

4

Avaliação do Processo PDS+T

Neste capítulo, detalhamos os estudos experimentais realizados com o propósito de avaliar o processo PDS+T. Ambos os estudos foram projetados com base na realização de tarefas de manutenção, pois esta atividade está fortemente relacionada com as qualidades, de rastreabilidade, detalhamento e dependência, que abordamos com o processo PDS+T. Cada estudo tem seu propósito específico. No primeiro, exploramos as características de rastreabilidade e detalhamento. Já no segundo, exploramos a característica de dependência. Foram realizadas análises qualitativas e quantitativas dos dados obtidos através do estudo.

4.1.

Estudos Empíricos

Escolhemos avaliar os artefatos resultantes do processo PDS+T sob a ótica de tarefas de manutenção, pois entendemos que estas são influenciadas diretamente pelas características de detalhamento, dependência e rastreabilidade. Segundo Mayrhauser (Mayrhauser, 95), o entendimento (detalhamento) do código fonte é um fator que influencia diretamente a eficácia da manutenção do software, facilitando, desta forma, sua evolução. Em (Banker et al., 1993) Banker também traça um paralelo entre o entendimento do software e as atividades de manutenção, porém o sentido de entendimento, neste caso, é mais amplo e não restrito apenas ao código. Seu estudo confirmou que os esforços de manutenção são afetados diretamente pela complexidade do código e, dentre os fatores que contribuem para complexidade, está a modularização. Sabemos que uma fragmentação demasiada dos módulos pode ter um efeito negativo sobre a manutenabilidade, já que o software se torna mais complexo.

A rastreabilidade, neste caso especificamente entre código e requisitos, também têm grande influência na execução de tarefas de manutenção. Isto foi demonstrado no experimento realizado por Mader (Mader e Egyed, 12), onde tarefas de manutenção foram executadas em média 21% mais rápido e foram criadas, na média, 60% mais soluções corretas quando disponibilizada a rastreabilidade entre requisitos e código.

Utilizaremos a relação entre tarefas de manutenção e as características supracitadas, para avaliar de forma indireta o processo PDS+T. Os estudos se baseiam na comparação dos resultados da execução de tarefas de manutenção, em um software descrito com os artefatos produzidos pelo PDS+T, em relação a um software sem estes artefatos. Se obtivermos melhores resultados na realização de tarefas de manutenção no software descrito com os artefatos do PDS+T, então podemos sugerir que o processo contribui positivamente com as características de detalhamento, dependência e rastreabilidade. Por conseguinte, concluiremos que o processo pode, potencialmente, melhorar a transparência do software que é produzido.

O projeto destes estudos foi baseado em (Mader e Egyed, 12), onde foi avaliado o impacto da rastreabilidade na realização de tarefas de manutenção. Neste trabalho, dois software livres foram utilizados: C&L e Biblioteca Digital. Cada um destes software possui duas versões. Uma onde nenhum processo definido foi utilizado em sua construção, e outra onde o PDS+T foi utilizado. As tarefas de manutenção representam tarefas reais, que foram resolvidas pelos desenvolvedores do software. O estudo proposto consiste na comparação do desempenho na realização de tarefas de manutenção na versão do software desenvolvida sem auxílio de processo, com aquela desenvolvida com o uso do PDS+T.

4.1.1. Sistemas Utilizados

Cel versão PHP (C&L, 2013b; Felicíssimo et al., 2004). Esta versão do C&L foi inicialmente desenvolvida como um trabalho acadêmico, mas com o passar do tempo progrediu para um projeto de Software Livre. Este software foi evoluído ao longo dos anos por estudantes de graduação, mestrado e doutorado. Por ser um Software Livre, durante seu desenvolvimento e evolução, procurou-se seguir a filosofia proposta em (Silva et al., 2003), para incluir cenários no código, com o intuito criar uma documentação para o código do software.

Cel versão Lua (C&L, 2013a). A versão em Lua do software C&L, foi criada a partir de uma reengenharia da versão em PHP, onde o processo PDS+T foi empregado. Portanto, esta versão conta com arquitetura MVC, e possuiu cenários que, ao mesmo tempo em que descrevem sua arquitetura e código, permitem a rastreabilidade entre estes artefatos e os requisitos do software.

Biblioteca Digital. Este software foi criado com a finalidade de permitir a organização e disponibilização de artigos de conferências e, atualmente, se encontra em uso em diversos locais (WERPapers, 2013; ClbSEPapers, 2013; FEESArtigos, 2013). Sua versão inicial possui problemas estruturais, o que a torna difícil de manter e evoluir. Desta forma, foi realizada uma reengenharia do software, onde o processo PDS+T foi empregado. A versão construída a partir deste processo ainda não foi utilizada em produção. Neste estudo, utilizaremos apenas a segunda versão, isto é, aquela desenvolvida com a utilização do processo PDS+T. A versão inicial, que se encontra atualmente em produção, não foi utilizada.

4.1.2. Estudo Biblioteca Digital

A seguir, descreveremos o primeiro estudo efetuado para avaliar o processo PDS+T. O estudo foi realizado com dois grupos distintos de participantes, cujos resultados foram avaliados separadamente. Ao final, os dados de ambos os estudos foram consolidados e uma nova avaliação dos resultados foi feita. Os participantes pertencentes ao primeiro grupo eram alunos de graduação, com experiência média inferior a 5 anos em desenvolvimento de software. O segundo grupo era composto, em sua maioria, de alunos de pós-graduação, com experiência média de 11 anos em desenvolvimento de software.

4.1.2.1. Objetivos Específicos

O objetivo deste estudo, em particular, é avaliar as qualidades **rastreabilidade** e **detalhamento** proporcionadas pelo processo PDS+T. Estas duas qualidades são implementadas pelo processo através dos cenários que descrevem os artefatos do software, o que inclui requisitos, arquitetura e código, e os rastros existentes entre estes cenários.

A avaliação do impacto da dependência está fora do escopo deste estudo específico. Isto porque ambas as versões do software Biblioteca Digital, utilizado neste estudo, contam com a mesma estrutura (modularização e arquitetura), já que apenas a omissão de informações não permite a remoção de tal característica. A dependência será avaliada através de outro estudo, que será detalhado mais adiante.

Desta forma, através deste estudo pretendemos responder as seguintes questões:

Q₁: A presença das qualidades de rastreabilidade e detalhamento no software, proporcionadas pelo processo PDS+T, contribuem para a realização de atividades de manutenção mais eficazes (corretas), quando comparadas a realização das mesmas tarefas em um software sem estas qualidades?

Q₂: A presença das qualidades de rastreabilidade e detalhamento no software, proporcionadas pelo processo PDS+T, contribuem para a realização de atividades de manutenção em menor tempo, quando comparadas a realização das mesmas tarefas em um software sem estas qualidades?

4.1.2.2.

Participantes

O estudo Biblioteca Digital contou com a participação de dois grupos distintos de participantes. O perfil dos participantes do estudo foi determinado através de um questionário (Tabela 17), cujo objetivo era, principalmente, determinar a experiência dos participantes em desenvolvimento de software e o seu conhecimento nas tecnologias utilizadas (neste caso, MVC e Lua).

O primeiro grupo (grupo 1) foi composto de 7 alunos de ciência da computação da PUC - Rio, que cursavam a disciplina de Engenharia de Requisitos. Em média, os indivíduos deste grupo relataram possuir aproximadamente 3,8 anos de experiência em desenvolvimento de software de maneira geral (universidade e indústria). Somente 2 dos indivíduos informaram ter tido experiência em desenvolvimento de software para indústria. Além disso, apenas um dos indivíduos relatou possuir experiência (1 ano) em desenvolvimento utilizando a linguagem Lua. De maneira geral, o conhecimento dos participantes quanto ao padrão MVC foi relatado como baixo (2 em uma escala de 1 a 5). Nenhum dos participantes tinha conhecimento prévio do código da Biblioteca Digital, utilizado no estudo.

O segundo grupo (grupo 2) era composto de 10 participantes que, em sua grande maioria, eram alunos de pós-graduação (mestrado e doutorado) em informática da PUC - Rio. Os indivíduos deste grupo possuem, em média, aproximadamente 11 anos de experiência em desenvolvimento de software de

maneira geral (indústria e universidade). Em termos de desenvolvimento de software específico para indústria, os indivíduos do grupo relataram possuir, em média, aproximadamente 8 anos de experiência. Somente 2 indivíduos informaram ter experiência em programação na linguagem Lua. De maneira geral, os indivíduos afirmaram possuir um bom conhecimento do padrão MVC (aproximadamente 4 em uma escala de 1 a 5). Além disto, 2 indivíduos informaram ter conhecimento do código do software Biblioteca Digital, utilizado no estudo.

Experiência

Gostaríamos de saber qual seu conhecimento prévio sobre tópicos de importância para o experimento. Por favor, a seguir informe para cada tópico quantos anos de experiência você possui na área.

	Experiência
Desenvolvimento de software (faculdade, trabalho, escola)	_____ anos
Desenvolvimento de software na indústria	_____ anos
Desenvolvimento com linguagem Lua	_____ anos

Qual seu conhecimento sobre o padrão de arquitetura MVC?

☐ Nenhum ☐ Pouco ☐ Razoável ☐ Bom ☐ Muito Bom

Você conhece o software Biblioteca Digital da perspectiva de desenvolvedor?

☐ Sim ☐ Não

Tabela 17. Questionário para determinar perfil do participante.

Totalizando os dois grupos, obtivemos uma média de 8,1 anos de desenvolvimento de software de maneira geral, e aproximadamente 4,9 anos especificamente na indústria. Somando o tempo de experiência em linguagem Lua dos participantes, obtivemos um total de 4 anos. E por fim, obtivemos uma média de conhecimento MVC razoável (3 em uma escala de 1 a 5). A Figura 30 apresenta um gráfico com o perfil dos participantes do estudo.

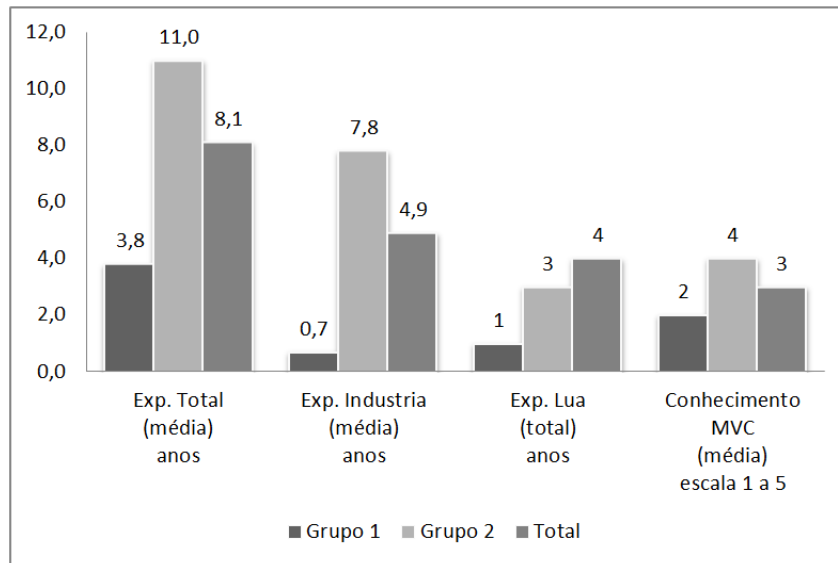


Figura 30. Perfil dos participantes do Estudo Biblioteca Digital.

4.1.2.3.

Projeto do Estudo

Ambientes

Dois ambientes distintos, que chamaremos de **A** e **B**, foram criados para execução do estudo. No ambiente **A**, os cenários resultantes do processo PDS+T, utilizado no desenvolvimento do software Biblioteca Digital, foram mantidos. Nestes **cenários**, estão **incluídos**: cenários de requisitos, cenários de camadas (MVC) e cenários operacionais integrados ao código. Os cenários foram inseridos em um projeto no software C&L, de forma que **é possível navegar entre os cenários** através de *links*, que mapeiam os relacionamentos entre cenários. Desta forma, é possível navegar dos cenários de requisitos até os operacionais, integrados ao código e, também, realizar o caminho inverso.

No ambiente **B**, todos os **cenários**, exceto os de requisitos, foram **suprimidos**. Os cenários integrados ao código foram removidos. O projeto no software C&L deste ambiente conta apenas com os cenários de requisitos.

Tarefas

A fim de avaliar as características de detalhamento e rastreabilidade, propomos a criação de **quatro tarefas de manutenção**, uma para cada tipo identificado na taxonomia proposta por Kitchenham (Kitchenham et al, 99). Desta

forma, foram criadas tarefas de correção, mudança de requisito, mudança de implementação e novo requisito. A fim de explicar cada tarefa, criamos um texto simulando o que seria apresentado para um desenvolvedor em um caso real. A seguir descreveremos as tarefas propostas. Apenas o enunciado informado foi apresentado aos participantes. Nenhuma informação adicional foi dada.

Tarefa 1: *Os usuários do sistema Biblioteca Digital estão reportando um erro na página de edições de um evento. Segundo as mensagens, o sistema está exibindo apenas o texto “descrição” ao invés de exibir a descrição do evento.*

Esta tarefa simula uma correção simples no software. Apenas a camada de visão é afetada. Porém, é uma modificação muito pontual, e exige que o indivíduo realize uma busca minuciosa no código e entenda a interface do software para encontrar o defeito.

