq 4$; $p\text{-value} : 0.0001254$), indicando que o grau do *feedback* foi acima de “4 - Concordo Parcialmente”. Isso indica uma percepção por parte dos alunos de que a ferramenta teve um bom impacto na sua aprendizagem.

5.3.4 Dados Qualitativos

Apesar de não ter sido realizada uma análise qualitativa dos dados, a pesquisa de usuário também recolheu respostas de três perguntas abertas sobre a experiência de usuário ($peou_5$, pu_6 e pu_7). As respostas dos alunos foram classificadas para apresentação (Tabela 5.7). A Tabela 5.8 apresenta exemplos de como as respostas foram classificadas.

5.3.5 Testagem de Correlações entre as Atividades

Como mencionado na Seção 5.1, existem algumas hipóteses interessantes a serem testadas a partir dos dados obtidos no estudo.

Um dessas hipóteses é se haveria diferença significativa entre a medida de aprendizagem antes e depois do uso da ferramenta, i.e., diferença significativa entre o desempenho no questionário teórico e o desempenho no relatório técnico. Porém como visto nos gráficos de caixa da Figura 5.6, não há dife-

Aluno	$peou_5$	pu_6	pu_7
1	Positivo	Positivo	Positivo
2	Principalmente Positivo	Principalmente Positivo	Principalmente Positivo
3	Neutro	Positivo	Principalmente Positivo
5	Principalmente Positivo	Positivo	Positivo
9	Neutro	Positivo	Principalmente Positivo
10	Negativo	Negativo	Principalmente Positivo
11	Neutro	Positivo	Neutro
12	Neutro	Positivo	Positivo
13	Neutro	Positivo	Principalmente Positivo
15	Positivo	Positivo	Positivo
16	Principalmente Positivo	Positivo	Positivo
18	Neutro	Positivo	Positivo
Média	3,42 (Neutro)	4,58 (Positivo)	4,42 (Principalmente Positivo)
σ	1,08	1,16	0,67

Tabela 5.7: Sumário do *feedback* subjetivo dos alunos sobre a ferramenta. A atitude dos alunos parece ter sido neutra em relação a usabilidade, mas positiva no quesito de utilidade.

Negativo	Talvez pelo pequeno escopo de opções e restrições que pudéssemos usar, a ferramenta aparentemente aplicou apenas o método guloso para escolher os resultados. Não consigo sequer dizer se o dark silicon foi levado em conta, mas parece não ter sido.
Principalmente Negativo	N/A
Neutro	Como não tinha tanta familiaridade com processadores, de certa forma não foi intuitivo a utilização da ferramenta, mas após ajuda consegui utilizar sem problema.
Principalmente Positivo	Pude compreender melhor como funciona essa área da computação, sua finalidade e seu processo. Simulando alguns processadores do mercado pude ver maneiras mais eficientes para montar processadores mais antigos e alternativas para alguns mais novos.
Positivo	Antes de fazer parte deste minicurso, não possuía conhecimento nenhum sobre a exploração de espaço de projeto, dark silicon e nenhum outro conceito apresentado. Com a disponibilização da ferramenta, consegui adquirir todos esses conhecimentos muito mais facilmente, mesmo sem conhecimento prévio algum.

Tabela 5.8: Essa tabela apresenta um exemplo da maneira que os *feedbacks* das questões abertas da pesquisa de experiência de usuário foram classificados para apresentação na Tabela 5.7.

relatório técnico? Para averiguar esse efeito agrupamos os dados das duas atividades (no caso dos alunos que entregaram ambas), os dados resultantes estão na Tabela 5.9. Analisamos de forma semelhante o efeito da performance no questionário teórico sobre a experiência de usuário (dados da Tabela 5.10). Os efeitos foram testados via Análise de Variância (**Anova**) ou testes Kruskal-Wallis (nos casos em que os resíduos dos modelos violaram gravemente a normalidade):

- $r \leftarrow Q$ p-value = 0.1178 (Kruskal-Wallis)
- $PEOU \leftarrow Q$ p-value = 0.5028 (Anova)
- $PU \leftarrow Q$ p-value = 0.3616 (Anova)

Tabela 5.10: Dados utilizados para testar o efeito do desempenho no questionário teórico sobre a experiência de usuário.

