



Programa de Pós-Graduação em

Computação Aplicada

Mestrado/Doutorado Acadêmico

Juliano Paulo Menzen

4Experts

Uma abordagem de recomendação de tarefas de desenvolvimento de software sensível a dados biométricos utilizando aprendizagem de máquina

São Leopoldo, 2020

Juliano Paulo Menzen

4EXPERTS:

Uma abordagem de recomendação de tarefas de desenvolvimento de software sensível a dados biométricos utilizando aprendizagem de máquina

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Kleinner Silva Farias de Oliveira

São Leopoldo
2020

M551f Menzen, Juliano Paulo.
4EXPERTS: uma abordagem de recomendação de tarefas de desenvolvimento de software sensível a dados biométricos utilizando aprendizagem de máquina / Juliano Paulo Menzen. – 2020.
189 f. : il. color. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2020.

“Orientador: Prof. Dr. Kleinner Silva Farias de Oliveira.”

1. Engenharia de software. 2. Eletroencefalografia. 3. Software - Desenvolvimento. 4. Aprendizado do computador. 5. Produtividade. I. Título.

CDU 004.41

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

Aos meus pais, por me oportunizarem a chegada até aqui.

AGRADECIMENTOS

O meu sucesso durante esta jornada de estudos e produção da dissertação foi possível graças ao auxílio de diversas pessoas as quais agradeço muito. Entretanto, faço um agradecimento especial a algumas delas, sem as quais tudo isso não teria sido possível.

Inicialmente agradeço a Deus por tudo e principalmente por ter colocado em meu caminho as pessoas certas no momento certo convertendo este objetivo de vida em realidade. Agradeço a minha esposa Rachel Bombardelli por ter me suportado e confortado durante todo este tempo, tendo paciência e me auxiliando em tudo o que precisei adotar durante este processo. Agradeço aos meus familiares e principalmente aos meus pais Natal Francisco Menzen e Naídes Fontanive Menzen por sempre terem me incentivado a estudar e a ser uma pessoa melhor. Faço um agradecimento especial ao meu orientador, Dr. Kleinner Silva Farias de Oliveira, por ter acreditado no meu potencial, me desafiado, orientado e auxiliado em um tema desafiador, tornando possível esta dissertação com resultados relevantes e motivadores. Agradeço também aos meus colegas e amigos Lucian Gonçalves e Vinicius Bischoff pela sua amizade, auxílio e motivação durante a dissertação. Agradeço a pró-reitora acadêmica do Centro Universitário UNIFTEC Dr. Débora Frizzo e a coordenadora do curso de Análise e Desenvolvimento de Sistemas Ms. Neiva Kuyven por terem aberto as portas da instituição, possibilitando a realização dos experimentos que tive a necessidade de executar para atingir os objetivos do meu estudo e por fim também agradeço ao meu amigo e sócio Renan Tedesco por ter me apoiado durante todo este período e auxiliando muito durante minhas ausências.

RESUMO

Estudos recentes relatam que o crescimento da necessidade de software é muito maior do que a capacidade de construção deles. Esta quantidade crescente de demanda por software faz com que esforços sejam direcionados em estudos e técnicas que possam melhorar a produtividade e o processo de construção de sistemas. Existem estudos em Engenharia de Software relacionadas à análise da produtividade na produção de software que avaliam aspectos como a produtividade associada às linhas de código em relação ao tempo, o impacto da organização sobre a produtividade, ferramentas e abordagens que auxiliam os desenvolvedores a melhorar o seu processo de desenvolvimento de sistemas. Porém, tais estudos não consideram uma análise do principal ator no processo de desenvolvimento de software, que é o próprio desenvolvedor. Por conta disso, este trabalho propõe o 4Experts, uma abordagem de recomendação de tarefas de desenvolvimento de software sensível aos dados biométricos do desenvolvedor utilizando aprendizagem de máquina. O 4Experts, baseia-se na análise por aprendizado de máquina dos dados biométricos produzidos pelo cérebro e coletados através de um equipamento de eletroencefalograma, com o propósito de prever em tempo real se a tarefa que está sendo executada pelo desenvolvedor será realizada com corretude, recomendando que as próximas tarefas a serem realizadas pelo desenvolvedor sejam adequadas a sua capacidade de resolução, caso a tarefa atual apresente indicadores de que foi executada incorretamente. Para viabilizar esta abordagem foram realizados dois estudos experimentais, onde o primeiro estudo procurou explorar os diferentes aspectos da carga mental gerada por 40 participantes enquanto estes atuavam em tarefas de programação. Por sua vez, o segundo experimento serviu para avaliar a abordagem 4Experts, no qual 20 participantes foram avaliados enquanto realizavam tarefas de programação. Os resultados do experimento sugerem que com o uso da abordagem 4Experts houve um aumento de 38,64% na corretude das tarefas e 49,02% de redução de esforço em relação ao tempo despendido em cada tarefa quando comparado a desenvolvedores que não foram submetidos a abordagem.

Palavras-chave: Engenharia de Software. Eletroencefalografia. Aprendizado de Máquina. Produtividade. Tarefas. Desenvolvimento de Software. EEG. Desenvolvedor.

ABSTRACT

Recent studies report that the growing need for software is greater than their building capacity. This growing amount for software demand means that efforts are directed at studies and techniques that can improve productivity and the process of building systems. There are studies in Software Engineering related to the analysis of productivity in software production that evaluate aspects such as the productivity associated with lines of code in relation to time, the impact of the organization on productivity, tools and approaches that help developers to improve their development process. However, such studies do not consider an analysis of the main actor in the software development process, which is the developer himself. By this, this study proposes 4Experts, an approach for recommending software development tasks sensitive to the developer's biometric data using machine learning. The 4Experts, is based on the analysis by machine learning of the biometric data produced by the brain and collected through an electroencephalogram equipment, with the purpose of predicting in real time whether the task being performed by the developer will be performed correctly, recommending that next task to be performed by the developer will be easier than the last one, if the current task presents indicators that it was performed with failure. To make this approach feasible, two experimental studies were carried out, where the first study explore the different aspects of the cognitive load generated by 40 participants while they performing programming tasks. In turn, the second experimental study served to validate the 4Experts approach, in which 20 participants were assessed while performing programming tasks. The results of the experiment suggest that the use of the 4Experts approach there was an increase of 38,64% in the correctness of the tasks and 49,02% of effort reduction in relation to the time spent in each task when compared to developers who were not submitted to approach.

Keywords: Software Engineering. Electroencephalography. Machine Learning. Productivity. Tasks. Software Development. EEG. Developer.

LISTA DE FIGURAS

Figura 1 – Visão geral do processo de um mapeamento sistemático.	46
Figura 2 – Formulário para extração de dados.	51
Figura 3 – Passos do processo de filtragem.	53
Figura 4 – Extratificação da busca baseada nas bases eletrônicas utilizadas.	54
Figura 5 – Uma taxonomia que suporta a classificação dos estudos que usam dados biométricos em atividades de desenvolvimento de software.	57
Figura 6 – Frequência de publicações.	64
Figura 7 – Publicações por países.	65
Figura 8 – Relacionamento entre fatores psicofisiológicos e melhoria de produtividade.	67
Figura 9 – Processo experimental.	81
Figura 10 – Emotiv Epoc+.	83
Figura 11 – Sistema internacional 10-20.	83
Figura 12 – Emotiv Test Bench.	83
Figura 13 – OpenSesame.	84
Figura 14 – Interface do Visual Studio Code.	86
Figura 15 – Participante durante a execução do experimento.	87
Figura 16 – Detalhamento do filtro espacial.	89
Figura 17 – Corretude por experimento.	92
Figura 18 – Carca cognitiva média por tarefa.	93
Figura 19 – Carca cognitiva média separada por nível de expertise.	100
Figura 20 – Boxplot carga cognitiva gerada em respostas corretas.	101
Figura 21 – Boxplot carga cognitiva gerada em respostas incorretas.	102
Figura 22 – Boxplot carga cognitiva gerada em respostas incorretas com limites de acerto.	103
Figura 23 – Pulseira antiestática.	106
Figura 24 – Visão geral do 4Experts.	108
Figura 25 – Arquitetura da ferramenta para a abordagem 4Experts.	112
Figura 26 – Aspectos técnicos do 4Experts.	117
Figura 27 – Plugin para visualizar tarefas no visual studio.	118
Figura 28 – Tarefas com parâmetros no <i>Redmine</i>	119
Figura 29 – Usuários do <i>Redmine</i>	120
Figura 30 – Relacionamento entre nível de expertise e nível de dificuldade.	120
Figura 31 – Processo experimental.	128
Figura 32 – Sistema internacional 10-20.	130
Figura 33 – Emotiv Pro.	131
Figura 34 – Participante durante a execução do experimento.	133
Figura 35 – Corretude por experimento.	138
Figura 36 – Tempo médio de execução por tarefa.	139

LISTA DE TABELAS

Tabela 1 – Formato padronizado de interpretação do sinal EEG.	40
Tabela 2 – Lista de questões de pesquisa investigadas.	47
Tabela 3 – Lista dos termos principais e seus sinônimos.	48
Tabela 4 – Lista de ajustes de string de busca nas bases eletrônicas.	49
Tabela 5 – Lista de bases eletrônicas utilizadas na pesquisa.	50
Tabela 6 – Esquema proposto para classificar os estudos selecionados.	52
Tabela 7 – Extratificação da pesquisa com estudos primários.	55
Tabela 8 – Artigos de referência.	56
Tabela 9 – Taxonomia para classificação de dados biométricos.	57
Tabela 10 – Medidas do corpo usadas para suportar atividades de desenvolvimento.	59
Tabela 11 – Fatores relatados a dados biométricos que podem influenciar nas atividades de desenvolvimento de software.	60
Tabela 12 – Tarefas diárias normalmente realizadas no contexto do desenvolvimento de software.	61
Tabela 13 – Classificação dos estudos primários baseada nos métodos de pesquisa.	62
Tabela 14 – Dados biométricos usados para melhorar a produtividade dos desenvolvedores.	63
Tabela 15 – Publicações por países em detalhes.	65
Tabela 16 – Autores por publicação.	66
Tabela 17 – Comparativo dos trabalhos relacionados	70
Tabela 18 – Checklist de validade de um mapeamento sistemático de literatura.	72
Tabela 19 – Resumo das hipóteses.	78
Tabela 20 – Descrição das variáveis de estudo.	78
Tabela 21 – Identificação da expertise através autorrelato, pelo formulário de pesquisa separado por classes de pontuação.	79
Tabela 22 – Identificação da expertise através da corretude das respostas.	80
Tabela 23 – Identificação da dificuldade das questões do experimento.	82
Tabela 24 – Resultados gerais do experimento.	90
Tabela 25 – Teste de normalidade.	91
Tabela 26 – Resultados agrupados por tarefa - Corretude.	92
Tabela 27 – Resultados agrupados por tarefa - Carga Cognitiva.	93
Tabela 28 – Relação das variáveis de quantificação.	94
Tabela 29 – Estatística descritiva e testes estatísticos.	95
Tabela 30 – Resultado do esforço - teste Wilcoxon.	96
Tabela 31 – Carga cognitiva separada por níveis de expertise - 1º etapa.	97
Tabela 32 – Carga cognitiva separada por níveis de expertise - 2º etapa.	98
Tabela 33 – Estatística sobre a carga cognitiva agrupada por níveis de expertise - 1º etapa.	98
Tabela 34 – Estatística sobre a carga cognitiva agrupada por níveis de expertise - 2º etapa.	98
Tabela 35 – Variação da carga cognitiva entre níveis de expertise.	99

Tabela 36 – Diferenças estatísticas da carga cognitiva entre acertos e erros em tarefas.	101
Tabela 37 – Estatística descritiva das medianas e quartis.	102
Tabela 38 – Bandas de frequência	111
Tabela 39 – Estrutura de persistência dos dados biométricos.	115
Tabela 40 – Resumo das hipóteses.	125
Tabela 41 – Descrição das variáveis de estudo.	126
Tabela 42 – Participantes do experimento.	127
Tabela 43 – Distribuição dos participantes entre os experimentos.	127
Tabela 44 – Relacionamento entre grau de dificuldade e nível de expertise.	129
Tabela 45 – Questões respondidas por participante.	129
Tabela 46 – Resultados gerais do experimento.	135
Tabela 47 – Resultados individuais por participante sem 4Experts.	136
Tabela 48 – Resultados individuais por participante com 4Experts.	136
Tabela 49 – Teste de normalidade.	137
Tabela 50 – Resultados agrupados por tarefa - Corretude.	137
Tabela 51 – Resultados agrupados por tarefa - Esforço.	138
Tabela 52 – Relação das variáveis de quantificação.	139
Tabela 53 – Estatística descritiva e testes estatísticos.	140
Tabela 54 – Resultado da corretude - teste McNemar.	141
Tabela 55 – Resultado do esforço - teste Wilcoxon.	142
Tabela 56 – Lista de Publicações Diretas	189

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo coletor EEG.	113
Algoritmo 2 – Algoritmo conversor de bandas multitaper.	114
Algoritmo 3 – Algoritmo mlp.	115
Algoritmo 4 – Algoritmo priorizador de tarefas.	116

LISTA DE ABREVIATURAS

Matlab	Matrix Laboratory
Mhz	Megahertz
QP	Questão de Pesquisa
OB	Objetivo
CT	Contribuições
GQ	Questões Gerais
FQ	Questão Focada
SQ	Questão Estatística
EC	Critério de Exclusão
IC	Critério de Inclusão
RW	Trabalhos Relacionados
CLT	Teoria da Carga Cognitiva
SMS	Mapeamento Sistemático da Literatura
ML	Aprendizado de Máquina
T	Tarefas
C	Corretude
CC	Carga Cognitiva
EC	Esforço de Codificação

LISTA DE SIGLAS

ESCoP	European Society for Cognitive Psychology
ECoG	Electrocorticography
RSSE	Recommendation systems for software engineering
fmRI	Ressônança Magnética Funcional
GQM	Goal Question Method
EPF	Eclipse Process Framework
ACM	Association for Computing Machinery
SMS	Sistematic Mapping Study
EEG	Eleetroencefalografia
BCI	Brain Computer Interface
MEG	Magnetoencephalography
MRI	Magnetic Resonance Imaging
GQM	Goal Question Method
EDF	European Data Format
CSV	Comma Separated Values
MLP	Multi-Layer Perceptron
IEEE	Institute of Electrical and Electronic Engineers
J.UCS	Journal of Universal Computer Science
ZORA	Zurich Open Repository and Archive
CoME	Categorização de EEG Móveis
IDE	Integrated Development Environment
SDK	Software Development Kit

LISTA DE SÍMBOLOS

=	Atribuição
==	Comparação de Igualdade
>=	Maior igual
<=	Menor igual
>	Maior
<	Menor
δ	Delta
ϑ	Theta
α	Alpha
β	Beta
γ	Gamma

SUMÁRIO

1	INTRODUÇÃO	29
1.1	Problemática	31
1.2	Questões de Pesquisa	33
1.3	Objetivos	34
1.4	Metodologia	35
1.5	Contribuição do Trabalho	36
1.6	Organização do Trabalho	38
2	FUNDAMENTAÇÃO TEÓRICA	39
2.1	Dados Biométricos	39
2.2	Eletroencefalografia (EEG)	39
2.3	Carga Cognitiva	40
2.4	Tarefas de Desenvolvimento	41
2.5	Expertise dos Desenvolvedores	41
2.6	Aprendizagem de Máquina	42
2.7	Sistemas de Recomendação	43
3	TRABALHOS RELACIONADOS	45
3.1	Plano de Pesquisa	45
3.1.1	Objetivos e Questões de Pesquisa	46
3.1.2	Estratégias de Busca	46
3.1.3	CrITÉrios de Inclusão e Exclusão	48
3.1.4	Extração de Dados	51
3.1.5	Execução do Filtro de Pesquisa	52
3.2	Resultados	54
3.2.1	GQ1: Como seria a taxonomia para classificação de dados biométricos em tarefas de desenvolvimento de software?	55
3.2.2	GQ2: Quais são os tipos de dados biométricos usados para suportar tarefas de desenvolvimento de software?	58
3.2.3	GQ3: Quais fatores relacionados aos dados biométricos podem influenciar as tarefas de desenvolvimento de software?	60
3.2.4	GQ4: Quais tarefas diárias são comumente realizadas no contexto do desenvolvimento de software?	61
3.2.5	GQ5: Quais são os métodos de pesquisa utilizados para avaliar os estudos?	62
3.2.6	FQ1: Como os dados biométricos podem ser utilizados para melhorar a produtividade dos desenvolvedores?	62
3.2.7	SQ1: Qual é o número e tipo de publicações por ano?	63
3.2.8	SQ2: Quais países apresentaram publicações na área de pesquisa?	64
3.2.9	SQ3: Quem são os principais autores que publicaram artigos sobre dados biométricos na Engenharia de Software?	65
3.2.10	Discussão Adicional	67
3.2.11	Trabalhos Relacionados	68
3.2.12	Análise Comparativa dos Trabalhos Relacionados	70
3.3	Oportunidades de Pesquisa	71
3.4	Ameaças a Validade	71
3.5	Conclusão	73

4 ANÁLISE EMPÍRICA DO GANHO DE EXPERTISE DOS DESENVOLVEDORES ATRAVÉS DO USO DE DADOS BIOMÉTRICOS	75
4.1 Objetivos e Questões de Pesquisa	75
4.2 Hipóteses Formuladas	76
4.3 Variáveis	78
4.4 Contexto e Seleção dos Participantes	78
4.5 Processo Experimental	80
4.6 Material	82
4.7 Execução	85
4.8 Procedimentos de Análise	88
4.9 Resultados Coletados	89
4.9.1 QP1: É possível quantificar a carga cognitiva de um desenvolvedor? (hipótese 1)	95
4.9.2 QP2: A carga cognitiva do desenvolvedor permite a classificação da sua expertise? (hipótese 2)	96
4.9.3 QP3: É possível identificar um limite de carga cognitiva, que quando ultrapassado, implique em redução de acuracidade durante a execução de tarefas de programação? (hipótese 3)	100
4.10 Discussão Adicional	103
4.11 Ameaças a Validade	105
5 ABORDAGEM 4EXPERTS	107
5.1 Visão Geral	107
5.2 Arquitetura	112
5.3 Aspectos da Implementação	117
5.3.1 Plugin 4Experts	117
5.3.2 Aplicação Web 4Experts	119
6 AVALIAÇÃO DA ABORDAGEM 4EXPERTS	123
6.1 Objetivo e Questões de Pesquisa	123
6.2 Hipóteses Formuladas	124
6.3 Variáveis	125
6.4 Contexto e Seleção dos Participantes	126
6.5 Processo Experimental	127
6.6 Material	130
6.7 Execução	132
6.8 Resultados Coletados	134
6.8.1 QP1: O emprego da abordagem 4Experts resulta em maior corretude das tarefas desempenhadas? (Hipótese 1)	141
6.8.2 QP2: As recomendações de tarefas propostas pela abordagem 4Experts reduz o esforço despendido pelos desenvolvedores nas tarefas desempenhadas? (Hipótese 2)	142
6.8.3 Conclusão e Discussão Adicional	143
6.8.4 Ameaças a Validade	144
7 CONCLUSÃO	147
7.1 Contribuições	148
7.2 Limitações do Trabalho	150
7.3 Trabalhos Futuros	150

REFERÊNCIAS	153
APÊNDICE A – INFORMAÇÕES COMPLEMENTARES	163
A.1 Termo de Consentimento	163
A.2 Questionário de Pesquisa	165
A.3 Questões Aplicadas no Experimento	169
A.4 Documento de Acompanhamento para a Validação da Abordagem 4Experts . .	174
A.5 Questões Aplicadas no Experimento de Validação da Abordagem 4Experts . .	176
ANEXO A – ARTIGOS PUBLICADOS	189
A.1 Publicações	189

1 INTRODUÇÃO

Estudos recentes relatam que o crescimento da necessidade de software é muito maior do que a capacidade de construção dos mesmos (MEYER ET AL., 2014). Esta quantidade crescente de demanda por software faz com que esforços sejam direcionados em estudos e técnicas que possam melhorar a produtividade e o processo de construção de sistemas. Existem estudos em Engenharia de Software relacionadas a análise da produtividade na produção de software que avaliam aspectos como a produtividade associada a linhas de código em relação ao tempo, o impacto da organização sobre a produtividade (isto é, o impacto do espaço de trabalho sobre a produtividade) e também ferramentas e abordagens que auxiliam os desenvolvedores a melhorar o seu processo de desenvolvimento de sistemas (como, Personal Software Process) (MEYER ET AL., 2014). Porém, tais estudos não consideram uma análise do principal ator no processo de desenvolvimento de software, que é o próprio desenvolvedor. Apenas estudos recentes estão aumentando a análise sobre o desenvolvedor, como é o exemplo do estudo realizado por Meyer (MEYER ET AL., 2017).

Um desenvolvedor, durante o seu dia de trabalho, não executa apenas atividades relacionadas a tarefas de codificação. Além de codificar, ele executa tarefas como: revisão de código, versionamento e resolução de conflitos de versão, planejamento das suas atividades, leitura e escrita de documentação, reuniões formais e informais, a navegação na internet associada a sua tarefa e em alguns casos sem relação com o trabalho, além de atividades remotas e outras atividades de menor impacto (MEYER ET AL., 2017). Além disso, a complexidade dos softwares atuais exige tarefas de colaboração, comunicação e busca por informações necessárias para a execução das mesmas (GONÇALVES; SOUZA; GONZÁLEZ, 2011). Devido a esse diversificado conjunto de atividades realizadas pelo desenvolvedor, o seu dia de trabalho pode apresentar uma grande fragmentação de tarefas, ocasionando constantes trocas de atividade, o que impede que o desenvolvedor possa manter o foco na execução de uma atividade por vez, até chegar no seu fim, ocasionando aumento no tempo de execução e maior chance de falhas, impactando na produtividade. Outro aspecto associado à redução da produtividade por parte do desenvolvedor são os sentimentos negativos do indivíduo, que podem ser provocados pela frustração decorrente do próprio desejo em conseguir executar com sucesso as tarefas a ele atribuídas (WROBEL, 2013).

Por sua vez, a identificação das expertises e competências técnicas que um desenvolvedor possui e a possibilidade de lhe atribuir as tarefas de acordo com estas expertises pode reduzir aspectos emocionais negativos como a frustração, apresentada por Wrobel (WROBEL, 2013), além de aumentar a própria velocidade e acuracidade da codificação, reduzindo a fragmentação das tarefas. Porém, a própria identificação da expertise do desenvolvedor é uma área de estudo na Engenharia de Software que não apresenta muita pesquisa, por ser subjetiva e exigindo um considerável número de estudos empíricos, que também apenas recentemente passaram a receber mais atenção através da Engenharia de Software Experimental (WOHLIN ET AL.,

2012). Para identificar a expertise (habilidades e competências técnicas) do desenvolvedor, são aplicados questionários ao desenvolvedor, o qual irá respondê-los e a partir da análise das respostas é possível avaliar o grau de expertise de cada desenvolvedor. Feigenspan (FEIGENSPAN ET AL., 2012) conduziu um estudo para identificar uma maneira padronizada de elaborar esse questionário, identificando cinco aspectos que devem ser considerados: autoavaliação, tempo de experiência, formação educacional, tamanho dos projetos envolvidos e alguma informação adicional, como a idade do desenvolvedor. Este método de avaliação é eficiente, porém, está condicionado à resposta descritiva do próprio desenvolvedor, a qual pode sofrer influências incapazes de serem detectadas pelo questionário.

Nos últimos anos, com o aumento dos estudos em Neurociências e utilização de dados biométricos para aplicações em outras áreas, além da medicina, surgiram estudos para analisar a utilização dos dados biométricos na Engenharia de Software. Alguns desses estudos estão analisando os dados biométricos coletados através do cérebro com uso de sensores como o eletroencefalograma (EEG) e a ressonância magnética funcional (fmRI) com o objetivo de identificar a carga mental e as emoções dos desenvolvedores enquanto esses realizam as suas atividades de codificação (PEITEK SIEGMUND, 2018; MÜLLER; FRITZ, 2015). A análise dos dados coletados, através do uso de técnicas de aprendizado de máquina, está auxiliando os pesquisadores a identificar dificuldades na compreensão de código pelos desenvolvedores e também identificar a expertise do desenvolvedor sem a utilização de testes escritos, apenas com a execução de atividades de desenvolvimento (LEE ET AL., 2017; CRK; KLUTHE; STEFIK, 2015).

O objetivo deste trabalho é desenvolver uma abordagem possa recomendar a próxima tarefa que o desenvolvedor deve realizar durante o seu dia a partir da análise, com o uso de técnicas de aprendizagem de máquina, dos dados biométricos produzidos pelo cérebro com o auxílio de um aparelho de eletroencefalograma (EEG) portátil enquanto ele executa uma tarefa. A técnica de aprendizagem de máquina será treinada a partir de uma base de dados biométricos e quando aplicada ao desenvolvedor irá prever a propensão da tarefa que está sendo performada apresentar falhas na sua execução, de forma a recomendar que a próxima tarefa a ser executada apresente características que permitam ao desenvolvedor executar a mesma com sucesso. Estas recomendações serão exibidas para o programador através da sua ferramenta de desenvolvimento, como o Visual Studio. A técnica utilizará dados biométricos provenientes de um equipamento EEG conforme já utilizado por CRK (CRK; KLUTHE; STEFIK, 2015).

Conseguir identificar a expertise do desenvolvedor utilizando os seus próprios dados biométricos pode garantir uma acuracidade superior em relação aos modelos tradicionais de avaliação, seja porque os dados obtidos são do próprio desenvolvedor como também por serem dados produzidos de forma autônoma pelo organismo, estando menos sujeitos a interferências externas, mesmo que involuntárias.

1.1 Problemática

A atividade de desenvolvimento de software não se resume apenas em codificar, conforme Meyer (MEYER ET AL., 2017), a codificação é apenas uma das atividades em meio a várias outras executadas pelos envolvidos no processo de construção dos sistemas, dentre estas tarefas, podem ser citadas: leitura e escrita de documentos, reuniões formais ou informais, envio e leitura de e-mails, navegação na internet, atividades colaborativas, entre outras. Todas estas atividades são executadas durante o dia de trabalho do desenvolvedor, sendo estas já previstas ou não, resultando na fragmentação do trabalho;(MARK; GONZALEZ; HARRIS, 2005).

Diversos autores apresentam evidências de que a fragmentação das tarefas implica em redução de produtividade (SANCHEZ; ROBBES; GONZALEZ, 2015; MARK; GONZALEZ; HARRIS, 2005; A diary study of task switching and interruptions, 2004), associada ao fato de que quando ocorre uma troca de contexto entre tarefas, para o desenvolvedor retornar à atividade anterior, é necessário que ele recapitule o que estava fazendo anteriormente. Este processo de recapitulação sempre irá necessitar de tempo extra do desenvolvedor e pode ocasionar falhas durante a execução da tarefa (SANCHEZ; ROBBES; GONZALEZ, 2015). Além disso, atividades não planejadas surgidas durante o dia tendem a causar frustração ao desenvolvedor, pois conforme Meyer (MEYER ET AL., 2014), o desenvolvedor se sente produtivo quando ele consegue executar sem interrupções as tarefas que lhe foram atribuídas.

O Eclipse Process Framework (EPF) (HAUMER, 2007) detalha o processo padrão, normalmente empregado no contexto de indústrias de software, para a execução de tarefas. Este processo é composto por: *role*, *task* e *work product*, os quais interagem entre si, onde as *roles* são os papéis que cada indivíduo possui no processo de desenvolvimento. Para cada tipo de papel existe um conjunto de expertises específicas necessárias para desempenhar a atribuição. As tarefas (*tasks*) possuem uma *role* associada a mesma, de acordo com o tipo de tarefa. Por exemplo, tarefas de codificação são atribuídas ao papel do desenvolvedor, enquanto tarefas de elaboração de requisitos são atribuídas ao papel do analista. Por fim, o conjunto de tarefas realizadas resulta em um produto funcional (*work product*).

A recomendação das tarefas neste processo de desenvolvimento é feita por um responsável, como o gerente de projetos. Considerando que para cada papel, possam existir na equipe vários colaboradores, além do gerente ter a função de identificar a tarefa adequada para cada papel, ele necessita identificar quem, dentro da equipe, está mais qualificado a realizar a tarefa de acordo com a complexidade dela. O conhecimento que o gestor possui para fazer esta atribuição está associado a indicadores e avaliações de capacidade técnica do desenvolvedor, além da experiência adquirida na execução de atividades anteriores (HUZEFA KAGDI1, MALCOM GETHERS2; HAMMAD, 2011).

A problemática deste tipo de recomendação de tarefas é o fato dela possuir um considerável grau de subjetividade e experiência do gestor, podendo provocar uma redução de qualidade e produtividade, caso não seja assertiva (HUZEFA KAGDI1, MALCOM GETHERS2; HAM-

MAD, 2011; KAGDI; POSHYVANYK, 2009; MA ET AL., 2009). Em um ambiente ideal, as atividades de maior complexidade devem ser direcionadas aos desenvolvedores com as maiores expertises e as de complexidade reduzida fiquem com os portadores de menor expertise (MA ET AL., 2009). Além dos aspectos técnicos apresentados, esta situação também impacta nos aspectos psicológicos dos desenvolvedores, os quais são afetados quando as especialidades técnicas do desenvolvedor não são utilizadas de forma adequada (WROBEL, 2013). Isso pode ocorrer quando um desenvolvedor com pouca expertise recebe uma tarefa muito complexa, a qual ele não possui competência para executar implicando em bloqueios durante a atividade, que consequentemente geram emoções negativas (GRAZIOTIN ET AL., 2018). Por outro lado, estas emoções negativas também podem ser sentidas pelo desenvolvedor, quando ele possui muita expertise, porém, ele apenas realiza tarefas de baixa complexidade. Ainda, este processo pode impactar na evolução progressiva das especialidades do desenvolvedor, por não identificar adequadamente esta evolução, distribuindo tarefas que dificultem esta evolução.

Diante do apresentado pelas técnicas atuais, este trabalho irá explorar quatro Problemáticas(P) descritas a seguir:

P1: A falta de uma visão geral do estado da arte no uso de dados biométricos na Engenharia de Software. O uso de dados biométricos em áreas além da medicina tem aumentado significativamente devido ao aumento da tecnologia, possibilitando coletar estes dados durante o dia a dia das pessoas (MINGUILLON; LOPEZ-GORDO; PELAYO, 2017) com o propósito de identificar estados cognitivos, físicos e emocionais (MINGUILLON; LOPEZ-GORDO; PELAYO, 2017; GIANNAKAKIS ET AL., 2017; CACIOPPO; TASSINARY; BERNTSON, 2016). Na Engenharia de Software, o uso de dados biométricos tem aumentado nas últimas décadas com o propósito de avaliar características como a compreensão de código, influência das emoções durante as atividades de programação e usabilidade de sistemas (ZÜGER ET AL., 2018; SIEGMUND ET AL., 2017; MÜLLER; FRITZ, 2015; MANHARTSBERGER; ZEL-LHOFER, 2005). Os desenvolvedores, em suas atividades de codificação, podem ser influenciados pelas suas emoções e necessitam de habilidades cognitivas para conseguirem desempenhar suas tarefas, tornando esta área de estudo importante para a Engenharia de Software (MÜLLER; FRITZ, 2015; FRITZ ET AL., 2014). Apesar de existirem diversos artigos que estudam o emprego de dados biométricos na Engenharia de Software, a área ainda necessita de uma revisão de literatura em formato de mapa sistemático, que possa fornecer uma visão geral de como estes dados podem ser utilizados gerando desta forma uma taxonomia para a área.

P2: Falta de conhecimento empírico sobre o uso de dados biométricos dos desenvolvedores.

Estudos como o realizado por Sebastian Muller (MÜLLER; FRITZ, 2015), Wrobel (WROBEL, 2018) e Siegmund (SIEGMUND, 2016) são conduzidos através da observação de como os desenvolvedores executam suas tarefas sob determinadas condições enquanto estes fornecem dados biométricos, possibilitando que seja criado um relacionamento entre os dados psicofisiológicos resultantes e ações ou atitudes dos desenvolvedores durante o seu trabalho através do

emprego de técnicas de aprendizagem de máquina.

Os estudos citados avaliam situações como a compreensão de código (SIEGMUND, 2016; NAKAGAWA ET AL., 2014) e a influência das emoções no resultado das atividades (WROBEL, 2018; MÜLLER; FRITZ, 2015), porém, conforme detalhado por Meyer (MEYER ET AL., 2017), o dia de trabalho do desenvolvedor é composto de variadas atividades, as quais sofrem influências do ambiente e das próprias características do desenvolvedor. O grande número de atividades de Engenharia de software relatado por Meyer associado a pesquisa dos trabalhos atuais sobre o emprego de dados biométricos nesta mesma área possibilita identificar uma aparente falta de conhecimento empírico sobre o uso de dados biométricos e a sua relação com as tarefas desempenhadas pelos desenvolvedores. A falta de conhecimento empírico resulta na necessidade de gerar este conhecimento sempre que o estudo proposto possuir a necessidade de avaliar algum aspecto da Engenharia de Software que ainda não tenha sido avaliado sob a ótica dos dados biométricos dos desenvolvedores.

P3: Inexistência de uma técnica sensível a informações psicofisiológicas para recomendar tarefas.

As técnicas de recomendação de tarefas atuais podem ser realizadas de forma manual ou automatizadas de acordo com o histórico do desenvolvedor (MA ET AL., 2009; KAGDI; POSHYVANYK, 2009). Caso seja empregado um processo manual, as tarefas serão distribuídas por um gerente de desenvolvimento ou líder técnico, o qual irá distribuir as atividades conforme a sua percepção de cada desenvolvedor e papéis que cada um pode assumir, além da capacidade técnica dos mesmos em realizar uma tarefa (KAGDI; POSHYVANYK, 2009; HUZefa KAGDI1, MALCOM GETHERS2; HAMMAD, 2011). Por sua vez, o processo automatizado, baseado no histórico do desenvolvedor, analisa os artefatos já modificados pelo desenvolvedor, recomendando tarefas com base nestas mudanças (KAGDI; POSHYVANYK, 2009). Apesar das duas formas de recomendação serem baseadas no desenvolvedor, elas não consideram o momento presente, ignorando qualquer estado psicofisiológico, que naquele momento pode influenciar a execução da tarefa com a desenvoltura necessária para evitar esforço desproporcional e garantir a qualidade adequada a tarefa, além de poder causar frustração no desenvolvedor por ele não conseguir desempenhar a atividade da forma que ele gostaria. Caso uma técnica de recomendação de tarefas possa considerar os dados psicofisiológicos do desenvolvedor durante o seu trabalho e com estes dados submetidos a uma técnica de aprendizagem de máquina, o sistema possa identificar o seu estado atual, esta técnica poderá identificar qual atividade o desenvolvedor está apto a executar, evitando a recomendação de algo que resultará em esforço excessivo ou que possa gerar uma falha de execução.

1.2 Questões de Pesquisa

Este trabalho busca investigar um novo enfoque para a recomendação de tarefas de desenvolvimento de software. Para isso, pretende-se desenvolver uma abordagem de recomendação

de tarefas que leve em conta informações do próprio desenvolvedor e faça uso de técnicas de aprendizagem de máquina. Frente ao exposto, a Questão de Pesquisa (QP) geral desta dissertação é apresentada a seguir:

QP_{Geral}: Como gerar recomendações de tarefas de desenvolvimento de software sensíveis a dados biométricos utilizando técnicas de aprendizagem de máquina?

Com a finalidade de responder a esta questão geral, foram elaboradas quatro questões de pesquisa específicas, as quais seguem:

- **QP1:** Qual é o estado da arte em relação ao uso de dados biométricos do desenvolvedor na Engenharia de Software?
- **QP2:** É possível identificar o nível de expertise dos desenvolvedores de software utilizando dados biométricos?
- **QP3:** Como melhorar a recomendação de tarefas utilizando dados biométricos e aprendizagem de máquina?
- **QP4:** Qual é o nível de eficácia da abordagem proposta comparando-se a corretude e o esforço durante a execução das tarefas recomendadas?

1.3 Objetivos

O objetivo geral desta pesquisa é construir uma abordagem para a recomendação de tarefas de Engenharia de Software, visando aumentar a corretude e reduzir o esforço dos desenvolvedores durante as atividades de codificação, promovendo desta forma aumento de produtividade. Para isso, quatro objetivos específicos foram construídos viabilizando a confecção e a avaliação desta abordagem em relação ao que é sugerido no estado da arte. Os objetivos (OB) específicos são listados a seguir:

- **OB1: Realizar um mapeamento sistemático para compreender o uso de dados biométricos do desenvolvedor nas atividades de Engenharia de Software.** Esse problema visa solucionar a falta de uma visão geral sobre o estado da arte na área de utilização de dados biométricos ou psicofisiológicos na Engenharia de Software. Para isso, é importante: (1) revisar os conceitos de dados biométricos, psicofisiológicos, atividades de Engenharia de Software, expertise e aprendizagem de máquina, as quais são atualmente empregadas em técnicas que fazem uso de dados biométricos; (2) criar uma visão panorâmica do estado da arte dos tipos de dados coletados e suas aplicações na Engenharia de Software; e (3) identificar oportunidades de pesquisa na área de recomendação de tarefas. Para cumprir este objetivo, o trabalho atual (1) fornece uma classificação da literatura através de um estudo de mapeamento sistemático relativo à utilização de dados biométricos na Engenharia de Software, resultando em uma taxonomia para a utilização destes dados, e (2) destaca as lacunas e direções de pesquisa para futuras investigações.

- **OB2: Produzir conhecimento empírico através do uso de dados biométricos dos desenvolvedores.** Devido à falta de conhecimento empírico relativo ao uso de dados biométricos em atividades de Engenharia de Software, este objetivo visa produzir conhecimento empírico através de um experimento controlado, executando o experimento em dois momentos intercalados por um período de tempo, com o propósito de analisar o comportamento dos dados biométricos em relação ao ganho de conhecimento dos participantes possibilitando identificar uma forma de avaliar as expertises adquiridas dos desenvolvedores.
- **OB3: Construir uma abordagem para recomendar tarefas de Engenharia de Software para o desenvolvedor utilizando dados biométricos e emprego de técnicas de aprendizagem de máquina.** Conforme apresentado anteriormente, algumas das atuais técnicas de recomendação de tarefas são subjetivas. Para resolver este problema, o trabalho tem por objetivo construir uma abordagem de recomendação de tarefas de Engenharia de Software, utilizando dados biométricos do desenvolvedor através do emprego de técnicas de aprendizagem de máquina. Esta abordagem é importante, pois o uso de dados biométricos do desenvolvedor levará em conta as características reais do desenvolvedor, sem o uso de subjetividade durante a recomendação. Neste processo, o emprego das técnicas de aprendizado de máquina é imprescindível, visto a grande quantidade de dados biométricos gerados, não apresentando condições de ser realizada uma análise manual.
- **OB4: Avaliar o sistema de recomendação através da corretude das tarefas realizadas e o esforço investido por desenvolvedores.** Com o propósito de avaliar o Objetivo 3, este objetivo propõe uma avaliação da abordagem desenvolvida. Esta avaliação será baseada em um experimento comparando taxa de acertos e o esforço empregado para realizar as atividades, quando estas tarefas são recomendadas a partir da abordagem criada em relação a recomendação manual de tarefas.

1.4 Metodologia

O principal objetivo deste trabalho é desenvolver uma abordagem de recomendação de tarefas de Engenharia de Software com o uso de dados biométricos do desenvolvedor e emprego de aprendizagem de máquina. Para isso, quatro questões de pesquisa foram elaboradas (Seção 1.2). Consequentemente, este trabalho utiliza vários métodos de pesquisa para alcançar os objetivos específicos estabelecidos na Seção 1.3, os quais são descritos abaixo.

A primeira questão de pesquisa (QP1) aborda a falta de uma visão geral do estado da arte na área de utilização de dados biométricos dos desenvolvedores nas tarefas de Engenharia de Software. Com isso, foi aplicado um protocolo de Estudo de Mapeamento Sistemático através de orientações bem estabelecidas na literatura (PETERSEN; VAKKALANKA; KUZNIARZ, 2015). Assim, os estudos devem ser obtidos através da busca nas bases de estudo mais utilizadas

no campo da ciência da computação. Especificamente, estes estudos foram obtidos aplicando critérios de exclusão e inclusão, enquadramento com as questões de pesquisa, a procura pelos estudos primários nos mecanismos de busca, e filtragem pelos estudos primários.

A segunda questão de pesquisa (QP2) tem por objetivo gerar dados empíricos sobre a efetividade do uso de dados biométricos na Engenharia de Software. Para isso, o método utilizado para esta questão foi um experimento controlado para a avaliação da evolução das expertises do desenvolvedor com o passar do tempo objetivando a identificação da sua expertise com o uso de dados biométricos. Os desenvolvedores foram submetidos a dois testes em momentos diferentes, de forma que durante o intervalo de tempo da realização dos testes, os desenvolvedores foram adquirindo novas especialidades, com o propósito de comparar três hipóteses definidas para esta questão. Ao total 42 pessoas participaram do teste com diversos níveis de experiência, permitindo analisar a efetividade da abordagem proposta.

A terceira questão de pesquisa (QP3) questiona as dificuldades e limitações das técnicas atuais de distribuição e recomendação de tarefas de Engenharia de Software. Para contornar isso, propõe-se uma abordagem que utilize os dados biométricos e psicofisiológicos do próprio desenvolvedor. Juntamente com a abordagem é apresentado um fluxo que define a interação entre os componentes que fazem parte do processo e como os envolvidos interagem durante as atividades diárias de desenvolvimento.

A quarta questão de pesquisa (QP4) procura validar o nível de eficácia da abordagem proposta em relação a corretude e esforço das tarefas performadas com o uso do sistema de recomendação. Esta validação também será feita através de um experimento controlado no qual os participantes receberão um conjunto de tarefas a serem desenvolvidas. Um grupo de participantes receberá as questões sem o emprego da abordagem, enquanto o outro grupo executará as tarefas distribuídas através da abordagem proposta. Após, será realizada a comparação do número de acertos e o tempo de resposta para cada uma das tarefas, objetivando avaliar o resultado do uso da abordagem.

Em resumo, as metodologias envolvidas neste trabalho buscam (1) a avaliação da efetividade da abordagem proposta de recomendação de tarefas baseada nos dados biométricos e aprendizagem de máquina, (2) o estudo do emprego de dados biométricos e aprendizagem de máquina na Engenharia de Software e (3) a geração de conhecimento empírico baseado no uso de dados biométricos em atividades de Engenharia de Software. As contribuições esperadas, provenientes da execução dessa etapa metodológica, são detalhadas na próxima seção.

1.5 Contribuição do Trabalho

Esta Seção descreve as Contribuições (CT) esperadas após a execução da metodologia definida na seção anterior. A principal contribuição esperada é uma ferramenta de recomendação de tarefas de Engenharia de Software que faça uso de dados biométricos do desenvolvedor e aplique técnicas de aprendizagem de máquina. Essa contribuição foi viabilizada através das

contribuições:

- **CT1: Uma visão geral do estado da arte sobre o uso de dados biométricos do desenvolvedor na Engenharia de Software.** Através do estudo de mapeamento sistemático foi possível contribuir com uma visão geral do estado da arte no que diz respeito: (1) as atividades performadas pelos desenvolvedores durante o seu dia a dia; (2) aos diferentes tipos de dados biométricos e sensores empregados para capturar estes dados dos desenvolvedores; e (3) ao relacionamento dos diferentes tipos de dados biométricos aos aspectos que impactam na realização das atividades de desenvolvimento; (4) o relacionamento destes dados biométricos com a melhoria da produtividade dos desenvolvedores; e (5) aos direcionamentos para futuros estudos na área de aplicação de dados biométricos em Engenharia de Software.
- **CT2: Produção de conhecimento empírico através do uso de dados biométricos dos desenvolvedores.** O experimento controlado utilizado para fornecer o conhecimento empírico relativo aos dados biométricos dos desenvolvedores procura revelar que: (1) a carga cognitiva do desenvolvedor será maior de acordo com o nível de dificuldade da atividade proposta; (2) quanto maior a carga cognitiva, maior a probabilidade do desenvolvedor cometer uma falha durante a execução da sua atividade; (3) conforme o desenvolvedor aumentar as suas expertises a quantidade de carga cognitiva necessária para executar uma atividade de desenvolvimento será menor; e (4) com base nestas identificações, conseguir identificar a expertise dos desenvolvedores.
- **CT3: Uma abordagem de recomendação de tarefas de Engenharia de Software baseada em dados biométricos do desenvolvedor com o emprego de técnicas de aprendizagem de máquina.** Devido à subjetividade de alguns dos processos atuais de recomendação de tarefas, foi necessário desenvolver a abordagem de recomendação de tarefas. Especificamente, esta etapa contribuí em: (1) uma abordagem empírica para o processo de recomendação de tarefas; (2) ênfase no desenvolvedor, o qual é o principal ator no processo de desenvolvimento de software; (3) o emprego de dados biométricos, os quais fornecem informações mais precisas em relação aos dados obtidos por outros meios, como avaliações escritas; e (4) utilização de aprendizagem de máquina em Engenharia de Software.
- **CT4: Apresentar resultados relativos ao nível de eficácia da abordagem proposta comparando-se a corretude e o esforço na recomendação de tarefas.** Com o experimento controlado necessário para avaliar o nível de eficácia da abordagem proposta será identificado que: (1) a recomendação das tarefas respeitando as expertises de cada desenvolvedor reduz o esforço empregado nos artefatos desenvolvidos, pois não expõe o desenvolvedor a tarefas que ele não tem condições de executar; e (2) a redução do número de falhas durante a execução das tarefas, aumentando a corretude.

1.6 Organização do Trabalho

Este trabalho está organizado da seguinte forma: O Capítulo 2 apresenta a fundamentação teórica, uma visão geral dos conceitos básicos sobre a utilização de dados biométricos na Engenharia de Software, as atividades dos desenvolvedores, a classificação da expertise e técnicas de aprendizagem de máquina que serão utilizadas neste trabalho. O Capítulo 3 apresenta os estudos relacionados com este trabalho através de um mapeamento sistemático sobre o uso de dados biométricos na Engenharia de Software. O Capítulo 4 aborda a produção de conhecimento empírico com base em um experimento controlado. Já o Capítulo 5 apresenta a abordagem proposta no trabalho, enquanto o Capítulo 6 aborda os resultados das avaliações realizadas sobre a abordagem proposta. O fechamento do trabalho é apresentado no Capítulo 7 com as considerações finais e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Para contextualizar este estudo, será apresentada uma definição de dados biométricos, eletroencefalografia, carga cognitiva, tarefas de desenvolvimento, expertise dos desenvolvedores, aprendizagem de máquina e sistemas de recomendação.

2.1 Dados Biométricos

Os dados biométricos são medidas fisiológicas obtidas através da pele, coração, olhos, cérebro e outros órgãos humanos. Estes dados são coletados por meio de sensores e as informações obtidas podem estar relacionados aos estados cognitivos e emocionais de uma pessoa (KRAMER, 1990). Na maioria dos casos, os sensores são não-invasivos e coletam dados fisiológicos produzidos de forma autônoma pelo corpo, que dificilmente podem manipulados pelo indivíduo. As medidas são capturadas de diferentes maneiras de acordo com a área do corpo a qual elas se relacionam.

A variação dos dados obtidos por essas medidas, em pessoas saudáveis, também está associada a fatores psicológicos como emoção e carga cognitiva. O estudo dessa relação entre dados biométricos e aspectos psicológicos é conhecida como psicofisiologia. A psicofisiologia vem recebendo atenção (SIEGMUND ET AL., 2017) na área de Engenharia de Software, pois a atividade de desenvolvimento de software envolve diversos processos cognitivos, como compreensão de linguagem, atenção e também aspectos emocionais como a frustração (WROBEL, 2013) e o estresse (OSTBERG ET AL., 2017). Sem o uso de sensores biométricos, o estudo dos aspectos psicológicos só poderia ser feito através de processos holísticos, limitando a precisão da análise (PEITEK ET AL., 2018a). Portanto, esta área é muito promissora na Engenharia de Software para promover uma melhor compreensão do processo mental do desenvolvedor durante o desempenho de suas atividades diárias (MÜLLER, 2016; ROBILLARD, 1999).

2.2 Eletroencefalografia (EEG)

Dentre os diferentes tipos de dados biométricos que podem ser utilizados na Engenharia de Software, o uso de dados provenientes do cérebro recebem uma atenção especial nos estudos. Estes dados são obtidos com o uso de equipamentos conhecidos como Interface Cérebro-Computador (BCI), que consistem em um conjunto de hardware e software que possibilitam humanos a controlar computadores e outros dispositivos a partir da atividade cerebral. Os BCI se classificam em invasivos, parcialmente invasivos e não invasivos (RAMADAN; VASILAKOS, 2017; WALDERT, 2016), de forma que nos métodos invasivos, microelétrodos são implantados diretamente no cérebro para coletar a atividade dos neurônios, enquanto que nos métodos parcialmente invasivos os eletrodos são implantados no crânio, mas não diretamente no cérebro e por fim na forma não invasiva os eletrodos não são implantados, permanecendo em contato

direto com o couro cabeludo do indivíduo. Neste último caso, a qualidade dos sinais capturados é menor quando comparada aos métodos invasivos, porém, é considerada a maneira mais segura e fácil de registrar a atividade cerebral.

O BCI permite diversas formas de analisar a atividade cerebral, das quais, a eletroencefalografia (EEG) é a forma mais proeminente de verificar a atividade cerebral por um BCI não invasivo. O primeiro uso do EEG em humanos ocorreu em 1924 (SWARTZ, 1998) e desde então vem sendo utilizado para capturar a atividade do sistema nervoso pelos sinais elétricos gerados pelo cérebro. Os dados de EEG podem ser caracterizados pela sua amplitude e frequência das ondas que variam de 0.01Hz até 64Hz. A Tabela 1 apresenta uma classificação das frequências baseada em bandas de frequência, nas quais cada banda de frequência está associada a determinados padrões cognitivos conforme relatado por Gopejenko e Buzsáki (GOPEJENKO; MIHAILOVS, 2015; BUZSÁKI, 2009).

Tabela 1 – Formato padronizado de interpretação do sinal EEG.

Identificação da Onda	Faixa de Frequência	Características
Delta	0.5 - 3Hz	Alta em sono profundo. Maior amplitude comparada com outras ondas.
Theta	4 - 7Hz	Associada a atividades cognitivas como meditação e carga cognitiva.
Alpha	8 - 14Hz	Frequência mais comum em adultos. Frequência dominante durante relaxamento. Reduz ou desaparece sob stress ou durante atividades cognitivas.
Beta	15 - 38Hz	Menor amplitude de todas as ondas. Associada com estado de alerta e ansiedade.
Gamma	38 - 100Hz	Forte durante momentos de foco, concentração e processamento de informação.

Fonte: Adaptado de (BUZSÁKI, 2009).

2.3 Carga Cognitiva

A carga cognitiva é um indicador que apresenta a quantidade de esforço mental exercido pela memória de trabalho enquanto uma tarefa é realizada. A teoria da carga cognitiva (CLT) (SWELLER, 1994) apresenta uma relação entre a capacidade da memória de trabalho e as demandas cognitivas de uma tarefa (ANDERSON ET AL., 2011). A teoria classifica a carga cognitiva em três categorias: (1) carga relevante ou pertinente (*germane cognitive load*); (2) carga intrínseca (*intrinsic cognitive load*); e (3) carga cognitiva irrelevante ou estranha (*extraneous cognitive load*). A carga intrínseca descreve o esforço mental experimentado devido a complexidade da tarefa e que não depende do design do problema. Por sua vez, a carga cognitiva relevante está relacionada a modulação dos recursos cognitivos do indivíduo ao se adaptar para realizar diferentes tarefas enquanto a carga irrelevante relaciona-se com o ruído nas tarefas, de forma que quanto menor for esta carga, melhor será o resultado da tarefa executada.

A capacidade da memória de trabalho é limitada e os indivíduos estão conscientes do que é mantido nesta memória. Segundo a teoria da carga cognitiva, a capacidade e o desempenho da memória de trabalho estão inversamente relacionados. Como a memória de trabalho desempenha um papel significativo na resolução de tarefas, medir o desempenho da memória de trabalho

implica medir a carga cognitiva do cérebro do usuário enquanto ele está performando a tarefa. A baixa carga de memória de trabalho está associada à experiência, enquanto a alta carga de memória de trabalho implica falta de experiência (CRK; KLUTHE, 2014). Wolfgang1 (KLI-MESCH, 1999) demonstra como os ritmos alfa e theta refletem o desempenho cognitivo e de memória.

2.4 Tarefas de Desenvolvimento

A construção de software é uma atividade de conhecimento, que não se limita à programação. Antes de o sistema ser codificado, o desenvolvedor deve descrever e organizar o conhecimento representado pelo programa a ser codificado (ROBILLARD, 1999). Este processo envolve uma série de atividades e interações como leitura e escrita de documentação, reuniões, navegação na internet, planejamento (MEYER ET AL., 2017) e programação. A atividade de programação pode ser classificada em subatividades, como depuração, controle de versão e teste (MAALEJ; HAPPEL, 2009). Todas essas tarefas podem ser desenvolvidas individualmente ou de forma colaborativa (GONÇALVES; SOUZA; GONZÁLEZ, 2011).