PUC-Rio - Certificação Digital Nº 0912908/CA

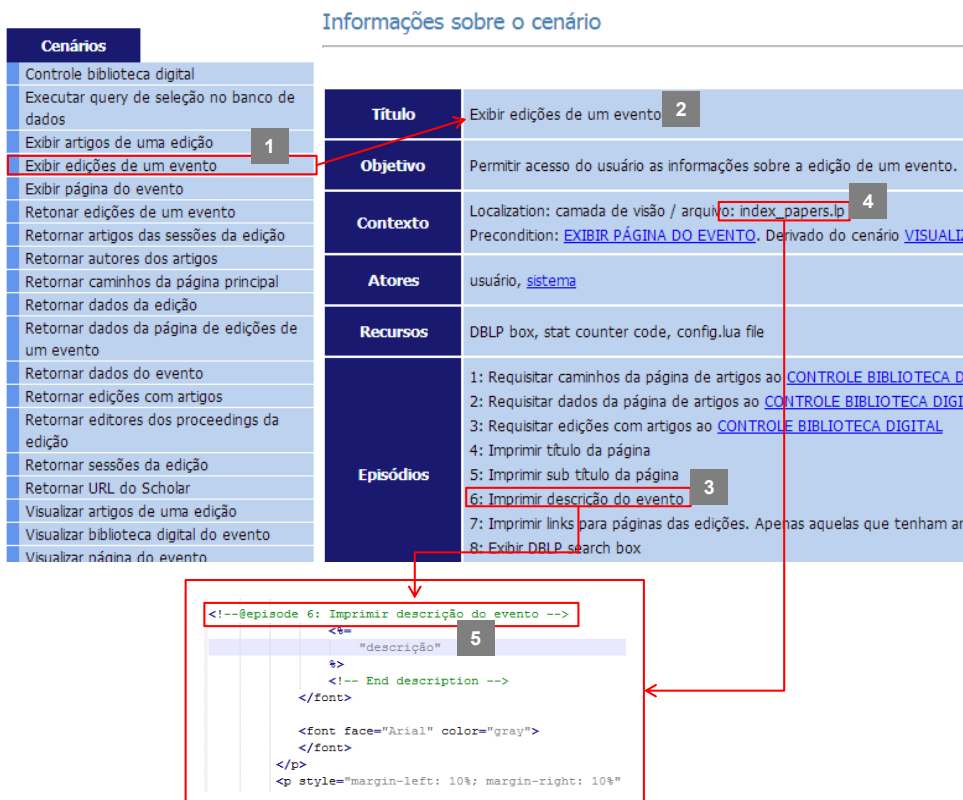


Figura 31. Passos para realizar a tarefa 1 do Estudo Biblioteca Digital com artefatos do PDS+T.

Para demonstrar os procedimentos necessários para encontrar o erro descrito na tarefa 1, criamos a Figura 31. Neste caso, simularemos a execução da tarefa por um participante no ambiente A, onde todos os artefatos do PDS+T foram mantidos e os participantes utilizaram o software C&L, para navegar entre os cenários.

Os passos 1 e 2, representados na Figura 31, referem-se à identificação do cenário que apresenta o erro. No passo 3, identificamos qual o episódio do cenário correspondente ao erro descrito. Uma vez encontrado o episódio, é preciso saber qual o arquivo em que se encontra o cenário que está sendo visualizado (passo 4). Após determinar o arquivo, basta abri-lo e procurar pelo episódio identificado no passo 3 (passo 5). Uma vez encontrado o episódio, é preciso determinar o erro. Após determinar o erro, o participante deveria preencher o questionário de acordo. A Tabela 18 mostra como deveria ser preenchido o questionário associado à tarefa 1, para que esta fosse considerada correta.

Artefato/posição	Tipo de modificação	Pseudocódigo/descrição
index_papers.lp linha 70	mudança	Substituir string "descrição" por <code>selected_paperspage["descricao"]</code>

Tabela 18. Resposta da tarefa 1 do Estudo Biblioteca Digital.

Para executar esta tarefa sem o apoio dos artefatos gerados pelo PDS+T, o participante teria que procurar o arquivo correto, abrindo um por um. Após encontrá-lo, o participante precisaria entender todo o código para descobrir em que parte se encontra o erro.

Tarefa 2: *Para tornar a busca no Google Scholar mais precisa, um dos desenvolvedores propõe adicionar a URL de busca no Scholar o último nome do segundo autor do artigo.*

Esta tarefa simula uma mudança de código sem alteração dos requisitos funcionais. Sua implantação exige a modificação apenas da camada de modelo do software. A mudança é simples, porém o participante deve entender o código para reutilizar os componentes já existentes, e saber que as modificações se limitam a apenas uma camada.

Tarefa 3: Um dos stakeholders do software Biblioteca Digital deseja a adição de uma nova funcionalidade ao software. Da mesma forma que um link é montado, permitindo pesquisa diretamente no Google Scholar para cada artigo da conferência, este usuário deseja que haja um link, ao lado do anterior, para uma busca direta no Microsoft Academic Search.

Esta tarefa simula a adição de um novo requisito ao software e tem um grande impacto, pois exige alteração em suas três camadas. Para realizá-la, o participante deve criar a lógica correta na camada de modelo, aproveitando a estrutura existente, e preparar a camada de visão e controle para permitir a utilização desta nova funcionalidade. A tarefa exige a compreensão do código, para que seja possível determinar onde, exatamente, cada camada deve ser alterada. Além disto, o participante precisa realizar as alterações necessárias nos cenários que descrevem o software.

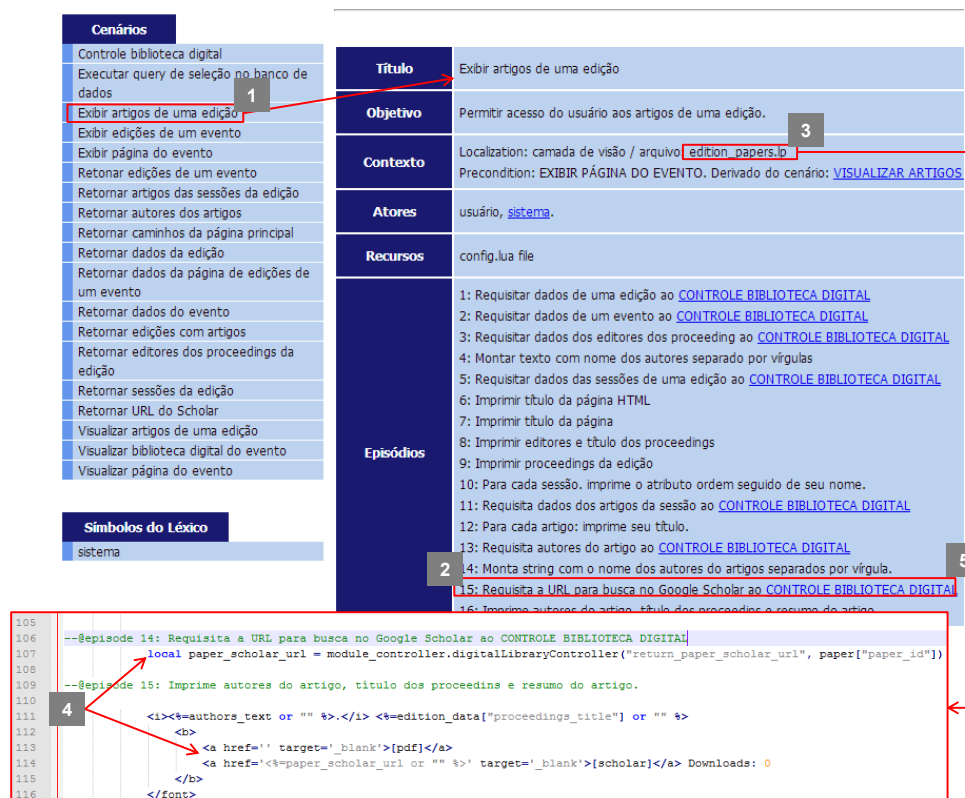


Figura 32. Passos 1 a 5 para realizar a tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.

Utilizamos as Figuras 32, 33 e 34, para explicar as etapas que deveriam ser percorridas pelos participantes, a fim de encontrar a resposta da tarefa 3. Isto para aqueles que a executaram no software com os artefatos do PDS+T, com o apoio do software C&L, isto é, no ambiente A.

O primeiro passo consiste em identificar o cenário que precisa ser alterado. Após sua identificação, é preciso determinar onde, em seus episódios, será feita a mudança (passo 2). Tendo o conhecimento de onde a mudança precisa ser feita, seguimos o rastro até o arquivo onde o cenário se encontra implementado (passo 3), para encontrar o código correspondente ao episódio que será alterado (passo 4). Em seguida, é preciso seguir o relacionamento com o cenário de controle (passo 5), a fim de determinar as modificações necessárias nesta camada.



Figura 33. Passos 6 a 9 para realizar a tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.

Após acessar o cenário da camada de controle (passo 6), é preciso identificar onde, em seus episódios, ocorrerá a mudança (passo 7). A seguir, é preciso encontrar onde o cenário de controle está implementado (passo 8), para estabelecer as mudanças necessárias no código de controle. Posteriormente, seguimos o relacionamento com o cenário da camada de modelo (passo 9), com o intuito de determinar as alterações necessárias nesta camada.

Depois de visualizar o cenário de modelo (passo 10), é preciso seguir o rastro até o código fonte que o implementa, para identificar as mudanças necessárias (passo 11 e 12) neste artefato.

Neste exemplo em particular, seguimos o fluxo do software para determinar como uma função semelhante foi implementada. A função "*returnPapersScholarURL*" não deve ser alterada para acomodar o novo requisito. Ela deve ser utilizada como base para a criação de um novo cenário (função) de modelo, a fim de contemplar a lógica necessária para montar a URL para busca no *Microsoft Academic Search*. Lembramos que, para a tarefa ser considerada correta, era preciso que os indivíduos indicassem também as alterações necessárias nos cenários. Com isso, estamos avaliando se a evolução dos artefatos do processo PDS+T, a fim de evitar sua erosão com o passar do tempo, é natural para os indivíduos durante a execução da tarefa. Também estamos avaliando o impacto desta evolução, isto é, se ela demandou mais tempo para ser realizada.



Figura 34. Passos 10 a 12 para realizar a tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.

Exibimos na Tabela 19, as respostas esperadas para a tarefa 3 do Estudo Biblioteca Digital. Como podemos observar, além das alterações no código em si, também era necessário informar as alterações relacionadas aos cenários resultantes do processo PDS+T.

Para realizar a mesma tarefa, sem o suporte dos artefatos do PDS+T, é preciso que o usuário identifique o fluxo do software através do seu código, e encontre o trecho que necessita de mudança em cada camada. Para que isto seja possível, o usuário deve encontrar os arquivos corretos e entender grande parte do código presente neles.

Artefato/posição	Tipo de modificação	Pseudocódigo/descrição
Cenário: exibir artigos de uma edição	inclusão	Incluir episódio 16: Requisita a URL para busca no Microsoft Academic Search (MAS) ao CONTROLE BIBLIOTECA DIGITAL
edition_papers.lp linha 107	incluir	Chamada a camada de controle para retornar a URL paper_msa_url = ("return_paper_mas_url", paper["paper_id"])
edition_papers.lp	incluir	URL na interface <a href='<%=paper_msa_url or "" %>' target='_blank'>[MAS]
controllerdigitallibrary.lua	incluir	Nova condição na camada de controle para tratar a requisição elseif (command == "return_paper_mas_url") then return model_paper.returnPapersMASURL (first_parameter)
Cenário: controle biblioteca digital	incluir	Episódio para descrever nova chamada Se comando igual a return_paper_mas_url então RETORNAR URL DO MAS
model_paper.lua	incluir	Função semelhante a returnPapersScholarURL para montar URL do MAS
Cenário: retornar URL do MAS	incluir	Incluir novo cenário para documentar a nova função criada na camada de modelo.

Tabela 19. Resposta da tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.

Tarefa 4: Houve uma mudança nos requisitos do software Biblioteca Digital. O cenário “Visualizar artigos de uma edição” sofreu a seguinte alteração:

Episódio 3

De: Exibe informações dos proceedings do evento, que contém o nome dos autores separado por vírgula, o título, a editora que publicou, o ISBN e o ano de publicação.

Para: Exibe informações dos proceedings do evento, que contém o nome dos autores separado por “-“, o título, a editora que publicou, o ISBN e o ano de publicação.

Esta tarefa tem como objetivo simular a mudança de um requisito. Seu impacto está restrito a camada de visão do software. Apesar de representar uma mudança simples na interface do software, sua implantação exige um conhecimento preciso do código, pois há comportamentos similares que podem confundir a identificação exata do ponto que deve ser alterado.

Métricas

A fim de avaliar o desempenho dos participantes na execução das tarefas de manutenção, utilizamos **duas** métricas: **tempo** de execução das tarefas (medido em segundos) e **correção** das alterações sugeridas. Para medir o tempo, pedimos aos participantes que registrassem a hora, os minutos e segundos em que começaram e terminaram cada uma das tarefas. A fim de avaliar a correção das atividades, utilizamos **três parâmetros**. O primeiro é relacionado aos artefatos que precisam ser modificados. Para que a tarefa esteja correta, o participante deve ter mencionado exatamente os artefatos que necessitam de modificação e, se for aplicável, a posição do artefato que precisa de modificação (linha, no caso de código e componente, no caso de cenário). O segundo parâmetro está relacionado ao tipo de modificação. Para que a tarefa esteja correta, o participante deve ter informado exatamente o tipo de modificação necessária em cada artefato. O terceiro, e último parâmetro, é mais subjetivo. Neste, iremos avaliar se o pseudocódigo informado resolve a tarefa correspondente. Não iremos considerar apenas a solução ótima. Qualquer solução que resolva de alguma forma a tarefa será considerada, contanto que a estrutura do software seja mantida. As respostas foram classificadas em corretas ou incorretas. Nenhuma classificação intermediária foi feita.