Aluno	Q	$PEOU$	PU
1	3.67	5	5
2	3	4.5	5
3	3.33	3.75	4.2
5	4	4.25	4.8
9	3.67	4.5	5
10	4	2.75	3.4
11	4	4	4.4
12	4	4.5	4.4
13	4.33	4	4
15	3.33	5	5
16	3.67	4	4.2
18	2.67	4	4.2

Utilizamos uma abordagem semelhante para averiguar os efeitos da experiência de usuário sobre o desempenho no relatório técnico, os dados podem ser vistos na Tabela 5.11. A ausência de efeito foi apontada por testes de Kruskal-Wallis ($r \tilde{PEOU}$ p -value = 0.6005; $r \tilde{PU}$ p -value = 0.1984).

5.4 Conclusões

De acordo com a análise de dados há uma forte correlação entre **PEOU** e **PU**. Essa conclusão não foi surpreendente, já que está alinhada com os resultados de muitos estudos que utilizaram o **TAM** [47]. Além disso, as medidas de aprendizagem e da experiência de usuário apontam que a ferramenta é adequada para utilização no ensino a nível de graduação, e que os tópicos abordados estão ao alcance dos estudantes. No geral, a usabilidade da ferramenta está entre regular e boa, e apesar de ser passível de críticas e possíveis

Tabela 5.11: Dados utilizados para testar os efeitos da experiência de usuário sobre o desempenho no relatório técnico.

Aluno	<i>PEOU</i>	<i>PU</i>	r
1	5	5	4
2	4.5	5	4
5	4.25	4.8	3
9	4.5	5	4
10	2.75	3.4	4
12	4.5	4.4	1
13	4	4	4
15	5	5	4
16	4	4.2	4
18	4	4.2	3

melhorias, os estudantes que participaram do estudo parecem convencidos dos benefícios da ferramenta para sua experiência de aprendizado.

Não foi possível estabelecer nenhum efeito entre da performance no questionário teórico sobre a performance no relatório técnico ou sobre a experiência de usuário. Também não foi possível encontrar nenhum efeito semelhante entre a experiência de usuário e a performance no relatório técnico. Devemos considerar que as amostras foram pequenas, e os valores de muitas das variáveis envolvidas na análise não seguiam uma distribuição normal, limitando os testes estatísticos que podiam ser utilizados, e também o poder desses testes. Sendo assim é desaconselhável o descarte definitivo dessas hipóteses.

5.5 Limitações e Ameaças à Validade

Existem algumas limitações e ameaças a validade do estudo identificadas pelo autor do trabalho. A primeira delas certamente é o pequeno tamanho da amostra. Apesar da coleta de dados demográficos e acadêmicos sobre os alunos, não foi possível estabelecer se os alunos constituem uma amostra representativa da população de alunos de graduação em programas de Ciência e Engenharia de Computação em geral. A ampliação da amostra via replicação poderia fortalecer as conclusões.

Outro ponto considerado após a conclusão do estudo foi o estabelecimento de critérios de avaliação mais objetivos, e um novo formato de avaliação. O questionário teórico foi composto por apenas três questões, de texto relativamente curto, abordando temas abrangentes. Um questionário que tivesse questões mais específicas permitiria que fosse estabelecido um gabarito e critérios mais detalhados de correção, facilitando a replicação. Outra questão a se considerar é a ausência de análise qualitativa. A análise qualitativa poderia permitir uma visão mais completa da experiência dos usuários com a

ferramenta.