Até poucos anos atrás essas atividades eram estudadas pela Engenharia de Software apenas com o objetivo de torná-las mensuráveis e lógicas através da aplicação de métodos, ferramentas, simbologias e linguagens (ROBILLARD, 1999). Entretanto, nos últimos anos, o número de artigos que utilizam a psicofisiologia para analisar os processos cognitivos e mentais dos desenvolvedores tem aumentado. Estes estudos focam na compreensão de como essas tarefas são realizadas pelos desenvolvedores, com um intuito de fornecer um melhor suporte durante sua realização (MÜLLER, 2016).

2.5 Expertise dos Desenvolvedores

Na Engenharia de Software, a identificação da expertise do desenvolvedor possui um papel relevante. A correta identificação da expertise permite relacionar adequadamente um desenvolvedor a uma determinada solicitação de codificação de um programa. Huzefa (HUZEFA KAGDI1, MALCOM GETHERS2; HAMMAD, 2011) cita em seu artigo que a tarefa de identificação da expertise é comum nas equipes de desenvolvimento, mas não é trivial, pois, de forma tradicional, ela envolve conhecimentos de projeto, organização e o balanceamento de diversos aspectos. Esta tarefa torna-se consideravelmente mais complexa nos casos de equipes grandes e distribuídas geograficamente.

A expertise está associada à compreensão que um determinado desenvolvedor possui sobre um determinado programa. A compreensão de um programa é um processo cognitivo do próprio desenvolvedor, o qual não pode ser observado diretamente. Por isso, para avaliar a expertise, são necessários experimentos e técnicas que permitam observar o comportamento do desenvolvedor, quando este for confrontado com um determinado problema de codificação.

Feigenspan (FEIGENSPAN ET AL., 2012) realizou uma revisão de literatura para identificar o estado da arte na medição e controle de expertise no desenvolvimento. A partir desta revisão, ele propôs um questionário com um modelo de 5 fatores para a classificação da expertise do desenvolvedor. Outra técnica que pode ser empregada para a identificação da expertise é a avaliação de códigos já realizados pelo desenvolvedor através dos sistemas de controle de versão. Esta técnica foi proposta por Huzefa (HUZEFA KAGDI1, MALCOM GETHERS2; HAMMAD, 2011) em um sistema de recomendação que selecionava os desenvolvedores com uma expertise adequada as tarefas de codificação propostas, em uma equipe geograficamente distribuída.

Conforme relatado anteriormente, o processo cognitivo do desenvolvedor não pode ser observado diretamente. Porém, com o avanço da neurociência, já é possível analisar sinais biométricos produzidos principalmente pelo cérebro para prever a expertise do desenvolvedor (CRK; KLUTHE; STEFIK, 2015; LEE ET AL., 2017). Nesta técnica, os sinais biométricos são coletados dos desenvolvedores com o uso de sensores, enquanto estes estão realizando alguma atividade cognitiva. Posteriormente, os dados coletados são classificados com o uso de métodos estatísticos ou de aprendizado de máquina.

2.6 Aprendizagem de Máquina

Aprendizagem de máquina é conceituada pela Enciclopédia de Ciência da Computação (DIETTERICH, 2003) como “*o estudo de métodos para construir e aprimorar sistemas de software, analisando exemplos do comportamento desejado em vez de programá-los diretamente*”. Ainda, conforme a enciclopédia, os métodos de aprendizado de máquina são adequados às aplicações em que não é possível fornecer especificações precisas para o comportamento desejado do programa, mas exemplos deste comportamento estão disponíveis. Exemplos destas situações são: reconhecimento óptico de caracteres, reconhecimento de fala, direção automatizada de automóveis e reconhecimento de padrões. A característica comum a todos estes exemplos é que as pessoas podem realizá-los sem dificuldade, mas não conseguem descrever com exatidão como se dá o processo (algoritmo) para a sua execução. Nestes casos, a própria execução destas tarefas pelas pessoas é o exemplo de entrada necessário para que a máquina possa desempenhar a tarefa com um grau de assertividade adequado.

Na Engenharia de Software o uso de técnicas de aprendizagem de máquina tem sido aplicado em diversas áreas, desde sistemas de recomendação (H ALI; EL DESOUKY, 2016) até a análise de dados biométricos em relação a compreensão de código (MÜLLER; FRITZ, 2016) e sistemas de segurança (BABLANI; EDLA, 2019). Especificamente as técnicas de aprendizado de máquina aplicadas aos dados biométricos coletados a partir do cérebro fornecem novos direcionamentos e descobertas na compreensão do funcionamento do cérebro (VU ET AL., 2018), sendo especificamente aplicada neste estudo a técnica supervisionada conhecida como Multi-layer Perceptron, a qual fornece bons resultados na predição de situações, sem necessitar de

uma quantidade elevada de dados para serem processados (GARDNER; DORLING, 1998).

2.7 Sistemas de Recomendação

O aumento do tamanho dos projetos de software traz consigo um aumento da complexidade em navegar nestes projetos. Neste contexto, uma mudança como a inclusão de uma mensagem pode envolver a descoberta da classe correta para fazer a mudança dentro centenas de classes em uma biblioteca de classes. Por conta de situações como esta, sistemas de recomendação de tarefas para Engenharia de Software (RSSEs) estão emergindo para auxiliar os desenvolvedores em diversas atividades. Conforme Gasparic (GASPARIC; JANES, 2016), sistemas de recomendação são aplicações de software que auxiliam no suporte à tomada de decisão dos usuários enquanto estes interagem com uma grande quantidade de informações em determinados contextos. Estes sistemas recomendam itens de interesse do usuário baseado em suas preferências expressas de forma explícita ou implícita.

Conforme apresentado por Robillard (ROBILLARD; WALKER; ZIMMERMANN, 2010), um RSSE precisa providenciar ou inferir os seguintes aspectos como parte do seu contexto:

- As características do usuário, como a descrição da sua função, nível de expertise, prioridades de trabalho e redes sociais;
- O tipo de atividade que é realizada, como a inclusão de uma nova funcionalidade, depuração ou otimização;
- As características específicas da tarefa como o código editado, código visualizado ou dependências do código;
- As atividades anteriores do usuário ou o rastro do usuário, como artefatos visualizados e artefatos explicitamente recomendados.

De forma genérica, os sistemas recomendadores de tarefa apresentam três funcionalidades principais:

- Um mecanismo de coleta de dados para coletar informações do processo de desenvolvimento e artefatos em um modelo de dados;
- Um motor de recomendação para analisar o modelo de dados e gerar recomendações a partir dele;
- Uma interface para disparar o ciclo de recomendações e apresentar os resultados.

Atualmente, os sistemas de recomendação são considerados sistemas que representam o estado da arte em ferramentas de desenvolvimento, porém, mesmo assim, eles apresentam algumas limitações. As duas principais limitações são a falta de informações suficientes para

permitir tomada de decisões relevantes pelo sistema, quando se inicia o uso, por possuir pouca informação sobre o projeto que está sendo executado. Para esta limitação, uma solução é prover o sistema de recomendação com dados de projetos anteriores. A outra limitação se refere ao RSSE não conseguir prever o que o desenvolvedor necessita em um determinado momento. O sistema recomendador apenas sugere o que normalmente seria utilizado naquela situação.

Os sistemas de recomendação atual focam principalmente apontamentos relacionadas ao próprio processo de codificação, principalmente sugestões de código fonte. Porém, os sistemas de recomendação poderiam auxiliar o desenvolvedor em outros aspectos como a recomendação de medidas de qualidade, ferramentas, gestão de projetos e até mesmo a recomendação de tarefas a serem desenvolvidas.

3 TRABALHOS RELACIONADOS

Este capítulo realiza uma análise do estado da arte no uso de dados biométricos na área da Engenharia de Software em especial no relacionamento dos dados biométricos com as atividades realizadas durante o processo de desenvolvimento de software. Apesar de terem sido lançados diversos trabalhos nas últimas décadas sobre o uso de dados biométricos, não existe nenhum estudo identificado que apresenta um relacionamento entre os diferentes dados biométricos que podem ser coletados do desenvolvedor e os diferentes aspectos das atividades a serem mensurados (compreensão de código, emoções, expertises do desenvolvedor, dificuldade das tarefas, interrupções) que podem ser utilizadas em sistemas de recomendação, como o proposto neste estudo.

Além disso, torna-se relevante identificar os desafios, tendências e oportunidades de pesquisa a partir da análise dos estudos já realizados. Para abordar estes pontos, o capítulo atual propõe um mapeamento sistemático da literatura (SMS) cuja etapas, estratégia de pesquisa e validação são baseadas em diretrizes conhecidas (PETERSEN; VAKKALANKA; KUZNIARZ, 2015; PETERSEN ET AL., 2008; ALI; USMAN, 2018a).

Desse modo, um mapeamento sistemático da literatura foi desenvolvido seguindo um protocolo cuidadosamente elaborado. O capítulo está organizado da seguinte forma: A Seção 3.1 apresenta a metodologia do estudo. Após, a Seção 3.2 demonstra os resultados através de resposta às questões de pesquisa formuladas na Seção anterior. Por sua vez, a Seção 3.3 descreve as oportunidades de pesquisa enquanto a Seção 3.4 elucida as ameaças à validade do estudo realizado. Por fim, a Seção 3.5 apresenta a conclusão da pesquisa realizada.

3.1 Plano de Pesquisa

Esta seção tem como objetivo descrever o protocolo de revisão. Como mencionado anteriormente na Seção 3, foi escolhido método de mapeamento sistemático para a realização desta pesquisa. De acordo com (COOPER, 2016), essa metodologia tende a produzir descobertas mais confiáveis, reduzindo o viés por meio de um rigoroso processo de revisão. A Figura 1 mostra uma visão geral do processo de revisão sistemática adotado. Composto por três fases, nas quais as atividades são executadas para criar artefatos, esse processo serve como um guia sobre como avançar com a revisão. A fase de planejamento (Seção 3.1) abrange todos os procedimentos para o processo de revisão. A fase de condução (Seção 3.1.5) detalha passo a passo como uma amostra de trabalhos potencialmente relevantes foi obtida e como essa amostra foi filtrada para identificar um conjunto de trabalhos representativos. A fase de resultados (Seção 3.2) se concentra em relatar os resultados, traçando algumas tendências e destacando alguns desafios que podem ser explorados pela comunidade científica.

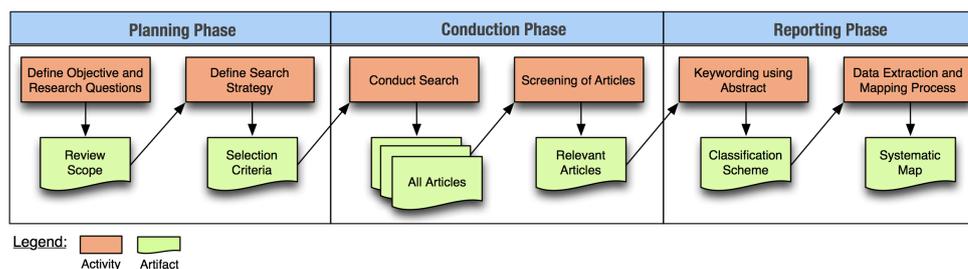


Figura 1 – Visão geral do processo de um mapeamento sistemático.

Fonte: Adaptado de (PETERSEN ET AL., 2008).

3.1.1 Objetivos e Questões de Pesquisa

Esta seção procura definir o objetivo e as questões de pesquisa para que esta revisão possa fornecer um escopo claro e objetivo.

Objetivo. O objetivo é apresentar uma visão geral da literatura existente, classificando os artigos publicados, discutindo as lacunas e apresentando algumas direções de pesquisa promissoras. Esta análise procura produzir uma síntese quantitativa dos resultados previamente relatados na literatura. Lau et al. (LAU; IOANNIDIS; SCHMID, 1997) reforçam que, combinando descobertas de um grande conjunto de estudos, é possível aumentar o poder estatístico e fornecer respostas que nenhum único estudo pode oferecer.

Questões de pesquisa. Foram definidas nove questões de pesquisa (QP) para abordar adequadamente os diferentes fatos desse objetivo. A Tabela 2 apresenta essas QPs, incluindo cinco Questões Gerais (GQ), uma Questão Focada (FQ) e três Questões Estatística (SQ). As GQs buscam revelar quais dados biométricos estão auxiliando nas atividades de desenvolvimento de software. As FQs visam identificar como determinados sensores biométricos são usados para melhorar a produtividade dos desenvolvedores durante as atividades de desenvolvimento de software. Por fim, o objetivo das SQs é divulgar dados estatísticos sobre onde e com que frequência os estudos foram publicados ao longo dos anos. Petersen et al. (PETERSEN; VAKKALANKA; KUZNIARZ, 2015; PETERSEN ET AL., 2008) mencionam que o mapeamento sistemático da literatura não tem como objetivo apontar com precisão problemas e soluções disponíveis. Em vez disso, o mapeamento sistemático deve descobrir poucos tópicos de pesquisa explorados, explorando tendências e lacunas ainda não reveladas, criando assim um mapa sistemático. Assim, as questões de pesquisa em um mapeamento sistemático da literatura precisam ser genéricas como as apresentadas na Tabela 2. Depois de definir o objetivo e as questões de pesquisa, a seção a seguir determina a estratégia de pesquisa.

3.1.2 Estratégias de Busca

A estratégia de busca baseia-se no método de busca automatizado seguindo diretrizes empíricas bem conhecidas (PETERSEN; VAKKALANKA; KUZNIARZ, 2015; PETERSEN ET AL.,

Tabela 2 – Lista de questões de pesquisa investigadas.

Referência	Questão
<i>Questões gerais</i>	
GQ1	Como seria a taxonomia para classificação de dados biométricos em tarefas de desenvolvimento de software?
GQ2	Quais são as medidas do corpo usadas para suportar tarefas de desenvolvimento de software?
GQ3	Quais fatores relacionados aos dados biométricos podem influenciar as tarefas de desenvolvimento de software?
GQ4	Quais tarefas diárias são comumente realizadas no contexto do desenvolvimento de software?
GQ5	Quais são os métodos de pesquisa utilizados para avaliar os estudos?
<i>Questões focadas</i>	
FQ1	Como os dados biométricos podem ser usados para melhorar a produtividade dos desenvolvedores?
<i>Questões estatísticas</i>	
SQ1	Qual é o número e tipo de publicações por ano?
SQ2	Quais países apresentaram publicações na área de pesquisa?
SQ3	Quem são os principais autores que publicaram artigos sobre dados biométricos na Engenharia de Software?

Fonte: Elaborado pelo autor.

2008; ALI; USMAN, 2018b), que auxiliaram a determinar uma string de busca embasada e identificar as bases eletrônicas efetivas.

String de busca. A string de busca é formada por termos identificados como *termos principais*, complementados pelos seus *sinônimos* mais relevantes. A string de busca desempenha um papel fundamental para identificar uma lista de estudos representativos para que o mapeamento sistemático da literatura possa ser formulado. Para este mapeamento, os termos principais são *Biometric*, *Device*, *Software* e *Tasks*. Os sinônimos para estes termos foram determinados com base no dicionário Collins e palavras chaves encontradas em trabalhos relacionados. A Tabela 3 mostra os principais termos e os seus sinônimos.

A string de busca foi produzida com base na combinação destes termos. Foram seguidos cinco passos para a definição da string de busca: (1) especificar os termos principais; (2) definir termos alternativos, sinônimos ou termos relacionados aos termos principais; (3) verificar se os termos principais estão contidos em artigos publicados anteriormente; (4) associar sinônimos, termos alternativos ou termos relacionados aos termos principais com o operador Lógico “OR”; e (5) relacionar os termos principais com o operador Lógico “AND”. A string de busca que

Tabela 3 – Lista dos termos principais e seus sinônimos.

Termos principais	Sinônimos
<i>Biometric</i>	<i>Biometric, Psychophysiological, Metric</i>
<i>Device</i>	<i>Sensor, Biosensor, Device</i>
<i>Software</i>	<i>Software Development, Software Engineering</i>
<i>Tasks</i>	<i>Task, Assignment, Work, Productivity</i>

Fonte: Elaborado pelo autor.

produziu os resultados mais significativos é apresentada a seguir:

(Biometric OR Psycho-physiological OR Metric) AND
(Sensor OR Biosensor OR Device) AND
(Software Development OR Software Engineering) AND
(Task OR Assignment OR Work OR Productivity)

Em algumas bases eletrônicas foi necessário ajustar a string de busca para ser possível chegar no número de artigos apresentados neste capítulo. A Tabela 4 apresenta em detalhes os ajustes realizados na string de busca e outros filtros específicos para cada base de dados eletrônica selecionada para este estudo. Além disso, os critérios de inclusão apresentados na Seção 3.1.3 também foram aplicados para reduzir o número de resultados da pesquisa.

Ali et al. (ALI; USMAN, 2018b) destacaram a importância do uso de um conjunto de estudos conhecidos para serem utilizados como base para avaliar o resultado obtido pela pesquisa em cada biblioteca eletrônica. Para este estudo cinco artigos foram utilizados como conjunto de estudos conhecidos (FRITZ ET AL., 2014; WROBEL, 2018; MÜLLER; FRITZ, 2016; CRK; KLUTHE; STEFIK, 2015; SIEGMUND ET AL., 2014). Esses artigos foram encontrados a partir do conhecimento prévio dos autores sobre o tema. Como resultado, todos os trabalhos utilizados como base de conhecimento foram identificados durante pesquisas realizadas em bibliotecas eletrônicas.

Bases eletrônicas. Após determinar a string de busca, o próximo passo foi identificar as bases eletrônicas que poderiam retornar estudos potencialmente relevantes. A Tabela 5 mostra as doze bases eletrônicas que foram usadas. Estas bases foram escolhidas por dois motivos. Primeiro, estas bases possuem um número elevado e representativo de artigos publicados, relacionados ao tópico pesquisado no mapeamento sistemático da literatura. Em segundo lugar, elas têm sido amplamente utilizadas em estudos anteriores de mapeamento sistemático (RODRÍGUEZ ET AL., 2017), o que significa que sua utilidade e eficácia já foram demonstradas.

3.1.3 Critérios de Inclusão e Exclusão

Esta seção procura estabelecer critérios para a filtragem dos artigos retornados pelas bases eletrônicas após aplicar a string de busca definida. Esses critérios de filtragem de artigos, ou seja, conhecidos como critérios de exclusão e inclusão, prescrevem regras para tornar o processo

Tabela 4 – Lista de ajustes de string de busca nas bases eletrônicas.

ACM Digital Library
(Biometric OR Psycho-physiological OR Metric) AND (Sensor OR Biosensor OR Device) AND (“Software Development” OR “Software Engineering”) AND (Task OR Assignment OR Work OR Productivity)
Google Scholar
(Biometric Sensor Task “Software Engineering” Psychophysiological OR Biosensor OR Productivity OR “Software Development”)
IEEE Xplore Digital Library
(Biometric OR Psycho-physiological OR Metric) AND (Sensor OR Biosensor OR Device) AND (“Software Development” OR “Software Engineering”) AND (Task OR Assignment OR Work OR Productivity)
Inspec
(Biometric OR Psycho-physiological OR Metric) AND (Sensor OR Biosensor OR Device) AND (“Software Development” OR “Software Engineering”) AND (Task OR Assignment OR Work OR Productivity)
Pubmed
(Biometric OR Psycho-physiological OR Metric) AND (Sensor OR Biosensor OR Device) AND (“Software Development” OR “Software Engineering”) AND (Task OR Assignment OR Work OR Productivity)
Science Direct
(Biometric OR Psycho-physiological) (Sensor OR Device) (“Software Development” OR “Software Engineering”) (Task OR Assignment OR Work OR Productivity)
Scopus
(Biometric OR Psycho-physiological OR Metric) AND (Sensor OR Biosensor OR Device) AND (“Software Development” OR “Software Engineering”) AND (Task OR Assignment OR Work OR Productivity)
Taylor & Francis OnLine
(Biometric OR Psycho-physiological OR Metric) AND (Sensor OR Biosensor OR Device) AND (“Software Development” OR “Software Engineering”) AND (Task OR Assignment OR Work OR Productivity)
Wiley Online Library
(Biometric OR Psycho-physiological) AND (Sensor OR Device) AND (“Software Development” OR “Software Engineering”) AND (Task OR Assignment OR Work OR Productivity)
Microsoft Academy
Biometric AND Sensor AND “Software Engineering” AND Task
Springer Link
Biometric OR Psycho-physiological Sensor OR Device “Software Development” OR “Software Engineering” Task OR Assignment OR Work OR Productivity
Semantic Scholar
text:(Biometric Sensor “Software Engineering” Task) AND abstract:(Biometric Sensor “Software Engineering” Task)

Fonte: Elaborado pelo autor.

de filtragem o mais objetivo e auditável possível, evitando qualquer tendência. Os seguintes Critérios de Exclusão (EC) foram usados para descartar trabalhos que:

- **EC1:** O título, abstract ou mesmo o seu conteúdo tenha sido relatado para os termos pesquisados, entretanto sem nenhum relacionamento semântico com o escopo do estudo.
- **EC2:** Não foram publicados em inglês, são descrições de patentes, ou podem ser considerados como um trabalho em estágio inicial, tipicamente representados por apenas sumário ou abstract.
- **EC3:** Não possuem similaridade com o tema da pesquisa, ou mesmo a intenção da pesquisa é oposta às questões abordadas pelas questões de pesquisa.
- **EC4:** Os estudos não são relacionados com as questões de pesquisa.
- **EC5:** Aparecem em duplicidade.

Tabela 5 – Lista de bases eletrônicas utilizadas na pesquisa.

Origem	Endereço eletrônico
ACM Digital Library	https://dl.acm.org
Semantic Scholar Library	https://www.semanticscholar.org/
Google Scholar	https://scholar.google.com
IEEE Xplore Digital Library	https://ieeexplore.ieee.org/Xplore/home.jsp
Inspec	http://digital-library.theiet.org
Microsoft Academy	https://www.microsoft.com/en-us/research
PubMed Central	https://www.ncbi.nlm.nih.gov/pmc
Science Direct	https://www.sciencedirect.com
Scopus	https://www.elsevier.com
Springer Link	https://link.springer.com
Wiley Online Library	https://onlinelibrary.wiley.com
Taylor & Francis OnLine	https://www.tandfonline.com

Fonte: Elaborado pelo autor.

- **EC6:** Não abordam questões relacionadas a dados biométricos em Engenharia de Software.

Razões para escolha dos critérios de exclusão. A utilização e eficácia destes critérios já foi validada em estudos anteriores (RODRÍGUEZ ET AL., 2017). Além disso, são listadas cinco razões pelas quais os estudos foram escolhidos. A primeira razão é a falta de qualquer sentido de considerar estudos que não possuam relação semântica com os tópicos de pesquisa abordados. Embora alguns estudos possam ter seu título ou resumo combinados com a string de pesquisa formulada, seu conteúdo pode não estar diretamente vinculado ao conteúdo explorado no estudo (EC1). Em segundo lugar, trabalhos em estágio inicial pouco ajudariam a elaborar o objetivo primário deste estudo, que é um mapeamento sistemático da literatura atual. Em vez disso, eles poderiam distorcer os dados coletados, prejudicando a identificação de tendências e lacunas (EC2). Terceiro, foi priorizada a identificação de estudos que pudessem contribuir em responder adequadamente às questões de pesquisa formuladas (EC3). Quarto, se o abstract de um artigo publicado ou mesmo o seu próprio texto, não relacionam fatos dos tópicos explorados nas questões de pesquisa, então ele não será considerado (EC4). Por fim, estudos duplicados foram descartados, pois não faz sentido contar um estudo mais de uma vez (EC5) ou não abordar questões relacionadas a dados biométricos em Engenharia de Software (EC6). Além disso, foram definidos quatro critérios para guiar a inclusão de trabalhos candidatos a serem analisados para este estudo. Os critérios de Inclusão (IC) são apresentados como segue:

- **IC1:** O estudo deve ter sido publicado em uma conferência, workshop, jornal científico, livro ou instituição educacional.
- **IC2:** O estudo deve estar relacionado ao uso de dados biométricos no contexto de execução de atividades de desenvolvimento de software.
- **IC3:** Trabalhos escritos, publicados ou disseminados em inglês.

Figura 2 – Formulário para extração de dados.

Fonte: Adaptado de (BISCHOFF ET AL., 2019).

- **IC4:** Estudos publicados de janeiro de 2002 até janeiro de 2019.

3.1.4 Extração de Dados

Esta seção tem por objetivo explicar como os dados dos trabalhos selecionados foram extraídos, e são apresentados na Seção 3.1.5.

Procedimentos de extração. Após ler cada um dos trabalhos cuidadosamente, os dados obtidos foram armazenados em uma planilha. A Figura 2 apresenta o formulário usado para extração dos dados para preencher a planilha, que foi inspirado em elementos já validados (FERNÁNDEZ-SÁEZ; GENERO; CHAUDRON, 2013). A Tabela 6 explica cada um dos rótulos apresentados no formulário de extração.

Razões para escolher este formulário. Este formulário foi utilizado por quatro motivos. Em primeiro lugar, orienta os autores, servindo de modelo para os procedimentos de extração, evitando vieses durante a identificação e armazenamento dos dados. Em segundo lugar, embora a extração manual de dados seja uma tarefa suscetível a erros, o formulário reduz esse risco padronizando a captura de dados. Terceiro, centralizar os dados em um formulário, auxilia na síntese, análise e plotagem das informações coletados. O uso do formulário resulta nas melhores descobertas em relação às questões de pesquisa, podendo ser geradas e plotadas. Além disso, a forma de extração aplicada é suportada por um esquema de classificação, mostrado na Tabela 6. Este esquema não apenas ajuda a gerar dados numéricos, nominais ou ordinais, mas também é essencial para qualquer tentativa de criar uma imagem consistente da literatura atual.

Tabela 6 – Esquema proposto para classificar os estudos selecionados.

Campo	Descrição
Título	Título do artigo.
Primeiro Autor	Autores do artigo. Esta informação é utilizada para responder a SQ3.
Origem	Bases onde os artigos foram encontrados. Esta questão será requerida para responder a SQ1.
Ano da Publicação	Ano da publicação usado para responder a SQ1.
País	País relacionado a instituição de ensino a qual o autor está associado. Esta informação é utilizada para responder SQ2.
Tipo de Publicação	Usado como critério de exclusão, permitindo apenas artigos publicados em conferências, revistas, livros, simpósios ou workshops.
Questões de Pesquisa	Lugar para responder as questões de pesquisa encontradas no artigo.
Data de Início	Local onde deve ser inserida a data em que o artigo foi pesquisado.
Total	Número total de artigos adicionados ao mapping.

Fonte: Elaborado pelo autor.

Para os resultados presentes neste estudo, o formulário apresentado na Figura 2 será complementado com a taxonomia a ser construída para responder a GQ1. Esta taxonomia será construída pela classificação dos dados biométricos utilizados nos estudos primários deste mapeamento e a sua relação com os tipos de atividades de desenvolvimento, sensores biométricos e melhorias obtidas com o uso destes dados.

3.1.5 Execução do Filtro de Pesquisa

Esta seção apresenta os detalhes de como o estudo foi conduzido, detalhando o processo usado para filtrar os artigos selecionados na Seção 3.1.2. O processo de filtragem dos artigos é composto de oito passos, nos quais os critérios de exclusão apresentados previamente, conforme mostrado na Figura 3 são aplicados. Cada um dos oito passos é descrito conforme segue:

Passo 1: Pesquisa Inicial. O objetivo deste passo é aplicar a string de busca as bases eletrônicas apresentadas na Tabela 5. No total, foram retornados 3.930 artigos potencialmente relevantes.

Passo 2: Remoção de Impurezas (EC1 & EC2). A remoção de impurezas consiste na aplicação dos critérios de exclusão 1 e 2 (EC1, EC2) detalhados na Seção 3.1.3. A remoção de impurezas refere-se à remoção de artigos que, embora tenham sido encontrados pela string de busca, pois os termos pesquisados estão presentes no artigo, esses termos não estão relacionados ao objetivo do presente estudo. Também foram removidos os estudos não escritos em inglês, patentes ou que não estejam completos. Neste estágio, 50,07% (1.968/3.930) dos artigos inicialmente pesquisados foram mantidos, totalizando 1.962 trabalhos excluídos.

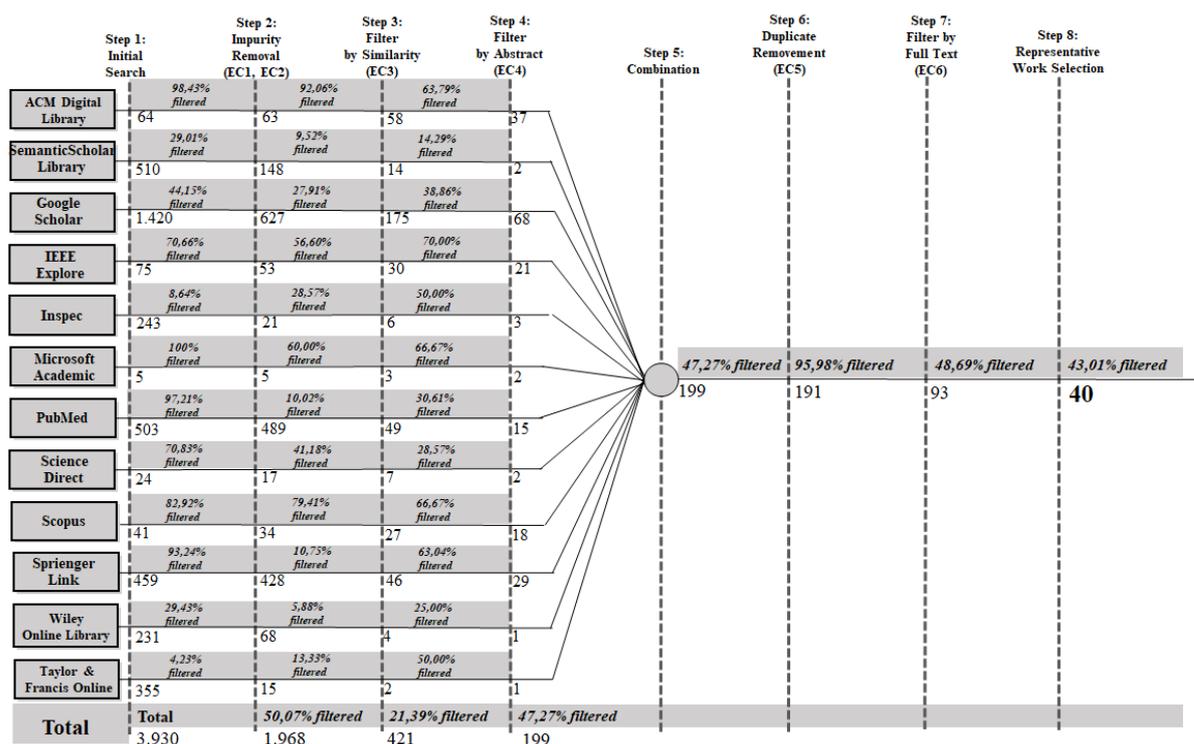


Figura 3 – Passos do processo de filtragem.

Fonte: Adaptado de (BISCHOFF ET AL., 2019).

Passo 3: Filtro por Similaridade (EC3). Neste passo 1.547 artigos foram removidos por não apresentarem similaridade em relação a string de busca. Somente 21,39% (421/1.968) dos artigos filtrados forma mantidos.

Passo 4: Filtro por Abstract (EC4). Este passo consiste em analisar o abstract dos trabalhos filtrados, verificando se ele tem relação com, pelo menos, uma das questões de pesquisa descritas na Tabela 2. Neste passo, 47,27% (199/421) dos artigos filtrados foram mantidos, com 222 estudos sendo removidos.

Passo 5: Combinação. Todos os artigos filtrados do passo anterior forma unificados, produzindo um total de 199 estudos.

Passo 6: Remoção de Duplicados (EC5). Estudos duplicados do passo 5 (Combinação) foram removidos. Somente oito artigos forma removidos neste passo, mantendo 95,98% (191/199) dos artigos filtrados.

Passo 7: Filtro pelo Texto Completo (EC6). Este passo consiste na leitura de todo o texto dos 191 artigos selecionados. Com a aplicação do EC6, 48,69% (93/191) dos artigos foram mantidos, excluindo os 98 artigos que não estavam adequados ao propósito deste estudo.

Passo 8: Seleção dos Trabalhos Representativos. Neste estágio do processo, os 93 artigos filtrados previamente foram lidos novamente e 43,01% dos estudos (40/93) foram selecionados como sendo os mais relevantes. Embora os 53 artigos excluídos tenham passado por todos os critérios de exclusão estabelecidos, eles não atenderam aos objetivos deste artigo relacionados na Seção 3.1.1. Alguns destes artigos excluídos utilizam dados biométricos nas atividades de

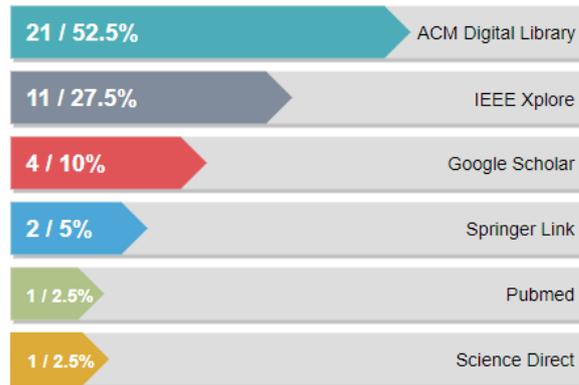


Figura 4 – Extratificação da busca baseada nas bases eletrônicas utilizadas.

Fonte: Elaborado pelo autor.

programação, mas eles não estão relacionados com a Engenharia de Software. Outros artigos apresentam revisão de literatura e forma relacionados na Seção 3.2.11, como trabalhos relacionados. Ao final deste passo, os estudos selecionados foram obtidos para este artigo.

A execução do processo de filtragem resultou em 40 artigos selecionados. Como estes trabalhos são os artigos escritos mais representativos em relação a este estudo, eles são chamados de estudos primários e sua lista é apresentada na Tabela 8.

Estratificação da pesquisa bases eletrônicas. Após discutir o processo de filtragem, o próximo passo é mostrar como os estudos selecionados foram estratificados nas bases de dados eletrônicas (Tabela 7) utilizadas. A Figura 4 mostra a estratificação dos estudos sobre as bases eletrônicas. Neste processo, após aplicar todos os critérios de inclusão e exclusão, somente seis das doze bases eletrônicas inicialmente pesquisadas retornaram estudos primários. Pode-se destacar que a *ACM Digital Library* foi a biblioteca digital com maior número de estudos (52,5%), seguida pela biblioteca *IEEE Explore* com 27,5%. A Tabela 7 a mesma estratificação apresentada na Figura 4, mas com detalhes de quais estudos foram identificados em cada base eletrônica. Estes estudos são apresentados em detalhes na Tabela 8.

Embora 12 bases eletrônicas tenham sido consideradas no estudo, em 6 destas bases, após aplicar o processo de filtragem, nenhum artigo foi mantido as quais são: *Semantic Scholar Library*, *Inspec*, *Microsoft Academic*, *Scopus*, *Wiley Online Library*, *Taylor & Fancis Online*. Por este motivo estas bases eletrônicas não foram apresentadas na Figura 4.

3.2 Resultados

Esta seção tem por objetivo apresentar as descobertas relacionadas às questões de pesquisa (Tabela 1), após examinar os 40 estudos primários (Tabela 8). Para facilitar o entendimento das descobertas identificadas pelas questões de pesquisa, foram empregadas representações gráficas e processamento numéricos conforme segue.

Tabela 7 – Extratificação da pesquisa com estudos primários.

Biblioteca Digital	Quantidade	Percentual	Lista de estudos primários
ACM Digital Library	21	52,5%	A01, A03, A05, A10, A14, A16, A17, A20, A22, A23, A24, A25, A26, A28, A29, A30, A31, A33, A35, A37, A39
IEEE Xplore	11	27,5%	A04, A06, A08, A09, A13, A15, A18, A21, A27, A32, A34
Google Scholar	4	10%	A02, A07, A12, A40
Springer Link	2	5%	A11, A36
Pubmed	1	2,5%	A38
Science Direct	1	2,5%	A19

Fonte: Elaborado pelo autor.

3.2.1 GQ1: Como seria a taxonomia para classificação de dados biométricos em tarefas de desenvolvimento de software?

Os estudos primários obtidos revelaram uma ampla e diversificada pesquisa sobre o uso de dados biométricos para apoiar atividades de Engenharia de Software. No entanto, tais pesquisas não podem ser classificadas, a menos que sejam acompanhadas por um esquema intuitivo de classificação. Por isso, foi construída uma taxonomia de apoio para auxiliar pesquisadores e profissionais na realização de atividades de classificação.

Para a questão de pesquisa GQ1, procurou-se definir uma taxonomia, conforme mostrado na Figura 5, com base nos artigos selecionados e analisados. Após realizar uma pesquisa na literatura por taxonomias relacionadas ou mesmo processos para construí-la, não foi encontrada nada que pudesse ser utilizado, por este motivo, a taxonomia foi definida com base nas seguintes etapas:

- **Passo 1: identificação dos principais grupos de termos.** Primariamente foram definidos os cinco grupos principais de termos (detalhados na Tabela 9) relacionados aos estudos que utilizam dados biométricos em tarefas de engenharia de software. Esses grupos macro estão estreitamente relacionados às questões de pesquisa: (1) medidas corporais (GQ2), (2) fatores (GQ3), (3) tarefas (GQ4), (4) métodos de pesquisa (GQ5) e (5) objetivos de pesquisa (FQ1).
- **Passo 2: detalhando os macro grupos identificados.** Os estudos identificados foram examinados profundamente para fornecer informações detalhadas sobre os macro grupos. Este processo foi realizado progressivamente a medida que as questões de pesquisa (GQ2-5 e FQ1) eram respondidas.
- **Passo 3: montagem e estilização da taxonomia.** Depois de ter os macro grupos e seus detalhes, o próximo passo foi montar a taxonomia em si e definir questões de estilo, incluindo linhas, cores, etc. O estilo da taxonomia foi inspirado pelo estudo proposto por Aghajani (AGHAJANI ET AL., 2019).

Tabela 8 – Artigos de referência.

Identificador	Autor, Ano	Publicador	Tipo
A01	Fritz et al, 2014 (FRITZ ET AL., 2014)	ACM	Conferência
A02	M. Gonçalves et al, 2011 (GONÇALVES; SOUZA; GONZÁLEZ, 2011)	J.UCS	Revista
A03	K. Kevic et al, 2015 (KEVIC ET AL., 2015)	ACM	Conferência
A04	W. Maalej et al, 2009 (MAALEJ; HAPPEL, 2009)	IEEE	Conferência
A05	A. Meyer et al, 2014 (MEYER ET AL., 2014)	ACM	Conferência
A06	A. N. Meyer et al, 2017 (MEYER ET AL., 2017)	IEEE	Revista
A07	S. C. Muller, 2016 (MÜLLER, 2016)	ZORA	Livro
A08	S. C. Muller et al, 2015 (MÜLLER; FRITZ, 2015)	IEEE	Conferência
A09	D. Perry et al, 2002 (PERRY; STAUDENMAYER; VOTTA, 2002)	IEEE	Revista
A10	J. Siegmund et al, 2017 (SIEGMUND ET AL., 2017)	ACM	Conferência
A11	T. Stankevic et al, 2011 (STANKEVIC ET AL., 2011)	Sprienger	Livro
A12	M. Wrobel, 2018 (WROBEL, 2018)	Applied Science	Revista
A13	M. Wrobel, 2013 (WROBEL, 2013)	IEEE	Conferência
A14	M. Züger et al, 2015 (ZÜGER; FRITZ, 2015)	ACM	Conferência
A15	J. Siegmund, 2016 (SIEGMUND, 2016)	IEEE	Conferência
A16	S. C. Müller et al, 2016 (MÜLLER; FRITZ, 2016)	ACM	Conferência
A17	I. Crk et al, 2015 (CRK; KLUTHE; STEFIK, 2015)	ACM	Revista
A18	T. Fritz et al, 2016 (FRITZ; MULLER, 2016)	IEEE	Conferência
A19	K. Kevic et al, 2017 (KEVIC ET AL., 2017)	Science Direct	Revista
A20	T. Nakagawa et al, 2014 (NAKAGAWA ET AL., 2014)	ACM	Conferência
A21	A. Begel, 2016 (BEGEL, 2016)	IEEE	Workshop
A22	J.P. Ostberg et al, 2017 (OSTBERG ET AL., 2017)	ACM	Workshop
A23	M. Konopka, 2015 (KONOPKA, 2015)	ACM	Conferência
A24	M. Züger et al, 2018 (ZÜGER ET AL., 2018)	ACM	Conferência
A25	R. Bednarik et al, 2006 (BEDNARIK; TUKIAINEN, 2006)	ACM	Simpósio
A26	D. Latoza et al, 2006 (LATOZA, THOMAS; VENOLIA; DELINE, 2006)	ACM	Conferência
A27	B. Sharif et al, 2010 (SHARIF; MALETIC, 2010)	IEEE	Conferência
A28	J. Siegmund et al, 2012 (SIEGMUND ET AL., 2012)	ACM	Simpósio
A29	H. Glücker et al, 2014 (GLÜCKER ET AL., 2014)	ACM	Conferência
A30	P. Rodeghero et al, 2014 (RODEGHERO ET AL., 2014)	ACM	Conferência
A31	J. Siegmund et al, 2014 (SIEGMUND ET AL., 2014)	ACM	Conferência
A32	T. Busjahn et al, 2015 (BUSJAHN ET AL., 2015)	IEEE	Conferência
A33	C. Palmer et al, 2016 (PALMER; SHARIF, 2016)	ACM	Simpósio
A34	J. Duraes et al, 2016 (DURAES ET AL., 2016)	IEEE	Simpósio
A35	B. Floyd et al, 2017 (FLOYD; SANTANDER; WEIMER, 2017)	ACM	Conferência
A36	S. Lee et al, 2017 (LEE ET AL., 2017)	Sprienger	Revista
A37	S. Fakhoury, 2018 (FAKHOURY, 2018)	ACM	Revista
A38	J. Castelhana et al, 2016 (CASTELHANO ET AL., 2018)	Pubmed	Revista
A39	N. Peitek et al, 2018 (PEITEK ET AL., 2018a)	ACM	Simpósio
A40	N. Peitek et al, 2018 (PEITEK ET AL., 2018b)	ACM	Simpósio

Fonte: Elaborado pelo autor.

Tabela 9 – Taxonomia para classificação de dados biométricos.

Grupo	Descrição
Dados biométricos	Providenciam métricas relacionadas às características humanas usadas para suportar os desenvolvedores de software nas suas atividades diárias (ver GQ2 na Seção 3.2.2)
Fatores	Importantes fatores de influência que impactam na execução das atividades diárias realizadas pelos desenvolvedores (ver GQ3 na Seção 3.2.3)
Atividades	Principais atividades diárias performadas pelos desenvolvedores (ver GQ4 na Seção 3.2.4)
Objetivos de pesquisa	Objetivo principal do trabalho realizado para aumentar o conhecimento sobre como melhorar a produtividade do desenvolvedor, gerenciando riscos, principalmente os relacionados a problemas que abrangem a compreensão de código, o estresse e a emoção, entre outros. (ver FQ1 na Seção 3.2.6)
Método de pesquisa	Estratégias empíricas usadas para planejar, executar e avaliar os estudos (ver GQ5 na Seção 3.2.7)

Fonte: Elaborado pelo autor.

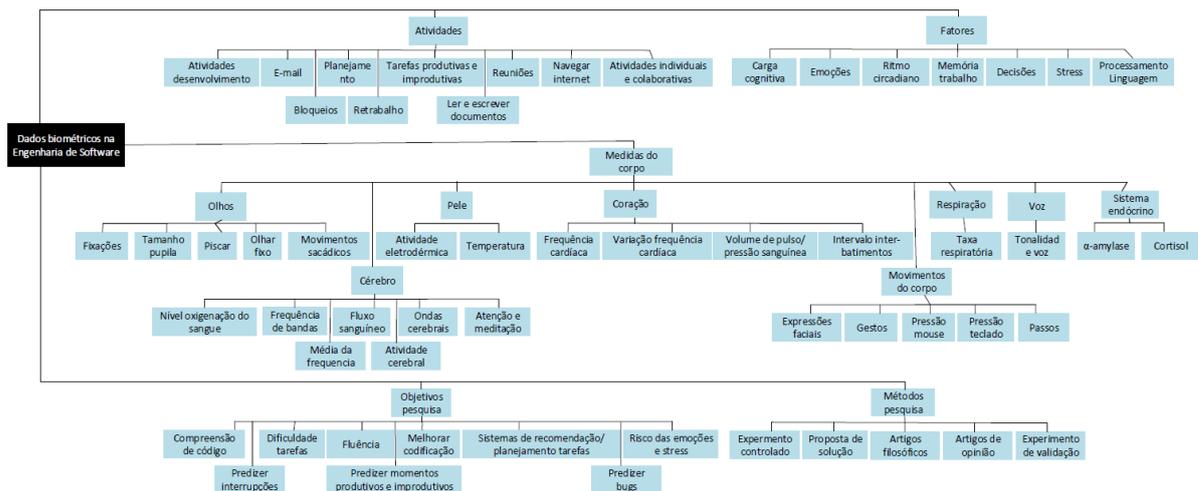


Figura 5 – Uma taxonomia que suporta a classificação dos estudos que usam dados biométricos em atividades de desenvolvimento de software.

Fonte: Adaptado de (AGHAJANI ET AL., 2019).

A análise da Figura 5 junto com a Tabela 9 torna possível compreender o relacionamento entre os principais elementos da taxonomia proposta em relação as questões de pesquisa deste estudo. Entretanto esta taxonomia é a primeira questão geral de pesquisa (GQ1), mas ela foi a última a ser respondida, pois ela necessitava da resposta das demais questões para a sua correta elaboração.

3.2.2 GQ2: Quais são os tipos de dados biométricos usados para suportar tarefas de desenvolvimento de software?

A GQ2 procura identificar o conjunto de medidas do corpo que normalmente são empregadas para suportar atividades de desenvolvimento de software. Para resolver a esta questão de pesquisa, foram analisados todos os estudos primários que exploram medidas corporais no contexto de execução das atividades diárias dos desenvolvedores de software, as quais são sumarizadas na Tabela 10. Ao examinar os artigos primários, foi possível localizar uma lista variada de tipos de métricas relacionadas a medidas corporais, incluindo aspectos físicos, comportamentais e fisiológicos. Cada registro geralmente reflete uma visão parcial de um sensor biométrico, geralmente sem permitir que os pesquisadores controlem e interajam com essas medições de forma integrada.

Existem vários tipos de dados biométricos envolvidos em quantificar as características dos desenvolvedores, como as ondas cerebrais, batimentos cardíacos e temperatura. Oito categorias foram propostas para um enquadramento mais refinado dos tipos de dados, incluindo Olhos, Pele, Cérebro, Coração, Respiração, Movimentos Corporais, Voz e Sistema Endócrino. Esses tipos de dados se referem a métricas relacionadas à características fisiológicas e comportamentais para rotular e descrever desenvolvedores de software.

Uma descoberta interessante é que a maioria dos estudos utiliza apenas um tipo de dado biométrico para realizar a análise das atividades realizadas pelos desenvolvedores (42,5%, 17/40), enquanto apenas (35%, 14/40) empregam informações de diferentes partes do corpo. A maioria dos estudos utiliza dados biométricos relacionados aos olhos (47,5%, 19/40), cérebro (40%, 16/40) e pele (22,5%, 9/40). Essas medidas são as mais utilizadas pelos pesquisadores, pois estão associadas ao sistema nervoso autônomo e, portanto, não podem ser controladas tão facilmente, garantindo maior precisão nos experimentos. Apenas alguns estudos foram encontrados utilizando movimentos corporais (10%, 4/40) e somente um estudo foi encontrado relacionando a voz e o sistema endócrino, correspondendo a 2,5% respectivamente.

Considerando-se os dados biométricos individualmente, a fixação dos olhos corresponde aos dados biométricos mais utilizados na análise das atividades do desenvolvedor (35%, 14/40), seguido do nível de oxigênio no sangue do cérebro (22,50%, 9 / 40) e a resposta galvânica da pele (22,50%, 9/40).

Tabela 10 – Medidas do corpo usadas para suportar atividades de desenvolvimento.

Grupo	Medidas do Corpo	Quantidade	Percentual	Lista de estudos primários
Olhos	Fixação	14	35%	A01, A07, A08, A18, A21, A25, A27, A29, A30, A32, A33, A37, A39, A40
	Tamanho da pupila	7	17,5%	A01, A07, A08, A18, A21, A36, A39
	Piscar dos olhos	6	15%	A01, A07, A08, A14, A18, A39
	Olhar fixo	6	15%	A03, A19, A23, A30, A32, A39
	Movimentos sacádicos	6	15%	A01, A21, A27, A36, A39, A40
Pele	Atividade eletrodérmica (phasic/tonic)	9	22,5%	A01, A07, A08, A11, A12, A14, A16, A18, A21
	Temperatura	5	12,5%	A07, A08, A11, A16, A18
Cérebro	Nível oxigenação do sangue	9	22,5%	A10, A15, A28, A31, A34, A35, A38, A39, A40
	Frequência de bandas	4	10%	A07, A08, A18, A36
	Média de frequências	3	7,5%	A07, A08, A18
	Ondas cerebrais	3	7,5%	A01, A15, A17
	Atenção & meditação	2	5%	A08, A14
	Fluxo sanguíneo	2	5%	A20, A37
	Atividades cerebrais	1	2,5%	A14
Coração	Frequência cardíaca	8	20%	A07, A08, A11, A12, A14, A16, A18, A24
	Variação da frequência cardíaca	5	12,5%	A07, A08, A16, A18, A24
	Pulso/pressão sanguínea	5	12,5%	A07, A08, A11, A14, A18
	Intervalo inter-batimentos	1	2,5%	A14
Respiração	Taxa respiratória	5	12,5%	A07, A12, A16, A18, A24
Movimentos do corpo	Expressões faciais	1	2,5%	A12
	Gestos	1	2,5%	A12
	Pressão mouse	2	5%	A11, A21
	Pressão teclado	1	2,5%	A12
	Passos	1	2,5%	A24
Voice	Tonalidade da voz	1	2,5%	A12
Sistema endócrino	Cortisol	1	2,5%	A22
	α -amylase	1	2,5%	A22

Fonte: Elaborado pelo autor.

Tabela 11 – Fatores relatados a dados biométricos que podem influenciar nas atividades de desenvolvimento de software.

Fatores fisiológicos	Quantidade	Percentual	Lista de estudos primários
Carga cognitiva	20	50%	A01, A06, A10, A14, A15, A16, A17, A18, A19, A20, A22, A25, A27, A28, A31, A35, A36, A37, A39, A40
Emoções	9	22,5%	A07, A08, A11, A12, A13, A14, A18, A21, A22
Ritmo circadiano	2	5%	A06, A24
Memória de trabalho	2	5%	A17, A34
Decisões	2	5%	A34, A38
Stress	1	4,17%	A22
Processamento de linguagem	1	4,17%	A34

Fonte: Elaborado pelo autor.

3.2.3 GQ3: Quais fatores relacionados aos dados biométricos podem influenciar as tarefas de desenvolvimento de software?

A GQ3 foca na identificação dos fatores que podem ser medidos através de dados biométricos coletados e identificados pela GQ2 que influenciam as atividades realizadas pelos desenvolvedores. A Tabela 11 apresenta os fatores que são relacionados às atividades performadas pelos desenvolvedores.

A carga cognitiva corresponde a 50% dos artigos pesquisados (20/40), sendo o aspecto fisiológico mais utilizado juntamente com as emoções (22,5%, 9/40). O aspecto da carga cognitiva é amplamente utilizado nos artigos, pois a atividade de codificação é uma tarefa intelectual que requer raciocínio e compreensão adequadas à programação que será realizada. Os estudos relatam que uma alta carga cognitiva está associada a dificuldades de entender e realizar uma tarefa (FRITZ ET AL., 2014; CRK; KLUTHE; STEFIK, 2015). As emoções também são estudadas porque estão associadas ao esforço empregado na execução de uma tarefa (MÜLLER; FRITZ, 2015). O ritmo circadiano tem recebido mais atenção apenas em estudos recentes, estando associado a maior atenção e produtividade em diferentes momentos do dia. Embora o estresse seja um estado emocional, ele recebeu uma categoria em separado das emoções, porque um estudo foi elaborado apenas analisando a influência do estresse na atividade de programação (OSTBERG ET AL., 2017). Da mesma forma a tomada de decisões, memória de trabalho e processamento de linguagem são processos cognitivos, porém foram categorizados separadamente por existirem estudos específicos direcionados para estas áreas.