Possíveis ameaças

Uma das possíveis ameaças ao estudo seria o conhecimento do código com cenários pelos participantes que começarem o experimento pelo ambiente A, fazendo com que tenham vantagem ao executar as tarefas no ambiente B. Porém, acreditamos que devido ao curto tempo de exposição dos participantes ao código com cenários, este conhecimento não tenha sido profundo a ponto de influenciar o segundo conjunto de tarefas. Além disto, as tarefas focam partes diferentes do código, de modo que o conhecimento de uma não tem muita influência na execução das outras.

Outra ameaça seria a execução das tarefas pelos próprios desenvolvedores dos software. Neste caso, acreditamos que obteríamos o mesmo resultado, pois mesmo tendo conhecimento do código, o desenvolvedor precisaria lembrar o que foi feito nos locais específicos afetados por estas tarefas. Porém, poderíamos ter um resultado diferente, já que as tarefas executadas nos software desenvolvidos através do PDS+T são mais complexas, uma vez que envolvem a atualização de outros artefatos (cenários) e não apenas do código.

Deixar a marcação do tempo para realizar as tarefas sob a responsabilidade dos participantes poderia representar um problema, pois estes poderiam esquecer-se de fazer isto no momento apropriado, distorcendo os resultados obtidos. Para mitigar este risco, cada sessão do estudo foi acompanhada de perto por um indivíduo que verificava se a marcação estava sendo feita corretamente.

4.1.2.4.

Realização do Estudo

Cada participante realizou as quatro tarefas de manutenção, apresentadas anteriormente, propostas para o Estudo Biblioteca Digital. Duas delas com o a presença de todos os artefatos resultantes do processo PDS+T, e duas apenas com a presença dos cenários de requisitos, inseridos no software C&L. Todos os participantes executaram as tarefas na mesma ordem, porém, alternamos a presença dos artefatos resultantes do PDS+T. Isto é, metade dos participantes executaram as tarefas 1 e 2 contando com os artefatos do PDS+T. Estes participantes executaram as tarefas 3 e 4 apenas com o apoio dos cenários de requisitos. A outra metade fez o inverso, isto é, executou as tarefas 1 e 2 apenas

com os cenários de requisitos, e as tarefas 3 e 4 com o apoio dos artefatos do processo PDS+T. Na Tabela 20, apresentamos a distribuição de tarefas entre os participantes do Grupo 1. Exibimos na Tabela 21 a distribuição de tarefas realizada entre os participantes do Grupo 2. Com esta distribuição, garantimos que, do total de tarefas executadas, metade foi realizada com apoio dos artefatos obtidos através do processo PDS+T e a outra metade apenas com cenários de requisitos. Porém, como podemos observar, os participantes 6 e 7 do Grupo 1 e os participantes 8, 9 e 10 do Grupo 2, receberam a mesma distribuição de tarefas. Isto aconteceu devido à desistência de dois participantes, um de cada grupo, que iriam executar as tarefas 1 e 2 no ambiente A, e 3 e 4 no ambiente B. Como a desistência ocorreu após a distribuição de tarefas, não pudemos equilibrá-la novamente.

	Ambiente A (com cenários)	Ambiente B (sem cenários)
Participante 1	Tarefas 1 e 2	Tarefas 3 e 4
Participante 2	Tarefas 3 e 4	Tarefas 1 e 2
Participante 3	Tarefas 1 e 2	Tarefas 3 e 4
Participante 4	Tarefas 3 e 4	Tarefas 1 e 2
Participante 5	Tarefas 1 e 2	Tarefas 3 e 4
Participante 6	Tarefas 3 e 4	Tarefas 1 e 2
Participante 7	Tarefas 3 e 4	Tarefas 1 e 2

Tabela 20. Distribuição de tarefas entre participantes do Grupo 1.

	Ambiente A (com cenários)	Ambiente B (sem cenários)
Participante 1	Tarefas 1 e 2	Tarefas 3 e 4
Participante 2	Tarefas 3 e 4	Tarefas 1 e 2
Participante 3	Tarefas 1 e 2	Tarefas 3 e 4
Participante 4	Tarefas 3 e 4	Tarefas 1 e 2
Participante 5	Tarefas 1 e 2	Tarefas 3 e 4
Participante 6	Tarefas 3 e 4	Tarefas 1 e 2
Participante 7	Tarefas 1 e 2	Tarefas 3 e 4
Participante 8	Tarefas 3 e 4	Tarefas 1 e 2
Participante 9	Tarefas 3 e 4	Tarefas 1 e 2
Participante 10	Tarefas 3 e 4	Tarefas 1 e 2

Tabela 21. Distribuição de tarefas entre participantes do Grupo 2.

Foi solicitado aos participantes que, para cada tarefa, indicassem os componentes e o local em que necessitam de modificação. Dentre estes componentes, podemos ter arquivos, funções ou cenários. Os locais podem ser indicados através de número da linha ou elemento do cenário. Pedimos também, que informassem que tipo de modificação necessária, dentre três possíveis: inclusão, modificação e exclusão. Por fim, foi solicitada uma breve explicação em pseudocódigo da modificação necessária. Após o breve treinamento inicial (20 minutos) sobre como responder o questionário do experimento e uso do software C&L, não foi mais permitido aos participantes tirar dúvidas. A realização do experimento foi acompanhada de perto para assegurar que os participantes não se esquecessem de anotar o tempo de início e fim de cada tarefa.

Os dados do estudo foram obtidos através de um questionário, que continha campos para que os participantes informassem o horário de início e fim de cada tarefa e todas as informações necessárias para indicar as modificações necessárias para sua implantação.

4.1.2.5.

Resultados Obtidos

A seguir, apresentaremos as análises dos dados do Estudo Biblioteca Digital. Como dito anteriormente, este estudo foi realizado com dois grupos, com perfis diferentes. Para avaliar melhor o comportamento de cada perfil, iremos analisar seus resultados separadamente e, ao final, iremos consolidar os dados e analisar como uma amostragem única de participantes.

Grupo 1

A seguir, apresentamos a análise dos dados do Estudo Biblioteca Digital, obtidos a partir do Grupo 1, que corresponde a alunos graduação da PUC - Rio.

Correção. A Figura 35 mostra um gráfico que relaciona o número de respostas corretas e incorretas com o tipo de abordagem avaliada. Cada barra do gráfico representa o número de tarefas executadas, utilizando cada abordagem, que neste caso foram 14 (28 no total). A quantidade de tarefas corretas, em cada caso, está representada pela cor mais escura e a de incorretas pela cor mais clara. Do total de 28 tarefas, apenas 9 foram executadas de maneira correta (32,14%). Das 9 tarefas corretas, a maioria foi realizada com

a ajuda de cenários (6 - o que significa 67,67% das corretas). Desta forma, concluímos que, a rastreabilidade e o detalhamento, proporcionados pelos cenários, seus relacionamentos e integração com o código, influenciaram positivamente a execução correta das tarefas.

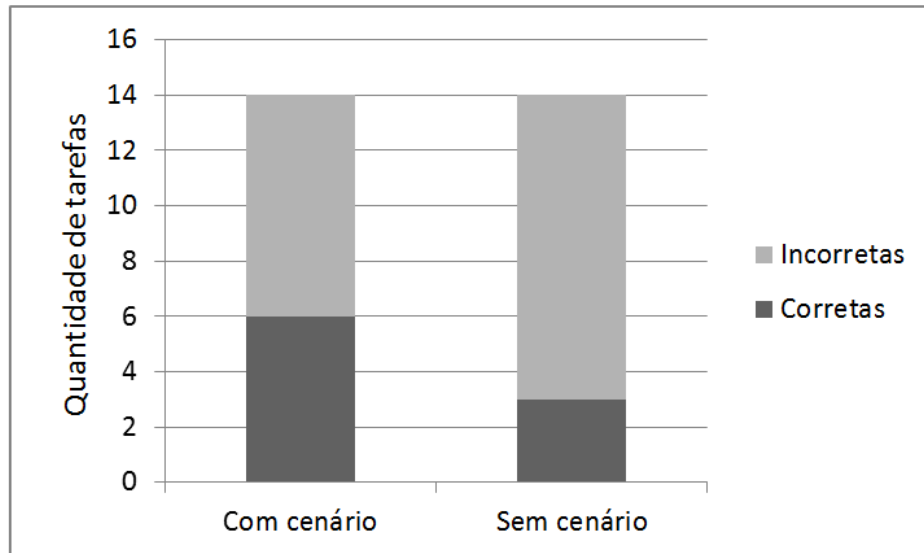


Figura 35. Grupo 1 gráfico de tarefas corretas.

Tempo de execução. Apresentamos na Figura 36 o gráfico que relaciona as tarefas com seu tempo de execução (em segundos), realizando um comparativo entre a sua execução com cenários oriundos do PDS+T e sem estes cenários. Neste gráfico, para cada tarefa, duas barras são exibidas. A mais escura, mostra a média dos tempos de execução da tarefa com apoio dos cenários. A segunda, mais clara, exibe a média dos tempos de execução da tarefa sem apoio dos cenários. Na média, tivemos uma redução de 7% no tempo de execução das tarefas, quando estas eram apoiadas por cenários. Realizando uma análise mais minuciosa do gráfico, é possível observar que, em algumas tarefas (2 e 4), o tempo de execução foi maior quando apoiadas por cenário. Porém, notamos que nenhuma destas tarefas foi realizada corretamente sem a ajuda de cenários. Desta forma, podemos sugerir que, dada a complexidade destas tarefas, os participantes não conseguiram identificar o impacto de sua implantação no software sem auxílio dos cenários. Por conseguinte, demoraram menos tempo para realizá-las, pois menos locais precisaram ser analisados, entretanto, a solução dada estava incorreta. Portanto, temos indícios que, de maneira geral, a rastreabilidade e o detalhamento, proporcionados pelos cenários, seus relacionamentos e integração com o código, diminuíram o tempo

de execução das tarefas. Entretanto, em vista da observação feita anteriormente, percebemos que alguns fatores influenciaram a medição do tempo, obtida através desta análise.

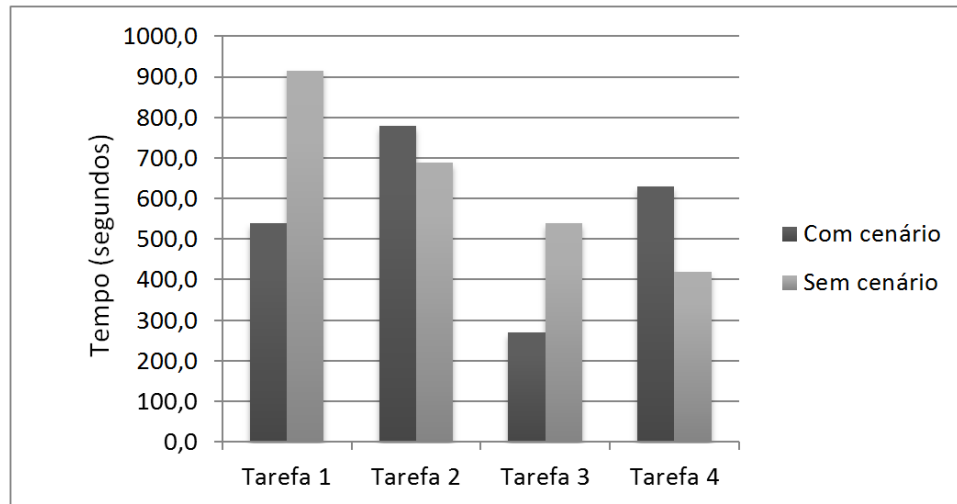


Figura 36. Grupo 1 gráfico do tempo de execução das tarefas.

Grupo 2

A seguir, apresentamos a análise dos dados do Estudo Biblioteca Digital, obtidos a partir do Grupo 2, que corresponde a alunos de pós-graduação (mestrado e doutorado) da PUC - Rio.

Correção. A Figura 37 apresenta um gráfico que relaciona o número de respostas corretas e incorretas com o tipo de abordagem avaliada. Cada barra do gráfico representa o número de tarefas executadas, utilizando cada abordagem, que neste caso foram 20 para cada uma (40 no total). A quantidade de tarefas corretas, em cada caso, está representada pela cor mais escura e a de incorretas pela cor mais clara. Do total de 40 tarefas, apenas 19, quase a metade, foram executadas de maneira correta (47,50%). Das 19 tarefas corretas, a maioria foi realizada com a ajuda de cenários, oriundos do PDS+T (14 - o que significa 73,68% das corretas). Podemos observar que a proporção de tarefas corretas foi, significativamente, superior ao obtido com o grupo 1 (que foi de 32,14%). Além disto, a proporção de tarefas corretas executadas com cenários também aumentou, ligeiramente, no grupo 2 (73,68% contra 67,67%). Acreditamos que isto seja consequência da maior maturidade e experiência dos participantes do Grupo 2, a qual permitiu aos participantes com este perfil

aproveitar melhor a presença dos cenários. Desta forma, temos indícios que, a rastreabilidade e o detalhamento proporcionados pelos cenários, seus relacionamentos e integração com o código, influenciaram positivamente a execução correta das tarefas, pois a maioria das tarefas corretas foi executada com a presença destas qualidades. Além disto, concluímos também que esta influência é maior quando os indivíduos possuem mais experiência, já que o percentual de tarefas corretas com auxílio destas qualidades foi maior no Grupo 2, que era composto de participantes com este perfil.