Entretanto, dado como os instrumentos do estudo foram bem documentados, a ferramenta está livremente disponível, e os dados quantitativos também foram apresentados na íntegra, apesar das conclusões serem questionáveis, não podemos descartar a validade do estudo como um todo. Todas as conclusões apresentadas estão amparadas nos dados que foram obtidos, e nas análises estatísticas realizadas. Uma expansão da amostra serviria para fortalecer as conclusões, ou poderia indicar a incorretude das mesmas, mas isso apenas indicaria a inadequação do estudo no que tange ao tamanho da amostra, e não a invalidade dos instrumentos e métodos utilizados.

Considerações Finais

A importância estratégica do setor industrial de microeletrônicos para o desenvolvimento econômico do país, bem como as recomendações da ACM, apontam para a relevância do tópico de Exploração de Espaço de Projeto como um componente curricular nos programas de graduação em Ciência da Computação, e sobretudo para os de Engenharia de Computação. Um aumento na frequência e profundidade da abordagem desse tópico nesses programas poderia enriquecer a qualificação dos futuros profissionais da área.

Não foi possível encontrar referências diretas ao tópico de Exploração de Espaço de Projeto nos projetos pedagógicos analisados na pesquisa, bem como também não foi encontrada uma ferramenta gratuita e *open source* que permitisse desenvolver atividades de ensino sobre DSE e explorar também o tópico de *Dark Silicon*, portanto este trabalho propôs a utilização da ferramenta MultiExplorer para esse fim.

Para isso, a ferramenta precisou antes passar por uma etapa de desenvolvimento de software, para implementar uma interface gráfica de usuário e outras melhorias. Essa etapa foi iniciada com uma atividade de prototipação, executada segundo a metodologia *sprint* [2]. Na sequência a implementação foi realizada em Python por um único desenvolvedor. A arquitetura de software foi devidamente documentada.

Um estudo de caso foi realizado utilizando a ferramenta como recurso educacional no contexto de um minicurso sobre Exploração de Espaço de Projeto ciente de *Dark Silicon*, e os resultados indicam que os alunos de graduação são capazes de operar a ferramenta, e alcançar um bom aproveitamento nas atividades avaliativas. Os alunos em geral concordaram que a ferramenta teve um impacto positivo em sua experiência de aprendizado.

Por meio do estudo de caso apresentado, avaliamos o potencial do MultiExplorer como uma ferramenta para ensino do tópico de Exploração de Espaço de Projeto aplicada a Arquitetura de Computadores. As análises do desempenho dos alunos e das avaliações da experiência de uso indicam que a ferramenta é adequada para uso no contexto de ensino e aprendizado a nível de graduação na área de computação. É importante considerar que o tamanho da amostra pode limitar parcialmente a confiabilidade das descobertas, e que a ferramenta ainda carece de algumas melhorias.

Por exemplo, no princípio da atividade prática alguns estudantes manifestaram dúvidas a respeito dos significados de certos indicadores informados pela ferramenta, como performance, densidade de potência, **D.S**, etc., mas essas questões foram esclarecidas sem muita dificuldade. Esses indicadores foram abordados na apresentação da ferramenta durante a introdução teórica, mas é possível que muitos alunos não tenham conseguido reter a informação. A possibilidade de obter essa informação diretamente da ferramenta pode vir ser útil para os usuários.

Trabalhos futuros também poderiam explorar o uso de novas versões da ferramenta, ou outras estratégias para sua aplicação, e.g.: utilizar a ferramenta em atividades de disciplinas obrigatórias ou eletivas na graduação, já que o curto espaço de tempo para realização de atividades e avaliações em um minicurso pode restringir a aprendizagem dos alunos.

O planejamento e execução de todo o trabalho contaram com a devida fundamentação teórica, e o desenvolvimento e avaliação da ferramenta como recurso educacional seguiram metodologias e técnicas bem estabelecidas, que foram rigorosamente executadas e documentadas.

Os esforços e resultados deste trabalho também foram parcialmente apresentados no artigo “*A Tool for Teaching and Learning of Design Space Exploration on Processor Architectures*”, que foi apresentado na conferência *2023 IEEE ASEE Frontiers in Education*¹, em College Station-Texas, em outubro de 2023, com apoio financeiro do Programa de Pós-graduação da Faculdade de Computação e Edital Mulheres na Ciência da UFMS.