Tabela 12 – Tarefas diárias normalmente realizadas no contexto do desenvolvimento de software.

<u>Categoria da tarefa</u>	<u>Quantidade</u>	<u>Percentual</u>	<u>Lista dos estudos primários</u>
Tarefas de desenvolvimento	37	92,5%	A01, A03, A04, A06, A07, A08, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20, A21, A22, A23, A24, A25, A26, A27, A28, A29, A30, A31, A32, A33, A34, A35, A36, A37, A38, A39, A40
E-mail	3	7,5%	A04, A06, A26
Planejamento	3	7,5%	A04, A06, A26
Reuniões	3	7,5%	A04, A06, A26
Navegação na internet	3	7,5%	A04, A06, A26
Ler e escrever documentação	3	7,5%	A04, A06, A26
Atividades colaborativas e individuais	1	2,5%	A02
Tarefas produtivas e improdutivas	1	2,5%	A05
Bloqueio de atividades	1	2,5%	A09
Atividades de retrabalho	1	2,5%	A09

Fonte: Elaborado pelo autor.

3.2.4 GQ4: Quais tarefas diárias são comumente realizadas no contexto do desenvolvimento de software?

A GQ4 identifica as atividades diárias realizadas pelos desenvolvedores. Estas atividades são apresentadas em detalhes na Tabela 12. Para auxiliar na identificação destas tarefas, Meyer (MEYER ET AL., 2014) conduziu uma pesquisa para identificar a percepção de produtividade pelos desenvolvedores. Esta pesquisa permitiu categorizar as atividades do desenvolvedor com a representatividade de cada tipo de tarefa durante o dia. Isso também permitiu categorizar estas atividades entre atividades produtivas e não produtivas sob o ponto de vista do desenvolvedor.

A maioria dos artigos estudados (95,5%, 37/40) somente generaliza as atividades de desenvolvimento, sem detalhar as mesmas, entretanto Meyer (MEYER ET AL., 2017), desenvolveu uma aplicação em sua pesquisa para monitorar as atividades diárias. Meyer (MEYER ET AL., 2014) chegou a uma categorização de atividades semelhantes às obtidas na pesquisa anterior. Ele também identificou uma grande fragmentação nas atividades diárias, além de identificar a influência do ritmo circadiano de cada desenvolvedor em relação à sua produtividade maior ou menor em determinados momentos do dia. De maneira similar, Maalej (MAALEJ; HAPPEL, 2009) detectou através dos textos usados nos *commits* de código fonte, os tipos de atividades realizadas pelos desenvolvedores, os quais foram os mesmos detectados por Meyer.

Outra importante classificação na qual as tarefas diárias podem ser submetidas está relacionada às atividades que exigem trabalho colaborativo ou atividades individuais e a intercalação dessas atividades durante o dia (GONÇALVES; SOUZA; GONZÁLEZ, 2011).

3.2.5 GQ5: Quais são os métodos de pesquisa utilizados para avaliar os estudos?

A GQ5 investiga os métodos de pesquisa utilizados nos estudos primários. De acordo com Wieringa (WIERINGA ET AL., 2006), os métodos de pesquisa podem ser classificados como: Experimento Controlado, Proposta de Solução, Artigos Filosóficos, Artigos de Opinião e Experimentos de Validação. Como diversos outros estudos (GONÇALES ET AL., 2019), (BISCHOFF ET AL., 2019), (GONÇALES ET AL., 2015) tem mostrado a utilidade desta classificação, ela também foi adotada neste estudo. A Tabela 13 apresenta a classificação dos estudos.

Tabela 13 – Classificação dos estudos primários baseada nos métodos de pesquisa.

Método de Pesquisa	Número de Estudos	Percentual	Estudos
Experimento Controlado	21	52.5%	A01, A06, A08, A11, A12, A07, A17, A18, A20, A22, A23, A24 A27, A28, A32, A34, A35, A37, A38, A39, A40
Proposta de Solução	10	25%	A01, A03, A07, A14, A25, A29, A30, A31, A33, A36
Artigo Filosófico	7	17.5%	A02, A04, A09, A13, A15, A19, A26
Artigo de Opinião	1	2.5%	A21
Experimento de Validação	1	2.5%	A10

Fonte: Elaborado pelo autor.

O método de pesquisa mais adotado é o Experimento Controlado (52,5%, 21/40), seguido do método de Proposta de Solução (25%, 10/40). Somente um dos estudos foi um Experimento de validação associado com a identificação do processo cognitivo durante a execução de tarefas de programação (SIEGMUND ET AL., 2017), demonstrando que a área de pesquisa ainda é nova. Uma direção de pesquisa interessante seria conduzir estudos de caso com profissionais em ambientes de desenvolvimento de software do mundo real.

3.2.6 FQ1: Como os dados biométricos podem ser utilizados para melhorar a produtividade dos desenvolvedores?

A FQ1 identifica como os dados biométricos podem ser usados para melhorar o processo de desenvolvimento de software, reduzindo os aspectos associados ao risco na atividade de desenvolvimento de software, como a compreensão de código, más práticas (code smells) e predição de bugs. A Tabela 14 apresenta um sumário do uso de dados biométricos identificados nos estudos relacionados com as atividades realizadas pelos desenvolvedores.

A compreensão do código é o assunto mais abordado pelos pesquisadores (37,5%, 15/40). Este tópico junto com a dificuldade em tarefas (20%, 8/40) estão sendo intensamente pesquisados por estarem relacionados com a atividade de programação, a qual é a principal atividade realizada pelos desenvolvedores, conforme mostrado na Tabela 12. A atividade de codifica-

Tabela 14 – Dados biométricos usados para melhorar a produtividade dos desenvolvedores.

Atividades de melhoria	Quantidade	Percentual	Lista de estudos primários
Compreensão de código	15	37,5%	A10, A15, A20, A25, A27, A28, A30, A31, A32, A34, A35, A37, A38, A39, A40
Dificuldade de tarefas	8	20%	A01, A08, A16, A17, A18, A19, A22, A36
Fluência	5	12,50%	A05, A06, A08, A18, A24
Melhorar a codificação	4	10%	A03, A23, A29, A32
Sistemas de recomendação/planejamento	4	10%	A02, A07, A11, A36
Riscos da emoção e stress	2	5%	A13, A22
Predizer interrupções	2	5%	A14, A18
Predizer momentos produtivos e improdutivos	1	2,5%	A06
Predizer bugs	1	2,5%	A34

Fonte: Elaborado pelo autor.

ção também está relacionada às melhorias de codificação (10%, 4/40), predição de bugs (2,5%, 1/40) e emoções e stress (5%, 2/40).

Por outro lado, os temas relacionados a organização e ordenação das atividades dos desenvolvedores foram menos pesquisados: Manter a fluência (12,5%, 5/40), Sistemas de recomendação e planejamento de tarefas (10%, 4/40), Predição de interrupções (5%, 2/40) e Predição de momentos produtivos e improdutivos (2,5%, 1/40).

3.2.7 SQ1: Qual é o número e tipo de publicações por ano?

O objetivo da SQ1 é revelar o número e o tipo de publicações dos estudos primários pelos anos. Esta classificação foi baseada no ano da publicação (i.e., de 2002 até 2019), tipo de publicação (revista, livro, conferência, workshop e simpósio) e o número de publicações pelos anos. A Figura 6 apresenta os dados obtidos de acordo com o critério mencionado.

A distribuição dos artigos nesta revisão sobre os anos ajuda a criar uma imagem do relacionamento da literatura em relação ao tempo. Pode-se identificar que a partir de 2014 houve um aumento no número de publicações que usam dados biométricos relacionados a Engenharia de Software e a partir dali este tema tem sido continuamente pesquisado. Isso demonstra que esta área de pesquisa é muito ativa e ainda em crescimento. A Figura 6 revela que existe uma preferência de publicações em conferências (55%, 22/40), correspondendo a mais da metade dos estudos primários. Em adição às conferências, muitos artigos foram publicados em revistas (20%, 8/40) e simpósios (15%, 6/40) e o restante está distribuído homogeneamente entre workshops e livros.

Esta questão também avalia os estudos primários em relação ao método de pesquisa. Os métodos de pesquisa foram classificados como: Experimento Controlado, Proposta de Solução, Artigos Filosóficos, Artigos de Opinião e Experimentos de Validação, como os categorizados por Wieringa (WIERINGA ET AL., 2006). A Tabela 13 apresenta a classificação dos estudos.

Tipo de Publicação	Percentual
Conferência	55.00% [22 de 40] Estudos Primários
Journal	20.00% [8 de 40] Estudos Primários
Simpósios	15.00% [6 de 40] Estudos Primários
Workshop	5.00% [2 de 40] Estudos Primários
Livros	5.00% [2 de 40] Estudos Primários

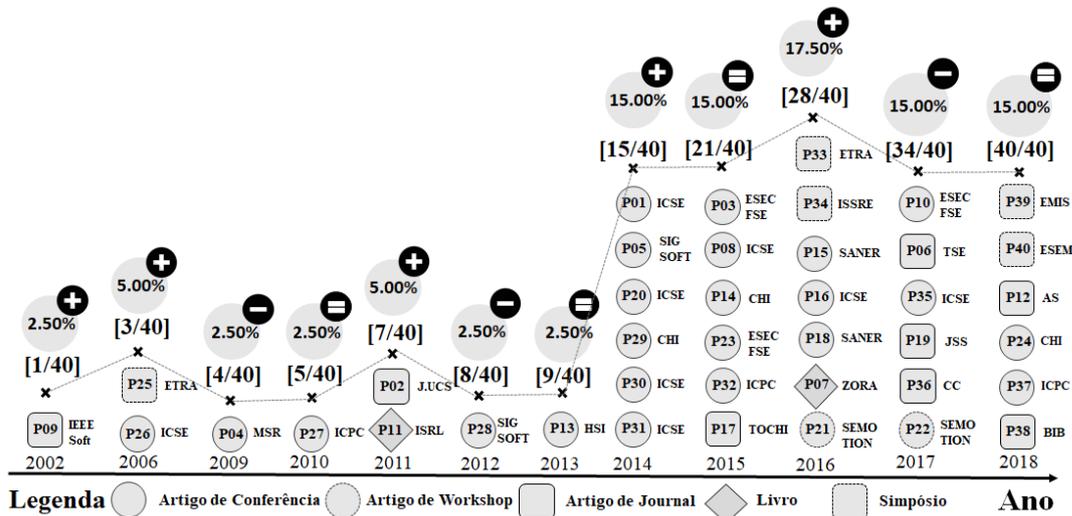


Figura 6 – Frequência de publicações.

Fonte: Adaptado de (BISCHOFF ET AL., 2019).

A maioria dos estudos primários foram Experimentos Controlados (52,5%, 21/40). Em segundo, foram Propostas de Solução (25%, 10/40). Somente um dos artigos foi um experimento de validação associado ao processo cognitivo durante a execução de tarefas de codificação (SI-EGMUND ET AL., 2017). Isto demonstra que esta área de pesquisa ainda é nova e requer replicação de experimentos em ambientes realísticos, como na indústria.

3.2.8 SQ2: Quais países apresentaram publicações na área de pesquisa?

A SQ2 apresenta a partir de quais países as publicações dos estudos primários se originaram. A análise dos países é relevante, pois permite identificar em quais regiões os autores estão realizando mais esforços na pesquisa do tema abordado.

A análise das publicações permitiu a criação de um mapa global apresentado na Figura 7. Este mapa revela que a maioria das publicações ocorre no hemisfério norte (97,5%, 39/40). A Suíça é o país com mais publicações (27,5%, 11/40), seguida pelos Estados Unidos (25%, 10/40) e Alemanha (22,5%, 9/40). Os países restantes, juntos contabilizam 25% das publicações, com 10 publicações. A Tabela 15 apresenta em detalhes os estudos primários organizados por países.



Figura 7 – Publicações por países.

Fonte: Elaborado pelo autor.

Tabela 15 – Publicações por países em detalhes.

País	Quantidade	Percentual	Lista de Estudos
Suíça	11	27,5%	A01, A03, A05, A06, A07, A08, A14, A16, A18, A19, A24
Estados Unidos	10	25%	A09, A17, A21, A26, A27, A30, A33, A35, A37, A40
Alemanha	9	22,5%	A04, A10, A15, A22, A28, A29, A31, A32, A39
Polónia	2	5%	A12, A13
Portugal	2	5%	A34, A38
Brasil	1	2,5%	A02
Finlândia	1	2,5%	A25
Japão	1	2,5%	A20
Lituânia	1	2,5%	A11
Eslováquia	1	2,5%	A23
Coreia do Sul	1	2,5%	A36

Fonte: Elaborado pelo autor.

3.2.9 SQ3: Quem são os principais autores que publicaram artigos sobre dados biométricos na Engenharia de Software?

Esta questão foi respondida a partir da análise dos autores relatados nos estudos primários. Nesta análise, apenas o principal e o segundo autor foram considerados, conforme apresentado na Tabela 16. Os demais autores não foram considerados pois usualmente as suas contribuições são menores.

A partir da Tabela 16 é possível identificar que alguns autores são ativos na área pesquisa, como Janet Siegmund (6 artigos), Thomas Fritz (6 artigos) e Sebastian Muller (5 artigos). Estes autores apresentam as seguintes características: *Janet Siegmund*: Doutor pesquisador na Universidade de Passau na Alemanha. Seus principais interesses são compreensão de código, engenharia de software experimental, fatores humanos na ciência da computação e ressonân-

Tabela 16 – Autores por publicação.

Autor	Primeiro Autor			Segundo autor		
	Quantidade	Percentual	Estudos primários	Quantidade	Percentual	Estudos primários
Janet Siegmund	4	10%	A10, A15, A28, A31	2	5%	A39, A40
Sebastian Muller	3	7,50%	A07, A08, A16	2	5%	A18, A24
Katja Kevic	2	5%	A03, A19	0	0%	-
Manuela Zuger	2	5%	A14, A24	0	0%	-
Michal R.Wrobel	2	5%	A12, A13	0	0%	-
Norman Peitek	2	5%	A39, A40	1	2,5%	A10
Thomas Fritz	2	5%	A01, A18	4	10%	A05, A08, A14, A16
André Meyer	2	5%	A05, A06	0	0%	-
Andrew Begel	1	2,5%	A21	1	2,5%	A01
Arturas Kaklauskas	1	2,5%	A11	0	0%	-
Bonita Sharif	1	2,5%	A27	0	0%	-
Benjamin Floyd	1	2,5%	A35	0	0%	-
Christopher Palmer	1	2,5%	A33	0	0%	-
Dewayaje E . Perry	1	2,5%	A09	0	0%	-
Hartmut Glücker	1	2,5%	A29	0	0%	-
Igor CRK	1	2,5%	A17	0	0%	-
J. Duraes	1	2,5%	A34	0	0%	-
Jan-Peter Ostberg	1	2,5%	A22	0	0%	-
Joao Castelhana	1	2,5%	A38	0	0%	-
Marcio Kuroki Goncalves	1	2,5%	A02	0	0%	-
Martin Konopka	1	2,5%	A23	0	0%	-
Paige Rodeghero	1	2,5%	A30	0	0%	-
Roman Bednarik	1	2,5%	A25	1	2,5%	A32
Sarah Fakhoury	1	2,5%	A37	0	0%	-
Seolhwa Lee	1	2,5%	A36	0	0%	-
Takao Nakagawa	1	2,5%	A20	0	0%	-
Teresa Busjahn	1	2,5%	A32	0	0%	-
Thomas D. LaToza	1	2,5%	A26	0	0%	-
Walid Maalej	1	2,5%	A34	0	0%	-
André Brechmann	0	0%	-	1	2,5%	A28
Bonita Sharif	0	0%	-	1	2,5%	A33
Braden M. Walters	0	0%	-	2	5%	A03, A19
Christian Kästner	0	0%	-	1	2,5%	A31
Cleudson R. B. de Souza	0	0%	-	1	2,5%	A02
Collin McMillan	0	0%	-	1	2,5%	A30
Danial Hooshyar	0	0%	-	1	2,5%	A36
Daniel Graziotin	0	0%	-	1	2,5%	A22
Felix Raab	0	0%	-	1	2,5%	A29
Gina Venolia	0	0%	-	1	2,5%	A27
H. Madeira	0	0%	-	1	2,5%	A34
Hans-Jorg Happel	0	0%	-	1	2,5%	A04
Isabel C. Duarte	0	0%	-	1	2,5%	A38
Laura E. Barton	0	0%	-	1	2,5%	A06
Markku Tukiainen	0	0%	-	1	2,5%	A25
Nancy .A. Staudenmayer	0	0%	-	1	2,5%	A09
Timothy Kluthe	0	0%	-	1	2,5%	A17
Tyler Santander	0	0%	-	1	2,5%	A35
Yasutaka Kamei	0	0%	-	1	2,5%	A20
Yuzhan Ma	0	0%	-	1	2,5%	A37

Fonte: Elaborado pelo autor.

cia magnética funcional (fmRI). *Sebastian Muller*: Pesquisador assistente na Universidade de Zurich durante o período de 2011 até 2016, concluindo seu pós doutorado em 2017. Também atuando como engenheiro de software na indústria até 2019. *Thomas Fritz*: Professor na Universidade de Zurich. Sua pesquisa está relacionada com a engenharia de software empírica e o uso de biometrias para melhor compreender e aumentar a produtividade dos desenvolvedores. Os demais autores apresentam menos publicações com no máximo três publicações cada autor.

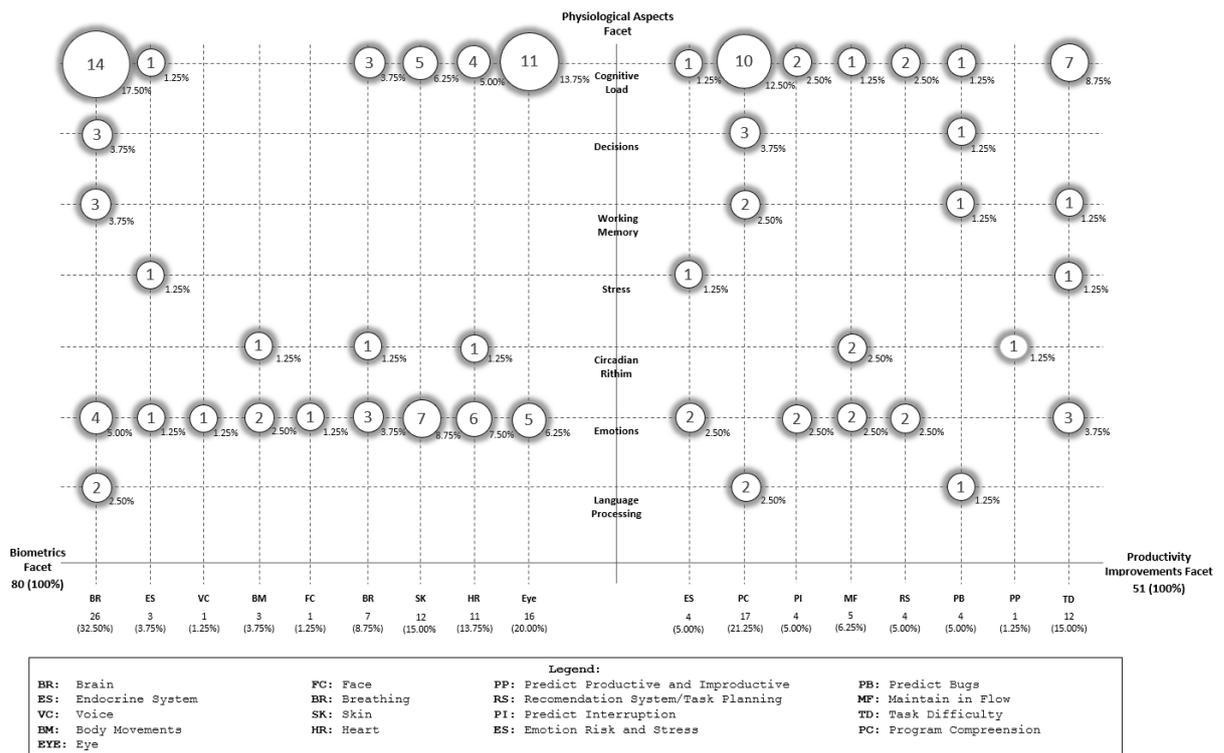


Figura 8 – Relacionamento entre fatores psicofisiológicos e melhoria de produtividade.

Fonte: Adaptado de (BISCHOFF ET AL., 2019).

3.2.10 Discussão Adicional

Esta seção apresenta uma discussão adicional ao fazer referência cruzada às questões de pesquisa GQ2, GQ3 e FQ1. A Figura 8 mostra a relação entre os dados biométricos e os fatores psicofisiológicos que influenciam a melhoria da produtividade. Cada bolha apresenta uma tripla (t_1 , t_2 , t_3) com os dados obtidos, onde t_1 representa as questões de pesquisa GQ2 e FQ1, t_2 é a questão de pesquisa GQ3 e t_3 consiste no número de estudos que exploram o cruzamento dessas perguntas. A partir das evidências quantitativas destacadas nesta figura, procura-se entender quais são os aspectos psicofisiológicos mais pesquisados em Engenharia de Software, identificados através de quais dados biométricos e aplicados sobre quais características podem melhorar a produtividade das tarefas diárias realizadas pelos desenvolvedores.

A Figura 8 mostra que a maioria dos estudos primários captura os dados biométricos obtidos pelos olhos, coração, pele e cérebro para analisar emoções e carga cognitiva, a fim de entender o processo de compreensão de código e dificuldade das tarefas. Os sensores associados ao cérebro e aos olhos foram usados principalmente para identificar a carga cognitiva (FRITZ ET AL., 2014; MÜLLER, 2016; SIEGMUND ET AL., 2017; ZÜGER; FRITZ, 2015; CRK; KLUTHE; STEFIK, 2015). Em muitos estudos, esses dois sensores foram usados juntos, produzindo maior precisão nos resultados (FRITZ ET AL., 2014; MÜLLER, 2016; ZÜGER; FRITZ, 2015; LEE ET AL., 2017; PEITEK ET AL., 2018a,b). Para analisar as emoções, foram relacionados sen-

sores a todas as partes do corpo, como mostra a Figura 8, no entanto, a maioria dos estudos utilizou medidas obtidas pela pele, coração, cérebro e olhos (OSTBERG ET AL., 2017; MÜLLER, 2016; MÜLLER; FRITZ, 2015; ZÜGER; FRITZ, 2015; WROBEL, 2018). Enquanto a carga cognitiva estava relacionada à compreensão e à dificuldade do código (FRITZ ET AL., 2014; SIEGMUND ET AL., 2017; MÜLLER; FRITZ, 2016; FRITZ; MULLER, 2016; NAKAGAWA ET AL., 2014), emoções e ritmo circadiano foram relacionados para interrupção, foco e planejamento de atividades de desenvolvimento (MEYER ET AL., 2017; MÜLLER, 2016; STANKEVIC ET AL., 2011; ZÜGER ET AL., 2018; FRITZ; MULLER, 2016).

Esse resultado sugere que a maioria das pesquisas na área concentram-se na compreensão de código e na dificuldade das tarefas, associados aos dados biométricos obtidos pelo cérebro e pelos olhos, que avaliam a carga cognitiva do indivíduo. A análise das emoções durante as atividades de desenvolvimento também é objetivo da pesquisa, mas com menos intensidade.

A SQ1 apresenta dois aspectos interessantes em relação ao local e ano de publicação. Através da Figura 6 é possível identificar que a partir de 2014 houve um avanço significativo no número de artigos publicados na área. Atribui-se que esse aumento esteja relacionado ao aumento da capacidade de processamento dos computadores, maior número de equipamentos vestíveis e melhorias nas técnicas de inteligência artificial. Essa questão estatística também aponta que a maioria dos artigos é publicada em conferências, sendo preferidas pelos autores de ciência da computação pelos seguintes motivos: 1) As conferências apresentam maior status, visibilidade e impacto; 2) As conferências têm critérios de seleção mais rígidos do que a maioria dos periódicos; 3) As conferências levam menos tempo para publicar do que a maioria dos periódicos. Em certas revistas, uma publicação pode levar anos para ser publicada e 4) As conferências têm um fator de novidade maior do que as revistas.

3.2.11 Trabalhos Relacionados

Após a análise do uso de dados biométricos na Engenharia de Software, realizou-se uma pesquisa dos trabalhos relacionados a este estudo, com o objetivo de identificar técnicas ou abordagens que de recomendação de tarefas de desenvolvimento de software. Nesta pesquisa foram identificados sete trabalhos relacionados (YU ET AL., 2019; JÚNIOR ET AL., 2018; MATTER; KUHN; NIERSTRASZ, 2009; ZHANG; LEE, 2012; ANVIK; MURPHY, 2011; ANVIK; HIEW; MURPHY, 2006; WU ET AL., 2011), apresentados conforme segue e na Seção 3.2.12.

Xu Yu et al. (YU ET AL., 2019) analisam um sistema de recomendação de desenvolvedores em *crowdsourcing* baseado em domínios, apresentando vantagens em relação a algoritmos anteriormente utilizados.

Júnior et al. (JÚNIOR ET AL., 2018) analisam e identificam um conjunto de atributos que são cruciais para melhorar as recomendações de integração de código fonte. A partir desta análise, os autores realizaram um experimento com 32 projetos de código aberto, aplicando

os atributos identificados na pesquisa, além daqueles considerados relevantes pelos autores. O experimento realizado conseguiu apresentar um sistema de recomendação que melhorou em 54% os índices obtidos pelas técnicas consideradas como o estado da arte. Desta forma, os autores trouxeram contribuições importantes para melhorar realização de tarefas de merge em projetos de código aberto que adotam um fluxo de trabalho baseado em *pull*.

Matter et al. (MATTER; KUHN; NIERSTRASZ, 2009) apresentam uma técnica de recomendação automática de desenvolvedores para a resolução de *bugs* através da identificação das expertises do desenvolvedor adequadas para a atividade. A identificação das expertises é feita através do cruzamento do vocabulário de palavras utilizados pelo profissional nas suas atividades anteriormente realizadas no código fonte em relação as palavras aplicadas no reporte do bug. Os autores executaram um experimento para validar a técnica, utilizando um período de 8 anos de histórico de código do desenvolvimento da plataforma Eclipse, conseguindo uma precisão de resultado de até 71%.

Zhang et al. (ZHANG; LEE, 2012) abordam a dificuldade de recomendar de forma adequada um desenvolvedor a uma tarefa. Como resultado, torna-se necessário reatribuir a tarefa, provocando aumento no tempo de resolução. Para auxiliar neste problema, os autores propõem um sistema de classificação e recomendação de desenvolvedores para tarefas de desenvolvimento de software, baseando-se nas redes sociais, no feedback de especialistas, além da eficiência de correções já realizadas pelo desenvolvedor e a experiência de cada um. Este sistema gera um ranking dos desenvolvedores associados a cada uma das tarefas a serem realizadas.

Anvik et al. (ANVIK; MURPHY, 2011) utilizam técnicas de aprendizagem de máquina para fazer recomendações. Os autores exploram a temática de como sugerir qual o desenvolvedor é mais adequado para a execução de uma atividade em específico. O resultado desta técnica foi uma precisão entre 70% e 98% em cinco projetos *open sources* em que a técnica foi aplicada.

Anvik et al. (ANVIK; HIEW; MURPHY, 2006): No estudo apresentado pelo autor, é apresentada uma abordagem semi-automatizada empregando um algoritmo de aprendizagem de máquina para priorizar registros de bug e sugerir alguns desenvolvedores para realizar a execução destas tarefas. Segundo o autor, os resultados da técnica proposta conseguiram atingir uma precisão de 57% e 64% nos projetos *open sources* Eclipse e Firefox respectivamente.

Wu et al. (WU ET AL., 2011) apresentam a abordagem DREX (*Developer Recommendation with k-nearest-neighbor search and Expertise ranking*) para recomendar desenvolvedores nas tarefas de correção de bugs. Em um experimento executado no repositório do Mozilla Firefox, os resultados apresentaram um desempenho superior da abordagem DREX em relação a métodos tradicionais de recomendação de desenvolvedores, quando utilizados 10 desenvolvedores alocados a um conjunto de 250 atividades de correção de bugs.

3.2.12 Análise Comparativa dos Trabalhos Relacionados

Após a identificação dos trabalhos relacionados, realizou-se uma análise comparativa entre estes estudos e a abordagem proposta conforme apresentado na Tabela 17. Os critérios de comparação aplicados foram elaborados com base nas características de sistemas de recomendação propostas por Robillard (ROBILLARD; WALKER; ZIMMERMANN, 2010) e são detalhados a seguir:

Tipo de recomendação. identifica se a recomendação acontece de forma automática ou não.

Utiliza Aprendizado de Máquina (ML). indica se foi aplicada uma técnica de aprendizado de máquina para fazer a classificação.

Tipo de tarefa. identifica que tipo de tarefas são recomendadas pelo método apresentado no estudo.

Dados de entrada. apresenta os dados de entrada utilizados pelos sistemas de recomendação.

Tipo de avaliação. descreve como a solução foi validada.

Tabela 17 – Comparativo dos trabalhos relacionados

Autor	Tipo de recomendação	Utiliza ML	Tipo de tarefas	Dados de entrada	Tipo de avaliação
Júnior et al. (JÚNIOR ET AL., 2018)	Automático	Não	Code	Domínio de Aplicação	Experimento
Júnior et al. (JÚNIOR ET AL., 2018)	Automático	Não	Merge	Atributos Gerais	Experimento
Matter et al. (MATTER; KUHN; NIERSTRASZ, 2009)	Automático	Não	Bugs	Vocabulário	Experimento
Zhang et al. (ZHANG; LEE, 2012)	Automático	Não	Code	Redes Sociais, Feedback Especialistas	-
Anvik et al. (ANVIK; MURPHY, 2011)	Automático	Sim	Code	Expertise	Experimento
Anvik et al. (ANVIK; HIEW; MURPHY, 2006)	Semiautomático	Sim	Code	Expertise	Experimento
Wu et al. (WU ET AL., 2011)	Automático	Não	Bug	Ranking Expertises	Experimento
4Experts	Automático	Sim	Code	Habilidades Cognitivas	Experimento

Fonte: Elaborado pelo autor.

A partir da Tabela 17 é possível identificar que todos os trabalhos relacionados foram avaliados a partir de um experimento e a recomendação de tarefas é feita de forma automática, com exceção de Anvik (ANVIK; HIEW; MURPHY, 2006) que emprega uma técnica de recomendação semiautomática. Apesar do amplo uso de aprendizado de máquina, apenas 28,50% dos trabalhos empregaram alguma técnica de aprendizado de máquina (ANVIK; MURPHY, 2011; ANVIK; HIEW; MURPHY, 2006)(ambos do mesmo autor). Os dados de entrada utilizados para a recomendação das tarefas baseou-se nas expertises dos desenvolvedores, exceto por

Júnior et al. (JÚNIOR ET AL., 2018) que utilizou atributos gerais do desenvolvedor para recomendar tarefas, enquanto que Matter et al. (MATTER; KUHN; NIERSTRASZ, 2009) empregou um dicionário de termos para avaliar a recomendação. O trabalho de Júnior et al. (JÚNIOR ET AL., 2018) recomenda tarefas de merge de código fonte, enquanto todos os demais autores recomendam tarefas de codificação que podem ser mudanças de código em geral (ZHANG; LEE, 2012; ANVIK; MURPHY, 2011; ANVIK; HIEW; MURPHY, 2006) ou exclusivamente correção de bugs (MATTER; KUHN; NIERSTRASZ, 2009; WU ET AL., 2011).

3.3 Oportunidades de Pesquisa

A partir dos trabalhos relacionados citados na Seção 3.2.11 e analisados de forma comparativa na Seção 3.2.12, foi possível identificar que apenas 33% dos estudos empregam alguma técnica de aprendizado de máquina para realizar a recomendação de tarefas. Além disso, percebeu-se que a literatura ainda carece de estudos que utilizem dados biométricos para auxiliar na recomendação de tarefas, principalmente dados relacionados às atividades cerebrais associados a atividade cognitiva, a qual influencia as atividades do desenvolvedor, conforme relatado na Seção 3.2.3, motivando a execução de experimentos para realizar uma análise empírica sobre o comportamento da atividade cognitiva do desenvolvedor enquanto este realiza tarefas de codificação. A abordagem 4Experts proporciona o preenchimento de uma lacuna identificada a partir dos trabalhos relacionados, ao propor uma abordagem sensível a dados biométricos de atividades cerebrais e baseada em técnicas de aprendizagem de máquina, para recomendar atividades de desenvolvimento de software.

3.4 Ameaças a Validade

Esta seção discute as medidas adotadas para reduzir as ameaças à validade dos resultados obtidos a partir do estudo realizado. Alguns aspectos podem ameaçar a validade deste estudo, incluindo a validade de condução da pesquisa, a validade interna, validade externa e a validade das conclusões estatísticas. Em parte, essas ameaças são devidas a decisões tomadas durante o processo de identificação e revisão da literatura. Por exemplo, definir uma string de busca que não resulte em documentos relevantes para a área ou a falta de adoção de critérios para selecionar artigos que tragam maior contribuição à pesquisa. Nesse sentido, foram analisadas ameaças relacionadas à validade interna, externa, construtiva e de conclusão.

Validade Interna. A principal ameaça foi a dificuldade em estabelecer uma relação entre as atividades diárias realizadas pelos desenvolvedores e os dados biométricos e aspectos psicofisiológicos que influenciam estas atividades. Esta ameaça foi caracterizada pela falta de detalhes das atividades realizadas pelos desenvolvedores nos artigos selecionados. Para reduzir essa ameaça, os artigos foram completamente filtrados pelos critérios de inclusão e exclusão apresentados na Seção 3.1.3 e revisados em detalhes pelo autor desta pesquisa.

Validade de construção. A classificação incorreta e exclusão de artigos relevantes são ameaças iminentes na revisão da literatura. Este problema foi minimizado a partir do estabelecimento de um protocolo de revisão, com critérios de inclusão/exclusão bem definidos e auditáveis e seguindo a estrutura proposta por Ali (ALI; USMAN, 2018b). Conforme apresentado por Ali, o número de revisões sistemáticas da literatura em Engenharia de Software aumentou e a confiabilidade dessas revisões depende do rigor aplicado durante a pesquisa a partir da estratégia de pesquisa adotada. A falta de rigor durante esse processo causa discrepância nos estudos primários incluídos na revisão as quais promovem revisões baseadas no mesmo assunto, mas realizadas por pesquisadores independentes que resultam em diferentes estudos primários. A Tabela 18 apresenta a lista de verificação proposta por Ali, destacando quais itens são seguidos por esse mapeamento sistemático.

Tabela 18 – Checklist de validade de um mapeamento sistemático de literatura.

Passo	Orientação relatada	Orientação aplicada?
Estratégia de pesquisa aplicada	[R1] Estratégia de pesquisa descrita	Sim
	[R2] Reporta os motivos da estratégia selecionada.	Não
Identificação artigos de referência	[R3] Descreve qual a abordagem utilizada para identificar o conjunto de artigos de referência.	Sim
	[R4] No caso de pesquisa manual para criar o conjunto de artigos de referência Descreve quais revistas e conferências foram consideradas para a pesquisa.	Não se aplica
	[R5] Documenta os artigos de referência.	Sim
	[R6] Documenta os motivos pela escolha das fontes dos artigos de referência.	Sim
Construção da string de busca	[R7] Documenta os termos, termos alternativos, grafias e sinônimos.	Sim
	[R8] Documenta as fontes e razões pela escolha e descarte de termos de busca.	Em partes
	[R9] Documenta a string de busca geral.	Sim
	[R10] Documenta refinamentos (e as razões para usar eles) na string de busca baseada em tentativas.	Em partes
	[R11] Documenta o período de tempo coberto pela pesquisa da literatura.	Sim
	[R12] Documenta as razões pela escolha do período do tempo empregado.	Sim
	[R13] Documenta se a pesquisa foi realizada no título, metadatos ou em todo o texto.	Sim
	[R14] Documenta os motivos por pesquisar no título, metadatos ou no texto todo.	Não
	[R15] Documenta o nível de recall e precisão obtido com a versão final da string de busca.	Não
	[R16] Documenta porque o nível de precisão/recall foi considerado aceitável.	Não
	Em particular, com recall menor que 100% reflete porque a string de busca é boa o suficiente.	Não
	Seleção de fontes de pesquisa	[R17] Documenta as bases utilizadas para a pesquisa.
[R18] Documenta os motivos da seleção das bases ser apropriado, e assegura a cobertura dos tópicos de interesse.		Sim
[R19] No caso de pesquisa manual para complementar a pesquisa, reporta o nome e ano dos locais pesquisados.		Não se aplica
[R20] Documenta as tentativas de reduzir o viés de publicação.		Não
[R21] Documenta a "literatura cinza" considerada para o estudo.		Não se aplica
[R22] Reporta trabalhos não publicados de especialistas e quais materiais foram identificados desta forma.		Não se aplica
Validação de protocolo	[R23] Documenta se o protocolo foi validade de forma independente e por quem.	Não
	[R24] Documenta as sugestões feitas pelos revisores sobre a estratégia de pesquisa, e as ações que foram tomadas para direcionar estas sugestões ao protocolo de pesquisa.	Sim
Condução da pesquisa	[R25] Documenta as strings de busca específicas por base de pesquisa.	Sim
	[R26] Documenta qualquer filtro adicional e razões pelo seu emprego.	Não
	[R27] Documenta qualquer diferença entre a string de busca geral em relação a string de busca específica da base de dados e os motivos desta mudança.	Sim
	[R28] Documenta o número de retorno por base de dados.	Sim
	[R29] Documenta o número de resultados da pesquisa por base de dados.	Sim

Fonte: Adaptado de (ALI; USMAN, 2018b).

Validade de conclusão. Essa ameaça está relacionada a problemas que podem afetar a confiabilidade das descobertas. Nesse contexto, os artigos selecionados podem ser influenciados

por interesses pessoais com os resultados do estudo. Para evitar essa situação, foram aplicados critérios de inclusão e exclusão durante o processo de filtragem dos artigos e foram monitoradas qualquer influência pessoal que pudesse impactar negativamente os resultados. Além dessa ameaça, outra ameaça a ser considerada é a classificação do artigo. Nesse caso, foi elaborada uma string de busca com os principais sinônimos de cada termo de pesquisa. Finalmente, todas as conclusões foram tiradas após a coleta dos resultados, evitando qualquer possibilidade de erro (WOHLIN ET AL., 2012).

Validade externa. Esta ameaça reflete sobre a validade dos resultados obtidos em outros contextos. Em termos práticos, deve-se preocupar em entender até que ponto os resultados deste estudo podem ser generalizados para outras realidades ou configurações da pesquisa. Poderiam surgir algumas perguntas a esse respeito, como se outras bases eletrônicas, diferentes *strings* de busca e critérios de exclusão e inclusão tivessem sido usados, os mesmos resultados seriam produzidos? Nesse sentido, foi analisado se a amostra de estudos poderia ser alterada significativamente se, por exemplo, os mecanismos de pesquisa fossem alterados. Como foram utilizadas 12 bases eletrônicas, acredita-se que a amostra não seria significativamente alterada.

Além disso, pode-se utilizar a teoria da similaridade proximal (proposta por Donald T. Campbell (CAMPBELL; RUSSO, 1999)) para traçar o grau de generalização dos resultados. Basicamente, o ponto focal é definir critérios que possam ser usados para situações ou configurações semelhantes, nas quais os resultados encontrados sejam os mesmos ou mesmo semelhantes. Alguns critérios são mostrados a seguir: Primeiro, os autores podem fazer uso de outras bases eletrônicas para recuperar artigos potencialmente relevantes. Segundo, os dados biométricos devem ser usados como um recurso para melhorar as práticas de engenharia de software, incluindo a compreensão do código-fonte. Em seguida, a string de pesquisa deve ser semelhante à usada no estudo, onde alterações nos termos usando outros sinônimos são permitidas. Como essas mudanças podem ocorrer em outros estudos, os achados podem ser generalizados ou aceitos, em algum momento, para outros contextos mais semelhantes a esses requisitos.

3.5 Conclusão

A análise de literatura realizada neste capítulo tornou possível classificar os dados biométricos em relação aos seus tipos, forma de aquisição, aplicação e aspectos psicofisiológicos, possibilitando a elaboração de uma taxonomia, conforme detalhado na Seção 3.2.1. Por sua vez, a análise comparativa dos trabalhos relacionados apresentada na Seção 3.2.12 tornou possível identificar oportunidades de pesquisa em sistemas de recomendação, visto que não existem referências ao emprego de dados biométricos e ainda há pouco uso de técnicas de aprendizado de máquina nestes sistemas.

A partir destas análises, é possível concluir que a abordagem 4Experts apresentada no Capítulo 5 é viável, por explorar lacunas ainda não empregadas em sistemas de recomendação, ao utilizar dados biométricos do cérebro, associados a atividade cognitiva do desenvolvedor, clas-

sificados com o uso de aprendizagem de máquina para recomendar tarefas de desenvolvimento.

4 ANÁLISE EMPÍRICA DO GANHO DE EXPERTISE DOS DESENVOLVEDORES ATRAVÉS DO USO DE DADOS BIOMÉTRICOS

Este capítulo reporta um estudo empírico sobre o processo de aquisição de expertises associadas às atividades de codificação dos desenvolvedores com o uso de dados biométricos coletados do cérebro por um equipamento de eletroencefalograma.

Experimentos controlados representam o método aplicado na maioria dos trabalhos que fazem uso de dados biométricos na Engenharia de Software, conforme identificado nos trabalhos relacionados deste artigo, na Questão de Pesquisa 6 (Seção 3.2.7), especificamente na Tabela 13. Os resultados obtidos através da análise do resultado deste experimento serão utilizados como base de dados para a abordagem 4Experts proposta nesta dissertação.

Este capítulo está organizado da seguinte forma: A Seção 4.1 apresenta o objetivo e as questões de pesquisa propostas para o experimento. A Seção 4.2 destaca as hipóteses que se pretende comprovar com a execução do experimento. As variáveis de estudo serão apresentadas na Seção 4.3. Os detalhes aplicados para a seleção dos participantes serão descritos na Seção 4.4, enquanto que na Seção 4.5 será descrito o processo experimental. Na Seção 4.6 serão destacados os principais materiais empregados durante a execução do procedimento, que será detalhada na Seção 4.7. O procedimento empregado para analisar os dados coletados será discutido na Seção 4.8. Por fim, na Seção 4.11 serão detalhadas as ameaças a validade do experimento, enquanto os resultados serão apresentados na Seção 4.9 com uma discussão adicional na Seção 4.10.

4.1 Objetivos e Questões de Pesquisa

Nesta seção são formalizados os objetivos e questões de pesquisa do experimento. Este experimento procura analisar os diferentes níveis de carga cognitiva resultantes do esforço empregado pelos desenvolvedores na execução de tarefas de codificação de diferentes complexidades de acordo com a expertise de cada um. Além disso, o experimento procura identificar um possível indicador de carga cognitiva que um desenvolvedor possa apresentar de acordo com a sua experiência. A identificação deste indicador possibilita prever momentos em que a carga cognitiva está fora do limite previsto, excedendo a capacidade do desenvolvedor em realizar a tarefa naquele instante (BROUWER ET AL., 2012), podendo incidir em falhas e dificuldades no processo de codificação. Seguindo a convenção Goal Question Method (GQM) proposta por Wohlin (WOHLIN ET AL., 2012), a seguir apresenta-se o propósito desta investigação.

Analisar o nível de carga cognitiva
com o propósito de identificar a expertise do desenvolvedor
com respeito a evolução da expertise com o passar do tempo
a partir da perspectiva dos desenvolvedores
no contexto de compreensão de código

Com o propósito desta investigação definido, foram elaboradas três questões de pesquisa (QP) para este experimento, apresentadas a seguir:

QP1: É possível quantificar a carga cognitiva de um desenvolvedor?
QP2: A carga cognitiva do desenvolvedor permite a classificação das suas expertises?
QP3: É possível identificar um limite de carga cognitiva, que quando ultrapassado, implique em redução de acuracidade durante a execução de tarefas de programação?

Estas questões de pesquisa a serem respondidas neste capítulo serão utilizadas para responder a QP2 da dissertação destacada na Seção 1.2.

4.2 Hipóteses Formuladas

Conforme o objetivo principal deste experimento de identificar a carga cognitiva conforme a expertise dos desenvolvedores, com o propósito de ser aplicado na construção da abordagem 4Experts proposta no Capítulo 5, foram propostas três diferentes hipóteses. A Hipótese 1 avalia a quantidade de carga cognitiva para responder questões de diferentes níveis de dificuldade. A segunda Hipótese verifica a evolução da carga cognitiva, conforme as suas expertises aumentam, enquanto a Hipótese 3 analisa a diferença de carga cognitiva do desenvolvedor quando este realiza uma tarefa com e sem acuracidade respectivamente.

Hipótese 1: A hipótese que pretende-se testar é que a aquisição de expertises ao longo do tempo pode ser percebida através de indicadores de carga cognitiva. Como a aquisição de expertise acontece em função do tempo e pela obtenção de conhecimento, conjectura-se que esta aquisição poderá ser percebida através da captura de indicadores neurais, durante a execução de tarefas (T) em momentos distintos - isto é, antes e após a aquisição de uma nova expertise. Esta suposição baseia-se no trabalho de CRK e Fritz (CRK; KLUTHE; STEFIK, 2015; FRITZ ET AL., 2014), no qual são identificadas variações na carga cognitiva dos desenvolvedores de acordo com as suas expertises. Com a análise desta hipótese será possível aferir a diferença de carga cognitiva entre as questões possibilitando identificar a existência de variação da carga cognitiva de acordo com o esforço empregado na realização da tarefa (T) conforme o desenvolvedor adquire expertise técnicas. Por outro lado, esta hipótese pode ser rejeitada caso a carga cognitiva gerada pelo desenvolvedor para resolver as tarefas não se altere, conforme apresentado a seguir:

Hipótese Nula 1, H_{1-0} : A carga cognitiva não se modifica conforme a evolução das expertises.

H_{1-0} : Carga Cognitiva (T)_{1º Etapa} = Carga Cognitiva (T)_{2º Etapa}

Hipótese Alternativa 1, H_{1-1} : A carga cognitiva aumenta conforme a evolução das expertises.

H_{1-1} : Carga Cognitiva (T)_{1º Etapa} > Carga Cognitiva (T)_{2º Etapa}

Hipótese 2: De acordo com a expertise de um desenvolvedor, acredita-se que exista diferença na carga cognitiva. Esta hipótese sugere que os desenvolvedores que apresentam maiores expertises necessitem de menos carga cognitiva para realizar as mesmas tarefas (T) que um desenvolvedor de menor expertise conforme constatado por Nakagawa e CRK (NAKAGAWA ET AL., 2014; CRK; KLUTHE; STEFIK, 2015). Esta hipótese poderá ser rejeitada caso não exista diferença de carga cognitiva entre os desenvolvedores de maior expertise em relação aos de menor expertise para executar as mesmas tarefas (T). A análise das expertises será feita conforme a evolução do conhecimento do participante com o passar do tempo, permitindo validar a abordagem 4Experts proposta no Capítulo 5, quando para um mesmo desenvolvedor, conforme ele vai adquirindo expertises, o sistema recomenda tarefas mais complexas.

Hipótese Nula 2, $H_{2.0}$: A carga cognitiva não apresenta diferença entre desenvolvedores de diferentes níveis de expertise.

$$H_{2.0}: \text{Carga Cognitiva (T)}_{\text{BaixaExpertise}} = \text{Carga Cognitiva (T)}_{\text{AltaExpertise}}$$

Hipótese Alternativa 2, $H_{2.1}$: A carga cognitiva dos desenvolvedores de menor expertise é menor do que a carga cognitiva dos desenvolvedores de maior expertise.

$$H_{2.1}: \text{Carga Cognitiva (T)}_{\text{BaixaExpertise}} < \text{Carga Cognitiva (T)}_{\text{AltaExpertise}}$$

Hipótese 3: Acredita-se que uma tarefa (T) que não apresente corretude tenha este resultado devido ao desenvolvedor não apresentar as expertises necessárias para executar a mesma, conforme evidenciado pelo trabalho de Fritz (FRITZ; MULLER, 2016). A hipótese 3 procura identificar a diferença de carga cognitiva das tarefas (T) executadas com corretude em relação as tarefas executadas sem corretude, sugerindo que possa ser estabelecido um limite de carga cognitiva comum aos desenvolvedores quando estes executam uma tarefa (T) corretamente. A rejeição desta hipótese acontecerá no caso da diferença da carga cognitiva não implicar em maior número de respostas incorretas.

Hipótese Nula 3, $H_{3.0}$: A carga cognitiva do desenvolvedor não se altera quando ele executa uma tarefa corretamente em relação a tarefa executada incorretamente

$$H_{3.0}: \text{Carga Cognitiva (T)}_{\text{Correta}} = \text{Carga Cognitiva (T)}_{\text{Incorreta}}$$

Hipótese Alternativa 3, $H_{3.1}$: A carga cognitiva do desenvolvedor se altera quando ele executa uma tarefa corretamente em relação a tarefa executada incorretamente .

$$H_{3.1}: \text{Carga Cognitiva (T)}_{\text{Correta}} \neq \text{Carga Cognitiva (T)}_{\text{Incorreta}}$$

A Tabela 19 apresenta um resumo das hipóteses e alternativas deste experimento, que objetivam identificar indícios que possam ser considerados suficientes para responder as questões de pesquisa deste capítulo. A partir dos testes de hipóteses, pretende-se confirmar se a carga cognitiva pode ser utilizada como ferramenta de análise durante a execução das tarefas de desenvolvimento, pelos desenvolvedores.

Tabela 19 – Resumo das hipóteses.

Hipótese	Tipo	Representação
H1	Nula	H_{1-0} : Carga Cognitiva (T) _{1º Etapa} = Carga Cognitiva (T) _{2º Etapa}
	Alternativa	H_{1-1} : Carga Cognitiva (T) _{1º Etapa} > Carga Cognitiva (T) _{2º Etapa}
H2	Nula	H_{2-0} : Carga Cognitiva (T) _{BaixaExpertise} = Carga Cognitiva (T) _{AltaExpertise}
	Alternativa	H_{2-1} : Carga Cognitiva (T) _{BaixaExpertise} < Carga Cognitiva (T) _{AltaExpertise}
H3	Nula	H_{3-0} : Carga Cognitiva (T) _{Correta} = Carga Cognitiva (T) _{Incorreta}
	Alternativa	H_{3-1} : Carga Cognitiva (T) _{Correta} \neq Carga Cognitiva (T) _{Incorreta}

Fonte: Elaborado pelo autor.

4.3 Variáveis

Esta pesquisa identifica como variáveis independentes os indicadores nominais de expertise e o grau de dificuldade das tarefas aplicadas, as quais incidem sobre as variáveis dependentes: Corretude (C) e Carga Cognitiva (CC) (WOHLIN ET AL., 2012; FARIAS ET AL., 2015; BISCHOFF, 2018). A Corretude (C) representa a assertividade, ou não da tarefa em análise, enquanto a Carga Cognitiva (CC) representa um indicador da atividade cerebral do participante enquanto este executa as tarefas do experimento (NAKAGAWA ET AL., 2014). A Tabela 20 apresenta o resumo das variáveis investigadas neste estudo.

Tabela 20 – Descrição das variáveis de estudo.

Variável	Escala	
Independente	Expertise	Nominal [Baixa Expertise, Alta Expertise]
	Dificuldade	Nominal [1..5]
Dependente	Corretude (C)	Nominal [correto, incorreto]
	Carga Cognitiva (CC)	Ordinal [0..1]

Fonte: Elaborado pelo autor.

A variável dependente da primeira hipótese é a carga cognitiva (CC) empregada pelo desenvolvedor durante a execução da tarefa proposta pelo experimento. Esta variável foi calculada através da aplicação da técnica de filtro espacial detalhada na Seção 4.7. Para a segunda hipótese, também foi utilizada a carga cognitiva (CC) como variável dependente. Por sua vez, na terceira hipótese foram empregadas duas variáveis dependentes: a corretude (C) da resposta associada a tarefa do experimento e a carga cognitiva (CC) do participante.

4.4 Contexto e Seleção dos Participantes

No total, 42 pessoas participaram do experimento, entretanto, dois participantes não conseguiram executar a segunda etapa do processo, sendo reduzido o número de participantes a 40. Destes 40 participantes, 35 são acadêmicos dos cursos de Análise e Desenvolvimento de Sistemas e Engenharia da Computação no Centro Universitário e Faculdade UNIFTEC, situado

em Caxias do Sul. Destes, 25 participantes são alunos da disciplina de Algoritmo, realizada no primeiro semestre dos cursos, enquanto os demais são provenientes da disciplina de Programação Web e Projetos de Sistemas para Web. Os 5 participantes restantes são provenientes da indústria, já formados em cursos de graduação na área de tecnologia, com pelo menos 7 anos de experiência na área de desenvolvimento de sistemas. Todos estes participantes, após serem capacitados para executar o experimento, responderam a um questionário com o propósito de identificar a expertise em relação a programação.