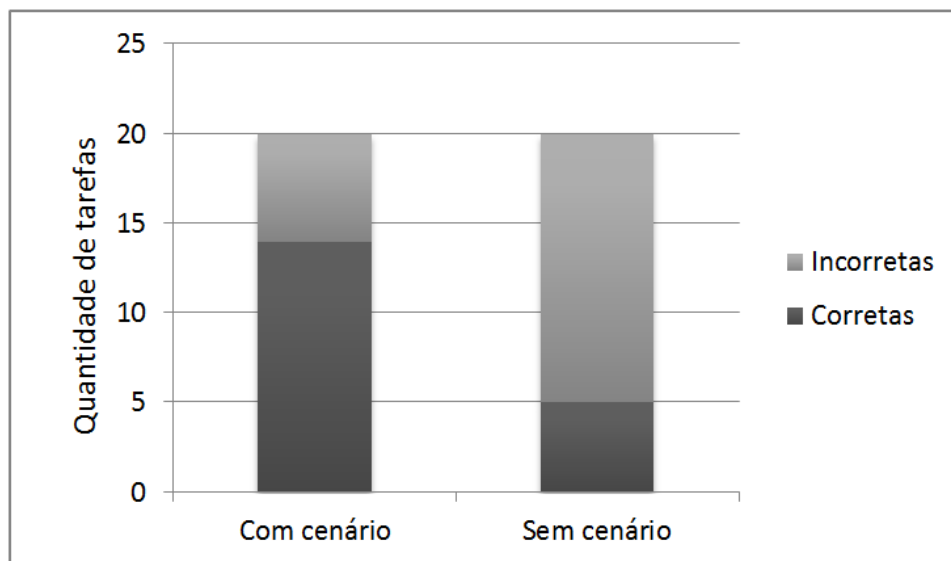


Figura 37. Grupo 2 gráfico de tarefas corretas.

Tempo de execução. Exibimos na Figura 38 o gráfico que relaciona as tarefas com seu tempo de execução (em segundos), realizando um comparativo entre a sua execução com apoio de cenários oriundos do PDS+T e sem seu apoio. Neste gráfico, para cada tarefa, duas barras são exibidas. A mais escura, mostra a média dos tempos para conclusão da tarefa com apoio dos cenários. A segunda, mais clara, exibe a média dos tempos para conclusão da tarefa, sem apoio dos cenários. Na média, tivemos uma redução de 50,45% no tempo de execução das tarefas, quando estas eram suportadas por cenários. Neste caso, como é possível observar através do gráfico, todas as tarefas tiveram seu tempo médio de execução reduzido, quando executadas com o apoio de cenários. Analisando o gráfico, vemos que a tarefa 1 teve uma redução significativamente maior que as demais (68,75%). Acreditamos que isto se deve a natureza da tarefa, que é muito pontual e, sem o auxílio dos cenários, exige do indivíduo o entendimento de todo o código, para que encontre o local exato onde o defeito

se encontra. Por conseguinte, podemos supor que a rastreabilidade e o detalhamento, proporcionados pelos cenários, seus relacionamentos e integração com o código, diminuam o tempo de execução das tarefas. Podemos inferir, comparando com o resultado do grupo 1, que quanto maior a experiência do indivíduo que está realizando a tarefa, maior será a redução do tempo (50,45% contra 7% do grupo 1), se o software onde ela está sendo executada for descrito por cenários.

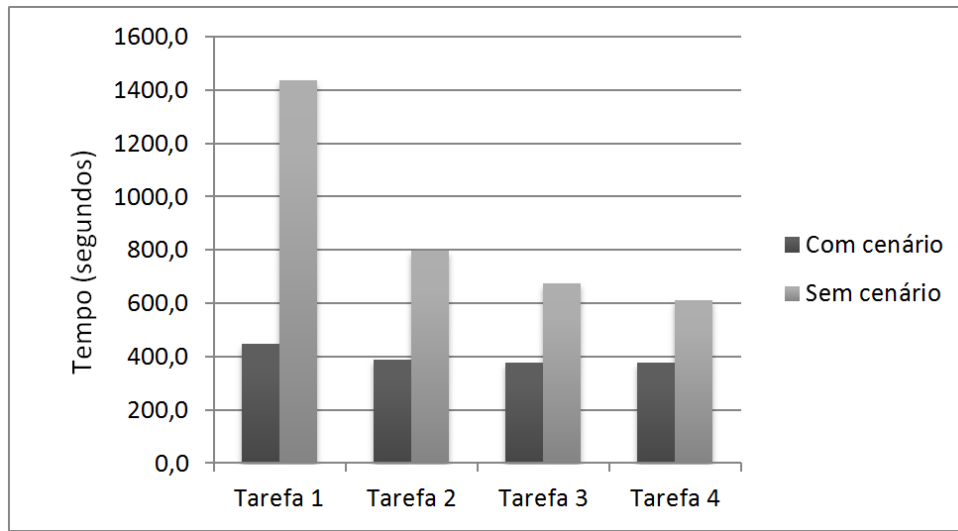


Figura 38. Grupo 2 gráfico do tempo de execução das tarefas.

Grupo1 + Grupo 2

A seguir, apresentamos os resultados do Estudo Biblioteca Digital obtidos a partir da análise do conjunto de dados do Grupo 1 e 2 consolidados. Esta é a análise principal a cerca dos dados obtidos neste estudo. Os grupos foram separados, apenas para fornecer um panorama do comportamento de dois perfis diferentes durante a realização das tarefas.

Correção. A Figura 39 apresenta um gráfico que relaciona o número de respostas corretas e incorretas com o tipo de abordagem avaliada. Cada barra do gráfico representa o número de tarefas executadas, utilizando cada abordagem, que neste caso foram 34 para cada uma (68 no total). A quantidade de tarefas corretas, em cada caso, está representada pela cor mais escura e a de incorretas pela cor mais clara. Do total de 68 tarefas, 28, o equivalente a 41,18%, foi realizado de maneira correta. Das 28 tarefas corretas, grande parte

foi realizada com a ajuda de cenários oriundos do PDS+T (20 - o que significa 71,43% das corretas). Portanto, temos indícios que, em um grupo de participantes com perfis diversos, a rastreabilidade e detalhamento, proporcionados pelos cenários, seus relacionamentos e integração com o código influenciaram positivamente a execução correta das tarefas de manutenção.

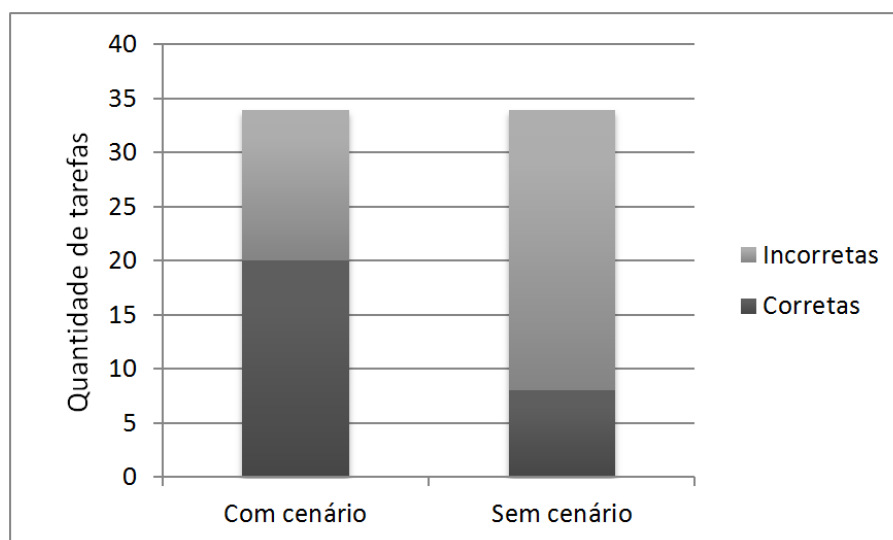


Figura 39. Gráfico de tarefas corretas Grupo 1 + Grupo 2.

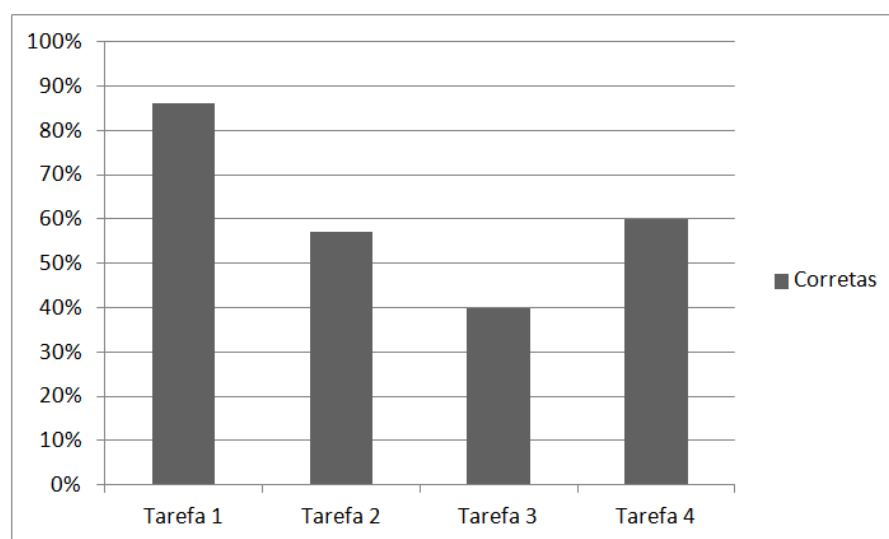


Figura 40. Percentual de tarefas corretas executadas no Biblioteca Digital com cenário.

As Figuras 41 e 40 apresentam os gráficos com percentual de tarefas executadas corretamente com e sem o auxílio dos cenários. Através destes gráficos, é possível observar que houve um aumento no percentual de tarefas

executadas corretamente em todos os casos, quando estas foram executadas com apoio de cenários oriundos do PDS+T. A tarefa 1, por exemplo, quanto executada com auxílio de cenários, teve uma taxa de acerto de, aproximadamente, 86%, isto é, de 7 tarefas executadas, 6 estavam corretas. Entretanto, na Biblioteca Digital sem apoio de cenários, a mesma tarefa teve uma taxa de acerto de 40%, isto é, de 10 tarefas executadas, apenas 4 foram realizadas de forma correta.

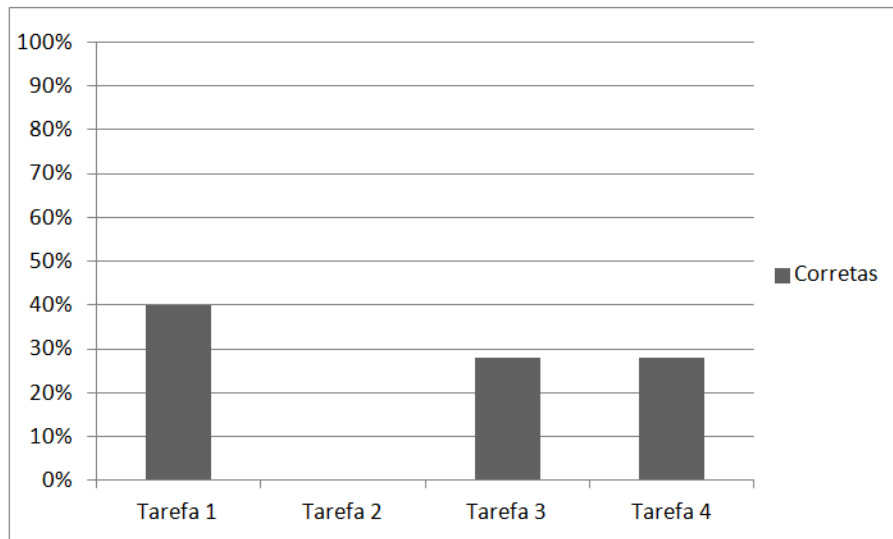


Figura 41. Percentual de tarefas corretas executadas no Biblioteca Digital sem cenário.

Tempo de execução. Exibimos, na Figura 42, o gráfico que relaciona as tarefas com seu tempo de execução (em segundos), realizando um comparativo entre a sua execução com e sem apoio de cenários provenientes do PDS+T. Neste gráfico, para cada tarefa, duas barras são exibidas. A mais escura, mostra a média dos tempos para conclusão da tarefa com apoio dos cenários. A segunda, mais clara, exibe a média dos tempos para conclusão da tarefa sem apoio dos cenários. Na média, tivemos uma redução de 35,43% no tempo de execução das tarefas, quando estas eram suportadas por cenários. Na tarefa 1, obtemos uma redução de 60,28%, a maior redução dentre todas. Já na tarefa 2 e 3, obtivemos, respectivamente, uma redução de 26,21% e 45,56%. E por fim, na tarefa 4, obtivemos a menor redução, que foi de apenas 9,68%. Assim, temos indícios que a rastreabilidade e detalhamento, proporcionados pelos cenários e sua integração com o código, possibilitaram a diminuição do tempo de execução das tarefas de manutenção no código do software.

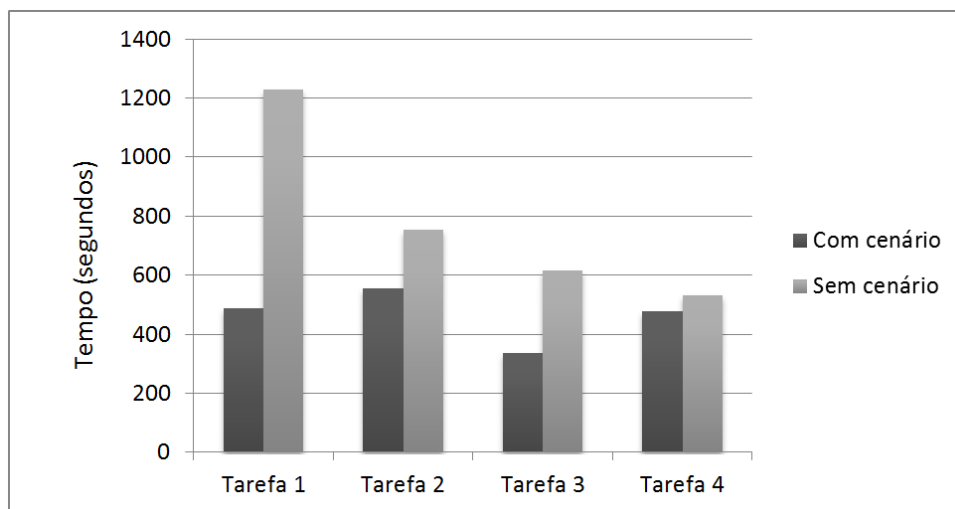


Figura 42. Gráfico do tempo de execução das tarefas Grupo 1 + Grupo 2.