¹<https://2023.fie-conference.org/>

Referências Bibliográficas

- [1] Giuseppe Ascia, Vincenzo Catania, Alessandro Di Nuovo, Maurizio Palesi, and Davide Patti. Efficient design space exploration for application specific systems-on-a-chip. *Journal of Systems Architecture*, 53(10):733–750, 2007. Citado nas páginas xiii, 7, e 8.
- [2] "John Knapp, John Zeratsky, and Braden Kowitz". *"Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days"*. "Simon & Schuster", 2016. Citado nas páginas xiii, 14, 15, 26, e 67.
- [3] Zai Jian Jia, Andy D. Pimentel, Mark Thompson, Tomás Bautista, and Antonio Núñez. Nasa: A generic infrastructure for system-level mp-soc design space exploration. In *2010 8th IEEE Workshop on Embedded Systems for Real-Time Multimedia*, pages 41–50, 2010. Citado na página 1.
- [4] Eunsuk Kang, Ethan Jackson, and Wolfram Schulte. An approach for effective design space exploration. In Radu Calinescu and Ethan Jackson, editors, *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*, pages 33–54, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. Citado na página 1.
- [5] Oscar S. Silva Filho. The ic brazil ecosystem. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems, MEDES '13*, page 270–277, New York, NY, USA, 2013. Association for Computing Machinery. Citado na página 1.
- [6] Bosko Nikolic, Zaharije Radivojevic, Jovan Djordjevic, and Veljko Milutinovic. A survey and evaluation of simulators suitable for teaching courses

in computer architecture and organization. *IEEE Transactions on Education*, 52(4):449–458, 2009. Citado na página 3.

- [7] Giuseppe Ascia, Vincenzo Catania, Alessandro G Di Nuovo, Maurizio Palesi, and Davide Patti. Performance evaluation of efficient multi-objective evolutionary algorithms for design space exploration of embedded computer systems. *Applied Soft Computing*, 11(1):382–398, 2011. Citado na página 3.
- [8] Rodrigo Devigo, Liana Duenha, Rodolfo Azevedo, and Ricardo Santos. Multiexplorer: A tool set for multicore system-on-chip design exploration. In *Proceedings of the 26th ASAP*, pages 160–161. IEEE, 2015. Citado na página 4.
- [9] Vittorio Zaccaria, Gianluca Palermo, Fabrizio Castro, Cristina Silvano, and Giovanni Mariani. Multicube explorer: An open source framework for design space exploration of chip multi-processors. In *23rd International Conference on Architecture of Computing Systems (ARCS)*, pages 1–7. VDE, 2010. Citado na página 7.
- [10] Eunsuk Kang, Ethan Jackson, and Wolfram Schulte. An approach for effective design space exploration. In *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*, pages 33–54. Springer, 2011. Citado na página 7.
- [11] Giuseppe Ascia, Vincenzo Catania, Alessandro G Di Nuovo, Maurizio Palesi, and Davide Patti. Performance evaluation of efficient multi-objective evolutionary algorithms for design space exploration of embedded computer systems. *Applied Soft Computing*, 11(1):382–398, 2011. Citado na página 8.
- [12] Robert R Schaller. Moore’s law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997. Citado na página 9.
- [13] R. H. Dennard, Fritz H. Gaensslen, Hwa nien Yu, V. Leo Rideout, Ernest Bassous, Andre, and R. Leblanc. Design of ion-implanted mosfets with very small physical dimensions. *IEEE Journal of Solid-Circuits*, pages 256–267, 1974. Citado na página 9.
- [14] Ganesh Venkatesh, Jack Sampson, and et. al. Goulding. Conservation cores: reducing the energy of mature computations. *SIGARCH Computuer Architeture News*, 38(1):205–218, 2010. Citado na página 9.