Na Engenharia de Software, a identificação da Expertise do desenvolvedor possui um papel relevante. A correta identificação permite relacionar adequadamente um desenvolvedor a uma determinada solicitação de modificação de um programa. Huzefa (HUZEFA KAGDI1, MALCOM GETHERS2; HAMMAD, 2011) cita em seu artigo que a tarefa de identificação da Expertise é comum nas equipes de desenvolvimento, mas não é trivial, pois, de forma tradicional, ela envolve conhecimentos de projeto, organização e o balanceamento de diversos aspectos. Esta tarefa torna-se consideravelmente mais complexa nos casos de equipes grandes e distribuídas geograficamente.

A expertise está associada à compreensão que um determinado desenvolvedor possui sobre um programa, sendo a compreensão de um programa, um processo cognitivo do próprio desenvolvedor o qual não pode ser observado diretamente. Por isso, para avaliar a expertise, são necessários experimentos e técnicas que permitam observar o comportamento do desenvolvedor, quando este for confrontado com um determinado problema de codificação. Feigenspan (FEIGENSPAN ET AL., 2012) realizou uma revisão de literatura para identificar o estado da arte na medição e controle de expertises no desenvolvimento, propondo um questionário com um modelo de cinco fatores para a classificação da expertise do desenvolvedor.

Tabela 21 – Identificação da expertise através autorrelato, pelo formulário de pesquisa separado por classes de pontuação.

Intervalo pontos	Quantidade Participantes	Participantes
0 - 38	14	P14, P20, P23, P24, P25, P27, P28, P30, P33, P35, P36, P37, P39, P40
39 - 78	24	P01, P02, P03, P04, P06, P08, P09, P10, P11, P12, P13, P15, P16, P17, P18, P19, P21, P22, P26, P29, P31, P32, P34, P38
79 - 117	2	P05, P07

Fonte: Elaborado pelo autor.

Para avaliar o nível da expertise através do questionário proposto por Feigenspan (FEIGENSPAN ET AL., 2012), foi utilizado um sistema de pontuação sobre as questões do formulário de pesquisa, sendo que todas as questões apresentadas pelo formulário possuem o mesmo peso, enquanto que a menor pontuação possível será zero quando nenhuma questão for respondida e 117 quando todas as questões forem respondidas em seu maior grau de resposta. Para

permitir esta categorização baseada em pontos, as questões foram construídas em escala *likert* conforme apresentado pelo Apêndice A.2. A Tabela 21 apresenta a classificação aplicada aos participantes quanto ao nível de expertise extraído das questões respondidas.

Entretanto, este questionário revela as expertises do desenvolvedor conforme o seu autorretrato, possibilitando divergências de expertise, quando estas expertises forem avaliadas a partir de tarefas reais de codificação. Para esta pesquisa, o emprego da expertise auto relatada poderá implicar em resultados inadequados, quando confrontados com a carga cognitiva obtida do desenvolvedor. Por isso, optou-se em categorizar os desenvolvedores em apenas dois níveis de expertise, classificados de acordo com a corretude das respostas relativas as tarefas executadas por eles durante o experimento. Desta forma os desenvolvedores são classificados em desenvolvedores de baixa expertise quando eles tiverem um índice de acerto igual ou inferior a 50% das tarefas propostas e de alta expertise quando obtiverem um índice de acerto superior a 50%, conforme apresentado na Tabela 22, onde os participantes são separados entre a primeira e segunda etapa do experimento.

Tabela 22 – Identificação da expertise através da corretude das respostas.

Expertise	Etapa Experimento	Participantes
Baixa Expertise	1º Etapa	P03, P14, P17, P18, P19, P20, P21, P22, P24, P25, P26, P27, P30, P31, P32, P33, P35, P36, P37, P40
	2º Etapa	P16, P19, P20, P21, P24, P26, P27, P28, P32, P33, P35, P37, P40
Alta Expertise	1º Etapa	P01, P02, P04, P05, P06, P07, P08, P09, P10, P11, P12, P13, P15, P16, P23, P28, P29, P34, P38, P39
	2º Etapa	P01, P02, P03, P04, P05, P06, P07, P08, P09, P10, P11, P12, P13, P14, P15, P17, P18, P22, P23, P25, P29, P30, P31, P34, P36, P38, P39

Fonte: Elaborado pelo autor.

4.5 Processo Experimental

Na Figura 25 é apresentado o processo experimental. O processo proposto é baseado nas práticas de WOHLIN (WOHLIN ET AL., 2012), sendo composto por três fases. Na primeira fase os participantes receberam treinamento para garantir que tenham compreendido o funcionamento do experimento, além de assinarem um termo de consentimento e responder um questionário sobre os seus conhecimentos técnicos e experiência na área de programação (Apêndice A.2) com o propósito de identificar a seu expertise. Na segunda fase, os participantes foram submetidos a execução do experimento, no qual lhes foi apresentado um conjunto de 10 questões de programação. Durante o período em que os participantes resolviam as questões os seus dados biométricos foram coletados através do equipamento EEG. A terceira fase teve por objetivo finalizar o processo de coleta e obter um feedback dos participantes em relação ao

experimento.

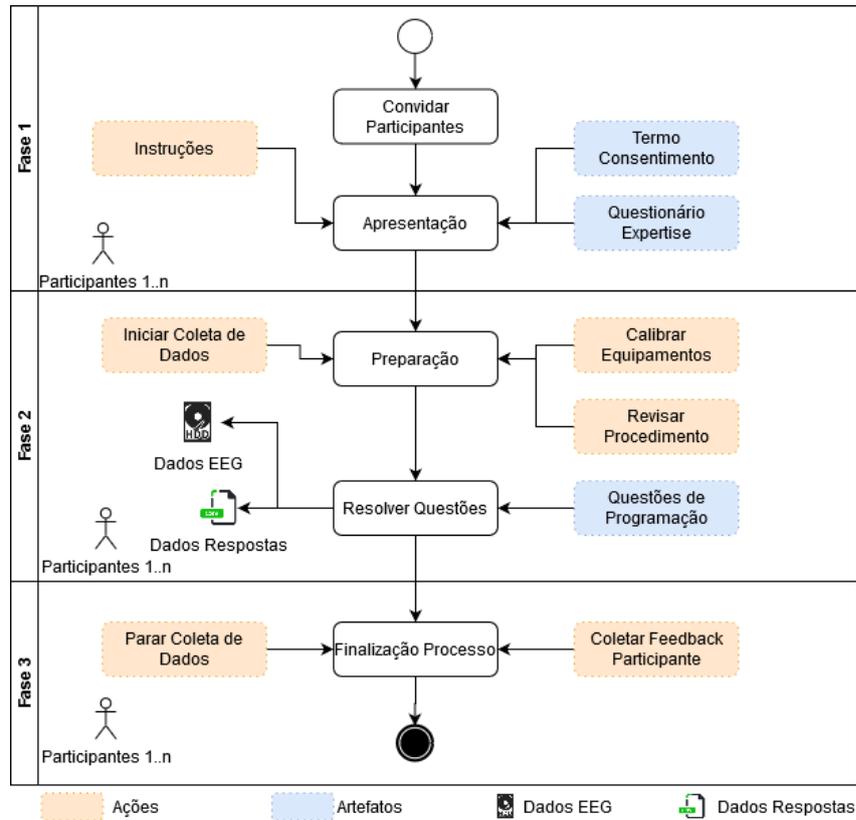


Figura 9 – Processo experimental.

Fonte: Adaptado de (FARIAS ET AL., 2019).

Durante a fase 3 os participantes foram orientados a analisar cada uma das questões propostas e escolher a alternativa correta para a questão, dentre 5 alternativas disponíveis. Nesta etapa foram propostas 10 questões de programação de diferentes níveis de dificuldade.

As questões aplicadas neste experimento fizeram uso de três estruturas de programação comuns a qualquer linguagem: atribuições, condições e repetição. A dificuldade atribuída a cada uma das questões está relacionada ao número de comandos associados as estruturas citadas, independentemente do tipo de instrução ser uma atribuição, condição ou repetição. O número de comandos pode ser associado ao grau de dificuldade a medida que o aumento de número de comandos do algoritmo exige maior uso da memória de trabalho do indivíduo (SWELLER, 1988). Para as questões deste experimento foram definidos 6 níveis de dificuldade. O nível zero não possui nenhum comando associado e está relacionado a questão de controle. Esta questão foi aplicada no experimento com o objetivo de verificar se o participante está atento a execução das tarefas. Por sua vez o nível de dificuldade 5 apresenta 8 comandos distribuídos entre as atribuições, condições e repetições. A Tabela 23 apresenta a identificação de cada uma das questões em conjunto com o seu nível de dificuldade, enquanto o Apêndice A.3 apresenta as questões aplicadas no experimento.

Devido ao desconforto que o uso prolongado do EEG pode causar no participante, as ques-

Tabela 23 – Identificação da dificuldade das questões do experimento.

Tarefa	Nível Dificuldade	Quantidade Atribuições	Quantidade Decisões	Quantidade Repetições	Quantidade Total
1	1	1	0	0	1
2	1	0	1	0	1
3	2	0	0	2	2
4	2	0	2	0	2
5	3	2	2	0	4
6	3	0	0	4	4
7	4	0	2	4	6
8	4	2	0	4	6
9	5	2	2	4	8
10	0	0	0	0	0

Fonte: Elaborado pelo autor.

tões propostas na Tabela 23 tiveram o seu tempo para resolução restrito a 1,5 minutos com um intervalo de descanso de 5 segundos entre as questões.

4.6 Material

Nesta seção são detalhados principais materiais empregados para o experimento em suas diversas etapas.

Emotiv Epoc+: O Emotiv Epoc+ (EMOTIV, 2011a) é um dispositivo de eletroencefalografia (EEG) portátil, desenvolvido pela empresa Emotiv e detalhado na Figura 10. Este dispositivo tem sido empregado em diversos experimentos em diversas áreas de pesquisa, dentre estas a Engenharia de Software (CRK; KLUTHE; STEFIK, 2015). O seu uso tem sido ampliado na área por ser um dispositivo portátil, de custo relativamente baixo, possuir um setup rápido durante a realização de experimentos e apresentar uma acuracidade superior em relação a outros dispositivos conforme apresentado por DAS (DAS ET AL., 2014) em uma análise comparativa com outro dispositivo de EEG portátil. A acuracidade superior se deve ao equipamento possuir 14 sensores para a coleta de dados EEG, seguindo o protocolo internacional 10-20 de posicionamento dos sensores EEG (MEYJES; ARTS, 1984) detalhado na Figura 12.

Além destas características, o Emotiv Epoc+ coleta dados EEG de forma bruta com uma taxa de amostragem de 256 Mhz, através de conexão bluetooth com um computador. A taxa de amostragem de 256Mhz é superior ao mínimo recomendado pelo teorema de Nyquist, o qual menciona que a taxa de amostragem deve ser no mínimo duas vezes a frequência a ser estudada. Os dados coletados pelo Emotiv Epoc+ foram processados em conjunto com o software Emotiv Xavier Test Bench versão 3.1.21, apresentado na Figura 12. Este software realiza a recepção dos dados brutos provenientes do EEG, mescla estes dados com marcadores recebidos através de porta serial e salva o resultado em um arquivo EEG no formato *European Data Format* (EDF).

OpenSesame (Versão 3.2.8): OpenSesame (MATHÔT, 2010) é um software open source



Figura 10 – Emotiv EPOC+.

Fonte: (EMOTIV, 2011a).

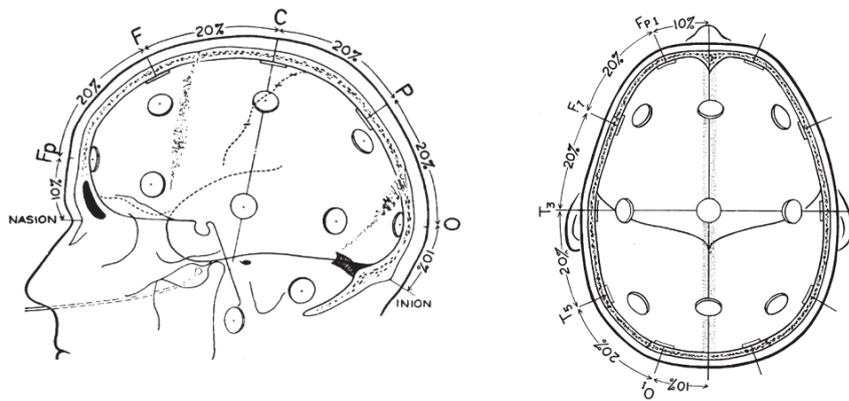


Figura 11 – Sistema internacional 10-20.

Fonte: (MEYJES; ARTS, 1984).

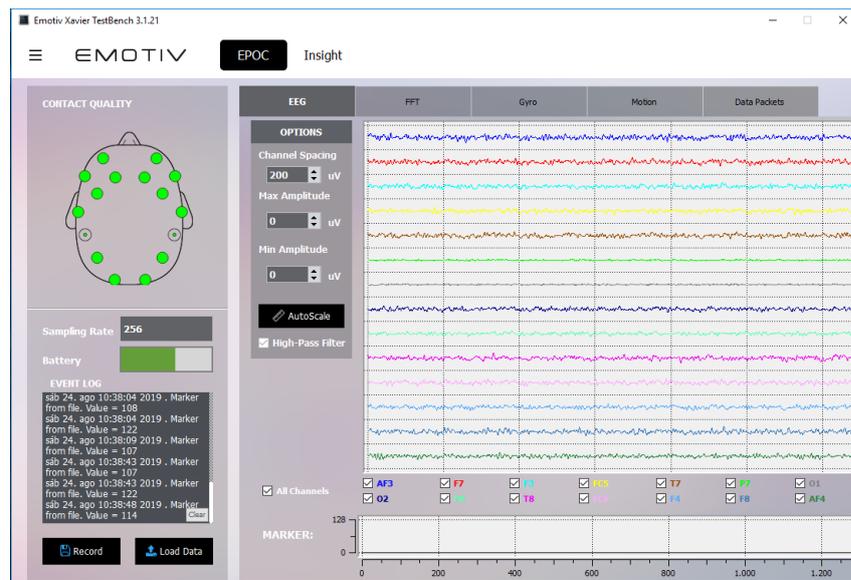


Figura 12 – Emotiv Test Bench.

Fonte: (EMOTIV, 2011a).

escrito na linguagem Python utilizado para a realização de experimentos em psicologia e neurociência. Este software é mantido pela *European Society for Cognitive Psychology* (ESCoP) e pela empresa SR Research, responsável por desenvolver sistemas de eye-track. O software OpenSesame foi escolhido para a execução deste experimento por possuir referências de uso em outros artigos (JIN; BORST; VUGT, 2019; OpenSesame: An open-source, graphical experiment builder for the social sciences, 2012), além de possuir uma interface visual intuitiva conforme apresentado na Figura 13 e permitir a programação de scripts em python para realizar o envio de marcadores através de porta serial para o dispositivo EEG.

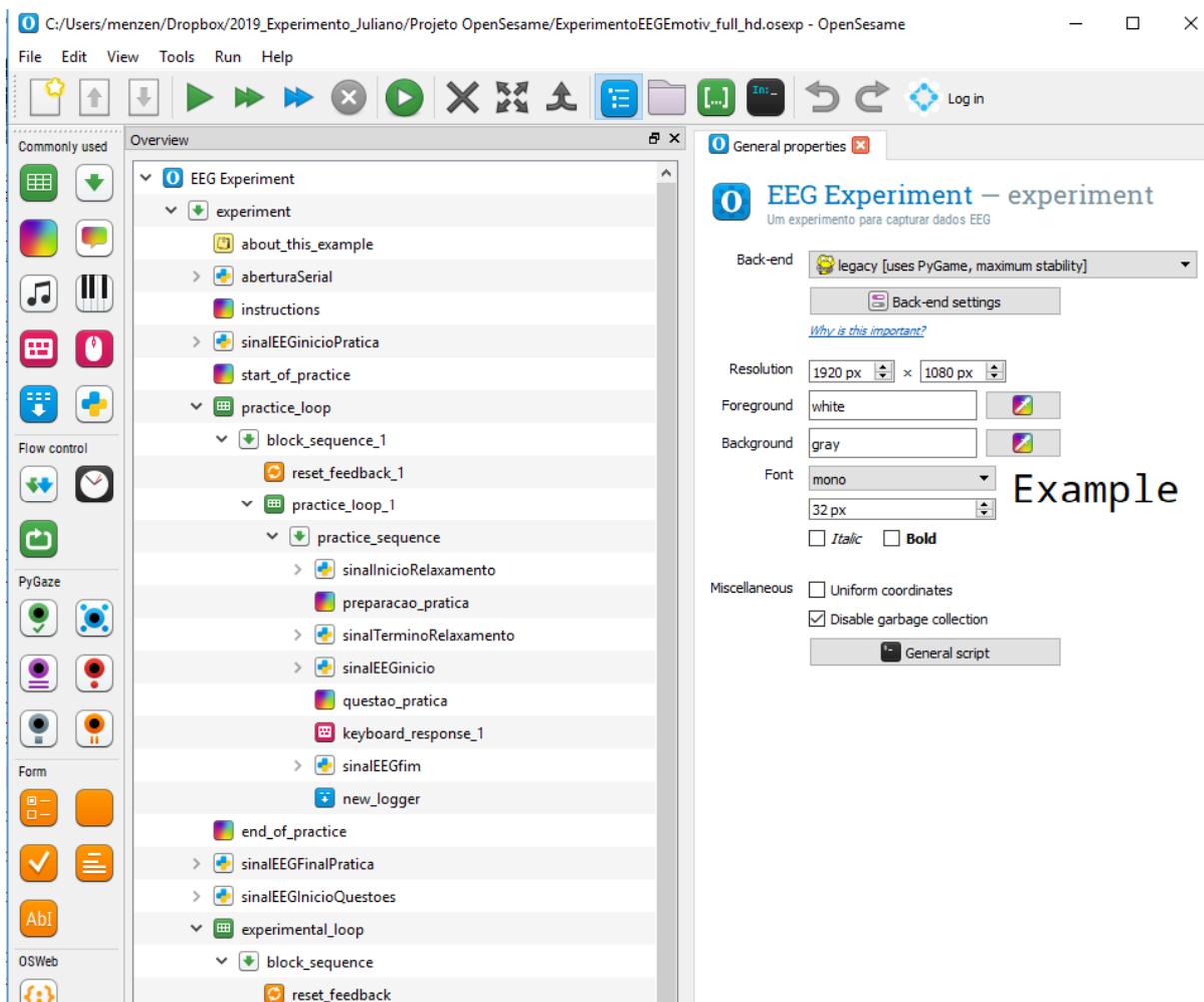


Figura 13 – OpenSesame.

Fonte: (MATHÔT, 2010)

Free Virtual Serial Ports (Versão 3.22): O software Free Virtual Serial Ports (LTD, 2015) foi utilizado para a criação de portas seriais virtuais no computador em que o experimento foi realizado. As portas seriais são utilizadas para o envio de sinais emitidos pelo software OpenSesame, sempre que um evento de início ou fim de questão ou intervalo fosse acontecer, para o software Xavier Test Bench, responsável pelo registro dos dados de EEG coletados pelo Emotiv EPOC+.

Python (Versão 3.7.3): A linguagem de programação Python (FOUNDATION, 2001) foi utilizada neste experimento em diversos momentos: (1) Para construção de script no software OpenSesame para envio de dados por porta serial, em eventos de início e fim das questões, para marcar os momentos destes eventos junto aos dados brutos coletados pelo EEG; (2) Para processar os dados brutos coletados pelo EEG, identificando os períodos do EEG em que o participante estava respondendo as questões, separando estes períodos em conjunto de dados independentes; (3) Processar estes conjuntos de dados com a técnica Multitaper para fornecer uma análise espectral relativa com o propósito de calcular a carga cognitiva. Para facilitar a realização dos procedimentos citados, foram utilizadas bibliotecas especializadas para o tratamento de dados que seguem: (1) **MNE versão 0.18.2** (DEVELOPERS, 2012) - é uma biblioteca especializada para tratamento e visualização de dados EEG. Neste experimento esta biblioteca foi utilizada para realizar a leitura dos arquivos EDF gerados pelo Xavier Test Bench, (2) **Pandas versão 0.25.0** (DEVELOPERS, 2001a) - esta biblioteca facilita o processamento e análise de grandes quantidades de dados no ambiente python. Neste projeto, a biblioteca foi utilizada para a manipulação dos dados de EEG, (3) **Scipy versão 1.3.0** (DEVELOPERS, 2001b) - esta biblioteca oferece um conjunto de rotinas matemáticas que podem ser utilizadas de forma amigável. A biblioteca foi utilizada para realizar o processamento dos dados coletados com o uso da técnica Multitaper.

Visual Studio Code (Versão 1.37.1): O Visual Studio Code (MICROSOFT, 2012) foi a ferramenta empregada para a codificação do experimento. Esta ferramenta foi escolhida por possuir muitos recursos baseados em plugins que facilitam o desenvolvimento na linguagem Python. A Figura 14 apresenta um trecho do código python empregado para processar os dados do EEG.

Matlab (Versão R2019a): O software Matrix Laboratory, conhecido como Matlab (INC., 1994) é uma plataforma de programação de alta performance, baseada em cálculo numérico. Por ser uma plataforma de alto desempenho e capaz de trabalhar com uma grande quantidade de dados ao mesmo tempo, esta plataforma é ideal para a análise de dados de EEG. Agregado ao Matlab, foi utilizado o plugin EEG Lab versão 2019 (CALIFORNIA, 2004) para a análise e plotagem dos dados de EEG utilizados neste experimento.

4.7 Execução

Toda a execução do processo experimental foi realizada seguindo os procedimentos definidos na Seção 4.5. Cada um dos participantes selecionados executou o experimento duas vezes, com um intervalo de tempo de dois meses entre as execuções. Os participantes foram avisados previamente sobre a data e local onde o experimento seria executado. Antes da primeira execução os participantes foram apresentados ao experimento. Nesta apresentação eles receberam instruções sobre o protocolo de execução do experimento e como eles deveriam se comportar durante a execução com o propósito de evitar movimentos desnecessários e comprometer a qua-

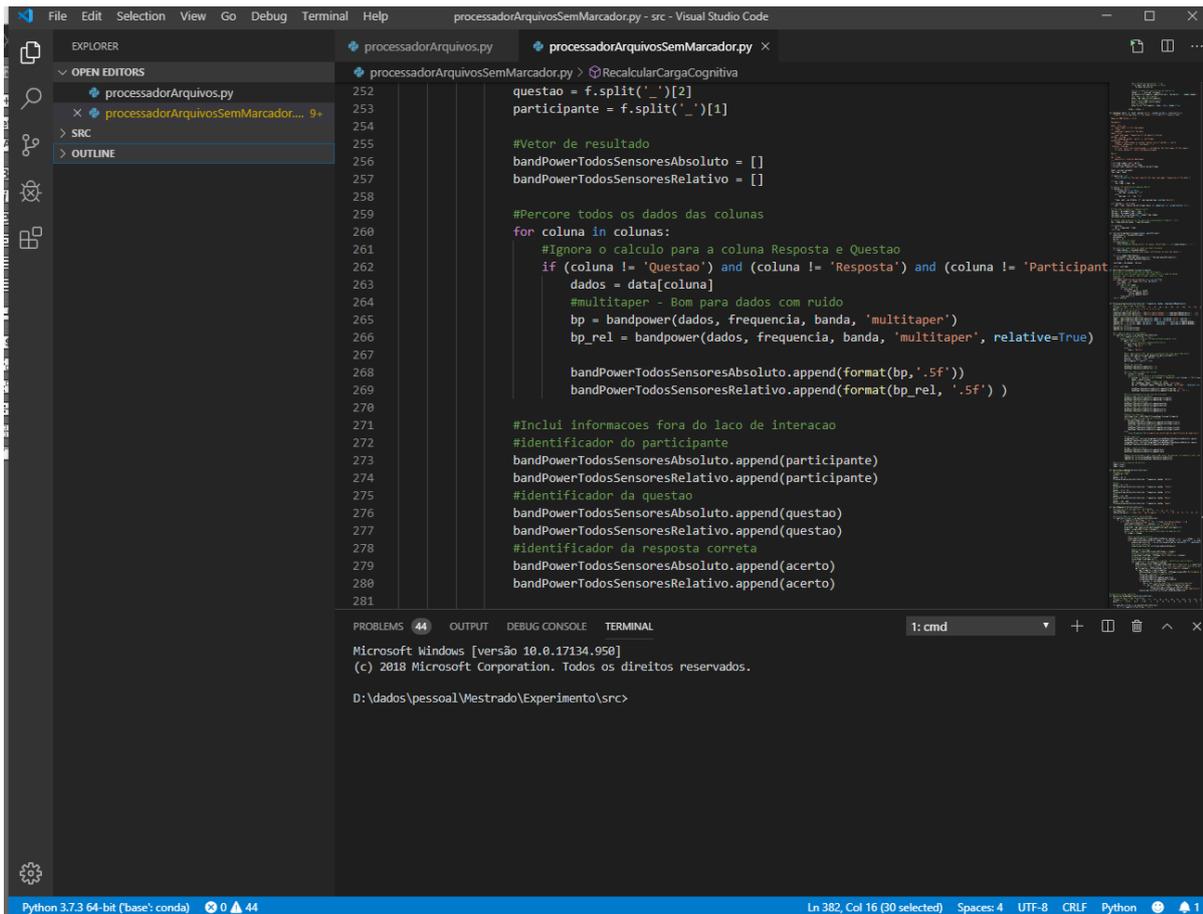


Figura 14 – Interface do Visual Studio Code.

Fonte: (MICROSOFT, 2012).

lidade dos dados EEG coletados. Neste mesmo momento os participantes receberam um termo de consentimento destacado no Apêndice A.1 no qual autorizam a sua participação no processo experimental. Após a autorização, cada voluntário preencheu um questionário demográfico, conforme apresentado no Apêndice A.2, utilizado posteriormente para detectar a expertise de cada um.

Após o questionário ser preenchido, o participante foi conduzido até o local do experimento, onde ele desligou qualquer equipamento eletrônico que possuía consigo. O participante também foi aconselhado a não utilizar roupas que pudessem provocar um aumento na eletricidade estática em seu corpo. Mesmo com o uso de pulseira antiestática, esta medida foi aconselhada para minimizar possíveis riscos de interferência durante o processo.

O participante é motivado a se posicionar de forma que se sinta confortável e possa evitar movimentos durante a execução do experimento, conforme apresentado na Figura 15. Na sequência, o Emotiv Epc+ é colocado sobre o escalpo e calibrado para que possa fornecer os dados de EEG corretamente. Após esta etapa ser concluída, é iniciada a gravação do EEG e em seguida é iniciado o processo de execução das questões apresentadas durante o processo. Na primeira etapa da exibição, é exibido ao usuário uma tela contendo as instruções a serem utili-

zadas no processo, ficando a conclusão desta interface para o início do experimento a critério do participante, no qual ele foi instruído a pressionar a barra de espaço do teclado para prosseguir. A seguir uma questão de teste, para ambientar o participante no experimento foi exibida, com um período de 1,5 minutos para ser respondida. Caso o tempo de resposta ultrapassasse este limite, a questão era concluída sem resposta e uma nova tela de instruções era exibida ao participante.

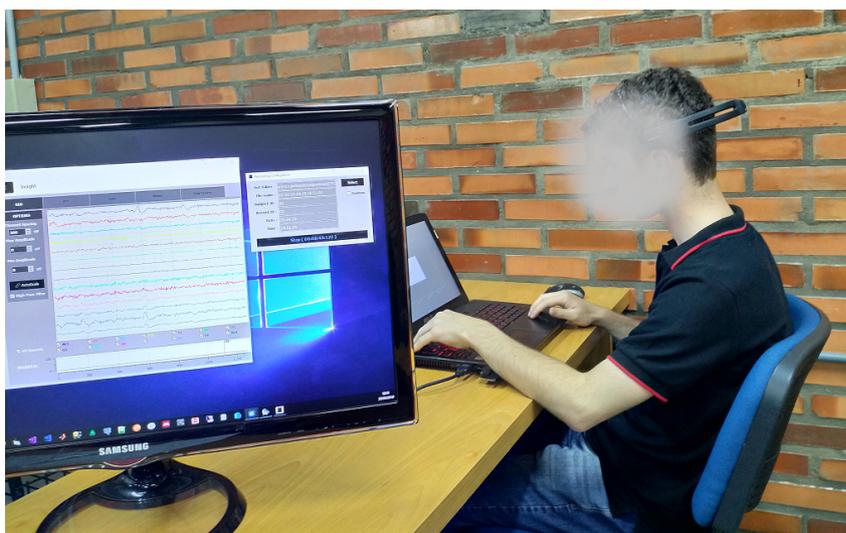


Figura 15 – Participante durante a execução do experimento.

Fonte: Elaborado pelo autor.

Ao concluir este processo, deu-se início a execução das questões do experimento. Cada questão apresentou um período de até 1,5 minutos intercaladas por um intervalo de 5 segundos entre cada questão. No total, o participante poderia levar até 16 minutos para a realização do experimento. Durante o processo de execução do experimento, eventuais falhas em sensores foram anotadas nos momentos em que aconteciam para que nos casos de falhas que pudessem interferir no resultado, o teste fosse reiniciado. Um dos participantes do experimento teve os seus dados descartados por não ser possível realizar as leituras biométricas necessárias devido aos seus cabelos, resultando nos 40 participantes do experimento.

Quando o participante chegava ao fim do experimento, a gravação dos dados de EEG era finalizada, o Emotiv EPOC+ removido do participante e os dados registrados eram verificados com o intuito de confirmar se tudo havia transcorrido adequadamente durante o processo. O participante também era indagado sobre as suas impressões durante o processo e eventuais desconfortos durante a execução.

Durante a segunda execução do processo experimental, o termo de consentimento, a apresentação do experimento e o questionário de identificação da expertise do participante não foram aplicados. Somente uma breve revisão do processo foi executada e deu-se início ao processo de coleta dos dados, mantendo-se inalterado o protocolo de coleta de dados relatado nos parágrafos anteriores.

4.8 Procedimentos de Análise

Após a conclusão do experimento, deu-se início ao processo de análise dos dados coletados. Cada participante do experimento forneceu um conjunto de dados de EEG e dados de resposta as questões realizadas. Os dados de EEG consistem em um arquivo único no formato EDF para cada um dos participantes, enquanto o arquivo de respostas contém todas as respostas de cada uma das questões aplicadas aos participantes, juntamente com um conjunto de informações adicionais, como o momento de início e término de cada questão e de cada intervalo de descanso entre as questões. Este arquivo de respostas foi gerado pela aplicação OpenSesame e salvo no formato CSV. O experimento resultou em 40 arquivos do tipo EDF e 40 arquivos do tipo CSV por execução, fornecendo no total 80 arquivos EDF e 80 arquivos CSV ao término das duas execuções.

O primeiro procedimento para analisar os dados coletados, foi processar o arquivo EDF de cada participante, extraindo apenas os dados do EEG do intervalo em que o participante estava respondendo a cada uma das questões do experimento. Para isso, foram utilizados os marcadores disparados pela aplicação OpenSesame aos dados do EEG coletados durante o experimento. Com o emprego dos marcadores, foram extraídos 10 intervalos de dados de EEG associados ao período em que o participante estava respondendo a cada uma das questões. A este tempo, foi descontado um segundo no início da marcação e um segundo ao final da marcação com o propósito de desconsiderar eventuais movimentos do participante no início e no final da questão.

A execução deste procedimento resultou em 800 amostradas de dados de EEG divididas entre a primeira e a segunda execução do experimento. Sobre estas amostras de dados, os sinais de EEG foram submetidos a técnica Multitaper (BABADI; BROWN, 2014). Esta técnica realiza a conversão de um conjunto de dados de cada um dos sensores do dispositivo EEG em um único valor, representando a energia de espectro (power spectrum) para cada sensor. Esta técnica já realiza a conversão dos dados brutos em frequências de banda. Conforme (DAS ET AL., 2014), uma das possibilidades de analisar a carga cognitiva de um indivíduo é através da análise de espectro da frequência de banda *beta*. A partir dos valores obtidos pela aplicação da técnica Multitaper sobre a frequência de banda *beta*, aplica-se um filtro espacial (spatial filtering) com os dados coletados para ser possível chegar a um valor único para quantificar a carga mental em relação a todos os sensores do equipamento de EEG (HIPPE ET AL., 2014). A Figura 16 detalha a forma como este algoritmo é aplicado aos dados do EEG.

O filtro espacial foi aplicado apenas aos sensores AF3, F7, F3, FC5, T7 e os demais sensores do equipamento EEG não foram empregados, pois em trabalhos anteriores (DAS ET AL., 2014), já foi realizado uma identificação de quais sensores apresentam maior representatividade para a identificação da carga cognitiva. A aplicação deste processo resultou em um valor que pode variar de zero (sem carga cognitiva) até 1 (carga cognitiva máxima) para medir a carga cognitiva do participante durante a execução de cada uma das questões propostas para o experimento.

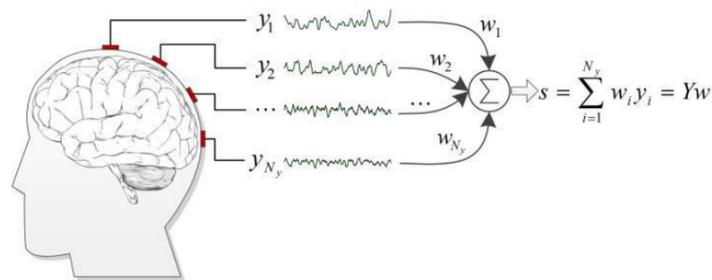


Figura 16 – Detalhamento do filtro espacial.

Fonte: (HIPPE ET AL., 2014).

4.9 Resultados Coletados

Após a conclusão do experimento foi iniciada a verificação dos dados, procurando investigar a relação existente entre a carga cognitiva, dificuldade das tarefas e expertise técnica dos participantes, além de quantificar a corretude das respostas. Os métodos estatísticos nortearam esta pesquisa, na qual procurou-se examinar todas as ocorrências dentro de uma população, isto é, a amostra a ser investigada para evidenciar o seu comportamento durante a execução do experimento. Os resultados coletados foram processados com o auxílio da ferramenta RStudio¹ e Excel², facilitando a visualização das variações dos resultados e a sua parametrização.

De acordo com a descrição apresentada na Seção 4.5 este experimento possui 10 tarefas, que foram analisadas e executadas pelos participantes gerando o resultado necessário para mensurar a carga cognitiva e corretude proveniente das tarefas executadas. A Tabela 24 apresenta o conjunto total de dados utilizados para as análises estatísticas aplicadas nesta Seção.

¹Site: <https://rstudio.com/>

²Site: <https://products.office.com/pt-br/excel>

Tabela 24 – Resultados gerais do experimento.

Participante	Etapa	T01		T02		T03		T04		T05		T06		T07		T08		T09		T10	
		Corretude (C)	Carga Cognitiva (CC)																		
P01	Parte 1	1	0,051249	1	0,048346	1	0,050561	1	0,066801	1	0,032609	0	0,047860	1	0,041820	1	0,038858	1	0,039004	1	0,041223
	Parte 2	1	0,123345	1	0,134496	1	0,064074	1	0,060469	1	0,097862	1	0,083234	1	0,094971	1	0,043319	0	0,064004	0	0,116786
P02	Parte 1	0	0,030310	1	0,032125	1	0,025372	1	0,024724	0	0,023039	0	0,030908	1	0,037964	1	0,029926	0	0,035132	1	0,030244
	Parte 2	0	0,076966	1	0,067030	1	0,072883	1	0,046613	0	0,052919	0	0,067242	1	0,045892	0	0,070672	1	0,060865	1	0,060931
P03	Parte 1	0	0,235053	1	0,191520	1	0,046490	0	0,250605	1	0,189087	1	0,156501	1	0,234096	0	0,140544	0	0,116446	0	0,117904
	Parte 2	1	0,153039	1	0,173214	1	0,196824	1	0,007046	0	0,082001	0	0,197046	1	0,167888	0	0,273241	0	0,186222	1	0,148496
P04	Parte 1	1	0,074321	1	0,076777	1	0,073130	1	0,128177	1	0,085325	1	0,082541	0	0,084503	0	0,060499	0	0,038625	0	0,072869
	Parte 2	1	0,070660	1	0,048460	1	0,075961	1	0,042792	0	0,053736	1	0,052682	0	0,072919	1	0,051544	1	0,056293	1	0,159861
P05	Parte 1	1	0,099179	1	0,048124	1	0,055316	1	0,106054	1	0,111306	0	0,157378	0	0,138579	1	0,086109	0	0,158741	1	0,134472
	Parte 2	0	0,055959	1	0,057701	1	0,058470	1	0,083835	1	0,067860	1	0,078771	0	0,063017	1	0,053374	1	0,076580	1	0,113974
P06	Parte 1	1	0,044599	1	0,074952	1	0,075479	1	0,087582	0	0,045778	1	0,047961	1	0,052589	1	0,063149	1	0,064341	1	0,074467
	Parte 2	1	0,053026	1	0,082366	1	0,045158	1	0,039541	0	0,063264	1	0,067044	0	0,060189	0	0,035878	0	0,058268	1	0,064956
P07	Parte 1	1	0,030349	1	0,042164	1	0,044802	1	0,039504	1	0,013989	1	0,049927	0	0,032769	1	0,028206	1	0,032685	1	0,080049
	Parte 2	1	0,110946	1	0,101103	1	0,103593	1	0,084598	0	0,126692	1	0,130807	1	0,118999	1	0,077806	1	0,093679	1	0,105344
P08	Parte 1	1	0,109089	1	0,118146	0	0,071450	1	0,037765	1	0,081931	1	0,090504	1	0,127599	1	0,073810	1	0,133129	1	0,044706
	Parte 2	1	0,087002	0	0,197389	1	0,041140	1	0,107780	0	0,102561	1	0,065187	1	0,042844	1	0,031559	0	0,082136	1	0,135261
P09	Parte 1	1	0,066688	1	0,135119	1	0,106824	1	0,120214	1	0,104268	1	0,115894	1	0,060601	1	0,099685	0	0,117008	1	0,115656
	Parte 2	1	0,061041	1	0,074608	1	0,079381	1	0,062400	1	0,077881	1	0,055066	0	0,063798	0	0,057234	1	0,050401	0	0,099579
P10	Parte 1	1	0,048194	1	0,016823	1	0,040471	1	0,016134	1	0,026051	0	0,011281	1	0,034424	0	0,012293	1	0,010525	1	0,013864
	Parte 2	1	0,041563	1	0,061665	1	0,021589	1	0,042767	1	0,016884	1	0,038856	0	0,043519	0	0,034612	0	0,093005	1	0,060119
P11	Parte 1	1	0,027611	1	0,076253	1	0,038990	1	0,054901	1	0,001261	1	0,019147	0	0,025246	1	0,056173	0	0,019945	1	0,002315
	Parte 2	1	0,020589	1	0,017638	1	0,056301	1	0,037948	1	0,016192	1	0,062966	0	0,054798	1	0,056596	1	0,020466	1	0,032116
P12	Parte 1	1	0,112840	1	0,115152	1	0,103921	1	0,099494	0	0,084661	1	0,101037	1	0,110536	0	0,113684	0	0,112839	1	0,103282
	Parte 2	1	0,098869	1	0,026309	1	0,045071	1	0,156636	0	0,115220	1	0,112820	0	0,129668	0	0,024859	0	0,093747	1	0,257884
P13	Parte 1	1	0,054381	1	0,059649	1	0,048184	1	0,053677	1	0,055507	1	0,050084	1	0,049729	0	0,048024	0	0,057133	1	0,049691
	Parte 2	1	0,042324	1	0,044051	1	0,044689	0	0,055291	1	0,051804	1	0,051944	0	0,051011	1	0,047534	0	0,049320	1	0,030556
P14	Parte 1	1	0,109781	1	0,079379	0	0,079057	1	0,092602	0	0,113427	1	0,086138	0	0,101895	1	0,085310	0	0,070527	1	0,097955
	Parte 2	1	0,138527	1	0,058715	1	0,009426	1	0,205536	1	0,000603	1	0,073714	0	0,052474	1	0,077919	0	0,081576	1	0,090144
P15	Parte 1	1	0,190874	0	0,161867	0	0,110596	1	0,139942	1	0,168013	1	0,185531	0	0,128658	1	0,168550	1	0,121339	1	0,112671
	Parte 2	1	0,078002	1	0,120589	1	0,090384	1	0,111770	1	0,142269	1	0,095412	1	0,125796	1	0,102950	1	0,093322	1	0,086601
P16	Parte 1	1	0,204769	1	0,123462	1	0,083069	1	0,022541	1	0,040706	0	0,167471	0	0,030020	1	0,019691	1	0,116524	0	0,061850
	Parte 2	0	0,033974	1	0,075191	1	0,050854	1	0,174635	0	0,066913	1	0,069176	0	0,120034	1	0,120737	0	0,031337	0	0,116617
P17	Parte 1	1	0,098344	0	0,144171	0	0,104540	0	0,125807	0	0,148778	0	0,144358	0	0,061897	1	0,138187	1	0,127674	1	0,111905
	Parte 2	1	0,015186	1	0,073884	0	0,102330	0	0,104191	0	0,019258	1	0,106611	1	0,011283	1	0,094439	0	0,005793	0	0,005793
P18	Parte 1	1	0,022339	1	0,037784	1	0,056177	0	0,017086	0	0,010208	0	0,005788	0	0,033439	0	0,016098	0	0,016092	1	0,068624
	Parte 2	1	0,100514	1	0,194586	1	0,093375	1	0,091229	0	0,099201	1	0,088136	0	0,101701	1	0,060253	1	0,153162	1	0,113460
P19	Parte 1	1	0,042191	0	0,127851	1	0,095917	0	0,046643	1	0,050553	1	0,104046	0	0,095139	0	0,085024	0	0,045350	1	0,173265
	Parte 2	0	0,052006	1	0,027858	1	0,019444	0	0,068326	1	0,026936	0	0,030138	0	0,052931	1	0,034234	1	0,010591	1	0,041979
P20	Parte 1	0	0,077604	0	0,119586	0	0,119607	0	0,091171	0	0,091521	0	0,073911	0	0,095832	0	0,114840	1	0,107217	1	0,118705
	Parte 2	0	0,071892	0	0,132416	1	0,070062	1	0,071462	1	0,073209	0	0,053414	0	0,007714	1	0,058639	0	0,184694	1	0,107539
P21	Parte 1	1	0,122444	1	0,221023	0	0,128973	1	0,118679	1	0,119690	1	0,121658	0	0,024193	0	0,102361	0	0,143637	1	0,047240
	Parte 2	1	0,065441	1	0,092954	0	0,095306	0	0,085999	1	0,136700	0	0,078394	0	0,080400	0	0,082679	0	0,079369	1	0,035841
P22	Parte 1	1	0,100941	1	0,078342	0	0,070198	1	0,094704	0	0,054279	1	0,086871	0	0,078957	0	0,066859	0	0,080586	1	0,068317
	Parte 2	1	0,065745	1	0,071186	1	0,088204	1	0,071610	1	0,082763	0	0,044759	1	0,035179	0	0,074594	1	0,037714	1	0,058658
P23	Parte 1	1	0,046217	1	0,092944	1	0,094841	1	0,004635	1	0,002726	1	0,003175	0	0,001619	0	0,002797	0	0,001627	1	0,071081
	Parte 2	1	0,084677	1	0,038034	1	0,099586	1	0,044496	1	0,115159	1	0,065432	0	0,081669	0	0,051700	1	0,014401	0	0,112716
P24	Parte 1	0	0,036721	1	0,016362	0	0,033607	1	0,076026	0	0,099024	0	0,070578	0	0,080656	1	0,041320	0	0,035134	1	0,046526
	Parte 2	1	0,052992	1	0,030604	0	0,053239	1	0,043579	0	0,054181	0	0,061844	0	0,057819	0	0,071452	0	0,052483	1	0,047857
P25	Parte 1	1	0,059242	1	0,209634	0	0,028417	1	0,190274	0	0,016155	0	0,048139	0	0,038992	0	0,169296	0	0,018273	0	0,047682
	Parte 2	1	0,072469	1	0,035182	1	0,075555	1	0,053639	0	0,088606	0	0,073546	0	0,077486	1	0,092565	1	0,080724	1	0,117639
P26	Parte 1	0	0,021325	1	0,000430	1	0,131241	0	0,006974	0	0,050070	0	0,148394	0	0,030458	0	0,063275	0	0,032795	1	0,016430
	Parte 2	1	0,072229	1	0,072463	1	0,054143	0	0,094707	1	0,086372	0	0,049404	0	0,105191	1	0,058381	0	0,055485	0	0,073308
P27	Parte 1	0	0,112832	0	0,115806	1	0,122931	1	0,139844	0	0,109921	1	0,134696	0	0,168719	0	0,098658	1	0,164499	1	0,134663
	Parte 2	0	0,097834	1	0,132655	1	0,120107	0	0,111391	0	0,135120	1	0,137769	0	0,137329	0	0,085906	0	0,105844	1	0,109611
P28	Parte 1	1	0,051999	1	0,080529	0	0,068829	1	0,063712	0	0,073023	0	0,079013	1	0,053061	1	0,058175	0	0,046290	1	0,053933
	Parte 2	1	0,132417	0	0,024691	1	0,038744	1	0,143144	0	0,028459	0	0,168208	0	0,163180	1	0,149080	0	0,023544	1	0,193491
P29	Parte 1	1	0,111782	1	0,114491	1	0,062210	1	0,087361	1	0,150169	1	0,068807	1	0,079511	1	0,100502	1	0,122301	0	0,087137

Os dados para a análise da corretude foram obtidos através da extração dos dados gerados pelo software *Open Sesame* enquanto este era utilizado na execução do experimento, enquanto os dados para análise da carga cognitiva foram obtidos pelo uso do equipamento de eletroencefalograma junto ao participante. A análise destes dados foi realizada com a utilização da estatística descritiva a qual verificou a distribuição dos mesmos, enquanto que para os testes de hipóteses foi aplicada a inferência estatística. O total de amostras deste experimento é igual a 800, ou seja, 400 amostras na primeira etapa do experimento e 400 amostras resultantes da segunda etapa do experimento, executada dois meses após a primeira execução. A análise da normalidade das amostras foi executada através do teste *Kolmogorov-Smirnov*, utilizado para grandes amostras (quando $n \geq 30$) (Levine et al., 2005).

A Tabela 25 apresenta o resultado do nível descritivo com o uso da técnica estatística de Kolmogorov-Smirnov, com um nível de significância de *Lilliefors* para teste de normalidade, os resultados retornados da significância, também conhecida como valor $p < 0,05$ para o teste nas amostras analisadas. Desta forma, pode-se afirmar que o nível de significância de 5% não provém de uma população normal, tanto pelos indicadores de corretude como de carga cognitiva.

Tabela 25 – Teste de normalidade.

Corretude	Kolmogorov-Smirnov*		
	Estatística	df	Sig.
1º Etapa	0,435	400	0,000
2º Etapa	0,415	400	0,000
Carga Cognitiva	Estatística	df	Sig.
1º Etapa	0,125	400	0,000
2º Etapa	0,116	400	0,000

* Correlação de Significância de Lilliefors

Fonte: Elaborado pelo autor.

A Tabela 26 e a Tabela 27 apresentam os resultados agrupados pelas tarefas aplicadas, demonstrando os resultados de: média, mediana desvio padrão, máximo, mínimo, primeiro quartil e terceiro quartil, separados por corretude e carga cognitiva. Por sua vez a Tabela 28 abstrai as informações por tarefa aplicada, corretude e carga cognitiva, demonstrando a média dos resultados individuais das variáveis e o percentual de diferença entre cada tratamento.

A partir de uma análise geral em relação a carga cognitiva e também corretude aplicadas as tarefas executadas pelos participantes a partir da evolução das expertises consideradas naturais devido a diferença de tempo entre a execução das duas etapas do experimento foi possível identificar um aumento na corretude de 7,14% e uma redução média da carga cognitiva entre a primeira e segunda etapa do experimento de 10,03%.

A Figura 17 exhibe o comparativo da corretude entre a primeira etapa da execução do experimento em relação a corretude obtida após a segunda execução do experimento, sendo produzido um total de 800 respostas, das quais 400 respostas foram geradas na primeira etapa e as outras 400 foram geradas com a segunda etapa. Pode-se observar que os participantes responderam a

Tabela 26 – Resultados agrupados por tarefa - Corretude.

		T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	
Participantes		40	40	40	40	40	40	40	40	40	40	
Corretude	1º Etapa	Média	0,7	0,85	0,67	0,75	0,5	0,45	0,27	0,4	0,2	0,87
		Mediana	1	1	1	1	0,5	0	0	0	0	1
		Desvio Padrão	0,46	0,36	0,47	0,44	0,51	0,50	0,45	0,50	0,41	0,33
		Mínimo	0	0	0	0	0	0	0	0	0	0
		Máximo	1	1	1	1	1	1	1	1	1	1
		1º Quartil	0	1	0	0,75	0	0	0	0	0	1
		3º Quartil	1	1	1	1	1	1	1	1	0	1
Corretude	2º Etapa	Média	0,72	0,82	0,85	0,75	0,45	0,62	0,22	0,5	0,22	0,87
		Mediana	1	1	1	0,75	0	0	0	0,5	0	1
		Desvio Padrão	0,45	0,38	0,36	0,43	0,50	0,48	0,42	0,50	0,42	0,33
		Mínimo	0	0	0	0	0	0	0	0	0	0
		Máximo	1	1	1	1	1	1	1	1	1	1
		1º Quartil	0	1	1	0,75	0	0	0	0	0	1
		3º Quartil	1	1	1	1	1	1	0	1	0	1

Fonte: Elaborado pelo autor.

227 tarefas de forma assertiva na primeira etapa, já durante a segunda etapa o número de acertos aumentou para 242. Desta forma é possível concluir que o ganho de expertise dos participantes entre o período da primeira etapa até a segunda etapa do experimento proporcionou uma melhora na corretude das respostas desempenhadas por eles.

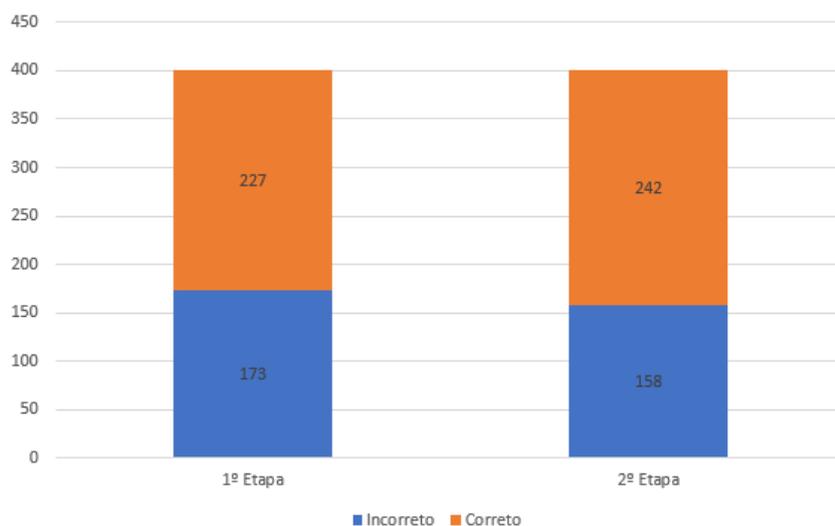


Figura 17 – Corretude por experimento.

Fonte: Elaborado pelo autor.

A Figura 18 exibe um comparativo da carga cognitiva média capturada dos participantes para cada uma das tarefas do experimento. Nesta figura é possível identificar que a carga cognitiva média que os participantes geraram para desenvolver as tarefas na segunda etapa do experimento

Tabela 27 – Resultados agrupados por tarefa - Carga Cognitiva.