Analisando os resultados obtidos através deste estudo, acreditamos ter indícios suficientes para afirmar que:

- A **presença** das qualidades de **rastreabilidade** e **detalhamento** no software, proporcionadas pelo processo PDS+T, **contribuem positivamente** para a realização de atividades de manutenção **mais eficazes** (corretas), quando comparadas a realização das mesmas tarefas em um software sem estas qualidades.
- A **presença** das qualidades de **rastreabilidade** e **detalhamento** no software, proporcionadas pelo processo PDS+T, **contribuem positivamente** para a realização de atividades de manutenção em **menor tempo**, quando comparadas a realização das mesmas tarefas em um software sem estas qualidades.

4.1.2.6.

Observações Gerais

A tarefa 4, proposta neste estudo, tem características particulares, que nos possibilitaram realizar algumas observações importantes. Nesta tarefa, é proposta uma modificação em um comportamento, que é muito similar a outros espalhados pelo código. O que acontece não é uma redundância de código, mas sim um comportamento que tem sutilezas, devido ao tipo de dado em que é

empregado. A tarefa citada, trata da mudança de exibição dos autores dos *proceedings* da conferência, que são diferentes dos autores dos artigos submetidos a conferência. No entanto, ambos são exibidos de maneira muito semelhante. O que podemos observar na consolidação dos dados, é que muitos indivíduos **confundiram os autores** dos *proceedings* com aqueles que pertenciam aos artigos e, portanto, sugeriram modificações erradas. Entretanto, **todos** os que **confundiram** fizeram a tarefa **sem o suporte dos cenários**. Nenhum daqueles que a realizou com os cenários cometeu este equívoco. Como exemplo, relatamos o caso de um dos indivíduos, que alegou possuir um bom conhecimento da linguagem Lua e padrão MVC. Este indivíduo acertou todas as tarefas, exceto a 4 - que foi realizada sem apoio de cenários - onde ele confundiu os autores dos artigos com os *proceedings*. O que percebemos é que os indivíduos normalmente utilizam a busca do editor de código, para procurar por uma palavra chave. Assim que encontram o que procuram, não analisam o código em volta para determinar se ali é realmente o ponto de interesse. Quando há cenários integrados, devido à proximidade do episódio com o código, o indivíduo acaba lendo o episódio e, desta forma, descobre se ali é ou não o ponto que lhe interessa. Esta observação é importante, pois nos dá indícios de que os cenários ajudam na compreensão do código, evitando que erros deste tipo sejam cometidos.

4.1.2.7.

Comentários dos Participantes do Estudo

Além do questionário para registrar a resposta para as tarefas de manutenção, foi disponibilizado aos participantes um espaço, onde puderam realizar comentários sobre o estudo realizado. Para proporcionar uma análise mais rica do estudo, assim como em (Simões, 2013), iremos expor os comentários que achamos relevantes e relatar nossas observações sobre eles.

"Os cenários me orientaram no código e no software, de modo que eu não perdi tempo tentando entender o que o código fazia e fui direto ao ponto" (P1)

Em seu comentário, o participante P1 fala sobre o direcionamento do seu esforço de entendimento ao que é realmente necessário, ao invés de precisar entender o código como um todo, apenas para encontrar o ponto de necessita de uma maior atenção. No nosso ponto de vista, isto evidencia a característica

de rastreabilidade do processo PDS+T, que permite um rastro mais pontual, entre episódios e código.

"Os cenários, ajudando a entender melhor o que acontecia, colaboraram para uma identificação mais rápida das mudanças pedidas." (P2)

No comentário do participante P2, há uma afirmação mais geral sobre os entendimento proporcionado pelos cenários. Acreditamos que este comentário aborde tanto a característica de rastreabilidade, quanto o detalhamento, que permite a explicação do código através dos cenários e seus episódios.

4.1.3. Estudo C&L

A seguir descreveremos o segundo estudo realizado para avaliar o processo PDS+T. O estudo foi realizado um grupo de 12 participantes, que são alunos de pós-graduação em informática da PUC - Rio. Os dados obtidos foram avaliados sob a ótica de **tempo** e **correção**.

4.1.3.1. Objetivos específicos

Neste estudo, o objetivo será avaliar o processo PDS+T, que tem como alguns de seus produtos a organização modular e arquitetura do software. A modularização é obtida através dos cenários iniciais, que são os requisitos do software. Estes passam por um refinamento para, posteriormente, compor a arquitetura do software, que segue o padrão arquitetural MVC.

Através deste estudo específico, pretendemos avaliar somente a influência da modularização e arquitetura (dependência), na realização de atividades de manutenção. Portanto, como será explicado a seguir, projetamos o estudo de modo que as outras características não tivessem influência.

Desta forma, através deste estudo pretendemos responder as seguintes questões:

Q₁: A presença da qualidade de **dependência**, proporcionada pelo processo PDS+T, contribui para a realização de atividades de manutenção mais eficazes (corretas), quando comparadas a realização das mesmas tarefas em um software sem esta qualidade?

Q₂: A presença da qualidade de **dependência**, proporcionada pelo processo PDS+T, contribui para a realização de atividades de manutenção em menor tempo, quando comparadas a realização das mesmas tarefas em um software sem esta qualidade?

4.1.3.2.

Participantes

O Estudo C&L contou com a participação de 12 indivíduos, alunos de pós-graduação (mestrado e doutorado) em informática da PUC - Rio. O perfil dos participantes do estudo foi determinado através de um questionário (Tabela 22), cujo objetivo foi determinar a experiência dos participantes em desenvolvimento de software, e o seu conhecimento nas tecnologias utilizadas (neste caso, linguagem Lua, PHP e MVC).

Experiência	
Gostaríamos de saber qual seu conhecimento prévio sobre tópicos de importância para o experimento. Por favor, a seguir informe para cada tópico quantos anos de experiência você possui na área.	
Desenvolvimento de software (faculdade, trabalho, escola)	Experiência anos _____
Desenvolvimento de software na indústria	anos _____
Desenvolvimento com linguagem Lua	anos _____
Desenvolvimento com linguagem PHP	anos _____
Qual seu conhecimento sobre o padrão de arquitetura MVC? <input type="checkbox"/> Nenhum <input type="checkbox"/> Pouco <input type="checkbox"/> Razoável <input type="checkbox"/> Bom <input type="checkbox"/> Muito Bom Você conhece o software C&L da perspectiva de desenvolvedor? <input type="checkbox"/> Sim <input type="checkbox"/> Não	

Tabela 22. Questionário para determinar perfil do participante.

Através da análise dos dados do questionário, pudemos observar que os participantes possuíam, em média, 8,1 anos de experiência de software no total (indústria e universidade) e 4,2 anos de desenvolvimento somente na indústria.

Vários participantes relataram já ter experiência em desenvolver com a linguagem PHP, totalizando 13 anos de experiência. Já na linguagem Lua, apenas 3 participantes relataram possuir experiência em seu uso, totalizando 3 anos.

Quanto ao conhecimento sobre o padrão MVC, os participantes informaram possuir, em média, um conhecimento razoável (aproximadamente 3 em uma escala de 1 a 5). A Figura 43 apresenta o perfil dos participantes do Estudo C&L de forma resumida.

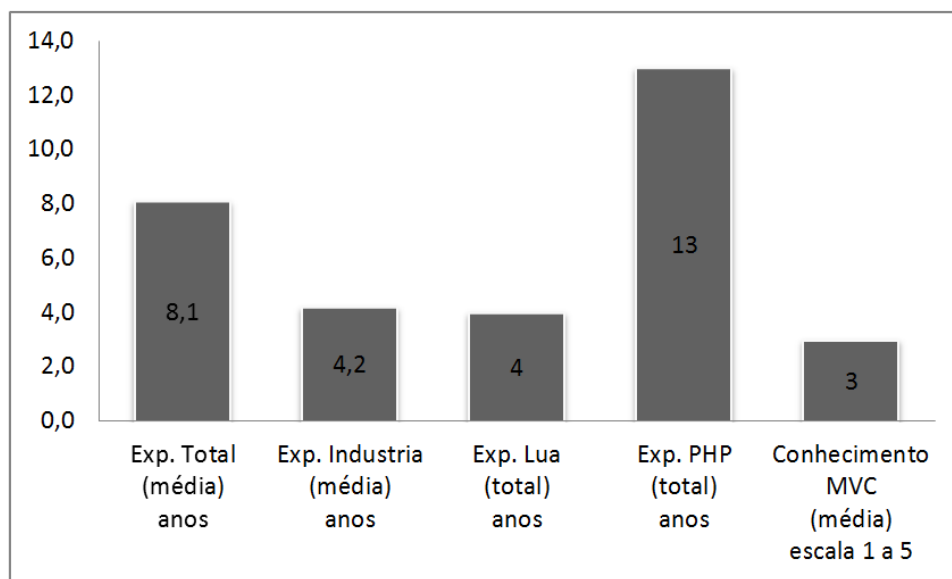


Figura 43. Perfil dos participantes do Estudo C&L.

4.1.3.3.

Projeto do experimento

Ambientes

Assim como anteriormente, criamos dois ambientes, **A** e **B**, para realização do estudo. O ambiente **A** é composto pela **versão PHP do software C&L**. Esta versão foi desenvolvida em PHP e possui cenários integrados ao código, embora não tenha modularização nem arquitetura semelhante àquelas propostas pelo processo PDS+T. Os cenários presentes em seu código foram colocados após sua implementação, como forma de documentação. Apesar de o software ter evoluído ao longo dos anos, sua arquitetura foi mantida a mesma.

O ambiente **B** é composto pelo C&L desenvolvido em Lua. Esta versão do software foi construída sem reutilizar nenhum componente da versão anterior.

Utilizou-se o processo PDS+T para construí-la a partir de cenários de requisitos. Desta forma, além dos cenários integrados ao código, temos, descrevendo o C&L em Lua, toda a estrutura resultante do processo PDS+T, que inclui a modularização, arquitetura, cenários de camadas e operacionais.

A principal diferença entre o C&L versão PHP e versão Lua é a modularização e arquitetura, já que ambos possuem cenários integrados ao código fonte.

Tarefas

A fim de avaliar a dependência (modularização e arquitetura), obtida através do processo PDS+T, propomos a criação de quatro tarefas de manutenção, uma para cada tipo identificado na taxonomia proposta por Kitchenham (Kitchenham et al, 99). Desta forma, assim como feito no estudo anterior, foram criadas tarefas de correção, mudança de requisito, mudança de implementação e novo requisito. A fim de explicar cada tarefa, criamos um texto simulando o que seria apresentado para um desenvolvedor em um caso real. A seguir descreveremos as tarefas propostas.

Tarefa 1: *Os usuários do sistema C&L estão reportando um erro ao cadastrar um símbolo no sistema. Segundo as mensagens, quando um símbolo é cadastrado no sistema, o nome informado pelos usuários não é considerado. Todos os símbolos estão sendo cadastrado com a string "nome" no lugar do nome.*

Esta correção simula um defeito simples no software. Apenas um componente é afetado. Entretanto, não é tão simples encontrar o defeito, pois este se encontra em um componente bem específico em ambas as versões. Para encontrá-lo, o indivíduo deve compreender o fluxo de informações do software.

Tarefa 2: *Para tornar a visualização dos sinônimos de um símbolo mais clara, um dos desenvolvedores sugeriu que eles sejam exibidos um em baixo do outro, no formato de lista, ao invés de lado a lado e separados por vírgula.*

Esta tarefa simula uma mudança de código sem alteração dos requisitos funcionais. Aparentemente, é uma simples mudança na interface do software,

porém devido a diferença entre como o dado é armazenado e apresentado, a informação requer uma lógica para sua apresentação. Neste caso, o indivíduo deve além de compreender o fluxo, entender uma boa parte da lógica que faz a montagem dos dados, a fim de realizar a mudança necessária.