- [15] IEEE-Institute of Electrical and Electronic Engineers. IRDS - International Roadmap for Devices and Systems. <https://irds.ieee.org>, Acessado em Abril-2019 2018. Citado na página 9.
- [16] Tony Santos, Ana Silva, Liana Duenha, Ricardo Santos, Edward Moreno, and Rodolfo Azevedo. On the dark silicon automatic evaluation on multi-core processors. In *2016 28th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 166–173. IEEE, 2016. Citado nas páginas 9, 11, e 13.
- [17] Hadi Esmaeilzadeh, Emily Blem, Renee St Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 365–376. IEEE, 2011. Citado na página 9.
- [18] Michael B Taylor. Is dark silicon useful? harnessing the four horsemen of the coming dark silicon apocalypse. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pages 1131–1136. IEEE, 2012. Citado nas páginas 9 e 10.
- [19] Ronald G Dreslinski, Michael Wieckowski, David Blaauw, Dennis Sylvester, and Trevor Mudge. Near-threshold computing: Reclaiming moore’s law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266, 2010. Citado na página 10.
- [20] Nikos Hardavellas, Michael Ferdman, Babak Falsafi, and Anastasia Ailamaki. Toward dark silicon in servers. *IEEE Micro*, 31(4):6–15, 2011. Citado na página 10.
- [21] Bingfeng Mei, Serge Vernalde, Diederik Verkest, Hugo De Man, and Rudy Lauwereins. Adres: An architecture with tightly coupled vliw processor and coarse-grained reconfigurable matrix. In *International Conference on Field Programmable Logic and Applications*, pages 61–70. Springer, 2003. Citado na página 10.
- [22] Arun Raghavan, Yixin Luo, Anuj Chandawalla, Marios Papaefthymiou, Kevin Pipe, Thomas Wenisch, and Milo Martin. Computational sprinting. In *Proceedings of IEEE 18th International Symposium on High Performance Computer Architecture (HPCA)*, pages 1–12. IEEE, 2012. Citado na página 10.
- [23] Efi Rotem, Alon Naveh, Doron Rajwan, Avinash Ananthakrishnan, and Eli Weissmann. Power management architecture of the 2nd generation intel® core microarchitecture, formerly codenamed sandy bridge. In *Hot*

Chips 23 Symposium (HCS), 2011 IEEE, pages 1–33. IEEE, 2011. Citado na página 10.

- [24] James Charles, Preet Jassi, Narayan S Ananth, Abbas Sadat, and Alexandra Fedorova. Evaluation of the intel® core™ i7 turbo boost feature. In *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pages 188–197. IEEE, 2009. Citado na página 10.
- [25] Muhammad Shafique and Siddharth Garg. Computing in the dark silicon era: Current trends and research challenges. *IEEE Design Test*, 34(2):8–23, 2017. Citado na página 10.
- [26] Daniela Catelan, Ricardo Santos, and Liana Duenha. Evaluation and characterization of approximate arithmetic circuits. *Concurrency and Computation: Practice and Experience*, n/a(n/a):e6865. Citado na página 10.
- [27] Indrani Paul, Srilatha Manne, Manish Arora, W. Lloyd Bircher, and Sudhakar Yalamanchili. Cooperative boosting: Needy versus greedy power management. 41(3), 2013. Citado na página 10.
- [28] Rhayssa Sonohata, Danilo Christi A. Arigoni, Eraldo Rezende Fernandes, Ricardo Ribeiro dos Santos, and Liana Dessandre Duenha. Performance predictors for graphics processing units applied to dark-silicon-aware design space exploration. *Concurrency and Computation: Practice and Experience*, n/a(n/a):e6877. Citado na página 10.
- [29] L. Wang and Kevin Skadron. Implications of the power wall: Dim cores and reconfigurable logic. *Micro, IEEE*, 33:40–48, 09 2013. Citado na página 10.
- [30] Jörg Henkel, Heba Khdr, Santiago Pagani, and Muhammad Shafique. New trends in dark silicon. In *Proceedings of the 52nd Annual DAC*, page 119. ACM, 2015. Citado na página 10.
- [31] Rodrigo Devigo, Liana Duenha, Rodolfo Azevedo, and Ricardo Santos. Multiexplorer: A tool set for multicore system-on-chip design exploration. In *Proceedings of the 26th ASAP*, pages 160–161. IEEE, 2015. Citado na página 11.
- [32] Heirman, Wim and Carlson, Trevor and Eeckhout, Lieven. Sniper: scalable and accurate parallel multi-core simulation. In *8th International Summer School on Advanced Computer Architecture and Compilation for*