		T01	T02	T03	T04	T05	T06	T07	T08	T09	T10
Participantes		10	40	40	40	40	40	40	40	40	40
Carca Cognitiva 1º Etapa	Média	0,0853	0,0947	0,0768	0,0869	0,0872	0,0941	0,0858	0,0793	0,0764	0,0914
	Mediana	0,0771	0,0867	0,0762	0,0893	0,0884	0,0843	0,0797	0,0703	0,0679	0,0890
	Desvio Padrão	0,0537	0,0539	0,0319	0,0537	0,0580	0,0555	0,0560	0,0491	0,0537	0,0600
	Mínimo	0,0026	0,0043	0,0067	0,0046	0,0012	0,0031	0,0016	0,0027	0,0016	0,0022
	Máximo	0,2350	0,2210	0,1312	0,2506	0,2525	0,2173	0,2340	0,1881	0,2225	0,3397
	1º Quartil	0,4558	0,0559	0,0499	0,0479	0,0445	0,0500	0,0387	0,0407	0,0327	0,0491
	3º Quartil	0,1126	0,1289	0,1045	0,1190	0,1198	0,1382	0,1191	0,1025	0,1166	0,1162
Carca Cognitiva 2º Etapa	Média	0,0712	0,0771	0,0673	0,0815	0,0739	0,0763	0,0808	0,0680	0,0756	0,0994
	Mediana	0,0712	0,0674	0,0642	0,0715	0,0753	0,0671	0,0683	0,0585	0,0675	0,0948
	Desvio Padrão	0,0365	0,0480	0,0393	0,0451	0,0421	0,0405	0,0441	0,0452	0,0486	0,0611
	Mínimo	0,0021	0,0024	0,0040	0,0057	0,0006	0,0018	0,0016	0,0021	0,0015	0,0034
	Máximo	0,1530	0,1973	0,1968	0,2055	0,1422	0,1970	0,1678	0,2732	0,1862	0,2778
	1º Quartil	0,0445	0,0438	0,0438	0,0460	0,0450	0,0515	0,0521	0,0424	0,0405	0,0573
	3º Quartil	0,0992	0,1003	0,0911	0,1050	0,1057	0,0937	0,1204	0,0817	0,0937	0,1220

Fonte: Elaborado pelo autor.

foi inferior em relação a execução das tarefas na primeira etapa. Desse modo, apresentam-se evidências de que a carga cognitiva do desenvolvedor sofre redução a medida que ele adquire expertises técnicas para a execução das tarefas.

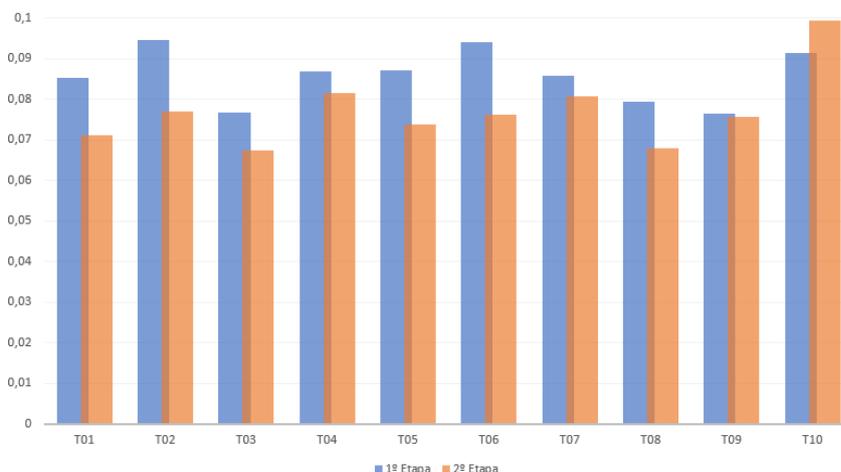


Figura 18 – Carca cognitiva média por tarefa.

Fonte: Elaborado pelo autor.

Aplicando os dados coletados com a ferramenta RStudio foi possível obter os cálculos de cada variável, as quais foram subdivididas conforme o tratamento, em categorias, tendo em vista diferentes resultados incorporados a Tabela 29. Nesta tabela são incluídos os resultados para as variáveis de quantificação, Corretude (C), que apresenta a corretude para a execução das tarefas, ou seja, quanto maior o índice, melhor será a capacidade de retornar a resposta correta. Da mesma forma a variável de Carca Cognitiva (CC), apresenta a carga cognitiva média que os participantes geraram durante a execução das tarefas. O principal motivo em aplicar os cálculos pela estatística descritiva com o uso da ferramenta RStudio advém deste retornar dos resultados

Tabela 28 – Relação das variáveis de quantificação.

Tarefas	Variáveis	Etapas	Média	% dif.
Todas	Corretude	1º Etapa	0,56	7,14
		2º Etapa	0,60	
	Carca Cognitiva	1º Etapa	0,0857	10,03
		2º Etapa	0,0771	
T01	Corretude	1º Etapa	0,70	2,85
		2º Etapa	0,72	
	Carca Cognitiva	1º Etapa	0,0853	16,52
		2º Etapa	0,0712	
T02	Corretude	1º Etapa	0,85	0,52
		2º Etapa	0,82	
	Carca Cognitiva	1º Etapa	0,0947	18,58
		2º Etapa	0,0771	
T03	Corretude	1º Etapa	0,67	26,86
		2º Etapa	0,85	
	Carca Cognitiva	1º Etapa	0,0768	12,36
		2º Etapa	0,0673	
T04	Corretude	1º Etapa	0,75	0,0
		2º Etapa	0,75	
	Carca Cognitiva	1º Etapa	0,0869	6,21
		2º Etapa	0,0815	
T05	Corretude	1º Etapa	0,50	10
		2º Etapa	0,45	
	Carca Cognitiva	1º Etapa	0,0872	15,25
		2º Etapa	0,0739	
T06	Corretude	1º Etapa	0,45	37,77
		2º Etapa	0,62	
	Carca Cognitiva	1º Etapa	0,0941	18,91
		2º Etapa	0,0763	
T07	Corretude	1º Etapa	0,27	18,51
		2º Etapa	0,22	
	Carca Cognitiva	1º Etapa	0,0858	5,82
		2º Etapa	0,0808	
T08	Corretude	1º Etapa	0,40	25
		2º Etapa	0,50	
	Carca Cognitiva	1º Etapa	0,0793	14,24
		2º Etapa	0,0680	
T09	Corretude	1º Etapa	0,20	10
		2º Etapa	0,22	
	Carca Cognitiva	1º Etapa	0,0764	1,04
		2º Etapa	0,0756	
T10	Corretude	1º Etapa	0,87	0,0
		2º Etapa	0,87	
	Carca Cognitiva	1º Etapa	0,0914	8,75
		2º Etapa	0,0994	

Fonte: Elaborado pelo autor.

Tabela 29 – Estatística descritiva e testes estatísticos.

Variáveis	Tratamento	DP	Mín	25th	MD	75th	Máx	Méd	%df	W p-v	N p-v
Corretude	1º Etapa	0,46	0,00	0,00	1,00	1,00	1,00	0,68	7,14	-	0,5108
	2º Etapa	0,48	0,00	0,00	1,00	1,00	1,00	0,64			
Carga Cognitiva	1º Etapa	0,0534	0,0012	0,0378	0,0655	0,1085	0,2525	0,0760	10,03	0,1487	-
	2º Etapa	0,0452	0,0006	0,0519	0,0675	0,1021	0,2732	0,0798			

Fonte: Elaborado pelo autor.

de testes estáticos e testes de hipóteses: desvio padrão, mínimo e máximo, média, mediana e por fim a aplicação dos testes de *Wilcoxon* e o *McNemar*.

O teste de normalidade de *Kolmogorov-Smirnov* - (Lilliefors) (BISCHOFF, 2018; OLIVEIRA, 2012; MOHD RAZALI; YAP, 2011) executado anteriormente indica que os dados não se encontram normalmente distribuídos, sendo necessário aplicar testes não paramétricos. Por isso, o teste de *Wilcoxon* é aplicado para analisar o esforço, enquanto o teste *McNemar* foi empregado para analisar a corretude da abordagem proposta. Estes testes serão empregados para analisar os testes de hipóteses abordados na Seção 4.2 e responder as questões de pesquisa descritas na Seção 4.1.

O teste de *Wilcoxon* é um teste de hipótese estático não paramétrico, aplicado para confrontar a média de duas amostras relacionadas, ou seja, é um teste de diferença emparelhado que pode ser utilizado como alternativa ao *t-test*. O *t-test* é considerado um teste de hipótese que utiliza conceitos estatísticos para rejeitar ou não uma hipótese nula, empregado para confrontar médias de amostras diferentes, normalmente o *t-test* é aplicado quando segue uma distribuição normal (BARROS; MAZUCHELI, 2005).

O teste *McNemar* é usado para analisar a eficiência de determinada técnica, isto é, tem por objetivo avaliar a eficiência de situações, "o antes e o depois" de cada amostra utilizada e mensurar se faz nível de escala nominal ou ordinal. Este teste é aplicado a variáveis dicotômicas, ou seja, à amostras que apenas tomam dois valores, por exemplo, 0 e 1 (BISCHOFF, 2018; FARIAS ET AL., 2015; FIRMINO, 2015).

4.9.1 QP1: É possível quantificar a carga cognitiva de um desenvolvedor? (hipótese 1)

Esta questão de pesquisa investiga a carga cognitiva, analisando a variação desta carga cognitiva dos desenvolvedores durante a execução das tarefas de desenvolvimento em um momento inicial e em um segundo momento com um intervalo de dois meses. Ao verificar os dados contidos na Tabela 28 foi diagnosticado que a carga cognitiva aplicada pelos participantes para realizar as tarefas foram em sua maioria, inferior durante a segunda etapa do experimento em relação primeira etapa. A Tabela 28 exibe a carga cognitiva média por tarefa, bem como o registro acumulado de todas as tarefas, o qual apresenta uma carga cognitiva de 0,0857 na primeira etapa e 0,0771 durante a segunda etapa. A redução da carga cognitiva gerada pelo desenvolvedor durante a resolução das tarefas pode também ser observada na Tabela 29, na qual é apresentada a diferença de 10,03% na carga cognitiva, ou seja, a carga cognitiva gerada durante

a segunda etapa do experimento foi 10,3% menor do que o registrado inicialmente.

Tabela 30 – Resultado do esforço - teste Wilcoxon.

Teste de Wilcoxon	
Pares(n)	400
Z	7574
p-Valor	<0,1487

Fonte: Elaborado pelo autor.

Analisando a H1-0 tem-se para a hipótese nula o teste de wilcoxon, não paramétrico que pode ser visto na Tabela 29 e Tabela 30. É possível verificar que a estatística coletada da significância é $< 0,1487$, com um intervalo de confiança de 95% em uma amostra de 400 pares, evidenciando que existe diferença de carga cognitiva entre as etapas de execução, rejeitando desta forma a hipótese nula.

Para analisar H1-1 são considerados os dados da Tabela 24, Tabela 28 e Tabela 29. Pode-se evidenciar que a carga cognitiva média para chegar a uma resposta é menor quando o desenvolvedor possuir mais tempo de experiência, ficando em 0,0771, enquanto na primeira execução a carga cognitiva foi de 0,0857. O percentual de carga cognitiva é detalhado na Tabela 28, na qual é possível identificar que a carga cognitiva gerada a partir da execução das tarefas no primeiro momento é superior ao valor gerado durante a execução das tarefas no segundo momento. Com isso, é possível identificar que os participantes ao possuírem mais tempo de experiência, conseguiram chegar a uma resposta com maior eficácia. Verifica-se nesta hipótese que a carga cognitiva para executar as tarefas de desenvolvimento é 10,03% menor após dois meses de tempo de experiência. Com isso, pode-se concluir que essa hipótese é válida, permitindo quantificar a carga cognitiva dos desenvolvedores.

Colocando em escala os resultados desta análise da carga cognitiva por tarefa de acordo com a Tabela 24 obtém-se o gráfico da Figura 18. A figura evidência mais nitidamente a diferença de carga cognitiva gerada para a resolução de tarefas entre os diferentes momentos de execução do experimento, permitindo identificar que em um momento inicial, onde o desenvolvedor apresenta menor experiência, a carga cognitiva despendida para a tarefa será maior, possibilidade identificar diferenças na carga cognitiva, tornando esta uma medida válida para a quantificação das expertises dos desenvolvedores.

4.9.2 QP2: A carga cognitiva do desenvolvedor permite a classificação da sua expertise? (hipótese 2)

Esta questão investiga a possibilidade de classificar as expertises do desenvolvedor de acordo com a sua carga cognitiva. Para viabilizar a resolução da hipótese necessária para responder a esta questão de pesquisa optou-se em dividir os participantes em dois grupos relativos a dois níveis de expertise a serem considerados como participantes com expertise técnica baixa e participantes com expertise técnica alta. A partir desta classificação, todos os participantes com

percentual de corretude geral sobre as tarefas realizadas igual ou inferior a 50% fazem parte do grupo de participantes com baixa expertise, enquanto que os participantes com um índice de corretude superior a 50% pertencem ao grupo de participantes alta expertise técnica conforme apresentado nas Tabelas 31 e 32, as quais apresentam esta classificação dos níveis de expertise dos participantes separadas pelas etapas de execução do experimento.

Tabela 31 – Carga cognitiva separada por níveis de expertise - 1º etapa.

Expertise	Participante	Carga Cognitiva									% Corretude	
		T01	T02	T03	T04	T05	T06	T07	T08	T09		T10
Expertise Baixa	P03	0,235053	0,191520	0,046490	0,250605	0,189087	0,156501	0,234096	0,140544	0,116446	0,117904	50
	P14	0,109781	0,079379	0,079057	0,092602	0,113427	0,086138	0,101895	0,085310	0,070527	0,097955	50
	P17	0,098344	0,144171	0,104540	0,125807	0,148779	0,144358	0,061897	0,188187	0,127674	0,111905	40
	P18	0,022339	0,037784	0,056177	0,017086	0,010208	0,005788	0,033439	0,016098	0,016092	0,068624	40
	P19	0,042191	0,127851	0,095917	0,046643	0,050553	0,104046	0,095139	0,085024	0,045350	0,173265	50
	P20	0,077604	0,119586	0,119607	0,091171	0,091521	0,073911	0,095832	0,114840	0,107217	0,118705	20
	P21	0,122444	0,221023	0,128973	0,118679	0,119690	0,121658	0,024193	0,102361	0,143637	0,047240	50
	P22	0,100941	0,078342	0,070198	0,094704	0,054279	0,086871	0,078957	0,066859	0,080586	0,068317	50
	P24	0,036721	0,016362	0,033607	0,076026	0,099024	0,070578	0,080656	0,041320	0,035134	0,046526	40
	P25	0,059242	0,209634	0,028417	0,190274	0,016155	0,048139	0,038992	0,169296	0,018273	0,047682	30
	P26	0,021325	0,004340	0,131241	0,006974	0,050070	0,148394	0,030458	0,063275	0,032795	0,016430	30
	P27	0,112832	0,115806	0,122931	0,139844	0,109921	0,134696	0,168719	0,098658	0,164499	0,134663	40
	P30	0,164882	0,149453	0,069697	0,105364	0,252519	0,165762	0,126858	0,151927	0,113359	0,177939	50
	P31	0,002682	0,024889	0,044015	0,048322	0,005403	0,147784	0,154686	0,124981	0,014874	0,002286	30
	P32	0,112622	0,130505	0,104653	0,190886	0,100634	0,217350	0,226499	0,022804	0,020159	0,339756	40
	P33	0,139089	0,130458	0,081794	0,146219	0,139309	0,064157	0,131993	0,052827	0,091716	0,091050	30
	P35	0,151784	0,128493	0,109811	0,097900	0,191649	0,197480	0,183656	0,181895	0,222589	0,148291	20
	P36	0,076698	0,058465	0,080784	0,025172	0,083026	0,071024	0,078328	0,034295	0,094620	0,153689	30
	P37	0,062330	0,077066	0,114767	0,095427	0,093686	0,072914	0,116572	0,090261	0,009012	0,130302	30
	P40	0,128941	0,142228	0,118181	0,125351	0,120514	0,115266	0,083516	0,103304	0,142966	0,101236	50
P01	0,051249	0,048346	0,050561	0,066801	0,032609	0,047860	0,041820	0,038858	0,039004	0,041223	90	
P02	0,030310	0,032125	0,025372	0,024724	0,023039	0,030908	0,037964	0,029926	0,035132	0,030244	60	
P04	0,074321	0,076777	0,073130	0,128177	0,085325	0,082541	0,084503	0,060499	0,038625	0,072869	70	
P05	0,099179	0,048124	0,055316	0,106054	0,111306	0,157378	0,138579	0,086109	0,158741	0,134472	70	
P06	0,044599	0,074952	0,075479	0,087582	0,045778	0,047961	0,052589	0,063149	0,064341	0,074467	90	
P07	0,030349	0,042164	0,044802	0,039504	0,013989	0,049927	0,032769	0,028206	0,032685	0,080049	90	
P08	0,109089	0,118146	0,071450	0,037765	0,081931	0,090504	0,127599	0,073810	0,133129	0,044706	80	
P09	0,066868	0,135119	0,106824	0,120214	0,104268	0,115894	0,060601	0,099685	0,117008	0,115656	90	
P10	0,048194	0,016823	0,040471	0,016134	0,026051	0,011281	0,034424	0,012293	0,010525	0,013864	70	
P11	0,027611	0,076253	0,038990	0,054901	0,001261	0,019147	0,025246	0,056173	0,019945	0,002315	80	
P12	0,112840	0,115152	0,103921	0,099494	0,084661	0,101037	0,110536	0,113684	0,112839	0,103282	70	
P13	0,054381	0,059649	0,048184	0,053677	0,055507	0,050084	0,049729	0,048024	0,057133	0,049691	80	
P15	0,190874	0,161867	0,110596	0,139942	0,168013	0,185531	0,128658	0,168550	0,121339	0,112671	60	
P16	0,204769	0,123462	0,083069	0,022541	0,040706	0,167471	0,030020	0,019691	0,116524	0,061850	60	
P23	0,046217	0,092944	0,094841	0,004635	0,002726	0,003175	0,001619	0,002797	0,001627	0,071081	70	
P28	0,051999	0,080529	0,068829	0,063712	0,073023	0,079013	0,053061	0,058175	0,046290	0,053933	60	
P29	0,111782	0,114491	0,062210	0,087361	0,150169	0,068807	0,079511	0,100502	0,122301	0,087137	90	
P34	0,014650	0,007054	0,006709	0,055558	0,134967	0,011610	0,005821	0,021202	0,010724	0,105971	80	
P38	0,084986	0,110117	0,096203	0,103682	0,124417	0,136153	0,112201	0,100687	0,088736	0,113269	70	
P39	0,081033	0,067401	0,077115	0,081540	0,091941	0,077009	0,080047	0,056859	0,065312	0,093709	70	

Fonte: Elaborado pelo autor.

Pode-se perceber através da análise das Tabelas 31 e 32, que durante a primeira execução do experimento 20 (50%) participantes foram classificados com expertise baixa e 20 (50%) com expertise alta, enquanto que na segunda execução 13 (32,5%) participantes foram classificados com expertise baixa e 27 (67,5%) com expertise alta, evidenciando um aumento de expertise dos participantes de 35% durante o intervalo de tempo de 2 meses utilizado para a execução das duas etapas do experimento.

Com base nos dados apresentados nas Tabelas 31 e 32, calculou-se a média e mediana da carga cognitiva dos participantes agrupadas por nível de expertise e por etapa do experimento conforme detalhado nas Tabelas 33 e 34, possibilitando a construção da Tabela 35, a qual apresenta a diferença entre a mediana da carga cognitiva entre participantes de expertise baixa e participantes de expertise alta nas duas etapas do experimento.

A Tabela 28 exibe a carga cognitiva média por tarefa separada por nível de expertise e etapa, bem como o registro acumulado de todas as tarefas, o qual apresenta uma carga cognitiva de 0,090926 para os participantes considerados com expertises baixas e 0,073143 para os participantes com expertises altas. A redução da carga cognitiva apresentada pelos participantes de

Tabela 32 – Carga cognitiva separada por níveis de expertise - 2º etapa.

Expertise	Participante	Carga Cognitiva										% Corretude
		T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	
Expertise Baixa	P16	0,033974	0,075191	0,050854	0,174635	0,066913	0,069176	0,120034	0,120737	0,031337	0,116617	50
	P19	0,052006	0,027858	0,019444	0,068326	0,026936	0,030138	0,052931	0,034234	0,010591	0,041979	50
	P20	0,071892	0,132416	0,070062	0,071462	0,073209	0,053414	0,007714	0,058639	0,184694	0,107539	50
	P21	0,065441	0,092954	0,095306	0,085999	0,136700	0,078394	0,080400	0,082679	0,079369	0,035841	40
	P24	0,052992	0,030604	0,053239	0,043579	0,054181	0,061844	0,057819	0,071452	0,052483	0,047857	40
	P26	0,072229	0,072463	0,054143	0,094707	0,086372	0,049404	0,105191	0,058381	0,055485	0,073308	40
	P27	0,097834	0,132655	0,120107	0,111391	0,135120	0,137769	0,137329	0,085906	0,105844	0,109611	40
	P28	0,132417	0,024691	0,038744	0,143144	0,028459	0,168208	0,163180	0,149080	0,023544	0,193491	50
	P32	0,045254	0,060722	0,077946	0,093566	0,050297	0,054553	0,061324	0,049403	0,085564	0,078400	20
	P33	0,030294	0,063419	0,123417	0,030774	0,137528	0,031859	0,155323	0,017427	0,018404	0,007771	30
	P35	0,102205	0,138566	0,121194	0,087731	0,052261	0,102868	0,121771	0,081427	0,150588	0,168634	20
	P37	0,092525	0,095594	0,052637	0,074952	0,100487	0,092789	0,075801	0,076349	0,041541	0,053484	40
	P40	0,102321	0,100136	0,007616	0,141064	0,129011	0,091714	0,130867	0,106286	0,071087	0,161433	40
	P01	0,123345	0,134496	0,064074	0,060469	0,097862	0,083234	0,094971	0,043319	0,064004	0,116786	90
	P02	0,076966	0,067030	0,072883	0,046613	0,052919	0,067242	0,045892	0,070672	0,060865	0,060931	60
	P03	0,153039	0,173214	0,196824	0,007046	0,082001	0,197046	0,167888	0,273241	0,186222	0,148496	60
	P04	0,070660	0,048460	0,075961	0,042792	0,053736	0,052682	0,072919	0,051544	0,056293	0,159861	80
	P05	0,055959	0,057701	0,058470	0,083835	0,067860	0,078771	0,063017	0,053374	0,076580	0,113974	70
P06	0,053026	0,082366	0,045158	0,039541	0,063264	0,067044	0,060189	0,035878	0,058268	0,064956	60	
P07	0,110946	0,101103	0,103593	0,084598	0,126692	0,130807	0,118999	0,077806	0,093679	0,105344	90	
P08	0,087002	0,197389	0,041140	0,107780	0,102561	0,065187	0,042844	0,031559	0,082136	0,135261	70	
P09	0,061041	0,074608	0,079381	0,062400	0,077881	0,055066	0,063798	0,057234	0,050401	0,099579	80	
P10	0,041563	0,061665	0,021589	0,042767	0,016884	0,038856	0,043519	0,034612	0,093005	0,060119	70	
P11	0,020589	0,017638	0,056301	0,037948	0,016192	0,062966	0,054798	0,056596	0,020466	0,032116	90	
P12	0,098869	0,026309	0,045071	0,156636	0,115220	0,112820	0,129668	0,024859	0,093747	0,257884	60	
P13	0,042324	0,044051	0,044689	0,055291	0,051804	0,051944	0,051011	0,047534	0,049320	0,030556	70	
P14	0,138527	0,058715	0,009426	0,205536	0,000603	0,073714	0,052474	0,077919	0,081576	0,090144	80	
P15	0,078002	0,120589	0,090384	0,111770	0,142269	0,095412	0,125796	0,102950	0,093322	0,086601	90	
P17	0,015186	0,073884	0,102330	0,104191	0,019258	0,106611	0,013283	0,094439	0,015286	0,005793	60	
P18	0,100514	0,194586	0,093375	0,091229	0,099201	0,088136	0,101701	0,060253	0,153162	0,113460	70	
P22	0,065745	0,071186	0,088204	0,071610	0,082763	0,044759	0,035179	0,074594	0,037714	0,058658	60	
P23	0,084677	0,038034	0,099586	0,044496	0,115159	0,065432	0,081669	0,051700	0,014401	0,112716	60	
P25	0,072469	0,053182	0,075555	0,053639	0,088606	0,073546	0,077486	0,092565	0,080724	0,117639	60	
P29	0,105517	0,141011	0,126916	0,147411	0,133753	0,160584	0,160279	0,118586	0,138037	0,215169	90	
P30	0,119092	0,067949	0,083513	0,153815	0,135554	0,115076	0,126201	0,039989	0,157069	0,203725	60	
P31	0,002108	0,002472	0,011356	0,005760	0,060345	0,001826	0,001697	0,002101	0,001500	0,003429	60	
P34	0,012275	0,002499	0,004036	0,064203	0,000701	0,050412	0,025479	0,043271	0,095351	0,207232	90	
P36	0,059939	0,037832	0,064346	0,064150	0,028639	0,038954	0,057805	0,075206	0,057066	0,083999	60	
P38	0,025093	0,046536	0,029919	0,068766	0,019291	0,028394	0,058394	0,010518	0,033214	0,070495	60	
P39	0,023880	0,043182	0,023486	0,026005	0,029394	0,025999	0,038004	0,026100	0,171989	0,025591	60	

Fonte: Elaborado pelo autor.

Tabela 33 – Estatística sobre a carga cognitiva agrupada por níveis de expertise - 1º etapa.

Expertise	Estatística	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10
Expertise Baixa	Média	0,093892	0,109368	0,087043	0,104253	0,101973	0,111641	0,107319	0,096703	0,083376	0,109688
	Mediana	0,099642	0,123718	0,088855	0,096664	0,099829	0,109656	0,095486	0,094460	0,086151	0,106570
Expertise Alta	Média	0,076765	0,080075	0,066704	0,069700	0,072584	0,076665	0,064365	0,061944	0,069598	0,073123
	Mediana	0,060624	0,076515	0,070139	0,065257	0,077477	0,072908	0,052825	0,057517	0,060737	0,073668

Fonte: Elaborado pelo autor.

Tabela 34 – Estatística sobre a carga cognitiva agrupada por níveis de expertise - 2º etapa.

Expertise	Estatística	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10
Expertise Baixa	Média	0,073183	0,080559	0,068055	0,093949	0,082883	0,078625	0,097668	0,076308	0,070041	0,091997
	Mediana	0,071892	0,075191	0,054143	0,087731	0,073209	0,069176	0,105191	0,076349	0,055485	0,078400
Expertise Alta	Média	0,070309	0,075470	0,066947	0,075567	0,069645	0,075279	0,072776	0,064016	0,078348	0,102982
	Mediana	0,070660	0,061665	0,064346	0,064150	0,067860	0,067044	0,060189	0,053374	0,076580	0,099579

Fonte: Elaborado pelo autor.

maior expertise é de 19,55% se comparada aos participantes com menos expertise.

Para analisar as hipóteses H2-0 e H2-1 são considerados os dados da Tabela 35, Tabela 33 e Tabela 34. Pode-se evidenciar que a carga cognitiva média para chegar a uma resposta é menor para os desenvolvedores com maior expertise, ficando em 0,073143, enquanto a carga cognitiva média dos desenvolvedores com menor expertise foi de 0,090926. Verifica-se nesta hipótese que a carga cognitiva para executar as tarefas de desenvolvimento é 19,55% menor nos participantes classificados como de expertises altas. Com isso, pode-se concluir que essa hipótese é válida, permitindo classificar a expertise de um desenvolvedor através da sua carga cognitiva gerada enquanto está codificando uma tarefa.

Tabela 35 – Variação da carga cognitiva entre níveis de expertise.

Tarefa	Expertise	Etapa	Carga Cognitiva	Mediana	% dif
Geral	Expertise Baixa	Etapa 1	0,100526	0,090926	19,55
		Etapa 2	0,081327		
	Expertise Alta	Etapa 1	0,071152	0,073143	
		Etapa 2	0,075134		
T01	Expertise Baixa	Etapa 1	0,093892	0,083538	11,97
		Etapa 2	0,073183		
	Expertise Alta	Etapa 1	0,076765	0,073537	
		Etapa 2	0,070309		
T02	Expertise Baixa	Etapa 1	0,109368	0,094963	18,10
		Etapa 2	0,080559		
	Expertise Alta	Etapa 1	0,080075	0,077772	
		Etapa 2	0,075470		
T03	Expertise Baixa	Etapa 1	0,087043	0,077549	13,82
		Etapa 2	0,068055		
	Expertise Alta	Etapa 1	0,066704	0,066825	
		Etapa 2	0,066947		
T04	Expertise Baixa	Etapa 1	0,104253	0,099101	26,70
		Etapa 2	0,093949		
	Expertise Alta	Etapa 1	0,069700	0,072633	
		Etapa 2	0,075567		
T05	Expertise Baixa	Etapa 1	0,101973	0,092428	23,05
		Etapa 2	0,082883		
	Expertise Alta	Etapa 1	0,072584	0,071115	
		Etapa 2	0,069645		
T06	Expertise Baixa	Etapa 1	0,111641	0,095133	20,14
		Etapa 2	0,078625		
	Expertise Alta	Etapa 1	0,076665	0,075972	
		Etapa 2	0,075279		
T07	Expertise Baixa	Etapa 1	0,107319	0,102494	33,09
		Etapa 2	0,097668		
	Expertise Alta	Etapa 1	0,064365	0,068570	
		Etapa 2	0,072776		
T08	Expertise Baixa	Etapa 1	0,096703	0,086505	27,19
		Etapa 2	0,076308		
	Expertise Alta	Etapa 1	0,061944	0,062980	
		Etapa 2	0,064016		
T09	Expertise Baixa	Etapa 1	0,083376	0,076709	3,56
		Etapa 2	0,070041		
	Expertise Alta	Etapa 1	0,069598	0,073973	
		Etapa 2	0,078348		
T10	Expertise Baixa	Etapa 1	0,109688	0,100843	12,68
		Etapa 2	0,091997		
	Expertise Alta	Etapa 1	0,073123	0,088052	
		Etapa 2	0,102982		

Fonte: Elaborado pelo autor.

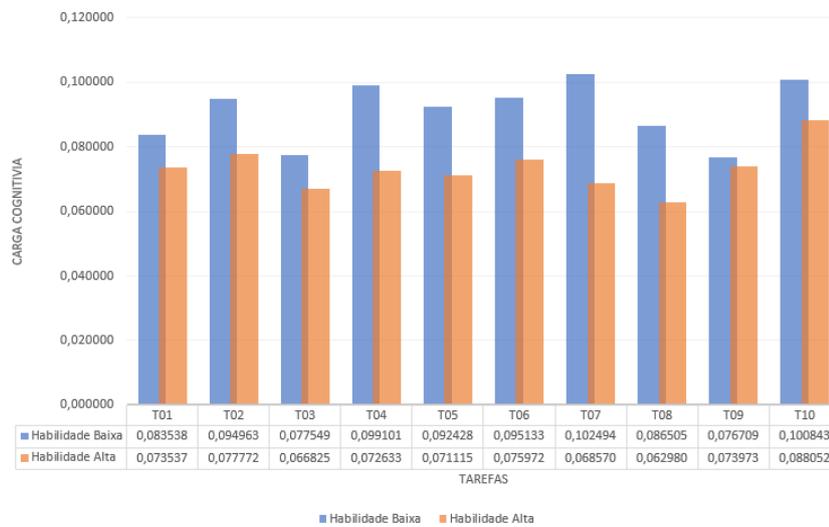


Figura 19 – Carca cognitiva média separada por nível de expertise.

Fonte: Elaborado pelo autor.

Colocando em escala os resultados desta análise da carga cognitiva por tarefa de acordo com a Tabela 35 obtém-se o gráfico da Figura 19. A figura evidencia mais nitidamente a diferença de carga cognitiva gerada para a resolução de tarefas entre os desenvolvedores com baixa expertise em relação aos desenvolvedores com alta expertise, gerando indícios que permitem utilizar a carga cognitiva como ferramenta para identificar a expertise de um desenvolvedor.

4.9.3 QP3: É possível identificar um limite de carga cognitiva, que quando ultrapassado, implique em redução de acuracidade durante a execução de tarefas de programação? (hipótese 3)

Esta questão procura identificar limites de carga cognitiva, que quando atingidos pelo desenvolvedor impliquem em maior possibilidade de falha durante a execução de uma tarefa de programação. Para responder a esta questão foram utilizadas as 800 amostras coletadas pelos participantes conforme apresentado na Tabela 24, as quais foram separadas em dois grupos distintos: o primeiro apresenta 331 amostras, sendo composto apenas da carga cognitiva das questões respondidas incorretamente, enquanto que o segundo grupo possui 469 amostras de carga cognitiva provenientes apenas das respostas corretas fornecidas pelos participantes.

As amostras foram então agrupadas pela corretude das tarefas e submetidas aos métodos de estatística descritiva para verificar a média, mediana, mínimo, máximo, 1º quartil, 3º quartil e desvio padrão conforme apresentado pela Tabela 36 e melhor visualizados através dos gráficos das Figuras 20 e 21.

Devido a esta questão procurar identificar um limite de carga cognitiva que possa indicar uma execução de tarefa que resulte em erro, foram analisadas as medianas e dispersão por intervalos de variação que indicam o valor do meio do conjunto de índices de carga cognitiva

Tabela 36 – Diferenças estatísticas da carga cognitiva entre acertos e erros em tarefas.

	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	
Acertos	Mínimo	0,002108	0,002472	0,004036	0,004635	0,000600	0,001826	0,013280	0,002100	0,010720	0,002315
	1º Quartil	0,051999	0,045294	0,044689	0,046084	0,029630	0,050248	0,042590	0,040700	0,039000	0,053596
	Mediana	0,076698	0,074608	0,060740	0,075489	0,083630	0,073714	0,056830	0,054810	0,064340	0,092380
	Média	0,081758	0,080858	0,067427	0,083870	0,083230	0,077437	0,084350	0,069990	0,070310	0,097730
	3º Quartil	0,109089	0,114822	0,093375	0,113497	0,123440	0,102542	0,120700	0,093030	0,093680	0,118439
	Máximo	0,204769	0,221023	0,196824	0,205536	0,252520	0,185531	0,234100	0,018819	0,016450	0,339756
	Desvio Padrão	0,042477	0,050185	0,036827	0,047767	0,058552	0,041624	0,055521	0,042170	0,038670	0,061236
Erros	Mínimo	0,002680	0,024690	0,028420	0,006970	0,005403	0,005788	0,001619	0,002797	0,001500	0,002290
	1º Quartil	0,030300	0,060720	0,070820	0,047900	0,050788	0,053414	0,052108	0,046015	0,033960	0,051220
	Mediana	0,055960	0,127850	0,083510	0,089450	0,082382	0,073911	0,078643	0,071062	0,071090	0,080220
	Média	0,069680	0,112020	0,087060	0,085420	0,078248	0,094350	0,083021	0,076676	0,077620	0,079160
	3º Quartil	0,095180	0,142230	0,110200	0,104480	0,102079	0,144358	0,120469	0,101106	0,116480	0,115640
	Máximo	0,235050	0,197390	0,128970	0,250610	0,191649	0,217350	0,226499	0,273241	0,222590	0,168630
	Desvio Padrão	0,052998	0,049911	0,028028	0,053558	0,042275	0,054969	0,048136	0,050698	0,053601	0,050068

Fonte: Elaborado pelo autor.

que se repetiram com mais frequência e o conjunto de índices dos 25% de registros inferior e 25% de registros superiores respetivamente (SANTOS, 2007), objetivando identificar os pontos de maior probabilidade de ocorrer uma falha durante o processo de codificação.

A partir da análise visual dos gráficos apresentados pelas Figuras 20 e 21 é possível perceber que a mediana da carga cognitiva de cada uma das respostas é mais dispersa durante as respostas incorretas (Figura 21) do que nas respostas corretas (Figura 20), estando em um valor médio próximo ao meio de 0,05 e 0,10 neste último caso.

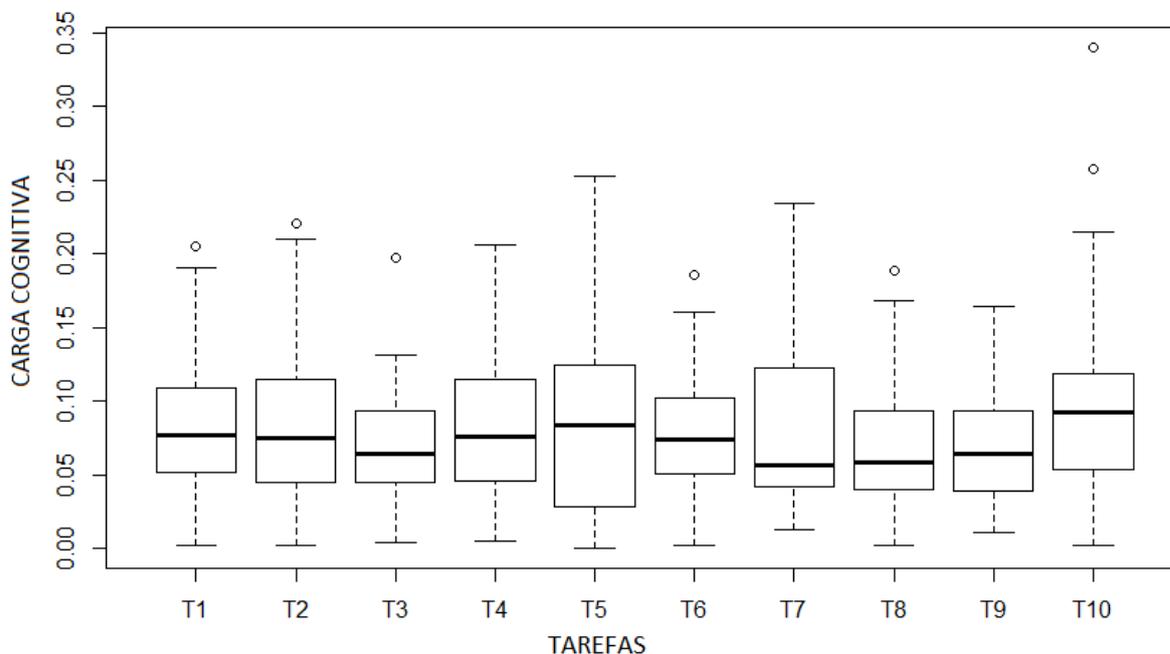


Figura 20 – Boxplot carga cognitiva gerada em respostas corretas.

Fonte: Elaborado pelo autor.

Como deseja-se obter o valor médio das medianas e quartis de cada uma das tarefas executadas, aplicou-se o resultado da Tabela 36 novamente aos métodos da estatística descritiva,

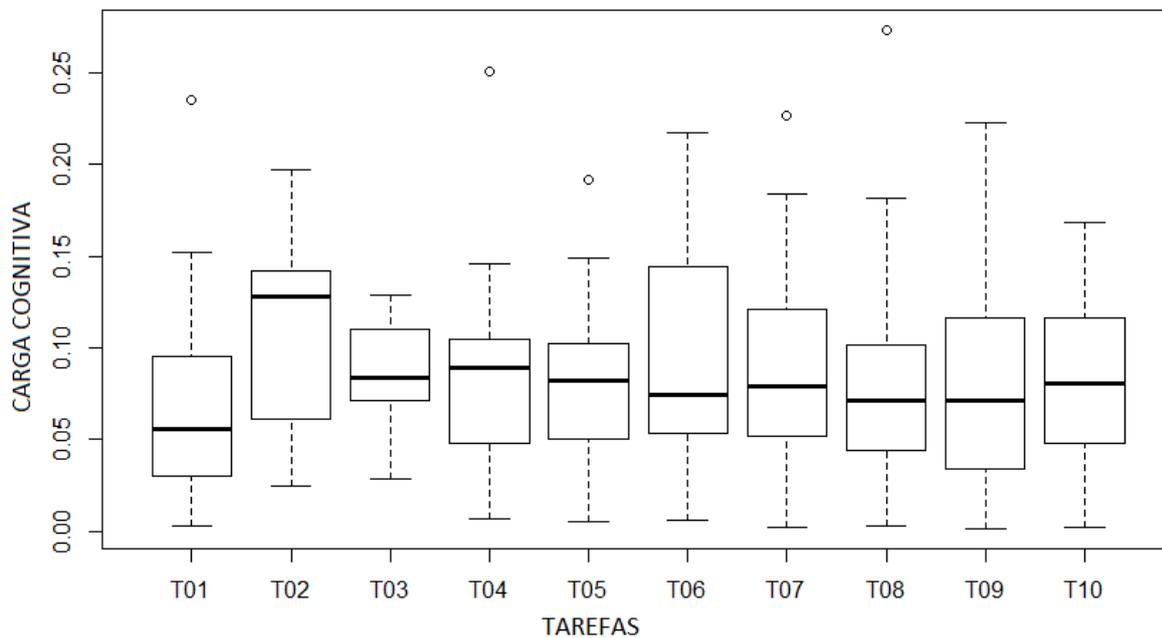


Figura 21 – Boxplot carga cognitiva gerada em respostas incorretas.

Fonte: Elaborado pelo autor.

resultando na Tabela 37, a qual exibe as informações estatísticas da mediana, 1º quartil e 3º quartil de todas as tarefas consolidados em um único registro. Ao utilizar-se a mediana do primeiro e terceiro quartil provenientes da carga cognitiva das respostas positiva tem-se os valores 0,04499 e 0,11129 respectivamente, os quais permitem que duas retas indicando estes valores sejam traçadas sobre o gráfico da carga conitiva gerada por tarefa respondida incorretamente, resultando na Imagem 22.

Tabela 37 – Estatística descritiva das medianas e quartis.

	Acerto			Erro		
	1º Quartil	Mediana	3º Quartil	1º Quartil	Mediana	3º Quartil
Mínimo	0,026930	0,054810	0,093030	0,030300	0,055960	0,095180
1º Quartil	0,041170	0,061640	0,095900	0,046490	0,071800	0,10268
Mediana	0,044990	0,074160	0,111290	0,051000	0,079430	0,112920
Média	0,044380	0,071320	0,108260	0,049720	0,081410	0,115220
3º Quartil	0,049210	0,076400	0,117530	0,053090	0,083230	0,119470
Máximo	0,053600	0,092380	0,123440	0,070820	0,127850	0,144360

Fonte: Elaborado pelo autor.

Para analisar as hipóteses H3-0 e H3-1 são considerados os dados da Tabela 37 sobre os dados da Tabela 24. A partir destes dados e com o uso da mediana do primeiro e terceiro quartis calculados pelas tarefas respondidas corretamente com os valores de 0,04499 e 0,111290 respectivamente e exibidos na Figura 22 identificou-se que do total de 331 respostas incorretas, 185 estavam fora deste intervalo, equivalendo a mais de 55% das respostas. Com isso, pode-se

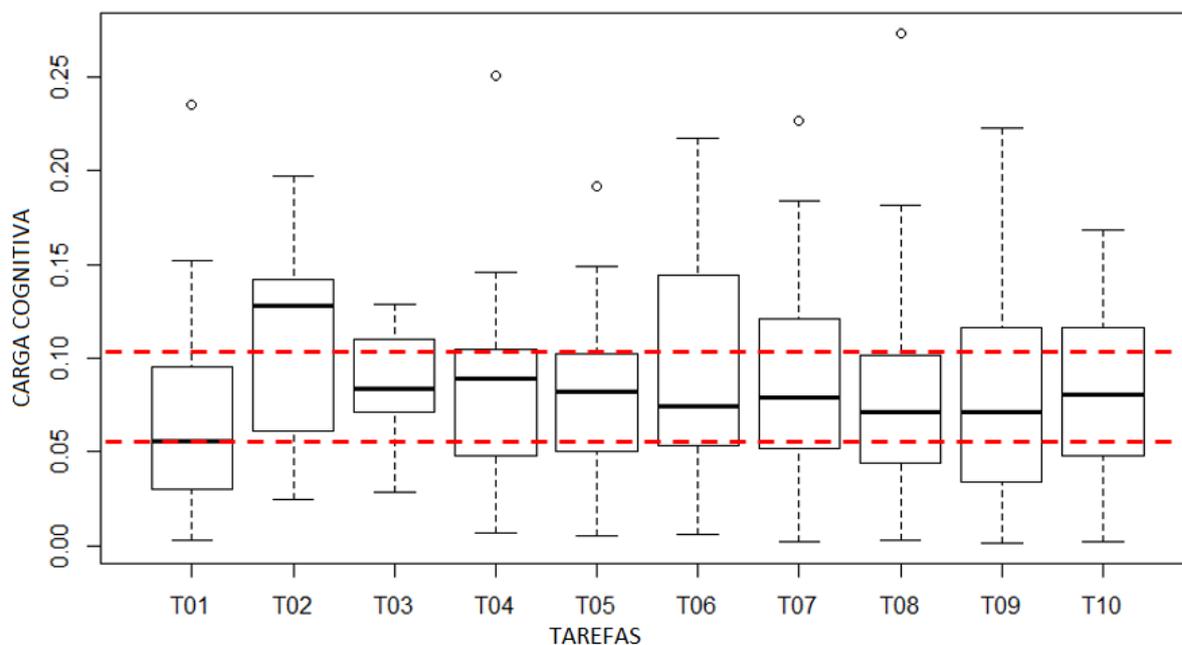


Figura 22 – Boxplot carga cognitiva gerada em respostas incorretas com limites de acerto.

Fonte: Elaborado pelo autor.

concluir que existem evidências que permitem definir um limite de carga cognitiva que após este limite ser ultrapassado a probabilidade de falha durante a execução da tarefa seja maior. A partir desta identificação, em conjunto do conceito de carga cognitiva (SWELLER, 1994) pode-se conjecturar que as tarefas respondidas incorretamente e que estiveram acima do limite identificado de carga cognitiva podem estar associadas ao alto esforço para realizar uma tarefa sem haver o conhecimento necessário para tal, enquanto que nos casos em que a atividade incorreta ficou no limite inferior de carga cognitiva, pode-se supor que o desenvolvedor executou a atividade sem procurar entender o que deveria ser realizado na tarefa.

4.10 Discussão Adicional

A partir dos resultados obtidos através da análise das hipóteses construídas na Seção 4.2, é possível traçar um relacionamento entre as hipóteses apresentadas. Na hipótese 1, buscou-se comprovar que a carga cognitiva pode ser uma ferramenta de análise para identificar as diferenças de expertise entre os desenvolvedores, possibilitando identificar que a carga cognitiva média de um desenvolvedor é menor quando ele possui mais experiência. Por sua vez, investigando a hipótese 2 foi possível identificar as expertises de um desenvolvedor a partir da carga cognitiva que ele gera enquanto executa as tarefas. Esta constatação foi identificada através da análise de desenvolvedores com diferentes níveis de experiência executando uma mesma tarefa, bem como através da análise da carga cognitiva de um mesmo desenvolvedor em intervalos de tempo distintos, evidenciando também a aquisição de expertises com o passar do tempo. Já a hipótese 3 permitiu a identificação de limites de carga cognitiva, que quando ultrapassados aumentam a

probabilidade de execução de uma tarefa com falhas.

A conclusão destas hipóteses permite considerar que a abordagem 4Experts apresentada no capítulo 5 possa apresentar resultados promissores, pois ela baseia-se na recomendação de tarefas de programação adequadas ao nível de expertise de cada indivíduo, gradualmente aumentando a complexidade das tarefas recomendadas conforme o indivíduo adquire conhecimento, estando esta premissa em conformidade a conclusão das hipóteses avaliadas neste experimento.

Além de ser possível concluir a viabilidade da implementação da abordagem 4Experts, este experimento permitiu identificar alguns direcionamentos futuros, para próximos trabalhos a serem realizados sobre este assunto, que são detalhados a seguir:

Replicar o experimento com outros equipamentos EEG: O processo de aquisição de dados a partir de EEG é um processo que requer muitos cuidados, sendo sempre executado em ambientes controlados, conforme detalhado na Seção 4.7. Com técnicas como a proposta nesta dissertação, o uso de dados neurológicos está passando a ganhar visibilidade em aplicações práticas. Nestas aplicações práticas nem sempre o ambiente poderá ser controlado e nestes casos torna-se necessário o emprego de equipamentos de EEG que atualmente são utilizados comercialmente em atividades de lazer e não apenas no meio acadêmico, como é o caso do Neurosky³ e Muse⁴. Em trabalhos futuros, recomenda-se replicar este experimento com estes equipamentos com o propósito de verificar se a acuracidade dos dados apresentados é a mesma dos equipamentos utilizados em ambientes controlados (laboratório).

Diferenças entre a expertise percebida e a expertise medida: Nas hipóteses apresentadas neste capítulo a expertise dos desenvolvedores foi mensurada através do número de tarefas desenvolvidas corretamente em relação as tarefas incorretas. Entretanto, cada participante do experimento respondeu a um questionário, a partir do qual ele identificou as suas expertises pelo seu autorrelato. Entretanto, nem sempre as expertises percebidas pelo indivíduo representam a realidade. Esta situação ocorre porque as questões propostas pelo questionário de avaliação verificam o participante de forma numérica em relação ao histórico de atividades desenvolvidas pelo mesmo durante o tempo, porém, não consideram o grau de efetividade destas atividades. Um trabalho futuro que pode ser estendido a partir deste experimento é tentar identificar um relacionamento entre as expertises autoatribuídas e expertises reais, identificadas a partir da carga cognitiva.

Aprimorar a análise de identificação das expertises de participantes com maior experiência: A hipótese 2 apresentou evidências da diferença de carga cognitiva entre participantes com maior e menor expertise, relacionando esta diferença com o ganho de conhecimento do participante conforme o tempo de experiência. Estas evidências são claras em participantes de menor expertise, porém, não se mostraram tão evidentes em participantes que já possuem expertises maiores com programação. Durante a revisão bibliográfica e análise de estudos realizada para a construção deste trabalho não foram identificados trabalhos que apresentassem

³Ver site: <http://neurosky.com/>.

⁴Ver site: <https://choosemuse.com/>.

técnicas diferenciadas para identificar a evolução da expertise em indivíduos que já apresentam expertises técnicas. Um estudo que tenha por objetivo identificar com maior grau de evidências o ganho de expertise em indivíduos que já possuem experiência poderá complementar técnicas que possam assistir aos desenvolvedores durante a execução das suas atividades promovendo melhores resultados.

4.11 Ameaças a Validade

Em estudos que envolvem a realização de experimentos, as ameaças a validade são riscos potenciais aos resultados obtidos. Este capítulo tem por objetivo relacionar riscos identificados durante o processo de execução do experimento identificando as possíveis ameaças relacionadas a cada risco e quais ações tomadas para mitigar os riscos identificados. Foram identificadas três ameaças principais, descritas conforme segue:

Processo de coleta de dados EEG: O processo de coleta de dados EEG representa uma ameaça significativa para o experimento. O EEG coleta ondas elétricas em frequências e voltagens muito baixas, sujeitas as interferências do ambiente e de outros dispositivos em possam emitir ondas em frequências similares. Em ambientes clínicos, conforme apresentado por Mecarelli (MECARELLI, 2019) no capítulo 7 (*EEG Laboratory: Patient Care and the Role of the EEG Technician*), diversos cuidados devem ser observados durante a execução do procedimento, como a posição dos eletrodos e ambiente livre de interferências e ruídos. Como o experimento não foi realizado em um ambiente clínico reconhecidamente controlado, foram empregados cuidados para minimizar as ameaças durante a coleta de dados. Para isso, foi empregado um ambiente livre de ruídos, reatores de lâmpadas e eventuais equipamentos eletrônicos foram mantidos desligados no local onde o experimento foi aplicado. Além disso, o participante do experimento utilizou uma pulseira antiestática, conforme apresentado na Figura 23, para evitar que alguma eventual energia estática pudesse interferir na coleta de dados. Durante a execução do experimento, também foi realizada a observação e marcação de sensores que eventualmente falhassem durante o processo.

Processamento dos dados EEG: Os resultados identificados na Seção 4.9 deste capítulo utilizam a carga cognitiva despendida por cada um dos participantes do experimento nas questões aplicadas. A obtenção da carga cognitiva é feita através da aplicação de técnicas de processamento e análise dos dados do EEG coletados. A aplicação da técnica incorreta neste processamento impacta diretamente no resultado do experimento, representando uma ameaça significativa a validade das hipóteses apresentadas. Para mitigar esta ameaça, foram identificados artigos associados a avaliação de carga cognitiva alinhados ao objetivo de identificar compreensão de código, com o propósito de identificar uma técnica de processamento adequada a forma como os dados foram coletados (BABADI; BROWN, 2014) e também pelo tipo de equipamento EEG utilizado (DAS ET AL., 2014).

Questões propostas: As questões aplicadas ao experimento representam uma ameaça a



Figura 23 – Pulseira antiestática.

Fonte: Elaborado pelo autor.

validade do experimento a medida que o seu escopo não reflete o objetivo da questão perante o desenvolvimento. Para o experimento, as questões devem refletir um gradual nível de dificuldade conforme elas são apresentadas aos participantes. Caso as questões propostas não representem um acréscimo ao nível de dificuldade, elas poderão não provocar um aumento da carga cognitiva nos participantes. Esta ameaça foi mitigada através do conceito associado a memória de trabalho dos participantes (SWELLER, 1988) em relação ao número de iterações existentes em cada questão. Neste contexto, questões com menos interações são consideradas questões simples por exigirem uma quantidade menor de memória de trabalho, em relação a questões com mais interações, as quais são consideradas questões complexas por exigirem maiores quantidades de memória para armazenar as informações do algoritmo. Esta evolução na dificuldade de cada uma das questões é apresentada na Tabela 23 localizada na Seção 4.5.