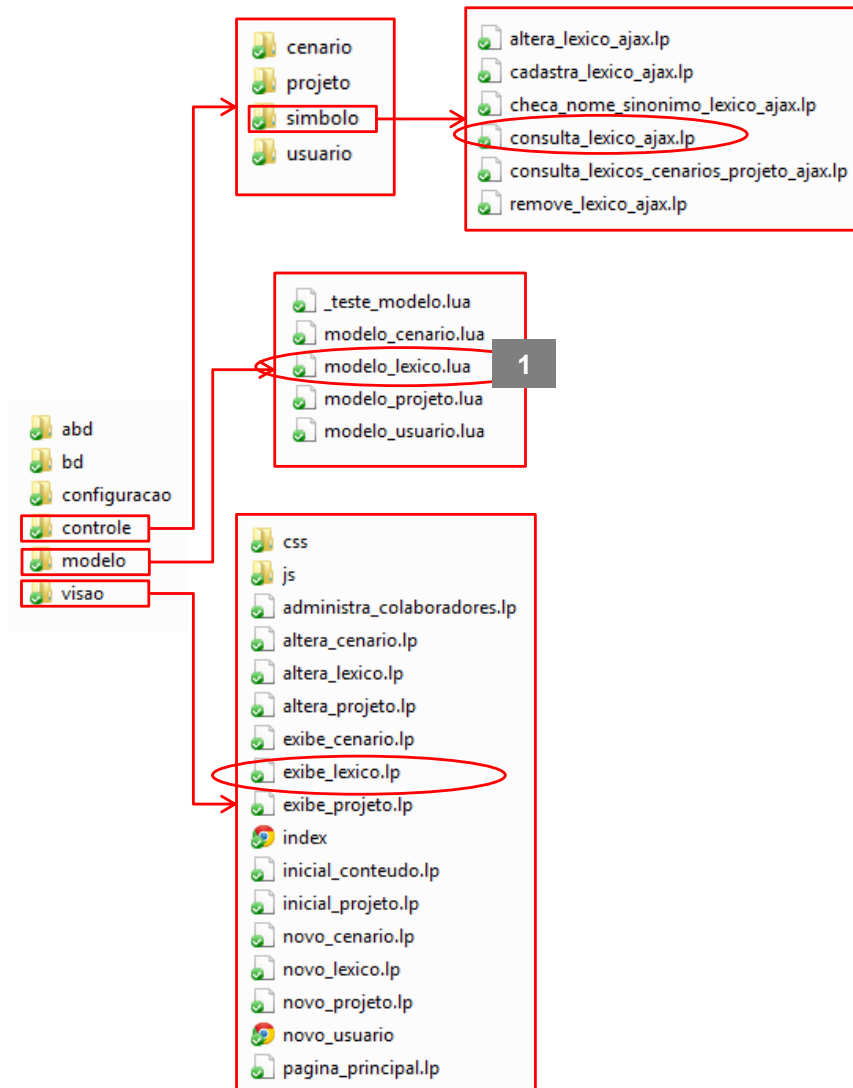


Figura 44. Passo 1 para realizar a tarefa 2 do Estudo B, no C&L em LUA.

As figuras 44 e 45 apresentam as etapas que são executadas para realizar a tarefa 2 no C&L em LUA. Utilizando a organização proporcionada pelo framework MVC, é possível identificar facilmente os arquivos relacionados as operações com os dados de símbolos do léxico. Como a mudança ocorre em um trecho de código que deve ter conhecimento sobre a estrutura de dados utilizada, identificamos que a mudança ocorrerá no arquivo de modelo (passo1).

Uma vez identificado o arquivo, procuramos o cenário responsável pela seleção dos sinônimos de um símbolo do léxico a partir do banco de dados (passo 2). Conhecendo o cenário que precisa ser alterado, basta olhar seus episódios para identificar exatamente o trecho de código que precisa ser modificado (passo 3). Assim que a mudança é realizada, é preciso verificar se o episódio que descreve o trecho necessitará de alteração, a fim de explicar corretamente o código. Neste caso, a mudança foi bem simples e o episódio teve que sofrer uma pequena alteração.

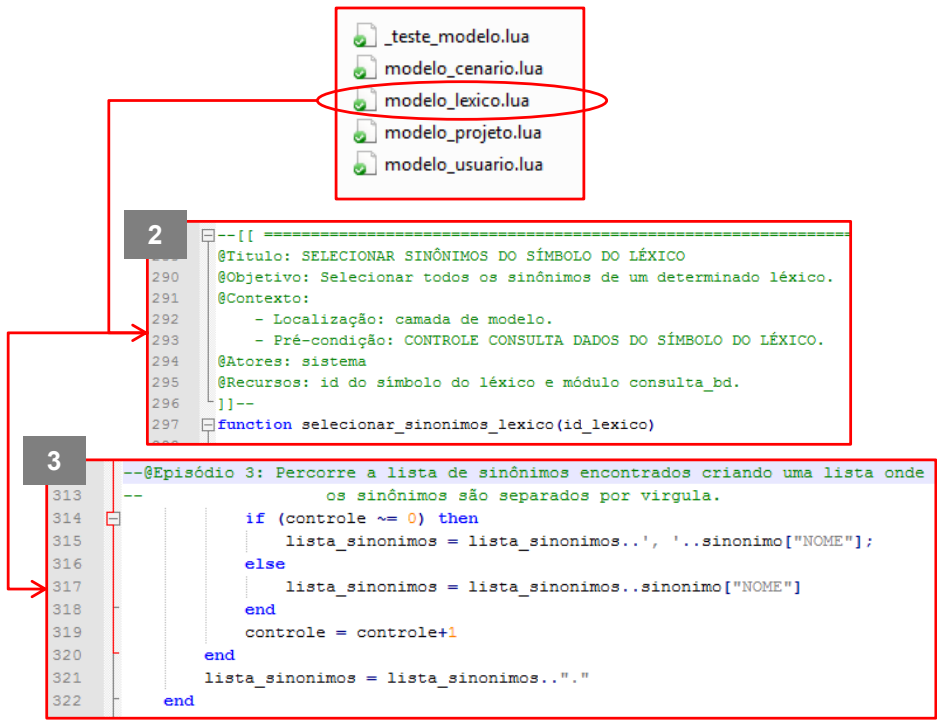


Figura 45. Passos 2 e 3 para realizar a tarefa 2 do Estudo C&L, no C&L em LUA.

Artefato/posição	Tipo de modificação	Pseudocódigo/descrição
Cenário: Selecionar sinônimos do símbolo do léxico - Episódio 3	modificar	Percorre a lista de sinônimos encontrados, criando uma lista onde os sinônimos são apresentados em linhas diferentes.
modelo_lexico.lp linha 315	incluir	Trocar "," por " "

Tabela 23. Resposta da tarefa 2 do Estudo C&L, para o C&L em LUA.

Na Tabela 23 exibimos o preenchimento esperado do questionário para que a tarefa 2 do Estudo C&L, executada no C&L em LUA, fosse considerada correta. Com podemos observar, além de informar a modificação necessária no código, o indivíduo também deveria informar a mudança no cenário. As respostas que não possuíam ambas as modificações descritas foram consideradas incompletas e, conseqüentemente, incorretas.

Através das figuras 46 e 47, descrevemos os passos necessários para executar a tarefa 2 no C&L em PHP. Como não temos uma arquitetura definida, o primeiro passo consiste em uma busca para encontrar o arquivo correto dentre todos os que compõem o software (passo 1).

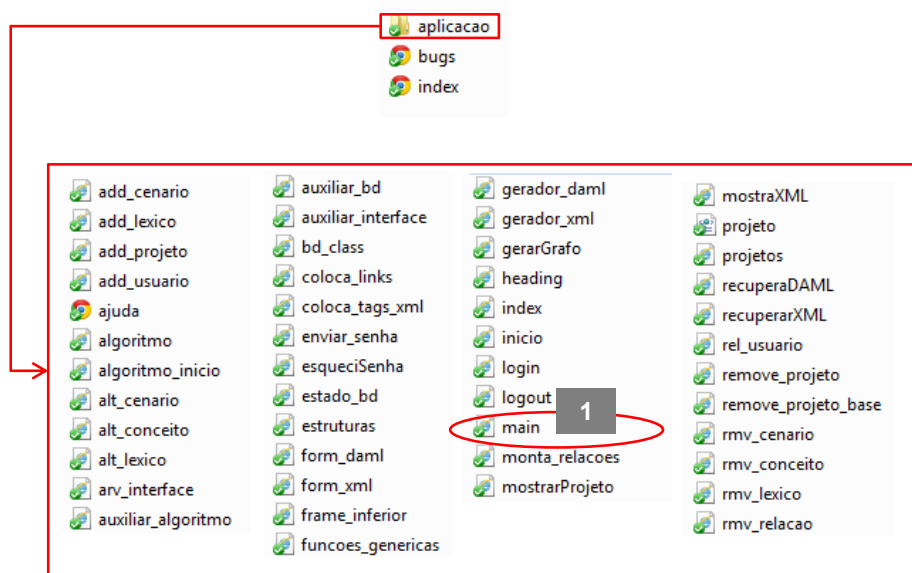


Figura 46. Passo 1 para realizar a tarefa 2 do Estudo C&L, no C&L em PHP.

Assim que o arquivo correto é encontrado, é preciso começar outra busca, desta vez pelo cenário que executa o comportamento procurado (passo 2). Novamente, como não temos uma arquitetura definida, os arquivos contêm funções relacionadas a todas as estruturas de dados do software. Uma vez encontrado o cenário, basta encontrar o episódio que descreve o trecho de código que deve ser mudado. Também é preciso verificar se o episódio que descreve o código necessitará de modificação.

Exibimos na Tabela 24 a resposta esperada para que a tarefa 2 do estudo B, realizada no C&L em PHP, fosse considerada correta. Esta resposta, como mostra a tabela, deveria ser composta de dois itens, um relacionado a modificação do código e o outro a modificação do cenário

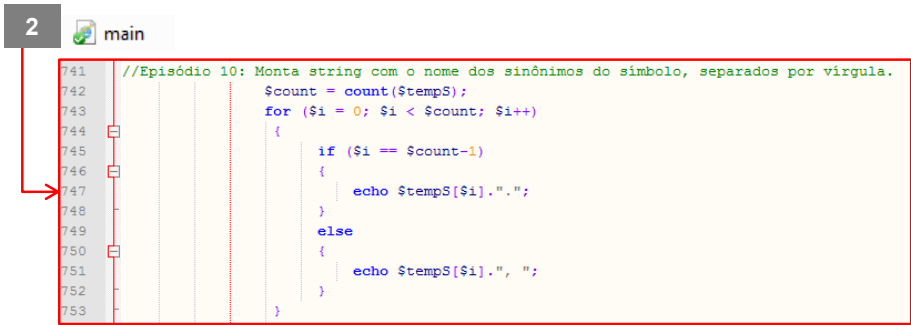


Figura 47. Passo 2 para realizar a tarefa 2 do Estudo C&L, no C&L em PHP.

Artefato/posição	Tipo de modificação	Pseudocódigo/descrição
Cenário: exibir léxico episódio 10	modificar	Monta string com o nome dos sinônimos do símbolo, um em cada linha.
Main.php linha 751	modificar	Trocar "," por " "

Tabela 24. Resposta da tarefa 2 do Estudo C&L, para o C&L em PHP.

Tarefa 3: Um dos stakeholders do software C&L deseja a adição de uma nova funcionalidade. O software deve permitir que o usuário importe cenários para um projeto específico a partir de arquivos externos. Neste caso, o usuário precisa selecionar o arquivo, que deve seguir um padrão específico.

Esta tarefa simula a adição de um novo requisito ao software. Seu impacto é grande, pois exige uma interface para selecionar o arquivo e a lógica para ler o arquivo e realizar as operações necessárias. Para executar esta tarefa, o indivíduo deve conhecer boa parte das funções do software relacionada aos cenários, a fim de reutilizá-las. Além disto, deve também conhecer a interface, para realizar corretamente a modificação requerida.

Tarefa 4: Houve uma mudança nos requisitos do software C&L. Agora, durante o cadastro de um novo usuário, o sistema deve exibir um campo para que o usuário informe a idade. Este novo dado deve ser armazenado no banco de dados do sistema, junto com os demais dados do usuário.

Esta tarefa tem como objetivo simular a mudança de um requisito. Esta é outra tarefa que aparenta ser simples, porém tem um grande impacto no software. Para realizá-la, é preciso conhecer o fluxo e entender o que é feito em cada etapa deste, a fim de realizar as modificações que são necessárias. Além disto, também é preciso alterar a interface, para permitir que o novo dado seja informado pelo usuário.

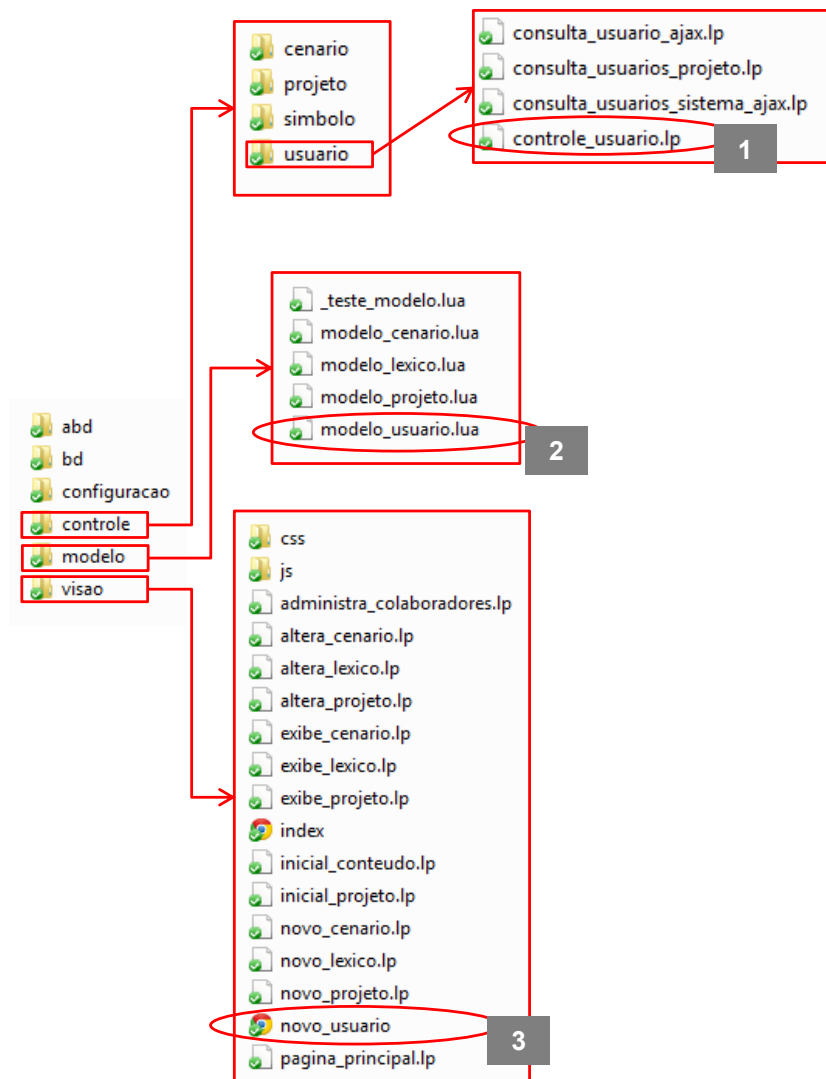


Figura 48. Passo 1 a 3 para realizar a tarefa 4 do Estudo C&L, no C&L em LUA.