High-Performance and Embedded Systems, Abstracts, pages 91–94. High-Performance and Embedded Architecture and Compilation Network of Excellence (HiPEAC), 2012. Citado na página 12.

- [33] Ali Bakhoda, George L Yuan, Wilson WL Fung, Henry Wong, and Tor M Aamodt. Analyzing cuda workloads using a detailed gpu simulator. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 163–174. IEEE, 2009. Citado na página 12.
- [34] Steven Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Singh, and Anoop Gupta. The splash-2 programs: Characterization and methodological considerations. *ACM SIGARCH Computer Architecture News*, 23:24–36, 1995. Citado na página 12.
- [35] Christian Bienia, Sanjeev Kumar, Jaswinder Singh, and Kai Li. The par-sec benchmark suite: Characterization and architectural implications. In *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, pages 72–81. ACM, 2008. Citado na página 12.
- [36] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020. Citado na página 12.
- [37] Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W. Sheffer, Sang-Ha Lee, and Kevin Skadron. Rodinia: A benchmark suite for heterogeneous computing. In *2009 IEEE International Symposium on Workload Characterization (IISWC)*, pages 44–54, 2009. Citado na página 12.
- [38] Sheng Li, Jung Ahn, Richard Strong, Jay Brockman, Dean Tullsen, and Norman Jouppi. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480. IEEE, 2009. Citado na página 12.
- [39] Ricardo Santos, Liana Duenha, Ana Caroline Silva, Matheus Sousa, Luiz Augusto Tedesco, João Carlos Melgarejo, Tony Santos, Rodolfo Azevedo, and Edward Moreno. Dark-silicon aware design space exploration. *Journal of Parallel and Distributed Computing*, 120:295–306, 2018. Citado na página 13.
- [40] M. T. d. Santos, R. Sonohata, C. Krebs, D. Segovia, R. Santos, and L. Duenha. Performance models for heterogeneous systems applied to the dark

silicon-aware design space exploration. In *2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 9–16, 2019. Citado na página 13.

- [41] Liana Duenha, Rhayssa Sonohata, Danillo Arigoni, and Ricardo Santos. Preditor de desempenho de gpus aplicado à exploração do espaço de projetos ciente de dark silicon. In *Anais do XXI Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 299–310. SBC, 2020. Citado na página 13.
- [42] Daniela Catelan, Ricardo Santos, and Liana Duenha. Accuracy and physical characterization of approximate arithmetic circuits. In *Anais do XXI Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 143–154. SBC, 2020. Citado na página 13.
- [43] Danillo Christi A Arigoni, Ricardo Ribeiro dos Santos, and Liana Dessandre Duenha. Exploração do espaço de projetos para alocação de recursos em nuvem. In *Anais da IV Escola Regional de Alto Desempenho do Centro-Oeste*, pages 12–16. SBC, 2021. Citado na página 13.
- [44] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 206–213, New York, NY, USA, 1993. Association for Computing Machinery. Citado nas páginas 16 e 26.
- [45] Fred D Davis. *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. PhD thesis, Massachusetts Institute of Technology, 1985. Citado na página 17.
- [46] Nikola Marangunić and Andrina Granić. Technology acceptance model: a literature review from 1986 to 2013. *Universal access in the information society*, 14:81–95, 2015. Citado na página 17.
- [47] Younghwa Lee, Kenneth A Kozar, and Kai RT Larsen. The technology acceptance model: Past, present, and future. *Communications of the Association for information systems*, 12(1):50, 2003. Citado nas páginas 17, 60, e 64.
- [48] William R King and Jun He. A meta-analysis of the technology acceptance model. *Information & management*, 43(6):740–755, 2006. Citado na página 17.