5 ABORDAGEM 4EXPERTS

Com base na revisão e análise da literatura sobre o uso dados biométricos e sistemas de recomendação na Engenharia de Software apresentados no Capítulo 3 e para atender aos objetivos deste trabalho apresentados na Seção 1.2, este capítulo apresenta a abordagem 4Experts a qual realiza a recomendação em tempo real de tarefas de desenvolvimento de software para o desenvolvedor. Para isso, o 4Experts coleta e classifica dados biométricos, utilizando um equipamento de eletroencefalograma (EEG) e técnicas de aprendizagem de máquina respectivamente, para identificar cada tarefa a ser recomendada.

Este capítulo é organizado da seguinte forma. A Seção 5.1 apresenta a visão geral da abordagem proposta, detalhando cada um dos passos. A Seção 5.2 detalha a arquitetura empregada para o desenvolvimento da abordagem, enquanto na Seção 5.3 são apresentados os aspectos da implementação para a execução da abordagem proposta.

5.1 Visão Geral

Os sistemas de recomendação de tarefas para a Engenharia de Software são amplamente utilizados para recomendar artefatos durante o processo de codificação, fornecendo suporte à tomada de decisão enquanto os desenvolvedores interagem com uma grande quantidade de informações sob determinados contextos (GASPARIC; JANES, 2016). A maioria dos sistemas de recomendação propostos baseiam-se em algoritmos para processar dados de software obtidos através da exploração de repositórios de código como os propostos por Junior e Matter (JÚNIOR ET AL., 2018; MATTER; KUHN; NIERSTRASZ, 2009). Por isso, os sistemas de recomendação de tarefas deste tipo são considerados sistemas centrados nas tarefas, sendo o oposto dos sistemas de recomendação utilizados em e-commerce por exemplo, os quais são centrados no usuário, proporcionando maior conhecimento sobre a tarefa a ser feita do que o desenvolvedor que irá executá-la (ROBILLARD; WALKER; ZIMMERMANN, 2010), revelando uma restrição de personalização dos atuais sistemas de recomendação (JÚNIOR ET AL., 2018; MATTER; KUHN; NIERSTRASZ, 2009; ZHANG; LEE, 2012; WU ET AL., 2011). Esta limitação, quando aplicada a recomendações de tarefas de programação pode contribuir para um maior esforço e probabilidade de falhas durante a execução da tarefa.

Por outro lado, a análise de informações biométricas dos desenvolvedores tem recebido atenção dos pesquisadores conforme pode ser observado na Figura 6. Este aumento no interesse em pesquisar a relação existente entre dados biométricos e desenvolvedores possibilitou trabalhos como o trabalho de Fritz (FRITZ ET AL., 2014), que avalia a relação entre carga cognitiva e a compreensão de código ou o trabalho de Lee e CRK (LEE ET AL., 2017; CRK; KLUTHE; STEFIK, 2015) que apresentam técnicas de identificação de expertises dos desenvolvedores através da análise de dados biométricos aplicados a uma técnica de aprendizagem de máquina.

Desta forma o 4Experts surge como uma abordagem centrada no usuário, procurando preencher desta forma uma lacuna dos sistemas de recomendação, por fazer uso de dados biométricos do desenvolvedor processados por uma técnica de aprendizado de máquina, para recomendar tarefas de programação em tempo real. A Figura 24 apresenta uma visão geral da abordagem proposta mostrando as etapas do processo de recomendação e a interação entre os diferentes envolvidos neste processo.

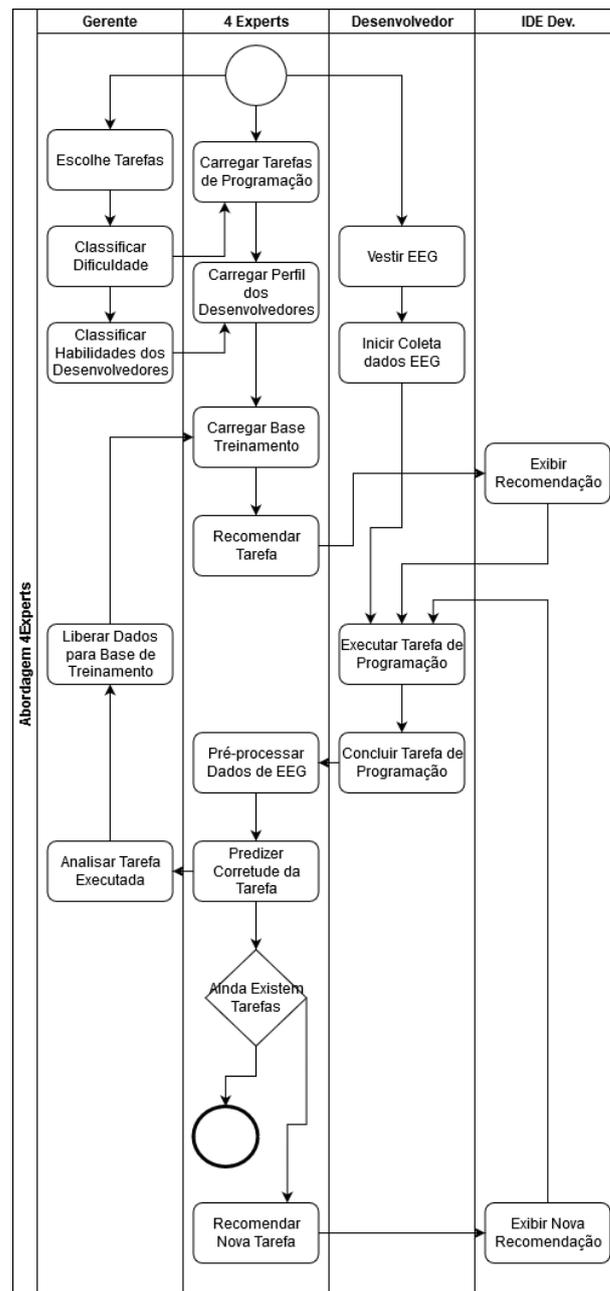


Figura 24 – Visão geral do 4Experts.

Fonte: Elaborado pelo autor.

As etapas descritas pela Figura 24 visam permitir que as tarefas de desenvolvimento e os desenvolvedores possam ser preparados visando a interação destes com a abordagem proposta.

Além disso, a própria abordagem necessita de preparação para que a técnica de aprendizagem de máquina possa ser utilizada adequadamente pelo 4Experts. Por fim, é possível ter a abordagem funcionando em conjunto com os desenvolvedores, fornecendo a recomendação da próxima tarefa que deve ser executada de forma integrada ao ambiente de desenvolvimento e retroalimentado a base de dados empregada pela técnica de aprendizado de máquina com base no feedback fornecido pelo gestor de equipe. Esta definição do processo é apresentada a seguir com base nos passos apresentados na Figura 24.

- **Escolher Tarefas.** O gerente da equipe identifica no sistema de gestão de projetos de software quais as tarefas de programação que estão disponíveis para serem performadas pelos desenvolvedores.
- **Classificar Dificuldade.** Neste passo o gerente da equipe classifica as tarefas escolhidas de acordo com o seu nível de dificuldade. Os níveis de dificuldade poderão ser definidos pelo gestor de acordo com as suas expertises e conhecimentos do projeto ao qual as tarefas se referem. Normalmente estas tarefas podem utilizar escala *likert* para definir a dificuldade, variando de 1 para tarefas fáceis até 5 para as tarefas difíceis, ou apenas classificar as tarefas com o grau de dificuldade fácil, médio ou difícil.
- **Classificar Expertises dos Desenvolvedores.** Da mesma forma que as tarefas são classificadas, os desenvolvedores também são classificados de acordo com as suas expertises pelo gestor da equipe. O gestor realiza esta classificação com base nas suas observações, percepções, acompanhamento e capacidade técnica de cada desenvolvedor.
- **Relacionar Dificuldade com Expertises.** Após a dificuldade das tarefas e as expertises dos desenvolvedores serem classificadas, o gestor precisa realizar um relacionamento entre o nível de dificuldade e o nível de expertise adequado que um desenvolvedor deve possuir para conseguir performar com acuracidade e menor esforço as tarefas. Este relacionamento é um requisito do mecanismo do 4Experts para recomendar as tarefas aos desenvolvedores com um nível de dificuldade proporcional as expertises para resolver a tarefa.
- **Carregar Base de Treinamento.** Na inicialização do 4Experts, uma das primeiras operações realizadas pelo sistema é o carregamento da base de dados de treinamento pela técnica de aprendizagem de máquina utilizada. O protótipo do 4Experts utiliza como base de dados de treinamento os dados biométricos coletados por EEG para o experimento do Capítulo 4 e preparados através do procedimento de análise apresentado na Seção 4.8.
- **Carregar Tarefas de Programação.** As tarefas de programação também são carregadas para o 4Experts a partir de uma integração com o sistema de gestão de projetos empregado pela equipe de desenvolvimento. Nesta integração o 4Experts importa os principais

dados da tarefa, com ênfase nas informações descritivas da tarefa que permitam ao desenvolvedor a partir da sua ideia de desenvolvimento saber detalhes do que precisa executar e o identificador de dificuldade da tarefa.

- **Carregar Perfil dos Desenvolvedores.** O 4Experts também importa da ferramenta de gestão de projetos os dados dos membros da equipe que atuam nas tarefas de programação com os seus dados de identificação e o nível de expertise de cada um.
- **Vestir EEG.** O desenvolvedor que participa do 4Experts precisa ter os seus dados biométricos produzidos pelo cérebro coletados através de um equipamento de eletroencefalograma (EEG). Este passo representa o momento do dia de trabalho em que o desenvolvedor deverá iniciar com o uso do equipamento citado. Devido a considerável quantidade de interferências que este equipamento sofre sob determinadas condições, é necessário que o desenvolvedor tome alguns cuidados com o uso do equipamento, conforme detalhado na Seção 4.11, na seção **Processo de coleta de dados do EEG**.
- **Iniciar Coleta de Dados.** Após o equipamento de EEG ser corretamente vestido no desenvolvedor, é iniciada a coleta dos dados biométricos brutos produzidos pelo cérebro. Os dados coletados passam a ser mantidos em memória para serem posteriormente utilizados pelo 4Experts.
- **Recomendar Tarefa/Nova Tarefa.** Quando o 4Experts é iniciado, a primeira tarefa a ser recomendada baseia-se apenas no relacionamento criado anteriormente entre a dificuldade das tarefas e o nível de expertise dos desenvolvedores devido ao fato dele ainda não possuir dados de execução de tarefas do desenvolvedor que permitam a ele recomendar uma tarefa aplicando a técnica de aprendizado de máquina. Esta limitação é considerada comum nos sistemas de recomendação, conforme apresentado no referencial teórico disponibilizado na Seção 2.7. No caso da recomendação de uma nova tarefa, após o desenvolvedor já ter performado uma tarefa anterior, a recomendação da tarefa será baseada na predição da corretude da tarefa anteriormente executada. A predição é importante neste momento, por considerar-se improvável que uma verificação da acuracidade da tarefa possa ser feita quando o desenvolvedor concluir a mesma. Desta forma, caso a predição indique que a resposta da tarefa seja incorreta, a tarefa a ser recomendada apresentará um nível de dificuldade inferior a dificuldade da tarefa que atualmente foi performada.
- **Exibir Recomendação/Nova Recomendação.** O resultado da recomendação da tarefa é exibido na ferramenta de desenvolvimento utilizada pelo desenvolvedor através do uso de um plugin de integração entre o ambiente de desenvolvimento e o 4Experts. Na ideia, a tarefa recomendada é exibida com o seu código de identificação utilizado pela ferramenta de gestão de projetos de desenvolvimento e por uma descrição breve do que necessita ser feito. Caso sejam necessários maiores detalhes, estes podem ser obtidos diretamente pela ferramenta de gestão de projetos de software utilizada pelos desenvolvedores.

- **Executar Tarefa de Programação.** Neste passo o desenvolvedor realiza o seu trabalho, codificando a tarefa recomendada. Enquanto ele codifica, os seus dados biométricos são coletados para serem analisados ao final da execução da tarefa.
- **Pré-processar Dados de EEG.** O 4Experts precisa executar o pré-processamento dos dados brutos de EEG coletados para remover ruídos coletados e converter estes dados em frequências de bandas normalizadas conforme apresentado na Tabela 38. A partir desta conversão, o 4Experts utiliza as ondas Beta, normalmente associadas ao desempenho de atividades cognitivas, convertendo as mesmas em espectro de força através da técnica Multitaper detalhada na Seção 4.8. O resultado deste processamento é submetido a técnica de predição apresentada no próximo item.

Tabela 38 – Bandas de frequência

Banda	Frequência (Hz)	Descrição
Delta (δ)	1-4	Ondas cerebrais associadas à inconsciência e ao sono profundo.
Theta (ϑ)	4-8	Estado de sonolência e consciência reduzida. Representa o estado de relaxamento físico e mental, embora com consciência do que está acontecendo ao redor do indivíduo. Uma ponte entre o pensamento consciente e a mente subconsciente.
Alpha (α)	8-12	Ondas cerebrais de baixa amplitude e alta frequência. Observadas quando o indivíduo está acordado. Relacionadas ao pensamento consciente e pensamento lógico. A frequência e amplitude destas ondas está relacionada ao foco e execução de tarefas escolares ou demais tarefas que necessitem de pensamento lógico.
Beta (β)	13-30	Relacionadas a tarefas de processamento e funcionamento cognitivo. Importante para aprendizagem, memória e processamento de informação
Gamma (γ)	30-100	

Fonte: Adaptado de (BUZSÁKI, 2009).

- **Predizer Corretude da Tarefa.** A partir dos dados pré-processados do EEG coletados do desenvolvedor enquanto este performava a tarefa atual é realizada a predição de acerto ou falha de execução desta tarefa. Este passo faz uso da técnica de aprendizagem de máquina MultiLayer Perceptron 2.6 que utiliza como base de treinamento um conjunto de dados de EEG que emprega como rótulo o acerto ou erro durante a execução de um conjunto de tarefas de programação que foram executadas e monitoradas anteriormente conforme apresenta o Capítulo 4.
- **Analisar Tarefa Executada.** Caso for desejado adicionar os dados de EEG da tarefa

executada pelo desenvolvedor, o gestor da equipe deverá analisar a tarefa em questão avaliando a acuracidade da tarefa executada e fornecendo este indicador ao 4Experts. Desta forma, é possível relacionar a acuracidade da tarefa com os dados do EEG coletados possibilitando que esta informação possa ser utilizada para retroalimentar a base de treinamento da abordagem.

- **Liberar Dados para Base de Treinamento.** Se a análise da tarefa for realizada, esta etapa libera os dados de EEG do desenvolvedor para a base de conhecimento e posteriormente estes dados passarão a ser utilizados durante o processo de predição, aprimorando os resultados.

5.2 Arquitetura

Para que possa ocorrer o processo descrito na Figura 24, o 4Experts faz uso de componentes. Estes componentes permitem a sua utilização de forma isolada, fornecendo interfaces de comunicação para os demais componentes da abordagem, permitindo o reuso dos componentes e facilitando o processo de manutenção do sistema gerado. Através da interface de comunicação fornecida por cada um dos componentes, estes são integrados entre si conforme apresentado na Figura 25. Cada um dos componentes elaborados possui um conjunto de atividades encapsuladas que quando utilizadas promovem o funcionamento do 4Experts conforme apresentado na Seção 5.1

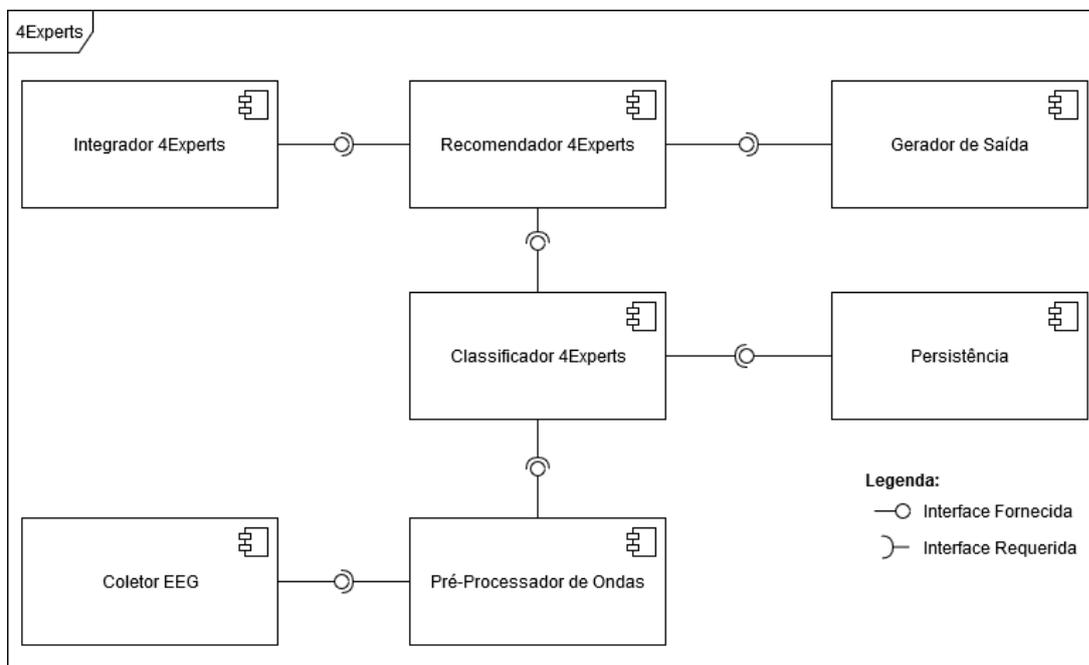


Figura 25 – Arquitetura da ferramenta para a abordagem 4Experts.

Fonte: Elaborado pelo autor.

A Figura 25 apresenta os dez componentes arquiteturais necessários para a abordagem 4Ex-

perts. A seguir estes componentes serão detalhados:

Coletor EEG. O objetivo deste componente é coletar os dados brutos associados ao equipamento de Eletroencefalograma enquanto o desenvolvedor executa uma tarefa. Os dados brutos coletados por este componente são provenientes dos 14 sensores disponibilizados pelo equipamento Emotiv Epor+, os quais são gerados com uma taxa de amostragem de 256 sinais por segundo para cada sensor. O componente pode receber dados de duas formas: (1) arquivos de dados de EEG coletados anteriormente em formato EDF ou CSV e (2) dados coletados do próprio sensor EEG através da sua SDK. Independente do formato obtido, estes dados sempre serão convertidos para o formato CSV, fornecendo a saída neste formato. O Algoritmo 1 apresenta o fluxo deste componente.

Algoritmo 1: Algoritmo coletor EEG.

Entrada: Dados EEG fornecidos por um arquivo EFD ou CSV e identificador do desenvolvedor

Saída: Arquivo CSV

```

1 início
2   tipoArquivo = IdentificarTipoArquivo(arquivoEntrada);
3   se tipoArquivo == edf então
4     | Arquivo = ConverterParaCsv(arquivoEntrada);
5   senão
6     | Arquivo = arquivoEntrada;
7   fim
8   enquanto existir linhas em Arquivo faça
9     | se linha possuir identificador de início de questão então
10      |   identificadorQuestao = ObterIdentificadorQuestao(linha);
11      |   arquivoQuestao = AbrirArquivoQuestao(identificadorQuestao,
12      |   identificadorParticipante);
13      |   fim
14      | se linha possuir identificador de fim de questão então
15      |   | FecharArquivoQuestao(arquivoQuestao) ;
16      |   fim
17      | AdicionarLinhaArquivoQuestao(arquivoQuestao, linha);
18 fim

```

Pré-processamento de Ondas. Após as ondas serem separadas em arquivos no formato CSV para a atividade realizada pelo desenvolvedor, que teve os sinais de EEG capturados durante a sua execução, é necessário tratar este sinal com o intuito de remover interferências que afetem o resultado do experimento e transformar o sinal em valores que possam ser utilizados posteriormente pelo classificador MLP. Uma técnica empregada para converter dados de EEG em um formato que possa ser aplicado a um classificador MLP é a técnica de Multitaper (BABADI; BROWN, 2014), a qual converte um conjunto de registros de um sensor EEG em um único valor, o qual é utilizado como no classificador. O Multitaper também é recomendado quando os dados provenientes do EEG possuem ruídos. O Algoritmo 2 apresenta a execução

deste processamento.

Algoritmo 2: Algoritmo conversor de bandas multitaper.

Entrada: Arquivos EEG brutos em formato CSV das questões de um participante

Saída: Arquivos de frequência de bandas processados

```

1 início
2   listaFrequenciaBandas = CarregarFrequenciaBandas();
3   arquivoProcessado = CriarArquivoProcessadoVazio();
4   enquanto existirem arquivos na listaArquivosEntrada faça
5     colunasArquivo = CarregarColunasArquivo(arquivo);
6     enquanto existirem bandas em listaFrequenciaBandas faça
7       vetorMultitaperProcessado = vazio;
8       enquanto existirem colunas em colunasArquivo faça
9         se coluna == Sensor EEG então
10          linhasColuna = CarregarTodasLinhasColuna(arquivoEntrada[coluna]);
11          energiaColuna = ProcessarMultitaper(linhasColuna, banda);
12          vetorMultitaperProcessado[coluna] = energiaColuna;
13        senão
14          vetorMultitaperProcessado[coluna] =
15            PrimeiraLinha(arquivoEntrada[coluna]);
16        fim
17      fim
18      AdicionarConteudoArquivo(arquivoProcessado, vetorMultitaperProcessado);
19    fim
20  FecharArquivo(arquivoProcessado);
21 fim

```

Classificador 4Experts. Este componente é a técnica de aprendizagem de máquina empregada no sistema. A técnica de Multilayer Perceptron (MLP) foi empregada por apresentar bons resultados quando aplicada a dados de EEG processados (GARDNER; DORLING, 1998) conforme apresentado na Seção 2.6. Este componente é inicializado quando o 4Experts inicia a sua execução, recebendo a base de treinamento a ser utilizada do componente de persistência. Enquanto os desenvolvedores executam as suas tarefas, este componente recebe do componente de pré-processamento os dados de EEG tratados de cada questão executada pelos desenvolvedores. Estes dados são submetidos ao MLP com o propósito de prever se a tarefa performada pelo desenvolvedor apresenta correteza ou falha em sua execução, fornecendo esta informação como saída do componente. O Algoritmo 3 descreve o processo de forma resumida que é aplicado para identificar a correteza da tarefa a partir da técnica de aprendizagem de máquina Multilayer Perceptron.

Persistência. Este componente é responsável por persistir todos os dados necessários para o funcionamento do 4Experts, realizando a interface entre o banco de dados contendo os dados de EEG utilizados para treinamento do componente *Classificador 4Experts*. Além disso, este componente também permite a ampliação da base de treinamento através da inclusão dos

Algoritmo 3: Algoritmo mlp.

Fonte: Elaborado pelo autor.

Entrada: Base de treinamento, arquivo de bandas processadas do Desenvolvedor

Saída: Indicador se a tarefa foi desenvolvida com acerto ou erro

```

1 início
2   camadasOcultas = Numero de Camadas Ocultas;
3   funcaoAtivacao = "tanh";
4   solver = "lbg";
5   taxaAprendizadoInicial = 0.001;
6   interacoesMaximas = 1000;
7   momentum = 0.9;
8   labels = CarregarLabels(baseTreinamento);
9   colunasValores = CarregarValores(baseTreinamento);
10  classificador = IniciarClassificadorMLP(camadasOcultas, funcaoAtivacao, solver,
11    taxaAprendizadoInicial, interacoesMaximas, momentum);
12  classificador.treinar(colunasValores, labels);
13  indicadorAcerto = classificador.predizerAcerto(arquivoBanda);
14 fim
  
```

próprios dados de EEG coletados dos desenvolvedores e previamente analisados pelo Gestor. A Tabela 39 apresenta a estrutura padrão de armazenamento dos dados biométricos coletados pelo EEG. Nesta tabela são armazenados todos os registros necessários para manter os dados do EEG persistidos para cada um dos participantes. Estes dados são separados entre dados de identificação, dados de valores, responsáveis por manter as informações biométricas e o rótulo, utilizada para a predição a partir da técnica de Aprendizado de Máquina utilizada.

Tabela 39 – Estrutura de persistência dos dados biométricos.

Identificador Campo	Tipo Informação	Tipo Dado
Momento	Identificador	Inteiro
Participante	Identificador	Inteiro
Tarefa	Identificador	Inteiro
AF3	Valor	Float
F7	Valor	Float
F3	Valor	Float
FC5	Valor	Float
T7	Valor	Float
P7	Valor	Float
O1	Valor	Float
O2	Valor	Float
P8	Valor	Float
T8	Valor	Float
F4	Valor	Float
F8	Valor	Float
AF4	Valor	Float
FC6	Valor	Float
Acerto	Rótulo	Boolean

Recomendador 4Experts. Este componente tem a responsabilidade de sugerir qual a próxima tarefa o desenvolvedor deve fazer com base na predição do acerto ou erro da tarefa atual gerada pelo componente *Classificador 4Experts*. Este componente recebe como entradas a predição de acerto da atividade atual, a identificação da tarefa atual, a lista de atividades que ainda não foram desenvolvidas com o seu indicador de dificuldade pelo componente *Integrador 4Experts*. O processamento deste componente consiste em identificar o nível de dificuldade da atividade atual e o indicador de acerto desta tarefa. Com estas informações, caso o indicador da tarefa aponte para uma falha na sua execução, o priorizador irá localizar uma tarefa com um nível de dificuldade inferior a tarefa atual para recomendar ao desenvolvedor. Caso contrário, uma tarefa de nível igual ou superior de dificuldade será recomendada ao desenvolvedor. O resultado deste módulo é a identificação da questão a ser recomendada ao desenvolvedor como próxima tarefa a ser executada. O Algoritmo 4 representa o fluxo que este componente executa para priorizar as atividades.

Algoritmo 4: Algoritmo priorizador de tarefas.

Entrada: Tarefa atual, Indicador de acerto da tarefa atual, Lista de tarefas disponíveis para execução

Saída: Próxima tarefa a ser executada pelo Desenvolvedor

```

1 início
2   dificuldadeTarefaAtual = LocalizarDificuldade(tarefaAtual);
3   se indicadorAcerto == tarefaDesenvolvidaIncorretamente então
4     listaQuestoesDisponiveis = LocalizarQuestoesNivelInferior(listaQuestoes,
5     dificuldadeTarefaAtual);
6   senão
7     listaQuestoesDisponiveis = LocalizarQuestoesNivelIgualSuperior(listaQuestoes,
8     dificuldadeTarefaAtual);
9   fim
10  proximaQuestao = ObterPrimeiraQuestaoLista(listaQuestoesDisponiveis);
11 fim

```

Integrador 4Experts. Este componente tem a responsabilidade de fazer toda a comunicação do 4Experts com o software de gestão de projetos utilizado pela equipe, fornecendo uma interface de serviço que realiza a leitura das tarefas que estão disponíveis para serem executadas pelos desenvolvedores a partir de um repositório de tarefas, frequentemente obtido a partir do sistema de gerenciamento de projetos utilizado pela equipe de desenvolvimento. Este módulo importa apenas as informações necessárias para a execução do 4Experts, focando nas seguintes informações: (1) identificador da atividade, (2) dificuldade associada à atividade, (3) perfil de responsável pela execução da atividade (isto é, desenvolvedor, tester, analista) e (4) recurso que está sendo modificado. Estas tarefas são utilizadas pelo componente *Recomendador 4Experts* e retornadas com a informação do responsável pela sua execução.

Este componente também importa a classificação de níveis de dificuldade utilizados nas tarefas. Como os níveis de dificuldade são subjetivos sob o ponto de vista da equipe de desen-

volvimento, cada equipe pode definir os seus próprios níveis de dificuldade das tarefas. Após definir estes níveis, eles serão associados aos diferentes níveis de expertise dos desenvolvedores que também é obtido neste momento permitindo o relacionamento entre os níveis de expertise dos desenvolvedores e os níveis de dificuldades das tarefas, que é necessário para o funcionamento do componente *Recomendador 4Experts*.

Gerador de Saída. Após o recomendador ser executado, este componente tem a tarefa de converter o resultado em um formato de dados padrão com o objetivo destes dados poderem ser utilizados por qualquer ferramenta de desenvolvimento. O formato escolhido para exibir o resultado é o formato Json, famoso por não apresentar um grande overhead de informações extras ao conteúdo da saída. Estes dados serão disponibilizados neste formato através de um serviço web.

5.3 Aspectos da Implementação

O protótipo do 4Experts à ser desenvolvido é formado por um plugin a ser executado em conjunto a ferramenta de desenvolvimento Visual Studio e uma aplicação web responsável por consumir e prover serviços necessários para a recomendação das tarefas que devem ser realizadas pelos desenvolvedores conforme apresentado na Figura 26.

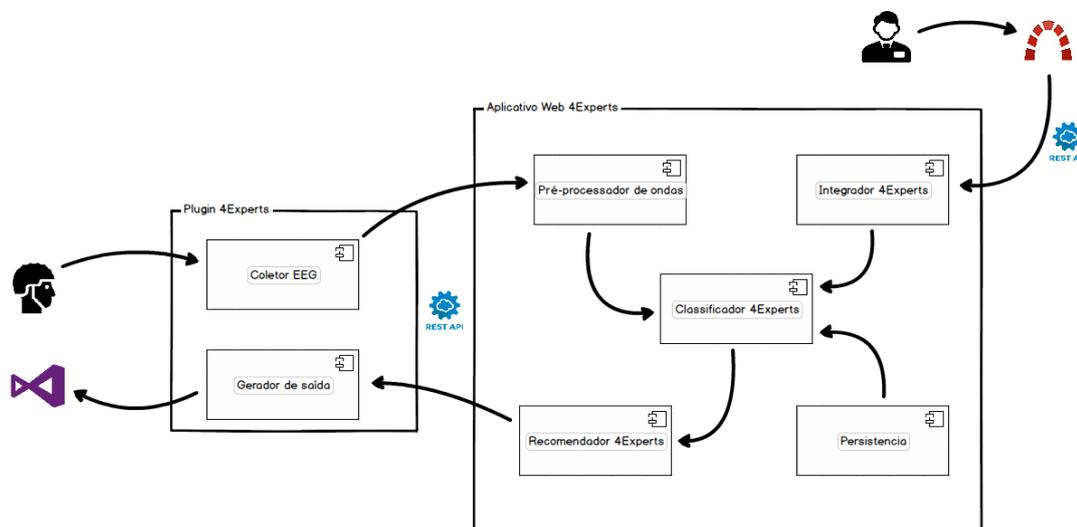


Figura 26 – Aspectos técnicos do 4Experts.

Fonte: Elaborado pelo autor.

5.3.1 Plugin 4Experts

O plugin 4Experts é o componente responsável por fazer a interação da abordagem com o desenvolvedor. Este plugin é integrado a ferramenta de desenvolvimento Visual Studio, realizando a coleta dos dados biométricos através do equipamento de eletroencefalograma utilizado

pele desenvolvedor e a recomendação das tarefas de forma integrada ao ambiente de desenvolvimento.

Para ser possível iniciar o uso do 4Experts é necessário ao desenvolvedor se identificar ao plugin, fornecendo ao mesmo as suas credenciais de identificação utilizadas junto a ferramenta de gestão de projetos empregada pela equipe. A partir desta identificação, é possível dar início a coleta dos dados de eletroencefalograma, necessários para o sistema de recomendação. O equipamento utilizado para a coleta destes dados é o *Emotiv Epoc+*¹ em conjunto com o seu kit de desenvolvimento (SDK) *EmotivPro*², que permite ao plugin obter os dados brutos da atividade cerebral a uma frequência de 256 amostras por segundo. Estes dados, após serem coletados pelo plugin, são enviados para a aplicação web através do consumo de um serviço fornecido pela própria aplicação.

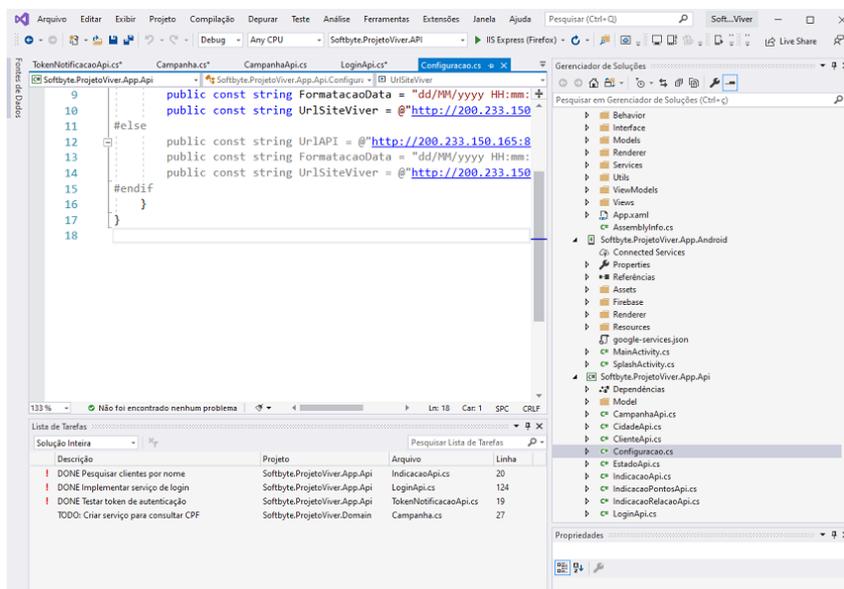


Figura 27 – Plugin para visualizar tarefas no visual studio.

Fonte: Elaborado pelo autor.

Além de coletar os dados do eletroencefalograma, este plugin também apresenta a saída das recomendações fornecidas pelo 4Experts para o desenvolvedor através de uma janela de atividades a ser visualizada no Visual Studio conforme detalhado na Figura 27. Nesta janela o desenvolvedor terá acesso ao identificador e detalhes da atividade atual que ele está desenvolvendo e poderá indicar o seu término. No momento em que o seu término for informado, após o 4Experts analisar os dados do eletroencefalograma e efetuar a predição da correteza da tarefa, será exibida a recomendação da próxima tarefa a ser executada.

¹Site: <https://www.emotiv.com/epoc/>

²Site: <https://www.emotiv.com/emotivpro/>

5.3.2 Aplicação Web 4Experts

A aplicação 4Experts é o responsável por realizar todas as integrações com a ferramenta de gestão de projetos de software da equipe e coleta dos dados de eletroencefalograma, processar e executar as técnicas de aprendizado de máquina responsáveis pela recomendação e fornecer as tarefas recomendadas para os desenvolvedores.

As integrações com a ferramenta de gerenciamento de projetos são feitas através do consumo das APIs fornecidas por estas ferramentas com o propósito de prover o 4Experts com dados necessários para que as recomendações possam ser feitas. Neste processo, serão requisitadas as informações de perfil dos desenvolvedores com uma identificação de expertise e as tarefas disponíveis para serem executadas com a identificação da dificuldade. Tanto a dificuldade da tarefa como a expertise do desenvolvedor precisam ser atribuídas pelo gestor da equipe diretamente na ferramenta de gestão de projetos, conforme pode ser visto nas Figuras 28 e 29 obtidas a partir da ferramenta *Redmine* (LANG, 2006) de gestão de projetos de Software.

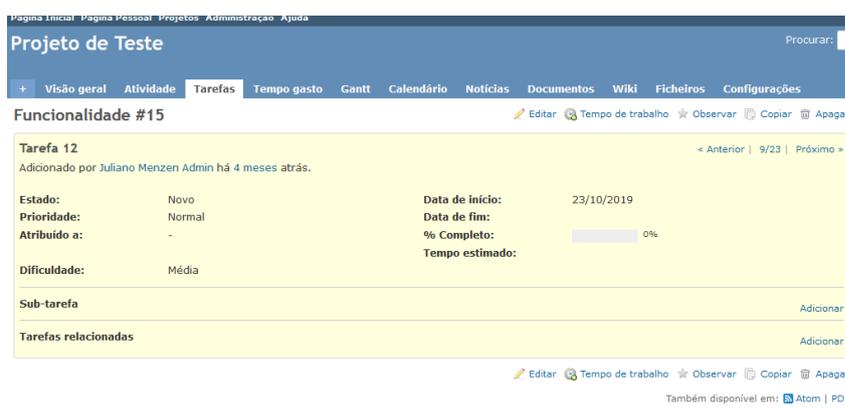


Figura 28 – Tarefas com parâmetros no *Redmine*.

Fonte: Elaborado pelo autor.

Os dados provenientes do eletroencefalograma de cada desenvolvedor são obtidos através do serviço disponibilizado pela aplicação e consumidos pelo Plugin, conforme detalhado na Seção 5.3.1. Conforme estes dados são recebidos, eles são salvos no SGBD *open-source PostgreSQL* até o momento do desenvolvedor concluir a tarefa que está em execução. A partir deste ponto, os dados do período são carregados em memória e inicialmente tratados com a técnica Multitaper (BABADI; BROWN, 2014) a fim de obter-se a energia de espectro (*power spectrum*) do conjunto de sensores do EEG, convertendo os dados de todos os sensores em um único valor. Após este processo, os dados resultados são aplicados a técnica de aprendizado de máquina *Multilayer Perceptron* (GARDNER; DORLING, 1998) já treinada, para gerar a predição de acerto da tarefa executada. Tanto o processamento Multitaper como a técnica de aprendizado de máquina foram desenvolvidos com o uso da linguagem de programação *Python*³ através da biblioteca NME (DEVELOPERS, 2012) e integrados a aplicação web, desenvolvido sob a

³Site: <https://www.python.org/>

Figura 29 – Usuários do Redmine.

Fonte: Elaborado pelo autor

plataforma *.Net Framework*⁴. O resultado da predição é submetido ao classificador 4Experts, que identifica a próxima tarefa a ser executada pelo desenvolvedor, disponibilizando esta informação para o plugin, integrado ao ambiente de desenvolvimento, conforme a Figura 27.

Figura 30 – Relacionamento entre nível de expertise e nível de dificuldade.

Fonte: Elaborado pelo autor.

A aplicação ainda disponibiliza o recurso para o gestor da equipe efetuar o relacionamento entre o nível de dificuldade das tarefas e o grau de expertises dos desenvolvedores, conforme apresentado pelo protótipo de tela da Figura 30.

Além disso, a aplicação permite ao gestor realizar uma análise sobre as tarefas já executadas, fornecendo o seu feedback da acuracidade da tarefa executada pelo desenvolvedor, a fim de disponibilizar os dados de EEG capturados durante a execução desta tarefa para a base

⁴Site: <https://dotnet.microsoft.com/>

de treinamento da técnica de aprendizagem de máquina, promovendo melhorias no índice de acuracidade da técnica.

6 AVALIAÇÃO DA ABORDAGEM 4EXPERTS

Este capítulo tem por objetivo avaliar o impacto da abordagem proposta no Capítulo 5 e para atender a parte dos objetivos específicos apresentados na Seção 1.2. Esta avaliação permitirá comparar a execução das atividades de desenvolvimento sem o uso de um sistema de recomendação e com o uso de um sistema de recomendação que emprega a 4Experts. Para isso foram executados dois experimentos contendo atividades de desenvolvimentos distribuídas em 30 tarefas das quais cada participante executou apenas 10. O primeiro experimento fez uso da abordagem 4Experts para recomendar as 10 tarefas de codificação a partir do conjunto de 30 tarefas disponíveis. Por sua vez, o segundo experimento foi realizado sem o uso da abordagem proposta, recomendando as 10 tarefas de forma sequencial, apenas categorizadas de acordo com a expertise autoidentificada do participante.

A organização deste capítulo começa pela Seção 6.1 que apresenta o objetivo da validação da abordagem 4Experts. A Seção 6.2 descreve as hipóteses à serem validadas ou refutadas pela abordagem proposta, enquanto que a Seção 6.3 detalha as variáveis utilizadas. Na Seção 6.4 é descrito como foi realizada a seleção dos participantes e nas Seções 6.5, 6.6 e 6.7 são apresentados o processo experimental, os materiais utilizados e os detalhes da execução do procedimento respectivamente. Por fim a Seção 6.8 descreve os resultados obtidos pela execução do experimento de validação da abordagem, além de apresentar uma discussão adicional sobre o tema e relatar as ameaças a validade que impactam este experimento.

6.1 Objetivo e Questões de Pesquisa

Nesta Seção são formalizados os objetivos e questões de pesquisa deste experimento. O experimento analisa a viabilidade de empregar a abordagem 4Experts em situações práticas durante as atividades de codificação dos desenvolvedores. Seguindo a convenção Goal Question Method (GQM) proposta por Wohlin (WOHLIN ET AL., 2012)), a seguir apresenta-se o propósito desta investigação.

Analisar a abordagem 4Experts
com o propósito de comprovar a sua eficiência
com respeito a melhoria da corretude e redução do esforço
a partir da perspectiva dos desenvolvedores
no contexto de tarefas de codificação

A partir da definição do propósito emergem duas questões de pesquisa (QP) que nortearão a formulação das hipóteses detalhadas na Seção 6.2 e que são apresentadas a seguir:

QP1: O emprego da abordagem 4Experts resulta em maior corretude das tarefas desempenhadas?

QP2: As recomendações de tarefas propostas pela abordagem 4Experts reduz o esforço despendido pelos desenvolvedores nas tarefas desempenhadas?

6.2 Hipóteses Formuladas

Conforme o objetivo principal deste experimento de comprovar a eficácia da abordagem 4Experts, proposta no Capítulo 5, foram propostas duas hipóteses. A primeira hipótese avalia a corretude das tarefas de programação, enquanto a segunda Hipótese verifica se há redução de esforço quando aplicada a abordagem 4Experts.

Hipótese 1: Esta hipótese supõe que a recomendação de atividades de codificação através da abordagem 4Experts irá garantir maior corretude nas tarefas (T) do que sem o uso da abordagem. A suposição baseia-se na premissa de que um desenvolvedor com maior expertise poderá executar atividades de codificação com maior dificuldade maior e mais assertividade do que um desenvolvedor menos experiente tentando resolver a mesma questão. Esta suposição é embasada pela análise do experimento realizado no Capítulo 4, o qual apresenta evidências de que a maior carga cognitiva despendida para resolver uma atividade de programação é proveniente da falta de expertise do desenvolvedor na tarefa proposta. Com base nesta apresentação, definem-se as hipóteses nula e alternativa comparando a corretude em relação ao uso da abordagem proposta conforme apresentado a seguir:

Hipótese Nula 1, H_{1-0} : A corretude não se altera com a aplicação da abordagem 4Experts para a execução de tarefas (T).

$$H_{1-0}: \text{Corretude}(T)_{4\text{Expert}} \geq \text{Corretude}(T)_{\text{sem } 4\text{Expert}}$$

Hipótese Alternativa 1, H_{1-1} : A corretude com o uso da abordagem 4Experts é menor em relação ao não emprego da abordagem na execução de tarefas (T).

$$H_{1-1}: \text{Corretude}(T)_{4\text{Expert}} < \text{Corretude}(T)_{\text{sem } 4\text{Expert}}$$

Quando analisada a primeira hipótese, se produz conhecimento empírico a respeito da recomendação de tarefas pela abordagem 4Experts, gerando impacto no nível de assertividade das tarefas executadas, resultando em aumento de produtividade e maior qualidade das tarefas executadas, e, por consequência acredita-se uma redução dos esforços de retrabalho para a resolução de tarefas incorretas.

Hipótese 2: Esta hipótese supõe que a recomendação de atividades de codificação através do 4Experts irá garantir menor esforço no processo de codificação das tarefas (T). A suposição baseia-se na premissa que um desenvolvedor que possuir mais aptidão a resolver uma tarefa

de codificação dispensará menos carga cognitiva para a sua execução, conforme detalhado no experimento executado na Capítulo 4 deste estudo.

Um desenvolvedor pode investir grande esforço para executar uma tarefa, e este esforço pode não ser convertido no resultado aguardado para a tarefa, ou seja, apresenta uma tarefa incorreta (BISCHOFF, 2018; FARIAS ET AL., 2015; OLIVEIRA, 2012). Portanto esta hipótese avalia se a abordagem 4Experts, com base nas tarefas propostas, reduz o esforço de codificação, auxiliando os desenvolvedores a codificarem uma tarefa em menor tempo. Com base nesta evidência, a hipótese nula e alternativa sobre o Esforço de Codificação (EC) são apresentadas a seguir:

Hipótese Nula 1, H_{1-0} : O esforço de codificação de uma tarefa (T) não se altera com o uso da abordagem 4Experts.

$$H_{1-0}: \text{Esforço}(T)_{4\text{Expert}} \leq \text{Esforço}(T)_{\text{sem } 4\text{Expert}}$$

Hipótese Alternativa 1, H_{1-1} : O esforço de codificação de uma tarefa (T) do desenvolvedor é maior quando utilizada a abordagem 4Experts.

$$H_{1-1}: \text{Esforço}(T)_{4\text{Expert}} > \text{Esforço}(T)_{\text{sem } 4\text{Expert}}$$

Ao ser analisada a segunda hipótese, pretende-se verificar se o uso do sistema de recomendação reduz o esforço dos desenvolvedores na execução de tarefas de codificação, gerando evidências empíricas sobre como o emprego da abordagem 4Experts impacta no esforço de codificação.

A Tabela 40 apresenta um resumo das hipóteses e alternativas deste experimento, que objetivam identificar indícios que possam ser considerados suficientes para responder as questões de pesquisa deste estudo. A partir dos testes de hipóteses, pretende-se confirmar se a abordagem e o protótipo proposto auxiliam na eficiência da execução das tarefas de codificação.

Tabela 40 – Resumo das hipóteses.

Hipótese	Tipo	Representação
H1	Nula	$H_{1-0}: \text{Corretude}(T)_{4\text{Expert}} \geq \text{Corretude}(T)_{\text{sem } 4\text{Expert}}$
	Alternativa	$H_{1-1}: \text{Corretude}(T)_{4\text{Expert}} < \text{Corretude}(T)_{\text{sem } 4\text{Expert}}$
H2	Nula	$H_{2-0}: \text{Esforço}(T)_{4\text{Expert}} \leq \text{Esforço}(T)_{\text{sem } 4\text{Expert}}$
	Alternativa	$H_{2-1}: \text{Esforço}(T)_{4\text{Expert}} > \text{Esforço}(T)_{\text{sem } 4\text{Expert}}$

Fonte: Elaborado pelo autor.

6.3 Variáveis

Esta pesquisa identifica como variáveis independentes os indicadores nominais de expertise autoatribuída dos participantes e o grau de dificuldade das tarefas aplicadas, as quais incidem sobre as variáveis dependentes: Corretude (C) e Esforço de Codificação (EC) (WOHLIN ET AL., 2012; FARIAS ET AL., 2015; BISCHOFF, 2018). A Corretude (C) representa a assertividade,

ou não da tarefa em análise, enquanto que o Esforço de Codificação (EC) representa o tempo médio (minutos) que os participantes levaram para solucionar as tarefas de codificação propostas. A Tabela 41 apresenta o resumo das variáveis investigadas neste estudo.

Tabela 41 – Descrição das variáveis de estudo.

Variável		Escala
Independente	Expertise	Nominal[nível1, nível2, nível3]
	Dificuldade	Nominal[fácil, média, difícil]
Dependente	Corretude (C)	Nominal[correto, incorreto]
	Esforço Codificação (EC)	Ordinal[0..1,5] (minutos)

Fonte: Elaborado pelo autor.

A variável dependente da primeira hipótese é a corretude (C) da tarefa a ser resolvida. Esta variável será identificada através da resposta correta ou incorreta da tarefa, enquanto que a variável dependente da segunda hipótese será o esforço, equivalente ao tempo de execução em minutos da tarefa.

6.4 Contexto e Seleção dos Participantes

Para executar o experimento de validação os participantes convidados foram os alunos da disciplina de Algoritmo do segundo semestre de 2019 do curso de Análise e Desenvolvimento de Sistemas do Centro Universitário UniFtec de Caxias do Sul. A turma é composta por 26 alunos, dos quais 20 participaram do experimento e os demais alunos não participaram por estarem ausentes durante os períodos de execução do experimento. Estes alunos tiveram a sua expertise identificada através do autorrelato obtido pelo preenchimento do questionário demográfico aplicado anteriormente no Capítulo 4 e disponibilizado no Apêndice A.5. A Tabela 42 apresenta relação de participantes do experimento com o seu nível de expertise, idade e curso que frequenta na instituição. As expertises dos participantes foram agrupadas em três níveis: O nível 1 representa os participantes com menor expertise técnica, enquanto o nível 2 representa os participantes com expertises médias e o nível 3 representa os alunos com as maiores expertises técnicas em programação.

A partir da identificação das expertises dos participantes pelo questionário, os alunos foram agrupados de acordo com as suas expertises. Esta classificação das expertises permitiu a subdivisão do grupo em dois grupos de 10 alunos, de maneira que as expertises dos participantes em cada um dos grupos fossem distribuídas de forma homogênea. Um dos grupos de participantes foi submetido a abordagem 4Experts enquanto o outro grupo executou o conjunto de tarefas sem a abordagem. A Tabela 43 apresenta a distribuição dos participantes de acordo com a sua expertise entre os grupos de execução do experimento.

Optou-se em escolher estudantes em lugar de profissionais da indústria devido ao contexto de sala de aula ser mais controlado do que o ambiente industrial, quando refere-se a e realização de experimentos, pelo relacionamento do autor desta dissertação com a instituição de ensino e

Tabela 42 – Participantes do experimento.

Participante	Curso	Idade	Nível Expertise*
P01	Gestão de TI	30	Nível 2
P02	Análise e Desenvolvimento de Sistemas	28	Nível 2
P03	Análise e Desenvolvimento de Sistemas	20	Nível 2
P04	Análise e Desenvolvimento de Sistemas	18	Nível 2
P05	Redes de Computadores	19	Nível 3
P06	Análise e Desenvolvimento de Sistemas	20	Nível 3
P07	Análise e Desenvolvimento de Sistemas	24	Nível 2
P08	Análise e Desenvolvimento de Sistemas	19	Nível 2
P09	Engenharia da Computação	28	Nível 2
P10	Análise e Desenvolvimento de Sistemas	28	Nível 2
P11	Análise e Desenvolvimento de Sistemas	21	Nível 3
P12	Análise e Desenvolvimento de Sistemas	21	Nível 3
P13	Análise e Desenvolvimento de Sistemas	29	Nível 2
P14	Gestão de TI	24	Nível 1
P15	Redes de Computadores	21	Nível 1
P16	Análise e Desenvolvimento de Sistemas	19	Nível 2
P17	Redes de Computadores	23	Nível 2
P18	Engenharia da Computação	21	Nível 2
P19	Engenharia da Computação	20	Nível 3
P20	Análise e Desenvolvimento de Sistemas	34	Nível 3

Fonte: Elaborado pelo autor.

Tabela 43 – Distribuição dos participantes entre os experimentos.

Nível Expertise	Quantidade Participantes	Grupos	
		Com abordagem 4Experts	Sem abordagem 4Experts
Nível 1	2	P15	P14
Nível 2	12	P02, P03, P04, P09, P16, P18	P01, P07, P08, P10, P13, P17
Nível 3	6	P11, P12, P19	P05, P06, P20

Fonte: Elaborado pelo autor.

também para evitar possíveis ameaças a validade do experimento.

A professora Neiva Kuiwen ministra a disciplina citada a mais de 10 anos, possuindo uma experiência em relação as atividades de codificação empregadas na disciplina necessária para a classificação das tarefas requeridas pela técnica proposta. Esta característica também exige o experimento de qualquer influência do autor sobre as questões propostas ao experimento, sendo empregadas as próprias questões utilizadas durante a disciplina.

6.5 Processo Experimental

O processo experimental para a validação da abordagem 4Experts é apresentado na Figura 31. Este processo é baseado nas práticas de WOHLIN (WOHLIN ET AL., 2012), sendo composto por três fases.

Na primeira fase os participantes receberam treinamento para garantir que tenham compreendido o funcionamento do experimento, além de assinarem um termo de consentimento

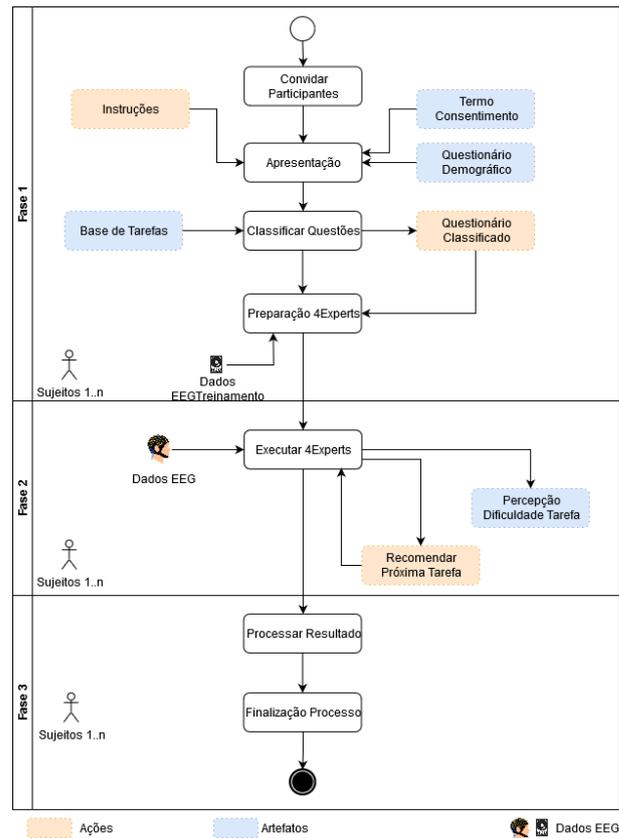


Figura 31 – Processo experimental.