Apresentamos através das figuras 48 e 49, os passos para realizar a tarefa 4, do Estudo C&L no C&L em Lua. Logo de início, é possível perceber que esta mudança tem impacto em todas as camadas do software MVC, pois exige a alteração de uma estrutura de dados. Sendo assim, iniciamos nossa busca na estrutura do software pelos componentes que serão afetados. Dada a arquitetura

e modularização, esta tarefa se torna menos difícil, como podemos ver através dos passos 1, 2 e 3.

Uma vez encontrados os arquivos, iniciamos a procura pelo código que deve ser alterado. Esta etapa também não é muito difícil. A interface do software para este fim é descrita através de um cenário. Assim, basta procurar o episódio que precisamos alterar (passo 4). O mesmo acontece com a camada de controle, que também é descrita por apenas um cenário (passo 4). Já a camada de modelo é composta de diversos cenários, todos manipulam informações de usuário. Neste caso, é preciso procurar pelo cenário que realiza o cadastro (passo 6). Assim que este cenário é encontrado, procuramos em seus episódios para determinar os que precisam ser alterados (passo 7).



Figura 49. Passo 4 a 7 para realizar a tarefa 4 do Estudo C&L, no C&L em LUA.

A Tabela 25 apresenta o preenchimento correto do questionário referente a tarefa 4, para o C&L em LUA. Como podemos observar, foi necessário realizar modificações em todas as camadas do software. Porém, apenas um cenário precisou ser alterado, aquele que descreve a interface que ganhou um novo campo.

Mostraremos através das Figuras 50 e 51, os passos necessários para realizar a tarefa 4, do Estudo B, no C&L em PHP. Iniciamos a tarefa através de uma busca para identificar os arquivos impactados pela mudança (passo1). Como não temos o auxílio de uma estrutura, temos que abrir e olhar cada um individualmente.

Artefato/posição	Tipo de modificação	Pseudocódigo/descrição
Cenário: exibir formulário para cadastro de usuário	incluir	Novo episódio: Exibir campo idade
Novo_usuario.html linha 66	incluir	<input name="input_idade_cd_usr" maxlength="100" size="60" type="text">
Controle_usuario.lp linha 31	incluir	["idade"] = cgilua.POST. input_idade_cd_usr,
Modelo_usuario.lua linha 29	incluir	..dados_usuario["idade"]

Tabela 25. Resposta da tarefa 4 do Estudo C&L, para o C&L em LUA.

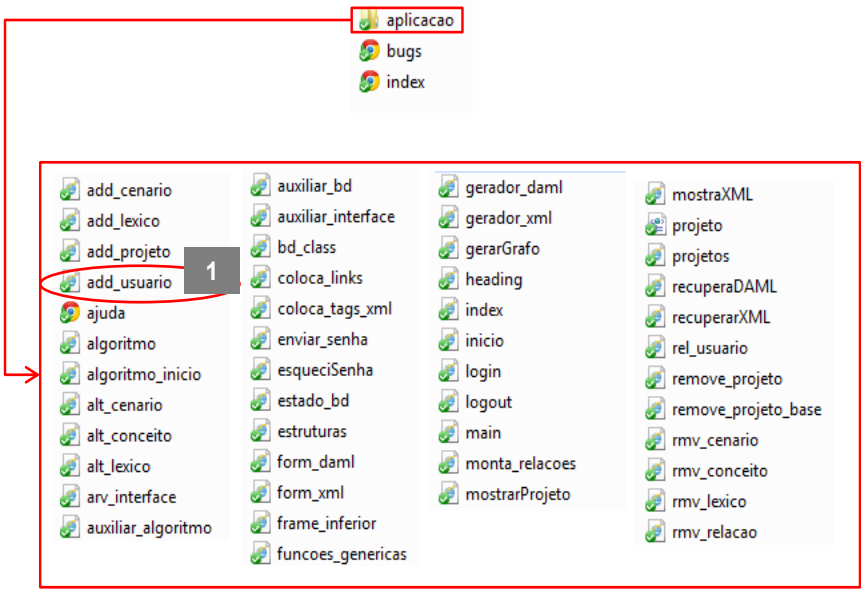


Figura 50. Passo 1 para realizar a tarefa 4 do Estudo C&L, no C&L em PHP.

Uma vez encontrados os arquivos, é preciso examiná-los para encontrar os cenários que devem ser alterados devido à mudança. Após encontrarmos

estes cenários, procuramos pelos episódios que descrevem o código que será modificado (passos 2 e 3).

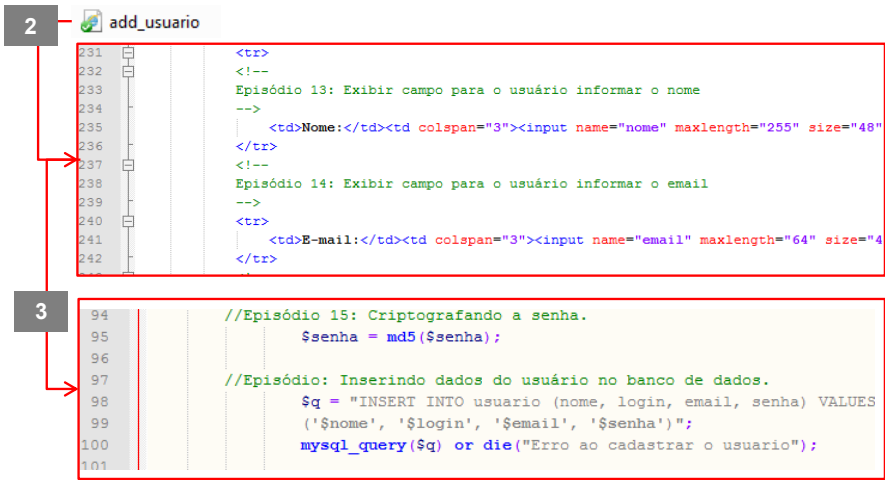


Figura 51. Passos 2 e 3 para realizar a tarefa 4 do Estudo C&L, no C&L em PHP.

No contexto do C&L em PHP, pode parecer que a tarefa 4 é mais simples, pois as alterações foram limitadas a um arquivo só. Porém, sabemos que isto não é totalmente verdade. A utilização de arquivos e cenários extensos prejudica a identificação das relações, pois estas não são expressas através de relacionamentos entre cenários. Desta forma, é preciso entender boa parte do código, para compreender o fluxo de execução do software.

Artefato/posição	Tipo de modificação	Pseudocódigo/descrição
Cenário: cadastrar usuário	incluir	Novo episódio: Exibir campo idade
Add_usuario.php linha 237	incluir	<input name="idade" maxlength="255" size="48" type="text" value="<?=\$idade?>">
Add_usuario.php Linhas 98-99	modificar	\$q = "INSERT INTO usuario (nome, idade, login, email, senha) VALUES ('\$nome', '\$idade', '\$login', '\$email', '\$senha')";

Tabela 26. Resposta da tarefa 4 do Estudo C&L, para o C&L em PHP.

Na Tabela 26, exibimos a resposta correta para a tarefa 4, do Estudo B, realizada no software em PHP. Como é possível observar, foi preciso modificar dois trechos do arquivo add_usuario.php, o responsável por exibir a interface e o

utilizado para executar a consulta de inserção no banco de dados. Além disto, foi preciso realizar a inclusão de um episódio, para descrever o novo campo incluído no formulário de cadastro de usuário.

Métricas

Assim com no estudo anterior, utilizaremos as métricas de **tempo** e **correção**, para avaliar o desempenho dos participantes na realização das tarefas. A medição do tempo será feita através do horário de início e fim, informado pelos participantes. Para avaliar se as tarefas foram corretas ou não, utilizaremos, como no estudo anterior, **três parâmetros**. O primeiro está relacionado ao impacto da tarefa. Para considerarmos a tarefa correta, o indivíduo deve ter citado todos componentes que necessitam de modificação. O segundo parâmetro está relacionado às modificações sugeridas. Para julgarmos a tarefa correta, o indivíduo deve ter indicado precisamente as modificações necessárias em cada componente. Por fim, será avaliado o pseudocódigo informado. Esta análise é mais subjetiva. Nela, iremos considerar qualquer solução que atenda a tarefa, mesmo aquelas que não sejam ótimas, desde que se mantenha a estrutura do software.

Possíveis ameaças

Uma das possíveis ameaças ao estudo seria o conhecimento adquirido pelo participante do software C&L, durante a execução da primeira parte do estudo, servir como vantagem para execução da segunda parte. Porém, acreditamos que isto não tenha causado interferência, pois o código das duas versões do software são completamente distintos. Além de utilizarem linguagens de programação distintas, a estrutura das duas versões é diferente.

Outra ameaça seria o conhecimento da linguagem influenciar a execução das tarefas. Como a linguagem PHP é muito mais utilizada que Lua para desenvolvimento Web, isto traria uma desvantagem para a versão desenvolvida em Lua. Esta característica pode ser observada nos participantes do estudo, que relataram possuir 11 anos (total) de experiência em PHP e apenas 4 anos (total) em Lua. Apesar de não termos tratado esta ameaça, conseguimos verificar, através da análise dos dados, que este fator não teve grande influência no estudo. Mesmo com maior conhecimento em PHP, na média, os participantes se saíram melhor nas tarefas executadas no código em Lua. Portanto, acreditamos

ter indícios de que este fator não tenha influenciado os resultados obtidos com o estudo.

4.1.3.4.

Realização do Estudo

Neste estudo, cada participante realizou as quatro tarefas de manutenção apresentadas anteriormente. Duas delas com utilizando como base o código fonte do C&L desenvolvido em PHP, e duas utilizando como base o código fonte do software C&L desenvolvido em Lua, que possui os artefatos gerados pelo PDS+T.

Desta forma, metade dos participantes executaram as tarefas 1 e 2 utilizando código do C&L em PHP, e as tarefas 3 e 4 utilizando código do C&L em Lua. A outra metade fez o inverso, isto é, executou as tarefas 1 e 2 com base no código do C&L em PHP, e as tarefas 3 e 4 com base no código do C&L em Lua. Desta forma, garantimos que, do total de tarefas, metade do foi realizada com base no código do C&L em PHP e a outra metade com base no código do C&L em Lua. Na Tabela 27, apresentamos a distribuição de tarefas entre os participantes do Estudo C&L. Como é possível observar nesta tabela, a distribuição de tarefas foi igual para os três últimos participantes. Isto aconteceu devido à desistência de um dos participantes durante o estudo, que realizaria as tarefas 1 e 2 no ambiente A. Como esta desistência ocorreu após a distribuição das tarefas, não foi possível equilibrá-la novamente.

Solicitamos aos participantes que, para cada tarefa, indicassem os componentes e o local em que necessitam de modificação. Dentre estes componentes, podemos ter arquivos, funções ou cenários. Os locais poderiam ser indicados através de número da linha ou elemento do cenário. Pedimos também, que informassem o tipo de modificação necessária, dentre três possíveis: inclusão, modificação e exclusão. Por fim, foi solicitada uma breve explicação em pseudocódigo da modificação necessária. Após o breve treinamento inicial (20 minutos) sobre como responder o questionário do experimento e uso do software C&L, não foi mais permitido aos participantes tirar dúvidas. A realização do experimento foi acompanhada de perto para assegurar que os participantes não se esquecessem de anotar o tempo de início e fim de cada tarefa.

Os dados do estudo foram obtidos através de um questionário, que continha campos para que os participantes informassem o horário de início e fim

de cada tarefa e todas as informações necessárias para indicar as modificações necessárias para sua implantação.

	Ambiente A (C&L PHP sem MVC)	Ambiente B (C&L Lua com MVC)
Participante 1	Tarefas 1 e 2	Tarefas 3 e 4
Participante 2	Tarefas 3 e 4	Tarefas 1 e 2
Participante 3	Tarefas 1 e 2	Tarefas 3 e 4
Participante 4	Tarefas 3 e 4	Tarefas 1 e 2
Participante 5	Tarefas 1 e 2	Tarefas 3 e 4
Participante 6	Tarefas 3 e 4	Tarefas 1 e 2
Participante 7	Tarefas 1 e 2	Tarefas 3 e 4
Participante 8	Tarefas 3 e 4	Tarefas 1 e 2
Participante 9	Tarefas 1 e 2	Tarefas 3 e 4
Participante 10	Tarefas 3 e 4	Tarefas 1 e 2
Participante 11	Tarefas 3 e 4	Tarefas 1 e 2
Participante 12	Tarefas 3 e 4	Tarefas 1 e 2

Tabela 27. Distribuição de tarefas entre participantes Estudo C&L.

4.1.3.5.

Resultados Obtidos

A seguir, mostraremos a análise dos dados do Estudo C&L. Os dados serão apresentados sob a ótica das métricas utilizadas: correção e tempo de execução. Em cada um dos casos, discutiremos sobre nossas conclusões.