- [49] Andrina Granić and Nikola Marangunić. Technology acceptance model in educational context: A systematic literature review. *British Journal of Educational Technology*, 50, 07 2019. Citado nas páginas 17 e 48.
- [50] Azam Beg and Walid Ibrahim. Perfpred: A web-based tool for exploring computer architecture design space. *Computer Applications in Engineering Education*, 17:305 – 313, 09 2009. Citado na página 18.
- [51] Sandro Neves Soares and Flávio Rech Wagner. Design space exploration of embedded processors in computer architecture education using t&d-bench. In *Proceedings. Frontiers in Education. 36th Annual Conference*, pages 19–24. IEEE, 2006. Citado na página 19.
- [52] G Esmeraldo and Edson Barbosa Lisboa. Uma ferramenta para exploração do ensino de organização e arquitetura de computadores. *International Journal of Computer Architecture Education*, 6(1):68–75, 2017. Citado na página 19.
- [53] Kenneth Vollmar and Pete Sanderson. Mars: an education-oriented mips assembly language simulator. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 239–243, 2006. Citado na página 19.
- [54] Liana Duenha and Rodolfo Azevedo. Utilização dos simuladores do mp-socbench para o ensino e aprendizagem de arquitetura de computadores. *International Journal of Computer Architecture Education (IJCAE)*, 5(1):26–31, 2016. Citado na página 20.
- [55] Liana Duenha, Henrique Almeida, Marcelo Guedes, Matheus Boy, , and Rodolfo Azevedo. Mpsocbench: A toolset for mpsoc system level evaluation. In *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, July 2014. Citado na página 20.
- [56] Liana Duenha, F Crominski, MT Santos, and R Ribeiro. Avaliação de preditores de desvios por meio de simuladores como parte do processo de ensino e aprendizagem de arquitetura de computadores. *International Journal of Computer Architecture Education (IJCAE)*, 6(1):1–9, 2017. Citado na página 20.
- [57] Fabrice Bellard. Qemu internals. *Jul*, 22:1–7, 2006. Citado na página 20.
- [58] Qemu. <https://www.qemu.org/>. Visitado em 04 de janeiro de 2022. Citado na página 21.

- [59] Avadh Patel, Furat Afram, Shunfei Chen, and Kanad Ghose. MARSSx86: A Full System Simulator for x86 CPUs. In *Design Automation Conference 2011 (DAC'11)*, 2011. Citado na página 21.
- [60] Jianxing Wang, Pooja Roy, Weng-Fai Wong, Xiuyuan Bi, and Hai Li. Optimizing mlc-based stt-ram caches by dynamic block size reconfiguration. In *2014 IEEE 32nd International Conference on Computer Design (ICCD)*, pages 133–138, 2014. Citado na página 21.
- [61] Antonio C Guimarães, Edson Borin, and Diego F Aranha. Extensão do conjunto de instruções para implementação segura de x25519. Citado na página 21.
- [62] Erion Çano. Simulation of l2 cache separation impact in cpu performance. In *International Conference on ICT Innovations*, pages 93–103. Springer, 2014. Citado na página 21.
- [63] Michel A Kinsky, Michael Pellauer, and Srinivas Devadas. Heracles: a tool for fast rtl-based design space exploration of multicore processors. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, pages 125–134, 2013. Citado na página 21.
- [64] Prateek Sharma, Lucas Chaufournier, Prashant Shenoy, and YC Tay. Containers and virtual machines at scale: A comparative study. In *Proceedings of the 17th international middleware conference*, pages 1–13, 2016. Citado na página 25.
- [65] Trevor E Carlson, Wim Heirman, and Lieven Eeckhout. Sniper: exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 52. ACM, 2011. Citado na página 25.
- [66] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlisside. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley Longman, Inc, 1998. Citado na página 29.