Fonte: Adaptado de (FARIAS ET AL., 2019)

e responder o questionário demográfico (Apêndice A.2) com o propósito de identificar a sua expertise conforme detalhado na Tabela 42. Além disso, nesta etapa, o professor da disciplina é convidado a classificar as questões do experimento de acordo com a sua percepção de dificuldade assumindo o papel equivalente ao Gerente, conforme proposto na abordagem 4Experts apresentada na Seção 5.1. No total foram utilizadas 30 tarefas detalhadas no Apêndice A.5, divididas em 10 de nível fácil, 10 de nível médio e 10 de nível difícil, com os níveis de dificuldade classificados através das diretrizes empregadas para a classificação de tarefas no Capítulo 4 deste estudo (*Análise Empírica da Aquisição de Conhecimento Através do Uso de Dados Biométricos*) formando desta forma a Base de Tarefas.

Além do banco de dados de questões, nesta etapa também é realizado o carregamento da base de dados EEG utilizada como base de treinamento para a técnica de Aprendizado de Máquina utilizada na abordagem 4Experts. Esta base de treinamento foi obtida durante a execução do experimento apresentado no Capítulo 4. Por fim, também foi criada uma relação entre o grau de dificuldade identificado para as questões aplicadas no experimento e o nível de expertise dos participantes. Esta relação foi executada pelo professor ministrante da disciplina de maneira empírica e detalhada na Tabela 44, sendo necessária para a recomendação das tarefas durante a execução da abordagem 4Experts.

Na segunda fase, os participantes foram submetidos a execução do experimento. Todos os

Tabela 44 – Relacionamento entre grau de dificuldade e nível de expertise.

Dificuldade Tarefas	Expertise Participantes
Fácil	Nível 1
Médio	Nível 2
Difícil	Nível 3

Fonte: Elaborado pelo autor.

participantes responderam a um conjunto de 10 tarefas de um total de 30 disponíveis no banco de dados. Os participantes que não utilizaram a abordagem 4Experts responderam a um conjunto de tarefas com um grau de dificuldade compatível com o grau de expertise identificado através do questionário demográfico, enquanto que os participantes submetidos a abordagem tiveram apenas a primeira tarefa recomendada com base na sua expertise e as demais passaram a ser recomendadas pelo 4Experts conforme este analisava os dados biométricos de EEG coletados com o uso de um Eletroencefalograma portátil. As sequências de tarefas recomendadas para cada um dos participantes do experimento é apresentada na Tabela 45.

Tabela 45 – Questões respondidas por participante.

Participante	Abordagem 4Experts	Tarefas Executadas
P01	Não	T11, T12, T13, T14, T15, T16, T17, T18, T19, T20
P02	Sim	T15, T11, T12, T13, T01, T02, T03, T04, T05, T06
P03	Sim	T15, T01, T02, T03, T04, T05, T06, T07, T08, T09
P04	Sim	T15, T11, T12, T13, T14, T16, T17, T18, T18, T20
P05	Não	T21, T22, T23, T24, T25, T26, T27, T28, T29, T30
P06	Não	T21, T22, T23, T24, T25, T26, T27, T28, T29, T30
P07	Não	T11, T12, T13, T14, T15, T16, T17, T18, T19, T20
P08	Não	T11, T12, T13, T14, T15, T16, T17, T18, T19, T20
P09	Sim	T15, T11, T12, T13, T14, T16, T17, T18, T19, T20
P10	Não	T11, T12, T13, T14, T15, T16, T17, T18, T19, T20
P11	Sim	T25, T11, T01, T02, T03, T04, T05, T06, T07, T08
P12	Sim	T25, T21, T22, T23, T24, T26, T27, T28, T29, T30
P13	Não	T11, T12, T13, T14, T15, T16, T17, T18, T19, T20
P14	Não	T01, T02, T03, T04, T05, T06, T07, T08, T09, T10
P15	Sim	T05, T01, T02, T03, T04, T06, T07, T08, T09, T10
P16	Sim	T15, T01, T02, T03, T04, T05, T06, T07, T08, T09
P17	Não	T11, T12, T13, T14, T15, T16, T17, T18, T19, T20
P18	Sim	T15, T11, T01, T02, T03, T04, T05, T06, T07, T08
P19	Sim	T25, T11, T12, T13, T01, T02, T03, T04, T05, T06
P20	Não	T21, T22, T23, T24, T25, T26, T27, T28, T29, T30

Fonte: Elaborado pelo autor.

O objetivo da terceira fase é realizar o processamento dos resultados obtidos, comparando a corretude e o esforço empregado pelos participantes com e sem a abordagem 4Experts, para ser possível apresentar os resultados na Seção 6.8.

6.6 Material

Para ser possível realizar o experimento de validação da abordagem 4Experts foi necessário construir um protótipo deste sistema. Nesta seção são detalhados os principais materiais utilizados para a construção deste protótipo em suas diversas etapas.

Emotiv Epoc+: O Emotiv Epoc+ (EMOTIV, 2011a) é um dispositivo de eletroencefalografia (EEG) portátil, desenvolvido pela empresa Emotiv e detalhado na Figura 10. Este dispositivo tem sido empregado em diversos experimentos em diversas áreas de pesquisa, dentre estas a Engenharia de Software (CRK; KLUTHE; STEFIK, 2015). O seu uso tem sido ampliado na área por ser um dispositivo portátil, de custo relativamente baixo, possuir um setup rápido durante a realização de experimentos e apresentar uma acuracidade superior em relação a outros dispositivos conforme apresentado por DAS (DAS ET AL., 2014) em uma análise comparativa com outro dispositivo de EEG portátil. A acuracidade superior se deve ao equipamento possuir 14 sensores para a coleta de dados EEG, seguindo o protocolo internacional 10-20 de posicionamento dos sensores EEG (MEYJES; ARTS, 1984) detalhado na Figura 32.

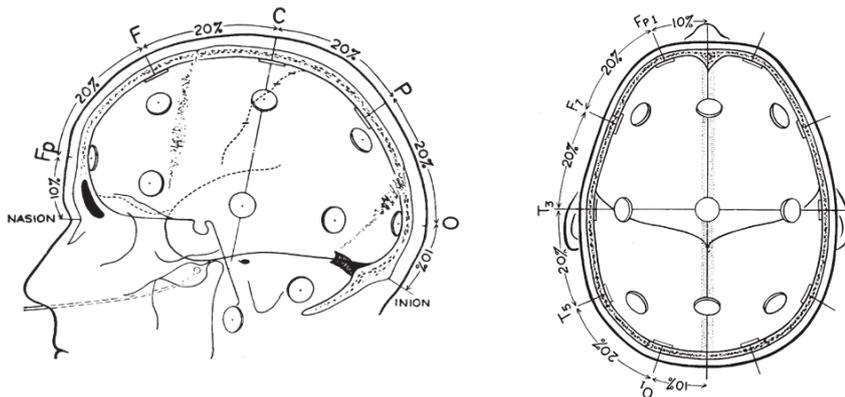


Figura 32 – Sistema internacional 10-20.

Fonte: (MEYJES; ARTS, 1984).

Além destas características, o Emotiv Epoc+ coleta dados EEG de forma bruta com uma taxa de amostragem de 256 Mhz, através de conexão bluetooth com um computador. A taxa de amostragem de 256Mhz é superior ao mínimo recomendado pelo teorema de Nyquist, o qual menciona que a taxa de amostragem deve ser no mínimo duas vezes a frequência a ser estudada. Como a abordagem 4Experts necessita de processamento em tempo real, em conjunto com o Emotiv Epoc+ foi utilizado o software EmotivAPP SDK (EMOTIV, 2011b) versão 2.3.212.83 em conjunto com o software EmotivPro versão 2.2.3, apresentado na Figura 33. O uso do EmotivAPP SDK licenciado permitiu a captura em tempo real dos dados EEG brutos diretamente a partir do software OpenSesame enquanto os participantes performavam as atividades do experimento.

OpenSesame (Versão 3.2.8): OpenSesame (MATHÔT, 2010) é um software open source escrito na linguagem Python utilizado para a realização de experimentos em psicologia e neu-

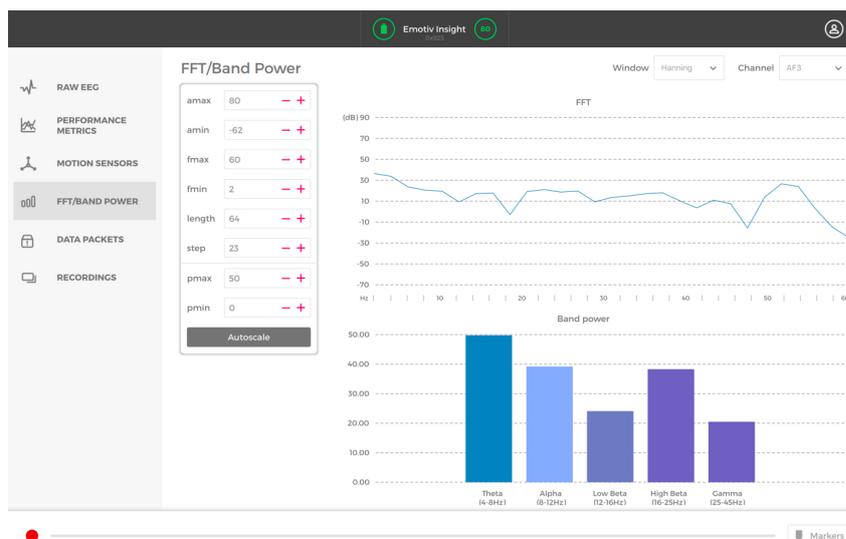


Figura 33 – Emotiv Pro.

Fonte: (EMOTIV, 2011b).

rociência. Este software é mantido pela *European Society for Cognitive Psychology* (ESCoP) e pela empresa SR Research, responsável por desenvolver sistemas de eye-track. O software OpenSesame foi escolhido para a execução deste experimento por possuir referências de uso em outros artigos (JIN; BORST; VUGT, 2019; OpenSesame: An open-source, graphical experiment builder for the social sciences, 2012), além de possuir uma interface visual intuitiva conforme apresentado na Figura 13 e permitir a programação de scripts em python. A possibilidade de programar Python na própria ferramenta OpenSesame permitiu aplicar a técnica de aprendizado de máquina empregada na abordagem 4Experts com os dados EEG coletados em tempo real enquanto o participante executava o questionário. O resultado do processamento foi diretamente aplicado na ferramenta alterando o fluxo de recomendação de questões enquanto o experimento executava.

Python (Versão 3.7.3): A linguagem de programação Python (FOUNDATION, 2001) foi utilizada neste experimento em diversos momentos: (1) Para construção de script no software OpenSesame para envio de dados por porta serial, em eventos de início e fim das questões, para marcar os momentos destes eventos junto aos dados brutos coletados pelo EEG; (2) Para processar os dados brutos coletados pelo EEG, identificando os períodos do EEG em que o participante estava respondendo as questões, separando estes períodos em conjunto de dados independentes; (3) Processar estes conjuntos de dados com a técnica Multitaper para fornecer uma análise espectral relativa com o propósito de calcular a carga cognitiva. Para facilitar a realização dos procedimentos citados, foram utilizadas bibliotecas especializadas para o tratamento de dados que seguem: (1) **MNE versão 0.18.2** (DEVELOPERS, 2012) - é uma biblioteca especializada para tratamento e visualização de dados EEG. Neste experimento esta biblioteca foi utilizada para realizar a leitura dos arquivos EDF gerados pelo Xavier Test Bench, (2) **Pandas versão 0.25.0** (DEVELOPERS, 2001a) - esta biblioteca facilita o processamento e análise de grandes

quantidades de dados no ambiente python. Neste projeto, a biblioteca foi utilizada para a manipulação dos dados de EEG, (3) **Scipy versão 1.3.0** (DEVELOPERS, 2001b) - esta biblioteca oferece um conjunto de rotinas matemáticas que podem ser utilizadas de forma amigável. A biblioteca foi utilizada para realizar o processamento dos dados coletados com o uso da técnica Multitaper.

6.7 Execução

A execução do processo foi realizada seguindo os procedimentos definidos na Seção 6.5. A execução do experimento foi conduzida na disciplina como uma atividade de revisão de conteúdo para a prova de avaliação do Grau B.

Na semana que antecedeu a execução do experimento, ele foi apresentado a todos os alunos. Nesta apresentação foi demonstrado o processo de execução do experimento, resolvidas dúvidas dos alunos, além das recomendações de não ingerir nenhuma substância contendo cafeína nas horas que antecederiam o experimento e a não utilizar roupas que pudessem ocasionar aumento de eletricidade estática. Além disso, neste dia foi coletado o termo de aceite e a execução do questionário demográfico para todos os alunos da turma.

No dia da execução do experimento, cada participante foi conduzido individualmente até o local do experimento, onde ele desligou qualquer equipamento eletrônico que possuía consigo. O participante também foi aconselhado a não utilizar roupas que pudessem provocar um aumento na eletricidade estática em seu corpo enquanto executava o experimento. Além disso, para reduzir as interferências os participantes fizeram uso de uma pulseira antiestática e as luminárias da sala que possuíam reator elétrico não foram ligadas.

O participante é motivado a se posicionar de forma que se sinta confortável e possa evitar movimentos durante a execução do experimento, conforme apresentado na Figura 34. Na sequência, o Emotiv Epoc+ é colocado sobre o escalpo e calibrado para que fornecer os dados de EEG corretamente.

Após esta etapa ser concluída, é iniciada a gravação do EEG e em seguida é iniciado o processo de execução das tarefas. Inicialmente é apresentada uma tela ao participante contendo as instruções e uma questão inicial utilizada como calibração e teste do processo. Depois desta questão ser respondida pelo participante, o experimento inicia de fato, exibindo a primeira questão de acordo com a expertise do participante identificada anteriormente e parametrizada no mecanismo da abordagem 4Experts. Ao término da questão, ocorrido quando o participante seleciona a resposta, a abordagem 4Experts realiza o processamento dos dados EEG coletados durante o período da questão, fazendo uso, neste momento da técnica de aprendizado de máquina parametrizada para a abordagem. Enquanto o processamento é executado é exibida uma tela informando ao participante para aguardar o processamento. Após, um intervalo de 5 segundos para relaxamento é aplicado e a nova questão é recomendada de acordo com o resultado do processamento da técnica de machine learning. Se a técnica predizer que o participante selecio-



Figura 34 – Participante durante a execução do experimento.

Fonte: Elaborado pelo autor.

nou uma opção correta, a próxima questão a ser respondida será do mesmo grau de dificuldade da questão anterior, caso contrário, será uma questão de nível de dificuldade inferior ao nível da questão atual. Para cada questão o participante dispõe de 1,5 minutos para analisar e responder a atividade. Caso ele não responda a questão, a mesma é considerada como não respondida, atribuindo uma resposta incorreta para a questão e o processo avança para a próxima tarefa.

Para os participantes que não foram submetidos à abordagem 4Experts o processo de execução é similar. Neste caso, os participantes também foram encaminhados individualmente para o local do experimento e motivados a se posicionar de forma confortável perante o computador onde o experimento é executado. Porém, não foram necessários os cuidados para a redução da eletricidade estática importantes para garantir a qualidade dos dados EEG coletados. A execução das tarefas do experimento se deu da mesma forma aplicada aos participantes que foram submetidos a abordagem 4Experts, porém, o mecanismo do experimento não realizava nenhum processamento entre a execução das tarefas e a recomendação das tarefas sempre foi baseada nas expertises declaradas pelos participantes.

Quando o participante chegava ao fim do experimento, para os participantes submetidos a abordagem 4Experts, a gravação dos dados de EEG era finalizada, o Emotiv Epoc+ removido e os dados registrados verificados com o intuito de confirmar se tudo havia transcorrido adequadamente durante o processo. Por sua vez, todos os participantes do experimento foram indagados ao final sobre as suas impressões durante o processo e eventuais desconfortos durante a execução.

6.8 Resultados Coletados

Após a conclusão do experimento foi iniciada a verificação dos dados, procurando investigar os impactos relativos ao esforço necessário para a execução das tarefas, bem como quantificar a corretude das respostas. Os métodos estatísticos nortearam esta pesquisa, na qual procurou-se examinar todas as ocorrências dentro de uma população, isto é, a amostra a ser investigada para evidenciar o seu comportamento na aplicação da abordagem 4Experts. Os resultados coletados foram processados com o auxílio da ferramenta RStudio¹, facilitando a visualização das variações dos resultados e a sua parametrização.

De acordo com a descrição apresentada na Seção 6.5 este experimento conta com uma base de 30 tarefas possíveis de serem utilizadas. Destas 30 tarefas apenas 10 são escolhidas através do sistema de recomendação para os participantes do experimento que fizeram uso da abordagem ou recomendadas manualmente para aqueles que não foram submetidos a técnica de recomendação. As questões escolhidas foram analisadas e executadas pelos participantes gerando o resultado necessário para mensurar o esforço e corretude aplicada nas tarefas. A Tabela 46 apresenta o conjunto total de dados utilizados para as análises estatísticas aplicadas nesta Seção, enquanto a Tabela 47 apresenta os resultados manuais, ou seja, sem o uso da recomendação de tarefa, onde é possível identificar o tempo que cada participante utilizou para realizar a tarefa e verificar se a conclusão da tarefa resultou em uma resposta correta ou incorreta.

¹Site: <https://rstudio.com/>

Por sua vez, a Tabela 48 mostra os resultados dos participantes que executaram tarefas recomendadas pela abordagem 4Experts, onde é possível identificar o tempo que cada participante aplicou na resolução da tarefa e se o resultado foi correto ou incorreto.

Tabela 47 – Resultados individuais por participante sem 4Experts.

Participante	Sem uso da abordagem 4Experts																			
	T01		T02		T03		T04		T05		T06		T07		T08		T09		T10	
	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)
P01	1.12	1	1.5	0	0.68	1	0.79	1	0.47	1	1.5	0	1.38	0	0.91	0	0.84	1	0.91	0
P05	1.25	0	1.33	1	1.5	0	1.17	1	0.65	1	0.96	0	1.29	0	1.18	1	1.08	1	1.26	0
P06	1.5	0	0.55	0	0.96	0	1.05	1	1.5	0	0.14	1	1.5	0	0.73	1	1.12	0	0.52	1
P07	1.17	0	1.5	0	1.2	0	0.87	1	1.09	1	0.47	1	1.1	0	0.93	0	0.71	0	0.74	0
P08	1.19	1	1.5	0	1.5	0	0.73	1	0.55	1	1.04	1	1.5	0	1.31	0	0.83	1	1.5	0
P10	0.67	1	1.08	1	0.67	1	0.61	1	0.6	1	0.85	1	1.39	0	1.18	0	0.85	0	1.17	1
P13	1.12	1	1.5	0	1.48	0	0.99	1	0.49	1	1.03	1	0.87	0	1.24	0	1.3	0	1.3	1
P14	0.62	0	1.5	0	1.26	1	0.54	0	1.45	0	0.97	0	1.14	1	1.11	0	0.74	1	0.62	0
P17	1.2	0	1.21	1	1.5	0	0.74	1	0.47	1	0.89	1	1.47	1	0.85	0	0.82	1	1.5	0
P20	0.97	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5	0	1.01	1	1.5	0	0.97	0	1.5	0

Fonte: Elaborado pelo autor.

Tabela 48 – Resultados individuais por participante com 4Experts.

Participante	Com uso da abordagem 4Experts																			
	T01		T02		T03		T04		T05		T06		T07		T08		T09		T10	
	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)	Esforço (EC)	Corretude (C)
P01	0.36	1	0.35	1	0.69	0	0.18	0	0.63	1	0.33	1	0.46	1	0.45	1	0.71	0	0.56	1
P02	0.78	1	0.37	1	0.31	1	0.77	1	0.31	1	0.22	1	0.24	1	1.12	0	0.35	1	0.38	1
P03	0.39	1	0.41	1	0.82	0	0.12	0	0.4	1	0.67	1	0.56	1	0.74	0	0.43	0	1.06	1
P04	0.22	1	0.46	1	0.41	0	0.17	0	0.35	1	0.83	0	0.62	0	0.77	1	0.15	1	0.57	0
P05	1.48	1	1.22	1	0.18	1	0.45	1	0.51	1	0.17	1	0.38	1	0.31	1	1	0	0.27	1
P06	1.27	0	1.04	1	1.5	0	0.75	0	1.5	0	1.49	1	0.84	1	1.01	0	0.67	1	1.5	0
P07	1.02	0	0.34	1	0.35	0	0.22	0	0.77	0	0.56	0	0.65	0	0.2	1	0.25	1	0.28	1
P08	0.49	1	0.19	1	0.38	0	0.55	0	0.52	1	0.68	0	0.27	1	0.25	0	0.25	1	0.4	1
P09	0.39	0	0.58	1	0.34	1	0.37	1	0.43	0	0.65	0	0.29	1	0.51	1	0.63	0	0.31	1
P10	0.86	0	0.56	1	0.68	0	0.23	0	0.57	1	0.33	1	0.37	1	0.23	1	0.27	0	0.39	0

Fonte: Elaborado pelo autor.

Os dados para a análise quantitativa foram obtidos através da extração dos dados gerados pelo software *Open Sesame* enquanto este era utilizado na execução do experimento. A análise destes dados foi realizada com a utilização da estatística descritiva a qual verificou a distribuição dos mesmos, enquanto para os testes de hipóteses foi aplicada a inferência estatística. O total de amostras deste experimento é igual a 200, ou seja, 100 amostras sem o uso do 4Experts e 100 amostras resultantes da recomendação de tarefas com o emprego da abordagem 4Experts. A análise da normalidade das amostras foi executada através do teste *Kolmogorov-Smirnov*, utilizado para grandes amostras (quando $n \geq 30$) (levine et al., 2005).

A Tabela 49 apresenta o resultado do nível descritivo com o uso da técnica estatística de Kolmogorov-Smirnov, com um nível de significância de *Lilliefors* para teste de normalidade, os resultados retornados da significância, também conhecida como valor $p < 0,05$ para o teste nas amostras analisadas. Desta forma, pode-se afirmar que o nível de significância de 5% não provém de uma população normal, tanto pelos indicadores de esforço como os de corretude.

A Tabela 50 e a Tabela 51 apresentam os resultados agrupados pelas tarefas aplicadas, demonstrando os resultados de: média, mediana, desvio padrão, máximo, mínimo, primeiro

Tabela 49 – Teste de normalidade.

Corretude	Kolmogorov-Smirnov*		
	Estatística	df	Sig.
Com 4Experts	0,407	100	0,000
Sem 4Experts	0,371	100	0,000
Esforço	Estatística		
	Estatística	df	Sig.
Com 4Experts	0,144	100	0,000
Sem 4Experts	0,118	100	0,001

* Correlação de Significância de Lilliefors

Fonte: Elaborado pelo autor.

quartil e terceiro quartil, separados por corretude e esforço. Por sua vez a Tabela 52 abstrai as informações por tarefa aplicada, corretude e esforço, demonstrando a média dos resultados individuais das variáveis e o percentual de diferença entre cada tratamento.

Tabela 50 – Resultados agrupados por tarefa - Corretude.

		T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	
Participantes		10	10	10	10	10	10	10	10	10	10	
Corretude	Sem 4Experts	Média	0,4	0,3	0,3	0,8	0,7	0,6	0,3	0,2	0,5	0,3
		Mediana	0	0	0	1	1	1	0	0	0,5	0
		Desvio Padrão	0,51	0,48	0,48	0,42	0,48	0,51	0,48	0,42	0,52	0,48
		Mínimo	0	0	0	0	0	0	0	0	0	0
		Máximo	1	1	1	1	1	1	1	1	1	1
		1º Quartil	0	0	0	1	0,25	0	0	0	0	0
		3º Quartil	1	0,75	0,75	1	1	1	0,75	0	1	0,75
Corretude	Com 4Experts	Média	0,6	1	0,3	0,3	0,7	0,6	0,8	0,6	0,5	0,7
		Mediana	1	1	0	0	1	1	1	1	0,5	1
		Desvio Padrão	0,51	0	0,48	0,48	0,48	0,51	0,42	0,51	0,51	0,48
		Mínimo	0	1	0	0	0	0	0	0	0	0
		Máximo	1	1	1	1	1	1	1	1	1	1
		1º Quartil	0	1	0	0	0,25	0	1	0	0	0,25
		3º Quartil	1	1	0,75	0,75	1	1	1	1	1	1

Fonte: Elaborado pelo autor.

A partir de uma análise geral em relação ao esforço e corretude aplicadas as tarefas executadas pelos participantes sem que estas fossem recomendadas, houve uma necessidade de esforço considerável, de forma que o tempo médio necessário para solucionar as tarefas propostas foi de 1,07 minutos com uma assertividade de 44%. Porém, quando empregada a recomendação de tarefas utilizando a abordagem 4Experts foi possível perceber uma melhora na eficiência e eficácia desempenhada pelos desenvolvedores. Neste caso, o esforço médio na resolução de todas as tarefas foi de 0,54 minutos com uma assertividade de 60%.

A Figura 35 exibe o comparativo da corretude sem o sistema de recomendação em relação a corretude aplicada com a abordagem de recomendação 4Experts, sendo produzido um total de 200 respostas de tarefas, das quais 100 respostas foram geradas sem o emprego da reco-

Tabela 51 – Resultados agrupados por tarefa - Esforço.

		T01	T02	T03	T04	T05	T06	T07	T08	T09	T10		
Participantes		10	10	10	10	10	10	10	10	10	10		
Esforço	Sem 4experts	Média	1,08	1,31	1,22	0,89	0,87	0,93	1,26	1,09	0,92	1,10	
		Mediana	1,14	1,50	1,37	0,83	0,62	0,96	1,33	1,14	0,84	1,21	
		Desvio Padrão	0,26	0,30	0,34	0,28	0,45	0,41	0,22	0,23	0,18	0,37	
		Mínimo	0,62	0,55	0,67	0,54	0,47	0,14	0,87	0,73	0,71	0,52	
		Máximo	1,50	1,50	1,50	1,50	1,50	1,50	1,50	1,50	1,50	1,30	1,50
		1º Quartil	1,00	1,24	1,02	0,73	0,50	0,86	1,11	0,91	0,82	0,78	
		3º Quartil	1,19	1,50	1,50	1,03	1,36	1,03	1,45	1,22	1,05	1,45	
Esforço	Com 4Experts	Média	0,72	0,55	0,56	0,38	0,59	0,59	0,46	0,55	0,47	0,57	
		Mediana	0,63	0,43	0,39	0,30	0,51	0,60	0,42	0,48	0,39	0,39	
		Desvio Padrão	0,42	0,32	0,38	0,24	0,34	0,38	0,19	0,33	0,27	0,40	
		Mínimo	0,22	0,19	0,18	0,12	0,31	0,17	0,24	0,20	0,15	0,27	
		Máximo	1,48	1,22	1,50	0,77	1,50	1,49	0,84	1,12	1,00	1,50	
		1º Quartil	0,39	0,35	0,34	0,19	0,40	0,33	0,31	0,26	0,25	0,32	
		3º Quartil	0,98	0,57	0,68	0,52	0,61	0,67	0,60	0,76	0,66	0,56	

Fonte: Elaborado pelo autor.

mendação e as outras 100 foram geradas com a recomendação de tarefas. Pode-se observar que os participantes responderam a 44 tarefas de forma assertiva sem a recomendação de tarefas, já com o auxílio da abordagem 4Experts o número de acertos aumentou para 61. Desta forma é possível concluir que a abordagem proposta auxilia os desenvolvedores a melhorarem a corretude das tarefas desempenhadas por eles.

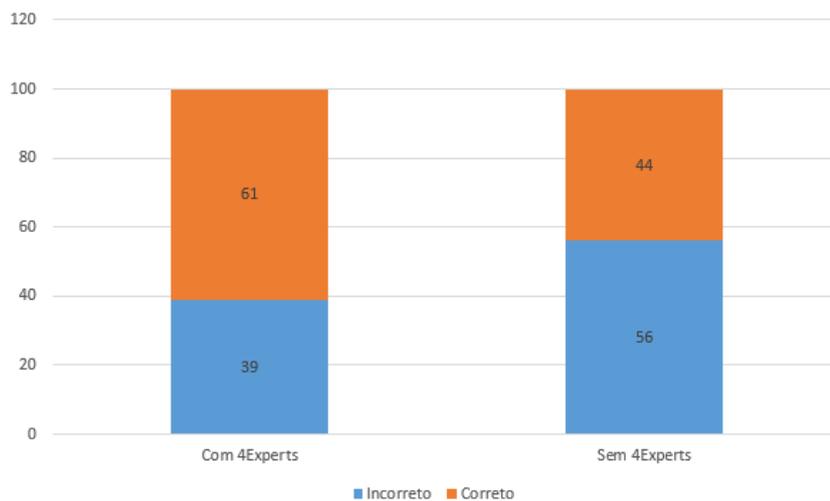


Figura 35 – Corretude por experimento.

Fonte: Elaborado pelo autor.

A Figura 36 exibe um comparativo em minutos do tempo médio para cada uma das tarefas do experimento. Nesta figura é possível identificar que o tempo médio que os participantes levaram para desenvolver as tarefas com o auxílio da recomendação das tarefas com a abordagem

Tabela 52 – Relação das variáveis de quantificação.

Tarefas	Variáveis	Uso abordagem 4Experts	Média	% dif.
Todas	Corretude	Sim	0,61	38,64
		Não	0,44	
	Esforço	Sim	0,54	49,02
		Não	1,07	
T01	Corretude	Sim	0,60	50,00
		Não	0,40	
	Esforço	Sim	0,72	33,33
		Não	1,08	
T02	Corretude	Sim	1,00	233,33
		Não	0,30	
	Esforço	Sim	0,55	58,02
		Não	1,31	
T03	Corretude	Sim	0,30	0,00
		Não	0,30	
	Esforço	Sim	0,56	54,10
		Não	1,22	
T04	Corretude	Sim	0,30	62,50
		Não	0,80	
	Esforço	Sim	0,38	57,30
		Não	0,89	
T05	Corretude	Sim	0,70	0,00
		Não	0,70	
	Esforço	Sim	0,59	32,18
		Não	0,87	
T06	Corretude	Sim	0,60	0,00
		Não	0,60	
	Esforço	Sim	0,59	36,56
		Não	0,93	
T07	Corretude	Sim	0,80	166,67
		Não	0,30	
	Esforço	Sim	0,46	63,49
		Não	1,26	
T08	Corretude	Sim	0,60	200,00
		Não	0,20	
	Esforço	Sim	0,55	49,54
		Não	1,09	
T09	Corretude	Sim	0,50	0,00
		Não	0,50	
	Esforço	Sim	0,47	48,91
		Não	0,92	
T10	Corretude	Sim	0,70	133,33
		Não	0,30	
	Esforço	Sim	0,57	48,18
		Não	1,10	

Fonte: Elaborado pelo autor.

4Experts foi inferior ao tempo em relação a execução das tarefas sem a recomendação. Desse modo, é possível afirmar que os participantes empenharam menos esforço em realizar as tarefas quando utilizaram o sistema de recomendação em relação ao não empregado da abordagem.

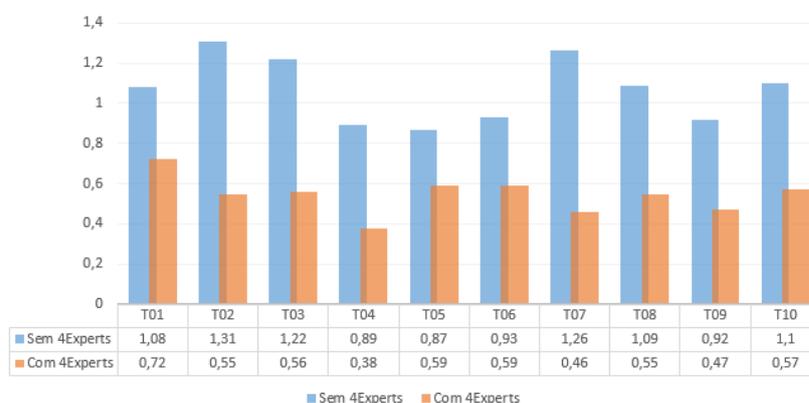


Figura 36 – Tempo médio de execução por tarefa.

Fonte: Elaborado pelo autor.

Aplicando os dados coletados com a ferramenta RStudio foi possível obter os cálculos de cada variável, as quais foram subdivididas conforme o tratamento, em categorias, tendo em vista diferentes resultados incorporados a Tabela 53. Nesta tabela são incluídos os resultados para as variáveis de quantificação, Corretude (C), que apresenta a corretude para a execução das tarefas, ou seja, quanto maior o índice, melhor será a capacidade de retornar a resposta correta. Da mesma forma a variável de Esforço de Codificação (EC), apresenta o tempo médio que os participantes levaram para responder as tarefas. O principal motivo em aplicar os cálculos pela estatística descritiva com o uso da ferramenta RStudio advém deste retornar dos resultados de testes estáticos e testes de hipóteses: desvio padrão, mínimo e máximo, média, mediana e por fim a aplicação dos testes de *Wilcoxon* e o *McNemar*.

Tabela 53 – Estatística descritiva e testes estatísticos.

Variáveis	Tratamento	DP	Mín	25th	MD	75th	Máx	Méd	%df	W p-v	N p-v
Corretude	Sem 4Experts	0,49	0,00	0,00	0,00	1,00	1,00	0,44	38,64	-	0,019
	Com 4Experts	0,48	0,00	0,00	1,00	1,00	1,00	0,63			
Esforço	Sem 4Experts	0,34	0,14	0,82	1,10	1,40	1,50	1,07	49,02	0,0001	-
	Com 4Experts	0,33	0,12	0,31	0,44	0,68	1,50	0,54			

Fonte: Elaborado pelo autor.

O teste de normalidade de *Kolmogorov-Smirnov* - (Lilliefors) (BISCHOFF, 2018; OLIVEIRA, 2012; MOHD RAZALI; YAP, 2011) executado anteriormente indica que os dados não se encontram normalmente distribuídos, sendo necessário aplicar testes não paramétricos. Por isso, o teste de *Wilcoxon* é aplicado para analisar o esforço, enquanto o teste *McNemar* foi empregado para analisar a corretude da abordagem proposta. Estes testes serão empregados para analisar os testes de hipóteses abordados na Seção 6.2 e responder as questões de pesquisa descritas na Seção 6.1.

O teste de *Wilcoxon* é um teste de hipótese estático não paramétrico, aplicado para confrontar a média de duas amostras relacionadas, ou seja, é um teste de diferença emparelhado que pode ser utilizado como alternativa ao *t-test*. O *t-test* é considerado um teste de hipótese que utiliza conceitos estatísticos para rejeitar ou não uma hipótese nula, empregado para confrontar médias de amostras diferentes, normalmente o *t-test* é aplicado quando segue uma distribuição normal (BARROS; MAZUCHELI, 2005).

O teste *McNemar* é usado para analisar a eficiência de determinada técnica, isto é, tem por objetivo avaliar a eficiência de situações, "o antes e o depois" de cada amostra utilizada e mensurar se faz nível de escala nominal ou ordinal. Este teste é aplicado a variáveis dicotômicas, ou seja, à amostras que apenas tomam dois valores, por exemplo, 0 e 1 (BISCHOFF, 2018; FARIAS ET AL., 2015; FIRMINO, 2015).

6.8.1 QP1: O emprego da abordagem 4Experts resulta em maior corretude das tarefas desempenhadas? (Hipótese 1)

A primeira questão de pesquisa investiga o impacto na corretude das tarefas quando aplicado a recomendação de tarefas baseada na abordagem 4Experts, analisando a assertividade das respostas relativas as tarefas recomendadas aos participantes do experimento com o uso da abordagem. Analisando os dados da Tabela 53 é possível observar que a média da corretude foi superior com o uso da abordagem 4Experts em relação a recomendação manual de tarefas, isto é, uma média de 0,44 (sem a abordagem), em comparação a média de 0,63 (com a abordagem 4Experts). Na Tabela 52 é possível visualizar que os percentuais de corretude com o emprego da abordagem 4Experts são superiores quando comparados a não aplicação da abordagem, demonstrando que os participantes apresentaram uma melhora na assertividade das tarefas superior a 38%.

Tabela 54 – Resultado da corretude - teste McNemar.

Teste de McNemar	
Pares(n)	100
X^2	5,4915
p-Valor	0,019

Fonte: Elaborado pelo autor.

Quando analisada a H1-0, tem-se para investigar a hipótese nula, o teste de *McNemar* não paramétrico que pode ser observado na Tabela 53 e Tabela 54. Pode-se verificar que a estatística coletada da significância é 0,019 em uma amostra de 100 pares. Este valor indica que a segunda hipótese nula (H1-0) pode ser rejeitada. Como o valor-p é menor que 0,05, pode-se concluir que há evidências de que a abordagem de recomendação de tarefas é significativamente mais eficaz do que a recomendação manual de tarefas. Analisando a H1-1, são considerados os dados da Tabela 50 e Tabela 53. Neste caso, pode-se concluir que o emprego da abordagem de recomendação de tarefas gerou em média 0,63 (acertos), sendo superior ao processo manual de recomendação de tarefas, o qual apresentou 0,44 (acertos). Os percentuais de corretude apresentados na Tabela 52 confirmam o melhor desempenho da abordagem de recomendação sobre o não uso da abordagem indicando um aproveitamento 38% superior. Portanto, pode-se perceber um aumento relevante na quantidade de participantes que concluíram as tarefas assertivamente permitindo concluir com isso que esta hipótese é válida.

Quando se coloca em escala os resultados da análise de corretude apresentados pela Tabela 47 e Tabela 48, é obtido o gráfico da Figura 35. A Figura evidencia mais nitidamente a diferença na obtenção de tarefas corretas aplicando-se ou não a abordagem 4Experts.

Portanto, para responder a QP1 é possível concluir que o protótipo da abordagem 4Experts afeta a eficácia das tarefas executadas pelos desenvolvedores positivamente, aumentando o número de tarefas respondidas com assertividade.

6.8.2 QP2: As recomendações de tarefas propostas pela abordagem 4Experts reduz o esforço despendido pelos desenvolvedores nas tarefas desempenhadas? (Hipótese 2)

A segunda questão de pesquisa investiga o impacto da abordagem 4Experts, analisando o esforço dos desenvolvedores durante a execução das tarefas de desenvolvimento. Ao verificar os dados contidos na Tabela 51 e Tabela 52 foi diagnosticado que o esforço aplicado pelos participantes para performar as tarefas foi inferior quando aplicada a abordagem de recomendação de tarefas proposta neste estudo em relação a não aplicação da abordagem. A Tabela 52 exibe o esforço médio por tarefa, bem como o registro acumulado de todas as tarefas, o qual apresenta o tempo médio de 1,07 (minutos) quando as tarefas executadas foram recomendadas de forma manual e 0,54 (minutos) quando as tarefas resolvidas foram recomendadas pela abordagem 4Experts. A redução do esforço aplicado a resolução das tarefas pode também ser observada na Tabela 53, na qual é apresentada a diferença de 49,02% no esforço, ou seja, o esforço empregado para atingir o resultado, com o uso da abordagem 4Experts é 49,02% menor do que o registrado com a recomendação manual de tarefas. Este esforço também pode ser observado comparando as medianas de 1,10 (minutos) sem o emprego da abordagem e apenas 0,44 (minutos) para a abordagem 4Experts.

Tabela 55 – Resultado do esforço - teste Wilcoxon.

Teste de Wilcoxon	
Pares(n)	100
Z	-7,0262
p-Valor	<0,00001

Fonte: Elaborado pelo autor.

Analisando a H2-0 tem-se para a hipótese nula o teste de wilcoxon, não paramétrico que pode ser visto na Tabela 53 e Tabela 55. É possível verificar que a estatística coletada da significância é $< 0,00001$, com um intervalo de confiança de 95% em uma amostra de 100 pares.

Para analisar H2-1 são considerados os dados da Tabela 51, Tabela 52 e Tabela 53. Pode-se concluir que o tempo médio para chegar a uma resposta é menor na aplicação da abordagem de recomendação, ficando em 0,54 (minutos), enquanto na recomendação manual o tempo médio para a execução da tarefa foi de 1,07 (minutos). O percentual de esforço é detalhado na Tabela 52, na qual é possível identificar que o esforço na execução das tarefas quando estas não foram submetidas ao sistema de recomendação é superior as tarefas executadas e recomendadas pela abordagem proposta. Com isso, é possível identificar que os participantes ao utilizar a abordagem 4Experts, conseguiram chegar a uma resposta com maior eficácia. Verifica-se nesta hipótese que o esforço para executar as tarefas de desenvolvimento com o sistema de recomendação é 49,02% menor em comparação ao não emprego da abordagem. Com iso, pode-se concluir que essa hipótese é válida e os desenvolvedores aplicam menos esforço com a aplicação da abordagem 4Experts.

Colocando em escala os resultados desta análise do esforço aplicado por tarefa de acordo com a Tabela 47 e Tabela 48 obtém-se o gráfico da Figura 36. A figura evidencia mais nitidamente a diferença de esforço aplicado para a resolução de tarefas entre a aplicação ou não da abordagem de recomendação de tarefas. Pode-se observar que no caso da não aplicação da abordagem, o esforço aplicado é superior, confirmando as discussões anteriores.

Respondendo a QP2, o emprego da abordagem 4Experts na recomendação de tarefas de execução influencia nos esforços investidos pelos desenvolvedores na resolução de tarefas de desenvolvimento, reduzindo o esforço empregado.

6.8.3 Conclusão e Discussão Adicional

Neste experimento foram exploradas situações durante o processo de recomendação de tarefas com o propósito de identificar oportunidades de aperfeiçoamento na abordagem proposta, além de identificar a diferença na quantidade de esforço e no grau de correteza quando empregada a abordagem 4Experts em relação ao emprego de recomendação manual.

Com base nos testes estatísticos executados neste experimento, foi possível responder as questões de pesquisa propostas deste capítulo através da avaliação das hipóteses relativas a correteza e ao esforço. O resultado do experimento proposto neste capítulo é utilizado para responder a QP4 geral deste estudo, detalhada na Seção 1.2. Os resultados apresentados demonstram a rejeição das duas hipóteses nulas e alternativas formuladas para esta validação, contribuindo desta forma para o atendimento dos objetivos específicos detalhados na Seção 1.3.

A partir de uma análise geral é possível concluir, de acordo com os resultados apresentados durante a execução das tarefas, que o protótipo do sistema utilizado para aplicar a abordagem de recomendação de tarefas 4Experts, baseada no uso de dados biométricos do Desenvolvedor aplicados a uma técnica de aprendizado de máquina aumentou o índice de respostas corretas em 38,64%, e reduziu o esforço empregado em 49,02%. Estes resultados geram evidências de um benefício no uso do sistema de recomendação proposto, porém, torna-se necessário aprofundar os estudos investigando a aplicação da abordagem realmente em um ambiente industrial, onde são utilizadas tarefas de maior complexidade, maior tempo de execução e menos controle sobre o ambiente. Por sua vez, a técnica de Aprendizado de Máquina aplicada para a abordagem apresentou uma taxa de acuracidade de aproximadamente 60%. Esta taxa pode ser ampliada, caso o tamanho da base de dados para treinamento for maior. Neste caso, os dados para treinamento foram coletados pelo experimento do Capítulo 4. Além disso, também se sugere a investigação para o uso de outros dados biométricos do desenvolvedor em conjunto com os dados de EEG já empregados e o emprego de outras técnicas de aprendizado de máquina objetivando aumentar a acuracidade do sistema de recomendação.

Todos os participantes do experimento inicialmente foram orientados sobre os procedimentos e cuidados necessários durante a realização do experimento. Este treinamento, além da distribuição de forma homogênea dos participantes de acordo com as suas expertises tornou

possível uma comparação mais assertiva dos dados coletados entre os grupos. O experimento contou com a participação de 20 pessoas, distribuídas em um grupo de controle de 10 participantes e um grupo de avaliação igualmente formado por 10 pessoas. Estes participantes são em sua maioria estudantes do curso de Análise e Desenvolvimento de Sistemas do Centro Universitário UniFTec, enquanto os demais são estudantes do curso de Engenharia da Computação, Redes e Gestão de TI, todos cursando a disciplina de Algoritmo. Para desempenhar uma análise mais detalhada, se sugere um aumento no número de participantes com maiores expertises e um maior número de participantes geral para executar o experimento.

Em relação a validade da conclusão estatística, afirma-se que as diretrizes experimentais foram seguidas com o propósito de eliminar ameaças pressupostas aos testes estatísticos (teste de *Wilcoxon* e o teste de *McNemar*). A escolha dos testes não paramétricos citados foi realizada a partir da análise dos dados colhidos durante o experimento através do teste *Kolmogorov-Smirnov*. Por sua vez, este teste foi utilizado devido a quantidade de amostras a serem utilizadas ser superior a 30 (200 amostras ao todo).

Finalizando, pode-se concluir que a abordagem 4Experts, proposta nesta pesquisa, demonstra através da execução deste experimento melhora na eficiência e efetividade das tarefas realizadas pelos desenvolvedores respondendo positivamente as questões de pesquisa propostas neste capítulo e por consequência a QP4 geral deste estudo.

6.8.4 Ameaças a Validade

Nesta seção são abordadas possíveis limitações e ameaças relacionadas a validade deste experimento. Para que um experimento possa ser realizado de forma válida torna-se necessário verificar a validade dos métodos e participantes selecionados, bem como a forma como estes métodos e participantes foram utilizados durante o processo experimental e os resultados depurados. De acordo com Wohlin et al. (2012), esta análise ocorre através da avaliação de quatro critérios de validade: validade de conclusão, validade da construção e as ameaças internas e externas.

Validade de conclusão. Esta validade considera a capacidade de obter a conclusão correta baseada nos resultados originados pelo experimento, através das escolhas dos métodos estatísticos a partir do tamanho da amostra e a confiabilidade das medidas (WOHLIN et al., 2012). Neste critério o experimento gerou uma amostra de 200 tarefas de desenvolvimento, cuja análise estatística descritiva identificou que os dados não se enquadram em uma distribuição normal, sendo necessário o emprego de testes não paramétricos, ou seja, o teste de *Wilcoxon*, para medir o esforço empregado durante a execução das tarefas e o teste *McNemar*, para avaliar a correção das tarefas executadas, com um intervalo de confiança de 95% conforme apresentado na Seção 6.8.

Validade de construção. Esta ameaça a validade considera a relação entre a teoria e a observação, ou seja, os resultados obtidos através de questionários ou experimentos e possuem

relação com as expectativas da teoria estudada (OLIVEIRA, 2012; FARIAS ET AL., 2015; WOHLIN ET AL., 2012). Assim, o experimento realizado nesta pesquisa foi planejado para mensurar o esforço durante a execução das tarefas, bem como quantificar a corretude das tarefas produzidas pelos participantes, sendo uma estratégia já adotada na literatura como, por exemplo, o trabalho de Bischoff (BISCHOFF, 2018). Os procedimentos de execução e correção do experimento foram cuidadosamente planejados, seguindo as boas práticas de quantificação conforme Wohlin (WOHLIN ET AL., 2012; KITCHENHAM ET AL., 2010; KITCHENHAM; BUDGEN; BRERETON, 2011).

Validade externa. Neste aspecto considera-se as condições que permitem generalizar os resultados do experimento para a prática industrial ou para o mais realista possível (WOHLIN ET AL., 2012; FARIAS ET AL., 2013). Neste contexto foram selecionados participantes com formação adequada ao tipo de experimento executado, ou seja, todos os participantes são estudantes de algum curso de computação (Análise e Desenvolvimento de Sistemas, Engenharia de Software, Redes de Computadores e Gestão de TI) e em alguns casos também atuantes na indústria conforme detalhado na Seção 6.4. Além disso, a linguagem de programação utilizada para os algoritmos é uma linguagem conhecida dos participantes, sendo que todo o experimento ocorreu nas dependências do Centro Universitário UniFTEC. Por fim, apesar dos cuidados observados em relação as ameaças de validade ao estudo, citados nesta Seção, não é possível afirmar que os resultados encontrados podem ser generalizados para modelos maiores, visto que as amostras selecionadas para este experimento são consideradas pequenas, de acordo com o detalhado na Tabela 43.

7 CONCLUSÃO

A recomendação de tarefas de desenvolvimento de software apresenta um considerável grau de subjetividade, sendo baseada na experiência e percepção do responsável pela equipe de desenvolvimento, provocando redução de qualidade e produtividade nos casos em que ela não for assertiva (HUZEFA KAGDI¹, MALCOM GETHERS²; HAMMAD, 2011; KAGDI; POSHY-VANYK, 2009; MA ET AL., 2009). Além de provocar redução de produtividade, a recomendação imprecisa exerce influência sobre o aspecto psicológico dos desenvolvedores (WROBEL, 2013), provocando sentimentos considerados negativos e que também acabam influenciando a qualidade e a produtividade do software desenvolvido (GRAZIOTIN ET AL., 2018). Para que a recomendação possa aproveitar adequadamente a capacidade de cada desenvolvedor e promover melhor eficiência, ela deveria levar em conta as expertises dos desenvolvedores, recomendando tarefas compatíveis as expertises de cada um (MA ET AL., 2009), porém foi identificado através dos trabalhos analisados na Seção 3.2.11, que existem poucas técnicas ou abordagens de recomendação de tarefas que consideram o desenvolvedor como o centro do processo de recomendação de tarefas.

Por conta disso, esta pesquisa foi construída com a proposta de desenvolver uma abordagem para a recomendação de tarefas de Engenharia de Software centralizada nas características do desenvolvedor, que puderam ser identificados com o uso dos próprios dados biométricos, coletados por um equipamento de eletroencefalograma e submetidos a uma técnica de aprendizado de máquina, visando aumentar a corretude e reduzir o esforço dos desenvolvedores durante as atividades de codificação, promovendo desta forma aumento de produtividade.

Para atingir este objetivo, inicialmente foi necessário realizar uma revisão das pesquisas feitas na área, através de um estudo de mapeamento sistemático da literatura que pode ser encontrado no Capítulo 3. Este estudo permitiu identificar o estado da arte no uso de dados biométricos na Engenharia de Software, possibilitando identificar os diferentes tipos de dados biométricos que podem ser obtidos e o seu relacionamento com os aspectos que influenciam o processo de codificação. Também foi possível identificar o que já está sendo realizado pelos sistemas de recomendação em relação a recomendar tarefas de programação. Com estas constatações, optou-se pelo uso dos dados biométricos produzidos pelo cérebro e coletados por um equipamento de eletroencefalograma e analisados com aprendizado de máquina para o sistema de recomendação proposto.

Com base na escolha do tipo de dado biométrico realizada após a análise do Capítulo 3, desenvolveu-se o Capítulo 4 para promover uma análise baseada em evidências, através de um experimento empírico, para identificar as influências que podem ser observadas nos dados do eletroencefalograma enquanto o desenvolvedor está executando tarefas de programação. O resultado deste experimento apresentou evidências que a carga cognitiva do desenvolvedor, calculada conforme apresentado na Seção 4.8, sofre influências de acordo com a expertise que o desenvolvedor possui em relação a um determinado tema, possibilitando concluir que a carga

cognitiva aumenta quando o desenvolvedor está com dificuldades em resolver uma determinada tarefa e reduz conforme as suas expertises forem aumentando. Com este resultado, optou-se em utilizar as informações utilizadas para o cálculo da carga cognitiva como dados de entrada para a técnica de aprendizado de máquina utilizada no 4Experts.

A partir destas escolhas, o Capítulo 5 foi desenvolvido para detalhar o processo de construção da abordagem de recomendação de tarefas 4Experts. Este capítulo descreve o relacionamento entre os diferentes atores e sistemas, detalhando os componentes que precisaram ser desenvolvidos e os detalhes técnicos considerados durante o processo de construção, que resultaram em um protótipo funcional da abordagem empregado no experimento do Capítulo 6 para avaliar os resultados do processo.