Correção. A Figura 52 mostra o gráfico que relaciona o número de respostas corretas e incorretas com cada caso avaliado. Cada barra do gráfico representa o número de tarefas executadas, em cada caso, que neste estudo foram 24 (48 no total). A quantidade de tarefas corretas está representada pela cor mais escura e a de incorretas pela cor mais clara. Do total de 48 tarefas, 35 foram executadas de maneira correta (72,92%). Das 35 tarefas corretas, a maioria foi realizada com no software C&L em Lua (31 - o que significa 60% das corretas). Desta forma, concluímos que, a dependência (modularização e arquitetura) do C&L desenvolvido em Lua, obtida através do uso do processo PDS+T, influenciaram positivamente a execução correta das tarefas de manutenção.

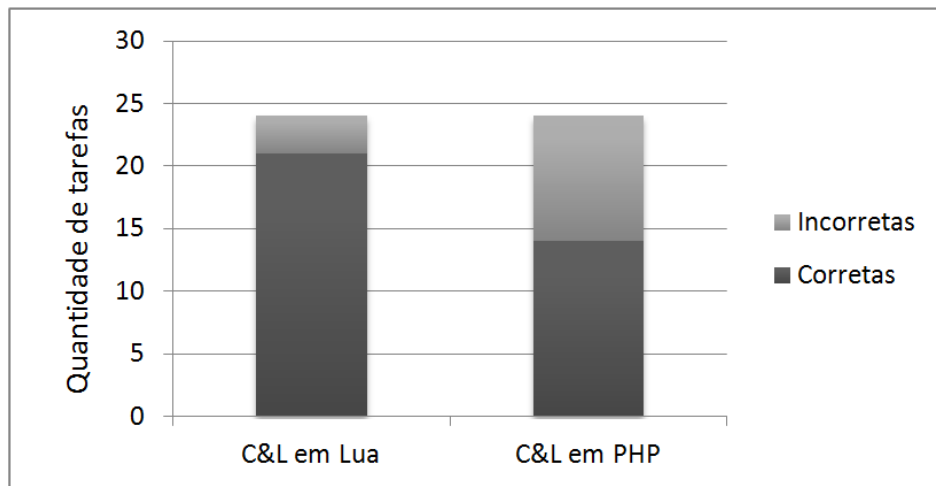


Figura 52. Gráfico de tarefas corretas do Estudo C&L.

As Figuras 53 e 54 apresentam os gráficos com percentual de tarefas executadas corretamente no C&L PHP e C&L Lua, respectivamente. Através destes gráficos, é possível observar que houve um aumento no percentual de tarefas executadas corretamente em todos os casos, quando estas foram executadas no C&L em Lua. A tarefa 4, por exemplo, quanto executada no C&L PHP teve uma taxa de acerto de 60%, isto é, das cinco vezes em que a tarefa 4 foi executada, em três vezes ela foi realizada de forma correta. Entretanto, no C&L Lua, a mesma tarefa teve uma taxa de acerto de 100%, isto é, foi realizada de forma correta nas sete vezes em que foi executada.

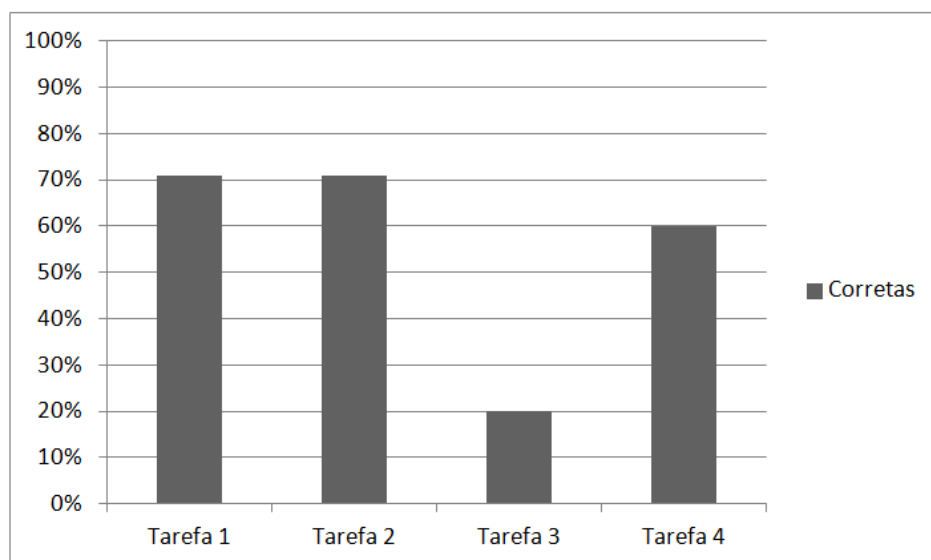


Figura 53. Percentual de tarefas corretas executadas no C&L PHP.

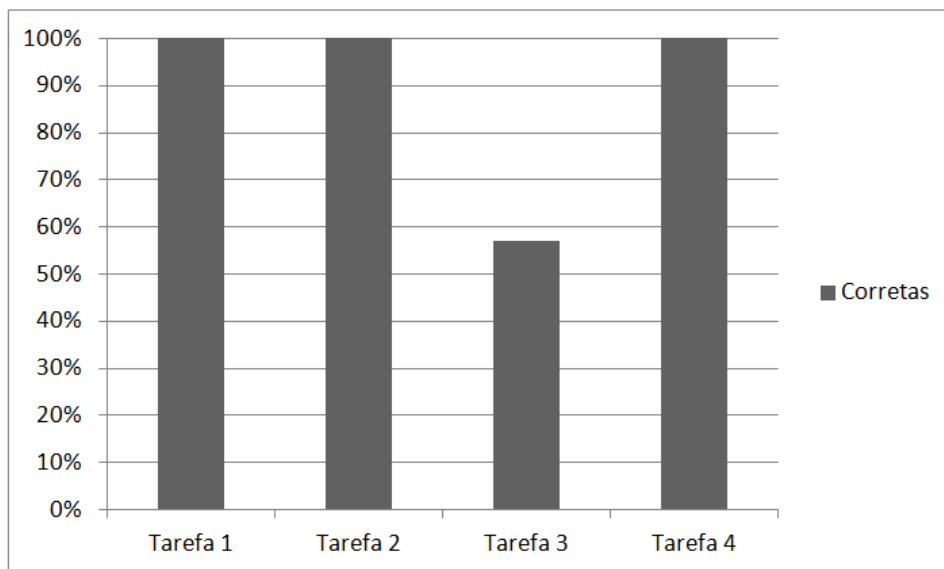


Figura 54. Percentual de tarefas corretas executadas no C&L Lua.

Tempo de execução. Apresentamos, na Figura 55, o gráfico que relaciona as tarefas com seu tempo de execução (em segundos), realizando um comparativo entre a sua execução com base no C&L em Lua e C&L em PHP. Neste gráfico, para cada tarefa, duas barras são exibidas. A mais escura, mostra a média dos tempos de execução da tarefa com base no código do C&L em Lua. A segunda, mais clara, exibe a média dos tempos de execução da tarefa com base no código do C&L em PHP. Na média, tivemos uma redução de 37,29% no tempo de execução das tarefas, quando estas foram executadas no C&L desenvolvido em Lua.

Realizando uma análise minuciosa do gráfico é possível observar que na tarefa 2, obtivemos uma redução menor no tempo que nas demais, apenas 17,50%. Acreditamos que isto tenha ocorrido porque, apesar de aparentar ser uma tarefa que exigiria modificação na camada de visão do software C&L em Lua, a alteração correta a ser feita era na camada de modelo. Isto porque, na arquitetura MVC, a visão não tem conhecimento da estrutura dos dados do software. Portanto, tal representação da informação não poderia ser construída nesta camada. Enquanto no C&L em PHP, não segue a arquitetura MVC, a modificação necessária deveria ser feita na própria interface do software.

Apesar desta particularidade, podemos concluir, de maneira geral, que a modularização, juntamente com arquitetura MVC, implementações da qualidade de dependência obtidas a partir do uso do processo PDS+T, diminuiriam o tempo

de execução das tarefas de manutenção, quando estas foram realizadas no software C&L desenvolvido em Lua.

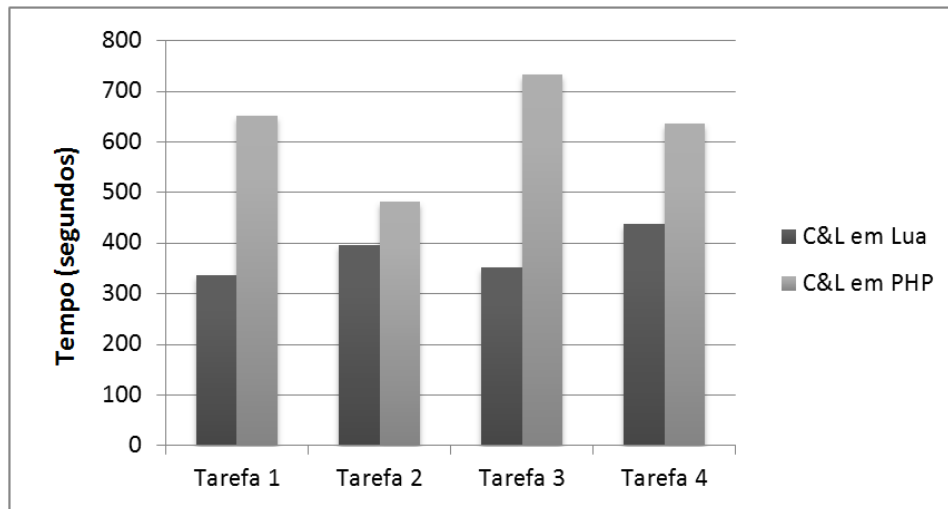


Figura 55. Gráfico do tempo de execução das tarefas do Estudo C&L.

Analizando os resultados obtidos através deste estudo, acreditamos ter indícios suficientes para afirmar que:

- A presença da qualidade de **dependência**, proporcionada pelo processo PDS+T, **contribui positivamente** para a realização de atividades de manutenção **mais eficazes** (corretas), quando comparadas a realização das mesmas tarefas em um software sem esta qualidade.
- A presença da qualidade de **dependência**, proporcionada pelo processo PDS+T, **contribui positivamente** para a realização de atividades de manutenção em **menor tempo**, quando comparadas a realização das mesmas tarefas em um software sem esta qualidade.

4.1.3.6.

Observações Gerais

Uma das características da versão do software C&L em PHP é que seus artefatos de código são muito extensos. Percebemos que mesmo com os cenários anotando o código, os participantes tinham dificuldade em identificar o

fluxo de ações do software. Os cenários não auxiliaram, pois o fluxo acontecia dentro dos próprios cenários, então não era possível utilizar os relacionamentos entre eles como suporte para identificá-lo. Isto evidenciou a importância da qualidade de dependência (modularização e arquitetura) na construção do software, além de seu relacionamento com a compreensão do software de uma maneira geral.

4.1.3.7.

Comentários dos Participantes do Estudo

Assim como no Estudo Biblioteca Digital, além do questionário para registrar a resposta para as tarefas de manutenção, foi disponibilizado aos participantes um espaço, onde puderam realizar comentários. Para proporcionar uma análise mais rica do estudo, assim como em (Simões, 2013), iremos expor os comentários que achamos relevantes e relatar nossas observações sobre eles.

"As mudanças no C&L em PHP dependem mais do conhecimento da linguagem" (P1)

Acreditamos que esta observação (P1) está relacionada aos cenários de camadas, que descrevem a estrutura do software. Como são em linguagem natural, não exigem que o desenvolvedor conheça profundamente a linguagem em que o software foi desenvolvido, para entender como ele está organizado. Para isso, basta que siga os relacionamentos entre cenários de camada.

"A arquitetura do C&L em Lua permite a rápida identificação das mudanças necessárias" (P2)

"A arquitetura MVC do C&L em Lua permite saber de antemão as mudanças necessárias" (P3)

"A organização em camadas (MVC) facilita a manutenção e navegação através do sistema" (P4)

As observações dos participantes P2, P3 e P4 tratam diretamente dos benefícios da arquitetura MVC implementada no C&L em Lua. Acreditamos que

facilidade de navegação se deve a presença dos relacionamentos entre cenários que detalham a estrutura do software, que através do software C&L podem ser seguidos através de um *link*.

"Os arquivos extensos do C&L em PHP dificultam a busca e identificação da linha a ser modificada" **(P5)**

Analisando o comentário do participante (P5) é possível perceber que, mesmo com anotações no software, indicando o relacionamentos de suas partes (relacionamentos dos cenários no código), se a qualidade de dependência não for atendida, o entendimento do software é prejudicado.

4.2. Conclusão

Neste capítulo, apresentamos os estudos feitos com o objetivo de avaliar o processo PDS+T. Os dois estudos realizados foram baseados na execução de tarefas de manutenção simuladas em dois software distintos. O primeiro estudo foi projetado de forma que as qualidades de rastreabilidade e detalhamento fossem avaliadas. Dois grupos de indivíduos, com perfis diferentes, participaram deste estudo. O primeiro grupo era composto por alunos de graduação, com pouca experiência em desenvolvimento de software. O segundo era composto, em sua maioria, por alunos de pós-graduação, com mais experiência em desenvolvimento de software. Os dados deste estudo foram, em um primeiro momento, analisados de forma separada, a fim de verificar o contraste dos resultados de acordo com o perfil dos participantes. Após esta avaliação, os dados foram consolidados e avaliados novamente. O segundo estudo foi projetado de maneira que a qualidade de dependência fosse avaliada. Este estudo foi realizado com apenas um grupo de participantes, que era composto de alunos de pós-graduação. O resultado dos estudos apontou uma melhora no desempenho das atividades manutenção nos software desenvolvidos através do PDS+T, o que nos permitiu concluir que o processo contribuiu positivamente para as qualidades avaliadas.