7.1 Contribuições

A principal contribuição deste trabalho foi a abordagem 4Experts para recomendar tarefas com o uso de dados biométricos aplicados a técnicas de aprendizagem de máquina. Os resultados obtidos foram subdivididos em quatro diferentes contribuições conforme detalhado na Seção 1.5 do Capítulo 1 e justificadas a seguir:

- **CT1: Uma visão geral do estado da arte sobre o uso de dados biométricos do desenvolvedor na Engenharia de Software.** A análise de literatura realizada a partir da análise inicial de 3.930, após a realização de critérios de filtragem detalhados na Seção 3.1, identificou 40 artigos que tornaram possível classificar os dados biométricos em relação aos seus tipos, forma de aquisição, aplicação e aspectos psicofisiológicos, possibilitando a elaboração de uma taxonomia, conforme detalhado na Seção 3.2.1. A análise destes dados permitiu identificar que a maioria dos estudos primários captura os dados biométricos obtidos pelos olhos, coração, pele e cérebro para analisar emoções e carga cognitiva, a fim de entender o processo de compreensão de código e dificuldade das tarefas. Os sensores associados ao cérebro e aos olhos foram usados principalmente para identificar a carga cognitiva (FRITZ ET AL., 2014; MÜLLER, 2016; SIEGMUND ET AL., 2017; ZÜGER; FRITZ, 2015; CRK; KLUTHE; STEFIK, 2015). Em muitos estudos, esses dois sensores foram usados juntos, produzindo maior precisão nos resultados (FRITZ ET AL., 2014; MÜLLER, 2016; ZÜGER; FRITZ, 2015; LEE ET AL., 2017; PEITEK ET AL., 2018a,b). Para analisar as emoções, foram relacionados sensores a todas as partes do corpo, conforme detalhado pela Figura 8, no entanto, a maioria dos estudos utilizou medidas obtidas pela pele, coração, cérebro e olhos (OSTBERG ET AL., 2017; MÜLLER, 2016; MÜLLER; FRITZ, 2015; ZÜGER; FRITZ, 2015; WROBEL, 2018). Enquanto a carga cognitiva estava relacionada à compreensão e à dificuldade do código (FRITZ ET AL., 2014; SIEGMUND ET AL., 2017; MÜLLER; FRITZ, 2016; FRITZ; MULLER, 2016; NAKAGAWA ET AL., 2014), emoções e ritmo circadiano foram relacionados para interrupção, foco e planejamento de atividades de desenvolvimento (MEYER ET AL., 2017; MÜLLER

LER, 2016; STANKEVIC ET AL., 2011; ZÜGER ET AL., 2018; FRITZ; MULLER, 2016).

Esse resultado sugere que a maioria das pesquisas na área concentram-se na compreensão de código e na dificuldade das tarefas, associados aos dados biométricos obtidos pelo cérebro e pelos olhos, que avaliam a carga cognitiva do indivíduo. A análise das emoções durante as atividades de desenvolvimento também é objetivo da pesquisa, mas com menos intensidade.

A partir dos trabalhos relacionados apresentados na Seção 3.2.11 e analisados de forma comparativa na Seção 3.2.12, foi possível identificar que apenas 33% dos estudos empregam alguma técnica de aprendizado de máquina para realizar a recomendação de tarefas e que ainda há carência de estudos que utilizem dados biométricos para auxiliar na recomendação de tarefas, principalmente dados relacionados às atividades cerebrais associados a atividade cognitiva, a qual influencia as atividades do desenvolvedor, conforme relatado na Seção 3.2.3.

- **CT2: Produção de conhecimento empírico através do uso de dados biométricos dos desenvolvedores.** A partir da execução do experimento controlado para buscar evidências sobre como os dados biométricos do cérebro são influenciados pela execução de tarefas de desenvolvimento, foi possível identificar indícios relatados na Seção 4.9, de que os desenvolvedores com menos experiência em determinadas tarefas de programação apresentam maior carga cognitiva quando executam tarefas específicas se comparados a desenvolvedores mais experientes executando as mesmas tarefas. A maior carga cognitiva, por sua vez, resulta em maior probabilidade de erros de execução das tarefas. Também foi possível observar que mesmo por períodos relativamente curtos é possível que um desenvolvedor adquira conhecimento, melhorando as suas expertises de execução de tarefas de desenvolvimento através da identificação da redução da sua carga cognitiva enquanto desenvolve as tarefas, conforme a sua aprendizagem aumenta.
- **CT3: Uma abordagem de recomendação de tarefas de Engenharia de Software baseada em dados biométricos do desenvolvedor com o emprego de técnicas de aprendizagem de máquina.** Esta contribuição foi obtida através do detalhamento e construção de um protótipo da abordagem 4Experts. O 4Experts apresenta uma abordagem empírica para recomendar tarefas de desenvolvimento de software centrada no desenvolvedor através do emprego de dados biométricos coletados por um equipamento de eletroencefalograma que analisam em tempo real as ondas cerebrais do desenvolvedor, procurando identificar momentos, através do aprendizado de máquina, em que o desenvolvedor cometerá algum erro na tarefa executada. Neste momento o 4Experts irá recomendar uma atividade de menor dificuldade para o desenvolvedor, objetivando promover maior eficiência e eficácia nas atividades realizadas.

- **CT4: Apresentar resultados relativos ao nível de eficácia da abordagem proposta comparando-se a corretude e o esforço na recomendação de tarefas.** Com base nos testes estatísticos executados na Seção 6.8 é possível concluir que o protótipo do 4Experts, aumentou o índice de respostas corretas em 38,64%, e reduziu o esforço empregado em 49,02%. Estes resultados geram evidências de um benefício no uso do sistema de recomendação proposto. Em relação a validade desta conclusão estatística, afirma-se que as diretrizes experimentais foram seguidas com o propósito de eliminar ameaças pressupostas aos testes estatísticos (teste de *Wilcoxon* e o teste de *McNemar*), garantindo desta forma a validade destes resultados.

7.2 Limitações do Trabalho

Apesar das contribuições listadas na Seção 7.1 apresentarem resultados positivos e todos os processos empregados para a execução deste trabalho serem condicionados a uma análise de ameaças a validade conforme observado nas Seções 3.4, 4.11 e 6.8.4, estes resultados podem apresentar limitações por conta do experimento executado e o protótipo construído.

Limitações Experimentais. Existe a necessidade de ampliar o número de participantes e também utilizar participantes com diferentes expertises e com maior participação na indústria, pois os participantes utilizados no experimento são alunos dos anos iniciais de cursos de graduação e que não possuem muita experiência na indústria. Além disso, as questões utilizadas no experimento também podem focar questões com maior duração de tempo e estarem relacionadas a tarefas performadas na indústria.

Limitações de Protótipo. Apesar do protótipo desenvolvido seguir todo o processo e componentes detalhados no Capítulo 5, ele não apresenta todos os detalhes de construção do Plugin e Aplicativo Web necessários para que o protótipo possa ser utilizado em um ambiente real, na indústria, necessitando ainda de alguma evolução para isso se tornar viável.

Conforme apresentado, estas limitações não implicam em desconsiderar os resultados obtidos, mas converter estas limitações em estudos futuros que possam dar continuidade a este projeto, promovendo uma ampliação do mesmo.

7.3 Trabalhos Futuros

Os resultados apresentados pelo 4Experts demonstraram que uma abordagem de recomendação de tarefas baseada em dados biométricos e com o auxílio de aprendizado de máquina gera resultados positivos, proporcionando melhor qualidade e eficiência no desempenho de tarefas de codificação pelos desenvolvedores. Porém, a partir do processo empregado para chegar nestes resultados, foi possível identificar melhorias no processo que podem resultar em trabalhos futuros, como: (1) pesquisar e utilizar outras técnicas de aprendizado além do Multilayer Perceptron, como a Deep Learning, para gerar a predição do resultado das tarefas. Estes técnicas

podem resultar em acuracidade superior a técnica empregada, melhorando ainda mais os resultados obtidos, (2) estender a abordagem para que ela analise outros dados biométricos além dos produzidos pelo cérebro com o propósito de ampliar a acuracidade dos resultados, (3) passar a considerar outros aspectos psicofisiológicos além da carga cognitiva, como as emoções, que são amplamente debatidas por Wrobel (WROBEL, 2018) como influenciadoras do processo de desenvolvimento e por isso, poderiam ser relacionadas a recomendação de tarefas, (4) acrescentar dados de contexto ao mecanismo de recomendação. Os dados de contexto dizem respeito a situações como o ambiente e local onde o desenvolvedor se encontra e a sua influência em relação a execução das suas tarefas, (5) passar a recomendar outras tarefas comuns a Engenharia de Software (MEYER ET AL., 2014), além das atividades de codificação, como reuniões, atividades de pesquisa, interações com clientes e até mesmo intervalos e, (6) realizar experimentos na indústria, utilizando tarefas que usualmente seriam realizadas neste meio, com o propósito de provar a eficácia da abordagem nestes ambientes.

REFERÊNCIAS

- A diary study of task switching and interruptions. **Conference on Human Factors in Computing Systems - Proceedings**, [S.l.], v. 6, n. 1, p. 175–182, 2004.
- AGHAJANI, E. et al. Software Documentation Issues Unveiled. **Proceedings - International Conference on Software Engineering**, [S.l.], v. 2019-May, p. 1199–1210, 2019.
- ALI, N. B.; USMAN, M. Reliability of search in systematic reviews: towards a quality assessment framework for the automated-search strategy. **Information and Software Technology**, [S.l.], v. 99, n. February, p. 133–147, 2018.
- ALI, N. B.; USMAN, M. Reliability of search in systematic reviews: towards a quality assessment framework for the automated-search strategy. **Information and Software Technology**, [S.l.], v. 99, p. 133 – 147, 2018.
- ANDERSON, E. W. et al. A user study of visualization effectiveness using eeg and cognitive load. **Computer Graphics Forum**, [S.l.], v. 30, n. 3, p. 791–800, 2011.
- ANVIK, J.; HIEW, L.; MURPHY, G. C. Who should fix this bug? **International Conference on Software Engineering**, [S.l.], p. 361–370, 2006.
- ANVIK, J.; MURPHY, G. C. Reducing the effort of bug report triage: recommenders for development-oriented decisions. **ACM Transactions on Software Engineering and Methodology**, [S.l.], v. 20, n. 3, 2011.
- BABADI, B.; BROWN, E. N. A review of multitaper spectral analysis. **IEEE Transactions on Biomedical Engineering**, [S.l.], v. 61, n. 5, p. 1555–1564, May 2014.
- BABLANI, A.; EDLA, R. Survey on Brain-Computer Interface: an emerging computational intelligence paradigm. **ACM Computing Surveys**, [S.l.], v. 52, n. 1, 2019.
- BARROS, E.; MAZUCHELI, J. Um estudo sobre o tamanho e poder dos testes t-student e wilcoxon. **Acta Scientiarum-technology - ACTA SCI-TECHNOL**, [S.l.], v. 27, p. 23–32, 03 2005.
- BEDNARIK, R.; TUKIAINEN, M. An eye-tracking methodology for characterizing program comprehension processes. **Proceedings of the 2006 symposium on Eye tracking research & applications - ETRA '06**, [S.l.], p. 125, 2006.
- BEGEL, A. Invited Talk : fun with software developers and biometrics. , [S.l.], p. 5–6, 2016.
- BISCHOFF, V. **Fmit**: feature model integration techniques. 2018. 147 p. Dissertação de Mestrado do Programa de Pós-Graduação em Ciência da Computação — Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, 2018.
- BISCHOFF, V. et al. Integration of feature models: a systematic mapping study. **Information and Software Technology**, [S.l.], v. 105, p. 209 – 225, 2019.
- BROUWER, A. M. et al. Estimating workload using EEG spectral power and ERPs in the n-back task. **Journal of Neural Engineering**, [S.l.], v. 9, n. 4, 2012.

BUSJAHN, T. et al. Eye Movements in Code Reading: relaxing the linear order. **IEEE International Conference on Program Comprehension**, [S.l.], v. 2015-Augus, p. 255–265, 2015.

BUZSÁKI, G. **Rhythms of the brain**. [S.l.: s.n.], 2009. xiv, 448 p. p.

CACIOPPO, J. T.; TASSINARY, L. G.; BERNTSON, G. G. Handbook of psychophysiology, fourth edition. **Handbook of Psychophysiology, Fourth Edition**, [S.l.], p. 1–716, 2016.

CALIFORNIA, U. of. **Eeg lab**. 2004.

CAMPBELL, D. T.; RUSSO, M. J. **Social experimentation**. [S.l.]: Sage Publications, Inc, 1999. v. 1.

CASTELHANO, J. et al. The role of the insula in intuitive expert bug detection in computer code: an fmri study. **Brain Imaging and Behavior**, [S.l.], n. CompTIA 2016, p. 1–15, 2018.

COOPER, I. What is a “mapping study?”. **Journal of the Medical Library Association : JMLA**, [S.l.], v. 104, p. 76–78, 01 2016.

Crk, I.; Kluthe, T. Toward using alpha and theta brain waves to quantify programmer expertise. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p. 5373–5376.

CRK, I.; KLUTHE, T.; STEFIK, A. Understanding Programming Expertise. **ACM Transactions on Computer-Human Interaction**, [S.l.], v. 23, n. 1, p. 1–29, 2015.

DAS, R. et al. Cognitive load measurement - A methodology to compare low cost commercial EEG devices. **Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014**, [S.l.], p. 1188–1194, 2014.

DEVELOPERS, M. **Mne**. 2012.

DEVELOPERS, P. **Pandas**. 2001.

DEVELOPERS, S. **Scipy**. 2001.

DIETTERICH, T. G. **Machine learning**. GBR: John Wiley and Sons Ltd., 2003. 1056–1059 p.

DURAES, J. et al. WAP: understanding the brain at software debugging. **Proceedings - International Symposium on Software Reliability Engineering, ISSRE**, [S.l.], p. 87–92, 2016.

EMOTIV. **Emotiv epoc+**. 2011.

EMOTIV. **Emotiv epoc+**. 2011.

FAKHOURY, S. The Effect of Poor Source Code Lexicon and Readability on Developers’s Cognitive Load. , [S.l.], p. 286–296, 2018.

FARIAS, K. et al. Analyzing the effort of composing design models of large-scale software in industrial case studies. In: **MODEL-DRIVEN ENGINEERING LANGUAGES AND SYSTEMS**, 2013, Berlin, Heidelberg. **Anais...** Springer Berlin Heidelberg, 2013. p. 639–655.

FARIAS, K. et al. Toward an architecture for model composition techniques. In: **INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING**, 27., 2015, Pittsburgh, USA. **Proceedings...** [S.l.: s.n.], 2015. p. 656–659.

Farias, K. et al. Uml2merge: a uml extension for model merging. **IET Software**, [S.l.], v. 13, n. 6, p. 575–586, 2019.

FEIGENSPAN, J. et al. Measuring programming experience. **2012 20th IEEE International Conference on Program Comprehension (ICPC)**, [S.l.], v. 2005, p. 73–82, 2012.

FERNÁNDEZ-SÁEZ, A. M.; GENERO, M.; CHAUDRON, M. R. Empirical studies concerning the maintenance of uml diagrams and their use in the maintenance of code: a systematic mapping study. **Information and Software Technology**, [S.l.], v. 55, n. 7, p. 1119–1142, 2013.

FIRMINO, M. J. d. A. C. d. S. **Testes de hipóteses: uma abordagem não paramétrica**. 2015. 107 p. Tese de mestrado, Matemática para Professores, Universidade de Lisboa, Faculdade de Ciências, 2015 — Universidade de Lisboa, Lisboa, 2015.

FLOYD, B.; SANTANDER, T.; WEIMER, W. Decoding the Representation of Code in the Brain: an fmri study of code review and expertise. **Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017**, [S.l.], p. 175–186, 2017.

FRITZ, T. et al. Using psycho-physiological measures to assess task difficulty in software development. **Proceedings of the 36th International Conference on Software Engineering - ICSE 2014**, [S.l.], p. 402–413, 2014.

FRITZ, T.; MULLER, S. C. Leveraging Biometric Data to Boost Software Developer Productivity. **2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)**, [S.l.], p. 66–77, 2016.

FUNDATION, P. S. **Python**. 2001.

GARDNER, M. W.; DORLING, S. R. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. **Atmospheric Environment**, [S.l.], v. 32, n. 14-15, p. 2627–2636, 1998.

GASPARIC, M.; JANES, A. What recommendation systems for software engineering recommend: a systematic literature review. **Journal of Systems and Software**, [S.l.], v. 113, p. 101 – 113, 2016.

GIANNAKAKIS, G. et al. Stress and anxiety detection using facial cues from videos. **Biomedical Signal Processing and Control**, [S.l.], v. 31, p. 89–101, 2017.

GLÜCKER, H. et al. EyeDE. **Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14**, [S.l.], p. 1555–1560, 2014.

GONÇALVES, M.; SOUZA, L. de; GONZÁLEZ, V. Collaboration, Information Seeking and Communication: an observational study of software developers' work practices. **J. Ucs**, [S.l.], v. 17, n. 14, p. 1913–1930, 2011.

GONÇALES, L. et al. Measuring the cognitive load of software developers: a systematic mapping study. In: INTERNATIONAL CONFERENCE ON PROGRAM COMPREHENSION, 27., 2019, Piscataway, NJ, USA. **Proceedings...** IEEE Press, 2019. p. 42–52. (ICPC '19).

GONÇALES, L. J. et al. Comparison of design models: a systematic mapping study. **International Journal of Software Engineering and Knowledge Engineering**, [S.l.], v. 25, n. 09n10, p. 1765–1769, 2015.

Gopejenko, V.; Mihailovs, I. Neurocomputer interface, the classification of the brain waves and virtual reality control. In: INTERNATIONAL CONFERENCE ON APPLICATION OF INFORMATION AND COMMUNICATION TECHNOLOGIES (AICT), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 433–436.

GRAZIOTIN, D. et al. What happens when software developers are (un)happy. **Journal of Systems and Software**, [S.l.], v. 140, p. 32–47, 2018.

H Ali, S.; El Desouky, A. I. A New Profile Learning Model for Recommendation System based on Machine Learning Technique. **Journal of Information Technology & Software Engineering**, [S.l.], v. 06, n. 01, p. 1–6, 2016.

HAUMER, P. Eclipse Process Framework Composer Part 1. **IBM Rational Software**, [S.l.], n. 2nd Revision, p. 1–25, 2007.

HIPPE, Z. S. et al. Human-Computer Systems Interaction: backgrounds and applications 3. **Advances in Intelligent Systems and Computing**, [S.l.], v. 300, n. July, 2014.

Huzefa Kagdi¹, Malcom Gethers², D. P.; HAMMAD, M. Assigning change requests to software developers. **Journal of Software: Evolution and Process**, [S.l.], v. 26, n. 12, p. 1172–1192, 2011.

INC., M. **Matlab**. 1994.

JIN, C. Y.; BORST, J. P.; VUGT, M. K. van. Predicting task-general mind-wandering with EEG. **Cognitive, Affective and Behavioral Neuroscience**, [S.l.], 2019.

JÚNIOR, M. L. d. L. et al. Automatic assignment of integrators to pull requests: the importance of selecting appropriate attributes. **Journal of Systems and Software**, [S.l.], v. 144, n. May 2017, p. 181–196, 2018.

Kagdi, H.; Poshyvanyk, D. Who can help me with this change request? In: IEEE 17TH INTERNATIONAL CONFERENCE ON PROGRAM COMPREHENSION, 2009., 2009. **Anais...** [S.l.: s.n.], 2009. p. 273–277.

KEVIC, K. et al. Tracing software developers' eyes and interactions for change tasks. **Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2015**, [S.l.], p. 202–213, 2015.

KEVIC, K. et al. Eye gaze and interaction contexts for change tasks – Observations and potential. **Journal of Systems and Software**, [S.l.], v. 128, p. 252–266, jun 2017.

KITCHENHAM, B. A.; BUDGEN, D.; BRERETON, O. P. Using mapping studies as the basis for further research – a participant-observer case study. **Information and Software Technology**, [S.l.], v. 53, n. 6, p. 638 – 651, 2011. Special Section: Best papers from the APSEC.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering - a tertiary study. **Inf. Softw. Technol.**, USA, v. 52, n. 8, p. 792–805, Aug. 2010.

KLIMESCH, W. Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. **Brain Research Reviews**, [S.l.], v. 29, n. 2, p. 169 – 195, 1999.

KONOPKA, M. Combining eye tracking with navigation paths for identification of cross-language code dependencies. **Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2015**, [S.l.], p. 1057–1059, 2015.

KRAMER, A. E. **Physiological metrics of mental workload**: a review of recent progress. 1990.

LANG, J.-P. **Redmine**. 2006.

Latoza, Thomas, D.; VENOLIA, G.; DELINE, R. Maintaining mental models: a study of developer work habits. **Proceedings of the 28th International Conference on Software Engineering**, [S.l.], p. 492–501, 2006.

LAU, J.; IOANNIDIS, J. P.; SCHMID, C. H. Quantitative synthesis in systematic reviews. **Annals of internal medicine**, [S.l.], v. 127, n. 9, p. 820–826, 1997.

LEE, S. et al. Mining biometric data to predict programmer expertise and task difficulty. **Cluster Computing**, [S.l.], n. January, p. 1–11, 2017.

LTD, H. S. **Free virtual port**. 2015.

Ma, D. et al. Expert recommendation with usage expertise. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE, 2009., 2009. **Anais...** [S.l.: s.n.], 2009. p. 535–538.

MAALEJ, W.; HAPPEL, H.-J. From work to word: how do software developers describe their work? In: IEEE INTERNATIONAL WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES, 2009., 2009. **Anais...** IEEE, 2009. p. 121–130.

MANHARTSBERGER, M.; ZELLHOFER, N. Eye tracking in usability research: what users really see. **Usability Symposium**, [S.l.], v. 198, p. 141–152, 2005.

MARK, G.; GONZALEZ, V. M.; HARRIS, J. No task left behind? Examining the nature of fragmented work. **CHI 2005: Technology, Safety, Community: Conference Proceedings - Conference on Human Factors in Computing Systems**, [S.l.], p. 321–330, 2005.

MATHÔT, S. **Open sesame**. 2010.

MATTER, D.; KUHN, A.; NIERSTRASZ, O. Assigning bug reports using a vocabulary-based expertise model of developers. **Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR 2009**, [S.l.], p. 131–140, 2009.

MECARELLI, O. **Clinical electroencephalography**. [S.l.]: Springer International Publishing, 2019.

MEYER, A. et al. Software Developers' Perceptions of Productivity. , [S.l.], 2014.

MEYER, A. N. et al. The Work Life of Developers: activities, switches and perceived productivity. **IEEE Transactions on Software Engineering**, [S.l.], v. 43, n. 12, p. 1178–1193, dec 2017.

MEYJES, F.; ARTS, R. Recommendations for the practice of clinical neurophysiology. **Journal of the Neurological Sciences**, [S.l.], v. 66, n. 2-3, p. 349, 1984.

MICROSOFT. **Visual studio code**. 2012.

MINGUILLON, J.; LOPEZ-GORDO, M. A.; PELAYO, F. Trends in eeg-bci for daily-life: requirements for artifact removal. **Biomedical Signal Processing and Control**, [S.l.], v. 31, p. 407 – 418, 2017.

MOHD RAZALI, N.; YAP, B. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. **J. Stat. Model. Analytics**, [S.l.], v. 2, p. 21–33, 01 2011.

MÜLLER, S. C.; FRITZ, T. Stuck and frustrated or in flow and happy: sensing developers' emotions and progress. **Proceedings - International Conference on Software Engineering**, [S.l.], v. 1, n. May, p. 688–699, 2015.

MÜLLER, S. C.; FRITZ, T. Using (bio)metrics to predict code quality online. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING - ICSE '16, 38., 2016, New York, New York, USA. **Proceedings...** ACM Press, 2016. p. 452–463.

Müller, S. Using biometric sensors to increase developers' productivity. , [S.l.], 01 2016.

NAKAGAWA, T. et al. Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: a controlled experiment. **Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014**, [S.l.], p. 448–451, 2014.

OLIVEIRA, K. S. F. D. **Empirical evaluation of effort on composing design models**. 2012. 281 p. Tese (Doutorado em Ciência da Computação) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2012.

OpenSesame: an open-source, graphical experiment builder for the social sciences. **Behavior Research Methods**, [S.l.], v. 44, n. 2, p. 314–324, 2012.

OSTBERG, J.-P. et al. Towards the assessment of stress and emotional responses of a salutogenesis-enhanced software tool using psychophysiological measurements. **Proceedings - 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering, SEmotion 2017**, [S.l.], n. SEmotion, p. 2–5, 2017.

- PALMER, C.; SHARIF, B. Towards automating fixation correction for source code. **Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16**, [S.l.], p. 65–68, 2016.
- PEITEK, N. et al. Toward conjoint analysis of simultaneous eye-tracking and fmri data for program-comprehension studies. **Proceedings of the Workshop on Eye Movements in Programming - EMIP '18**, [S.l.], p. 1–5, 2018.
- PEITEK, N. et al. Simultaneous Measurement of Program Comprehension with fMRI and Eye Tracking : a case study. , [S.l.], 2018.
- PEITEK SIEGMUND, A. A Look into Programmers' Heads. **IEEE Transactions on Software Engineering**, [S.l.], p. 1–1, 2018.
- PERRY, D.; STAUDENMAYER, N.; VOTTA, L. People, organizations, and process improvement. **IEEE Software**, [S.l.], v. 11, n. 4, p. 36–45, 2002.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: EASE, 2008. **Anais...** [S.l.: s.n.], 2008. v. 8, p. 68–77.
- PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: an update. **Information and Software Technology**, [S.l.], v. 64, p. 1–18, 2015.
- RAMADAN, R. A.; VASILAKOS, A. V. Brain computer interface: control signals review. **Neurocomputing**, [S.l.], v. 223, p. 26 – 44, 2017.
- ROBILLARD, M. P.; WALKER, R.; ZIMMERMANN, T. Recommendation Systems for Software Engineering. **IEEE Software**, [S.l.], v. 27, n. 4, p. 80–86, 2010.
- ROBILLARD, P. N. The role of knowledge in software development. **Communications of the ACM**, [S.l.], v. 42, n. 1, p. 87–92, 1999.
- RODEGHERO, P. et al. Improving automated source code summarization via an eye-tracking study of programmers. **Proceedings of the 36th International Conference on Software Engineering - ICSE 2014**, [S.l.], p. 390–401, 2014.
- RODRÍGUEZ, P. et al. Continuous deployment of software intensive products and services: a systematic mapping study. **Journal of Systems and Software**, [S.l.], v. 123, p. 263–291, 2017.
- SANCHEZ, H.; ROBBES, R.; GONZALEZ, V. M. An empirical study of work fragmentation in software evolution tasks. **2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering, SANER 2015 - Proceedings**, [S.l.], n. Dcc, p. 251–260, 2015.
- SANTOS, C. **Estatística descritiva**. [S.l.: s.n.], 2007. v. 2.
- SHARIF, B.; MALETIC, J. I. An eye tracking study on camelcase and under-score identifier styles. **IEEE International Conference on Program Comprehension**, [S.l.], p. 196–205, 2010.
- SIEGMUND, J. Program Comprehension: past, present, and future. **2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)**, [S.l.], p. 13–20, 2016.

SIEGMUND, J. et al. Toward Measuring Program Comprehension with Functional Magnetic Resonance Imaging. **Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering**, [S.l.], p. 24:1–24:4, 2012.

SIEGMUND, J. et al. Understanding understanding source code with functional magnetic resonance imaging. **Proceedings of the 36th International Conference on Software Engineering - ICSE 2014**, [S.l.], p. 378–389, 2014.

SIEGMUND, J. et al. Measuring neural efficiency of program comprehension. **Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2017**, [S.l.], v. 11, p. 140–150, 2017.

STANKEVIC, T. et al. Engineering Applications of Artificial Intelligence Web-based Biometric Computer Mouse Advisory System to Analyze a User ' s Emotions and Work Productivity. , [S.l.], v. 24, p. 928–945, 2011.

SWARTZ, B. E. The advantages of digital over analog recording techniques. **Electroencephalography and Clinical Neurophysiology**, [S.l.], v. 106, n. 2, p. 113 – 117, 1998.

SWELLER, J. Cognitive load during problem solving: effects on learning. **Cognitive Science**, [S.l.], v. 12, n. 2, p. 257–285, 1988.

SWELLER, J. Cognitive load theory, learning difficulty, and instructional design. **Learning and Instruction**, [S.l.], v. 4, n. 4, p. 295 – 312, 1994.

VU, M. A. T. et al. A shared vision for machine learning in neuroscience. **Journal of Neuroscience**, [S.l.], v. 38, n. 7, p. 1601–1607, 2018.

WALDERT, S. Invasive vs. non-invasive neuronal signals for brain-machine interfaces: will one prevail? **Frontiers in Neuroscience**, [S.l.], v. 10, p. 295, 2016.

WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. **Requirements Engineering**, [S.l.], v. 11, n. 1, p. 102–107, 2006.

WOHLIN, C. et al. **Experimentation in software engineering**. [S.l.]: Springer Publishing Company, Incorporated, 2012.

WROBEL, M. Applicability of Emotion Recognition and Induction Methods to Study the Behavior of Programmers. **Applied Sciences**, [S.l.], v. 8, n. 3, p. 323, feb 2018.

WROBEL, M. R. Emotions in the software development process. **2013 6th International Conference on Human System Interactions (HSI)**, [S.l.], n. June 2013, p. 518–523, 2013.

WU, W. et al. DREX: developer recommendation with k-nearest-neighbor search and expertise ranking. **Asia-Pacific Software Engineering Conference, APSEC**, [S.l.], p. 389–396, 2011.

YU, X. et al. Cross-domain developer recommendation algorithm based on feature matching. In: **COMPUTER SUPPORTED COOPERATIVE WORK AND SOCIAL COMPUTING**, 2019, Singapore. **Anais...** Springer Singapore, 2019. p. 443–457.

ZHANG, T.; LEE, B. How to recommend appropriate developers for bug fixing? **Proceedings - International Computer Software and Applications Conference**, [S.l.], p. 170–175, 2012.

ZÜGER, M. et al. Sensing Interruptibility in the Office: a field study on the use of biometric and computer interaction sensors. **Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18**, [S.l.], p. 1–14, 2018.

ZÜGER, M.; FRITZ, T. Interruptibility of Software Developers and its Prediction Using Psycho-Physiological Sensors. In: ANNUAL ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS - CHI '15, 33., 2015, New York, New York, USA. **Proceedings...** ACM Press, 2015. p. 2981–2990.

APÊNDICE A – INFORMAÇÕES COMPLEMENTARES

A.1 Termo de Consentimento

Termo de Consentimento Livre e Esclarecido

Eu, _____ estou sendo convidado a fazer parte de um estudo intitulado de: **técnica para avaliar a aquisição de conhecimento com o uso de EEG**, cujos objetivos e justificativas são: **identificar padrões de carga cognitiva em diferentes níveis de complexidade a fim entender o processo de aquisição de conhecimento dos desenvolvedores em suas atividades de programação**. A minha contribuição para este estudo é a de: **observar, interpretar e responder perguntas referentes a atividades de programação e realizar pequenas tarefas de programação que serão capturados pelo equipamento de EEG**.

Fui informado que posso me recusar de participar do estudo, ou até de retirar meu consentimento a qualquer momento durante o estudo, sem precisar justificar ou dar esclarecimentos.

Foram me repassados todos os esclarecimentos necessários sobre quaisquer possíveis desconfortos e riscos em relação a este estudo, levando-se em conta que é uma pesquisa, e os resultados positivos ou negativos somente serão obtidos após a sua realização. Assim, **desconfortos devido a utilização do EPOC+ da marca Emotiv podem ser notados após longo tempo de exposição, sem nenhum outro risco envolvido**. Também estou ciente de que minha privacidade será respeitada, ou seja, meu nome ou qualquer outro dado ou elemento que possa, de qualquer forma, me identificar, será mantido de forma confidencial. De qualquer forma, caso qualquer tipo de dano decorrente da minha participação no estudo, conforme determina a lei, eu deverei ser indenizado.

Os pesquisadores que participam deste projeto são: **o mestrando Juliano Paulo Menzen, o doutorando Lucian José Gonçalves e o orientador Kleinner Farias, todos da Universidade do Vale do Rio do Sinos (Unisinos)**. Eu poderei manter contato com o mestrando através do telefone **+55 54 99174-2009** ou simplesmente pelo e-mail **juliano.menzen@gmail.com**.

O livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências é assegurada a assistência durante toda a pesquisa, tendo sido também orientado quanto ao teor de todo o aqui mencionado e compreendido a natureza e o objetivo do já referido estudo, exponho meu livre consentimento em fazer parte, estando completamente ciente de que não há nenhum valor econômico, a receber ou a pagar, por minha participação, ou seja, não irá ocorrer nenhum tipo de benefícios, lícito ou ilícito, a não ser os resultados que o trabalho do mestrando venha a atingir no fim deste estudo.

Recebi a informação que em caso de reclamação, abuso, ou denúncia sobre este estudo devo mandar e-mail para **pipca@unisinos.br** ou ligar para o PIPCA **+55 51 93590-8161**.

Participant Acceptance:

() Eu concordo em fazer parte deste estudo como participante.

Participant Name:

Date: / /

A.2 Questionário de Pesquisa

Questionário Demográfico

Tempo: _____ Hora de Início: ____:____ Hora de Fim: ____:____

Id:		Nome:			
	Idade:		Profissão:		
	Cargo Atual:			Empresa:	
	Quanto tempo estás no cargo atual?			Sexo:	<input type="checkbox"/> M. <input type="checkbox"/> F. <input type="checkbox"/> Outro

1) Qual é o seu maior grau de escolaridade ?	<input type="checkbox"/>	Técnico
	<input type="checkbox"/>	Graduação
	<input type="checkbox"/>	Mestrado
	<input type="checkbox"/>	Doutorado
	Outro: Qual ?	

2) Por quanto tempo você estudou (tem estudado) em universidade/faculdade?	<input type="checkbox"/>	Nunca estudei
	<input type="checkbox"/>	Menos de 2 anos
	<input type="checkbox"/>	De 2 a 4 anos
	<input type="checkbox"/>	De 5 a 6 anos
	<input type="checkbox"/>	De 7 a 8 anos
	<input type="checkbox"/>	Mais de 8 anos

3) Qual é a sua formação acadêmica ? (em andamento ou concluída)	<input type="checkbox"/>	Sistemas de Informação
	<input type="checkbox"/>	Ciência da Computação
	<input type="checkbox"/>	Engenharia de Sistemas
	<input type="checkbox"/>	Análise e Desenvolvimento de Sistemas
	<input type="checkbox"/>	Sistemas para Internet
	<input type="checkbox"/>	Engenharia da Computação
	Outro: Qual?	

4) Em uma escala de 1 a 10 como você estima o seu conhecimento em programação (1 – Muito inexperiente 10 – Muito experiente)	Escala de Likert
	1[] 2[] 3[] 4[] 5[] 6[] 7[] 8[] 9[] 10[]

5) Como você estima sua experiência de programação comparada com especialistas com 10 ou mais anos de experiência (1 – Muito inexperiente 5 – Muito experiente)	Escala de Likert
	1[] 2[] 3[] 4[] 5[]
	Escala de Likert

6) Como você estima sua experiência de programação comparada com os seus colegas. (1 – Muito inexperiente 5 – Muito experiente)	1[<input type="checkbox"/>] 2[<input type="checkbox"/>] 3[<input type="checkbox"/>] 4[<input type="checkbox"/>] 5[<input type="checkbox"/>]
--	---

7) Qual a sua experiência nas seguintes línguas de programação? (1 – Muito inexperiente 5 – Muito experiente)	Item	Escala de Likert				
	JavaScript	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Linguagem R	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Python	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	SQL (ANSI)	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Linguagem C	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Linguagem C#	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Linguagem Java	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Lógica de Programação	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
Língua Inglesa	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]	

8) Quantas línguas de programação você conhece além das apresentadas até o momento (Apenas marcar se o conhecimento for mediano em diante).	Número de línguas: _____
--	--------------------------

9) Qual a sua experiência nos seguintes paradigmas de programação? (1 – Muito inexperiente 5 – Muito experiente)	Item	Escala de Likert				
	Funcional	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Imperativa	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Lógica	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]
	Orientação a Objetos	1 [<input type="checkbox"/>]	2 [<input type="checkbox"/>]	3 [<input type="checkbox"/>]	4 [<input type="checkbox"/>]	5 [<input type="checkbox"/>]

10) Por quantos anos você programa em alguma língua de programação?	[<input type="checkbox"/>]	Até 1 anos
	[<input type="checkbox"/>]	De 1 a 2 anos
	[<input type="checkbox"/>]	De 2 a 4 anos
	[<input type="checkbox"/>]	De 5 a 6 anos
	[<input type="checkbox"/>]	De 7 a 8 anos
	[<input type="checkbox"/>]	Mais de 8 anos

11) Por quantos anos você programa profissionalmente em alguma linguagem de programação ?	<input type="checkbox"/>	Até 1 anos
	<input type="checkbox"/>	De 1 a 2 anos
	<input type="checkbox"/>	De 2 a 4 anos
	<input type="checkbox"/>	De 5 a 6 anos
	<input type="checkbox"/>	De 7 a 8 anos
	<input type="checkbox"/>	Mais de 8 anos

12) Qual o tamanho médio dos projetos de software que você trabalha profissionalmente ? (em linhas de código)	<input type="checkbox"/>	Não sei
	<input type="checkbox"/>	Até 900 linhas
	<input type="checkbox"/>	Entre 900 e 40.000 linhas
	<input type="checkbox"/>	Acima de 40.000 linhas

Este questionário tem como objetivo avaliar e prover dados para a implementação do sistema proposto pela dissertação do Juliano Paulo Menzen. Desse modo, as respostas para as questões devem ser baseadas na experiência do participante. O questionário possui duas partes, uma para caracterizar o participante e a outra para coletar as informações. Os participantes não estão sendo avaliados e seus dados não serão divulgados. As questões pessoais acima servem somente para agrupar os dados de acordo com o conhecimento adquirido na compreensão de artefatos de software.

Muito obrigado pela sua participação!

A.3 Questões Aplicadas no Experimento

Questões propostas para o experimento

Questão 1:

```
1 Algoritmo questao
2 var
3 varA: inteiro
4 inicio
5     varA <- 4 * 2 + 6 / 2
6     escreva(varA)
7 fimalgoritmo
```

Opções de resposta:

a) 11 b) 7 c) 16 d) 1 e) 12

Questão 2:

```
1 Algoritmo questao
2 var
3 varA, varRes: inteiro
4 inicio
5     varA <- 125
6     se varA > 200 entao
7         varRes <- 1
8     senao
9         varRes <- varA
10    fimse
11    escreva(varRes)
12 fimalgoritmo
```

Opções de resposta:

a) 1 b) 484 c) 2 d) 125 e) 9

Questão 3:

```
1 Algoritmo questao
2 var
3 varB: inteiro
4 inicio
5     varB <- 3
6     enquanto varB < 10
7         varB <- varB * 2
8     fimenquanto
9     escreva(varB)
10 fimalgoritmo
```

Opções de resposta:

a) 11 b) 3 c) 10 d) 5 e) 12

Questão 4:

```
1 Algoritmo questao
2 var
3 varA, varB, varF, varH: inteiro
4 inicio
5     varA <- 4
6     varB <- 2
7     varF <- 15
8     varH <- 3
9     se (varB < varA) AND (varB > varH) entao
10         varF <- varB
11     senao
12         varF <- varH
13     fimse
14     escreva (varF)
15 fimalgoritmo
```

Opções de resposta:

- a) 4 b) 2 c) 0 d) 15 e) 3

Questão 5:

```
1 Algoritmo questao
2 var
3 f, sal, b: real
4 inicio
5     f <- 2
6     sal <- 1000
7     se (f > 0) AND (f <= 3) entao
8         b <- sal * 0.35
9     senao
10        b <- sal * 0.15
11    fimse
12    b <- b + 15
13    escreva (b)
14 fimalgoritmo
```

Opções de resposta:

- a) 150 b) 365 c) 15 d) 165 e) 350

Questão 6:

```
1 Algoritmo questao
2 var
3 varA: inteiro
4 inicio
5     varA <- 10
6     enquanto varA > 2
7         varA <- varA - 2
8     fimenquanto
9     escreva(varA)
10 fimalgoritmo
```

Opções de resposta:

- a) 0 b) -1 c) 2 d) 10 e) 3

Questão 7:

```
1 Algoritmo questao
2 var
3 varA, varT: inteiro
4 inicio
5     varA <- 1
6     enquanto varA < 5
7         varA <- varA + 1
8         se (varA > 1) AND (varA < 2) entao
9             varT <- 0
10        senao
11            varT <- varA
12        fimse
13    fimenquanto
14    escreva(varT)
15 fimalgoritmo
```

Opções de resposta:

- a) 10 b) 5 c) 4 d) 1 e) 0

Questão 8:

```
1 Algoritmo questao
2 var
3 varA, varT: inteiro
4 inicio
5     varA <- 1
6     enquanto varA < 5
7         varT <- varA * 2
8         varT <- varT / 2 + 5
9         varA <- varA + 1
10    fimenquanto
11    escreva(varT)
12 fimalgoritmo
```

Opções de resposta:

- a) 0 b) 11 c) 9 d) 1 e) 5

Questão 9:

```
1 Algoritmo questao
2 var
3 t, i, q: inteiro
4 inicio
5     i <- 0
6     q <- 4
7     t <- 1
8     enquanto i < q faça
9         se (i < 2) OR (i = 4) entao
10            t <- t + 5
11        senao
12            t <- t + 1
13        fimsenao
14        i <- i + 1
15    fimenquanto
16    t <- t + 1
17    escreval(t)
18 fimalgoritmo
```

Opções de resposta:

- a) 5 b) 25 c) 11 d) 14 e) 1

Questão de Controle:

```
1 Algoritmo questao
2 inicio
3     escreval(4)
4 fimalgoritmo
```

Opções de resposta:

- a) 1 b) 4 c) 11 d) 10 e) 31

A.4 Documento de Acompanhamento para a Validação da Abordagem 4Experts

Documento de Controle de Execução

Id:		Nome:	
Utilizando Abordagem 4Experts?		<input type="checkbox"/> SIM <input type="checkbox"/> NÃO	

Questão 1	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 2	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 3	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 4	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 5	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 6	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 7	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 8	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 9	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Questão 10	ID Questão:			
	Início:		Fim:	
	Resposta:	<input type="checkbox"/> Correta <input type="checkbox"/> Incorreta		

Muito obrigado pela sua participação!

A.5 Questões Aplicadas no Experimento de Validação da Abordagem 4Experts

Questões propostas para o experimento

Questão 1:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float varA;
6
7      varA = 4 * 3 + 6 / 3;
8      printf("%.2f", varA);
9  }
```

Questão 2:

```
1  #include <stdio.h>
2
3
4  int main()
5  {
6      float varA, varRes;
7
8      varA = 125;
9      if (varA < 200){
10         varRes = 1;
11     }
12     else{
13         varRes = varA;
14     }
15     printf("%.2f", varRes);
16 }
17
```

Questão 3:

```
1  #include <stdio.h>
2
3
4  int main()
5  {
6      float varB;
7
8      varB = 10;
9      while(varB < 10){
10         varB = varB * 2;
11     }
12     printf("%.2f", varB);
13 }
14
15
```

Questão 4:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int varA;
6
7     varA = 10 % 3 + 7;
8     printf("%d", varA);
9 }
10
11
12
```

Questão 5:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float varA;
6
7     varA = 2 + 4 / 4 - 1 * 3;
8
9     printf("%.2f", varA);
10 }
11
12
```

Questão 6:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6     float varA;
7
8     varA = 2 * (3 - 1) + pow(2,2);
9     printf("%.2f", varA);
10 }
11
12
13
```

Questão 7:

```
1 #include <stdio.h>
2
3
4 int main()
5 {
6     int varS;
7     int varRes;
8
9     varS = 512;
10    varS = varS / 4;
11    varRes = (varS % 127);
12    printf("%d", varRes);
13 }
14
15
16
```

Questão 8:

```
1 #include <stdio.h>
2
3
4 int main()
5 {
6     int varA;
7
8     varA = 3 * 3;
9     if (varA > 15){
10        varA = 3;
11    }
12    else{
13        varA = 6;
14    }
15    varA = varA * 3;
16
17    printf("%d", varA);
18 }
19
20
```

Questão 9:

```
1 #include <stdio.h>
2
3
4 int main()
5 {
6     int varC;
7
8     varC = 11;
9     varC = (3 * 5) + (varC * 1);
10    varC = varC - 2;
11
12    printf("%d", varC);
13 }
14
15
```

Questão 10:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int varZ;
6      float varRes;
7
8      varRes = 4;
9      varZ = 2;
10     varZ++;
11     varZ++;
12     varZ++;
13
14
15     if (varZ == 6){
16         varRes = varZ;
17     }
18     else{
19         varRes = 4 / varRes;
20     }
21     printf("%.2f", varRes);
22 }
23
24
```

Questão 11:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float varA, varB, varF, varH;
6
7      varA = 4;
8      varB = 3;
9      varF = 16;
10     varH = 7;
11     if (varB < varA && varB > varH){
12         varF = varB;
13     }
14     else{
15         varF = varH;
16     }
17     printf("%.2f", varF);
18 }
19
20
```

Questão 12:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float f, sal, b;
6
7     f = 2;
8     sal = 2000;
9     if (f > 0 && f <= 3){
10         b = sal * 0.35;
11     }
12     else{
13         b = sal * 0.15;
14     }
15     b = b + 15;
16     printf("%.2f", b);
17 }
18
19
20
```

Questão 13:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float varA;
6
7     varA = 12;
8     while (varA > 2){
9         varA = varA - 2;
10    }
11    printf("%.2f", varA);
12 }
13
14
15
```

Questão 14:

```
#include <stdio.h>
int main()
{
    float varA, varB, varC, varRes, total;

    varA = 6;
    varB = 6;
    varC = 6;
    total = 0;
    if ((varA != varB) && (varB != varC)) {
        total = total + 1 ;
        varRes = 10;
        total = total +varRes;
    }
    else{
        total = total + 2 ;
        varRes = 20;
        total = total +varRes;
    }
    printf("%.2f", total);
}
```

Questão 15:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int varA, varB, varF;
6
7      varA = 4;
8      varB = 2;
9
10     if (varA > varB || varB > varA){
11         varF = 2;
12     }
13     else{
14         varF = 4;
15     }
16     printf("%d", varF);
17 }
18
19
```

Questão 16:

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      float sal, b;
6      char categoria;
7
8      categoria = 'B';
9      sal = 200;
10
11     switch(categoria)
12     {
13         case 'A':
14             b = sal * 0.2;
15             break;
16         case 'B':
17             b = sal * 0.3;
18             break;
19         case 'C':
20             b = sal * 0.4;
21             break;
22         default:
23             b = 0;
24     }
25     sal = sal + b;
26     printf("%.2f", sal);
27 }
28
29
```

Questão 17:

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      float varB;
6      int x;
7
8      varB = 3;
9      for (x=1 ; varB <6 ; x++ )
10     {
11         varB = varB * 2;
12     }
13     printf("%d", x);
14 }
15
16
```

Questão 18:

```
1  #include <stdio.h>
2
3  main(void)
4  {
5      float a, s;
6      char area;
7      bool flag;
8      s = 1000.00;
9      area = 'G';
10     flag = false;
11     a = 0;
12     if (s <= 1000) {
13         a = 0;
14     }
15     else
16     {
17         switch(area)
18         {
19             case 'A':
20                 a = 20;
21                 break;
22             case 'B':
23                 a = 30;
24                 break;
25             default:
26                 flag = true;
27                 a = 40;
28         }
29     }
30     if (flag == true)
31     {
32         printf("%.2f", a);
33     }
34     else
35     {
36         printf("%.2f", a+s);
37     }
38 }
```

Questão 19:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float varA, varB, varC, varRes, total;
6
7     varA = 6;
8     varB = 8;
9     varC = 4;
10    total = 0;
11    if ((varA != varB) && (varB != varC)) {
12        total = total + 2 ;
13        varRes = 10;
14        total = total +varRes;
15    }
16    else if (varA > varB){
17        total = total + 4 ;
18        varRes = 20;
19        total = total +varRes;
20    }
21    else {
22        total = 10;
23    }
24    printf("%.2f", total);
25 }
26
```

Questão 20:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float varY;
6     int varX, varRes;
7
8     varX = 9;
9     varY = 14;
10
11    if (varX % 3 == 0){
12        varRes = varY / 2;
13    }
14
15    varRes = varX % varRes;
16    varRes++;
17    varRes--;
18    varRes--;
19
20    printf("%d", varRes);
21 }
22
```

Questão 21:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float varA, varT;
6
7     varA = 3;
8     while (varA < 7){
9         varA = varA + 1;
10        if (varA > 1 && varA < 2){
11            varT = 0;
12        }
13        else{
14            varT = varA;
15        }
16    }
17    printf("%.2f", varT);
18 }
19
20
```

Questão 22:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float varA, varT;
6
7     varA = 3;
8     while (varA < 5){
9         varT = varA * 3;
10        varT = varT / 3 + 5;
11        varA = varA + 1;
12    }
13    printf("%.2f", varT);
14 }
15
16
```

Questão 23:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int t, i, q;
6
7     i = 1;
8     q = 6;
9     t = 2;
10    while (i < q){
11        if (i < 2 || i == 4){
12            t = t + 5;
13        }
14        else{
15            t = t + 1;
16        }
17        i = i + 1;
18    }
19    t = t + 1;
20    printf("%d", t);
21 }
22
23
```

Questão 24:

```
1 #include <stdio.h>
2
3 main (void){
4     int contador,numero, m1, m2;
5
6     numero = 8;
7     m1 = numero;
8     m2 = numero;
9     numero =1;
10    contador =1;
11
12    do
13    {
14        numero = (m1-2);
15
16        if (numero > m1)
17        {
18            m1 = numero;
19        }
20        if (numero < m2)
21        {
22            m2 = numero;
23        }
24        contador ++;
25    }
26    while (contador <5);
27
28    printf( "\n %d %d ", m1,m2);
29 }
30
31
```

Questão 25:

```
1 #include <stdio.h>
2
3 main(void)
4 {
5     int a, b;
6     a = 3;
7     b = 4;
8     while( (a > 0) && (b > 0))
9     {
10        if (a > b)
11        {
12            a = a - 3;
13        }else
14        {
15            b = b - 1;
16        }
17    }
18    printf( "\n %d %d ", a,b);
19 }
20
21
22
```

Questão 26:

```
1 #include <stdio.h>
2
3 main(void)
4 {
5     int h;
6     float x,s,w;
7     s = 0;
8     x = 0;
9     for (int h = 4; h <= 7; h++)
10    {
11        if(h % 3 == 0){
12            s = s + h;
13            x = x + 1;
14        }
15    }
16    w = s/x;
17    printf("%.2f", w);
18 }
19
20
```

Questão 27:

```
1 #include <stdio.h>
2
3 main(void )
4 {
5     int linha, coluna;
6     linha = 1;
7     while (linha < 4)
8     {
9         coluna = 1;
10        while (coluna < linha)
11        {
12            printf( " # " );
13            coluna ++;
14        }
15        linha++;
16    }
17 }
18
19
20
```

Questão 28:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int varX, varY, varT;
6
7     varX = 2;
8     varY = 5;
9     while (varX < varY){
10         if (varX > varY || varX < 2){
11             varT = 0;
12         }
13         else{
14             varT = varX;
15         }
16         varX++;
17     }
18     varT = varT * 2;
19     printf("%d", varT);
20 }
21
22
```

Questão 29:

```
1 #include <stdio.h>
2
3 main(void)
4 {
5     int soma, subtracao, condA, condB;
6
7     condA = 10;
8     condB = 20;
9     soma = 0;
10    subtracao = 0;
11
12    while( (condA > 0) && (condB > 0))
13    {
14        soma++;
15        subtracao--;
16        condB = condB - condA;
17    }
18    printf( "\n %d %d ", soma,subtracao);
19 }
20
```

Questão 30:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int v1, v2, v3;
6     int vRes;
7
8     v2 = 7;
9     v3 = 2;
10    vRes = 0;
11    for (v1 = 0; v1 < v2; v1++){
12        v3 = v3 + 1;
13        if (v3 == (v2 - 1)){
14            vRes = (v3 * 2);
15        }
16    }
17    for (v1 = 0; v1 < v2; v1++){
18        vRes++;
19    }
20
21    printf("%d", vRes);
22 }
23
24
```


ANEXO A – ARTIGOS PUBLICADOS

A.1 Publicações

A longo deste período de pesquisa, foram publicados alguns artigos científicos para publicações em periódicos e anais de eventos. Além dos artigos referentes a este estudo, foram produzidos outros artigos em colaboração com pesquisadores do grupo de pesquisa da linha de engenharia de *software* e ambientes inteligentes desta universidade. A Tabelas 56 exhibe as publicações diretas, e as questões de pesquisa relacionadas a este estudo. O status das publicações segue como: publicado, aceito, submetido, e preparando.

Tabela 56 – Lista de Publicações Diretas

1. Juliano Paulo Menzen, Kleinner Farias, Vinicius Bischoff. <i>Using Biometric Data in Software Engineering: A Systematic Mapping Study</i> . Behaviour & Information Technology , 2020. (Aceito)	QP-1
2. Juliano Paulo Menzen, Kleinner Farias. <i>4Experts: Uma Abordagem de Recomendação de Tarefas Usando Aprendizagem de Máquina e Dados Biométricos</i> . XVI Brazilian Symposium on Information Systems (SBSI'20), 2019. (Submetido)	QP-3
3. Gabriel Lopes Nunes, Kleinner Farias, Juliano Paulo Menzen. <i>Parthenos: Uma Abordagem de Injeção de Código-Fonte para Transformação de Software</i> . XVI Brazilian Symposium on Information Systems (SBSI'20), 2019.(Submetido)	-

Fonte: Elaborado pelo autor.