

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**ABORDAGEM DE ANÁLISE DO TEMPO DE RESPOSTA PARA  
TESTE DE DESEMPENHO EM APLICAÇÕES WEB**

PRISCILA GUARIENTI

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre em Ciência da  
Computação na Pontifícia Universidade Católica  
do Rio Grande do Sul.

Orientador: Prof. Dr. Avelino Francisco Zorzo

**Porto Alegre  
2014**



## Dados Internacionais de Catalogação na Publicação (CIP)

G915a Guarienti, Priscila  
Abordagem de análise do tempo de resposta para teste de desempenho em aplicações WEB / Priscila Guarienti. - Porto Alegre, 2013.  
130 p.

Diss. (Mestrado) – Fac. de Informática, PUCRS.  
Orientador: Prof. Avelino Francisco Zorzo.

1. Informática. 2. Engenharia de Software. 3. Software – Avaliação. I. Zorzo, Avelino Francisco. II. Título.

CDD 005.1

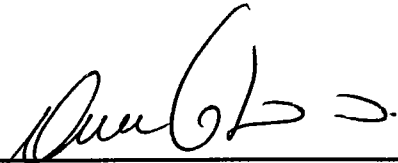
**Ficha Catalográfica elaborada pelo  
Setor de Tratamento da Informação da BC-PUCRS**

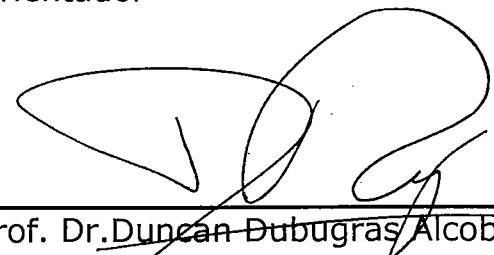


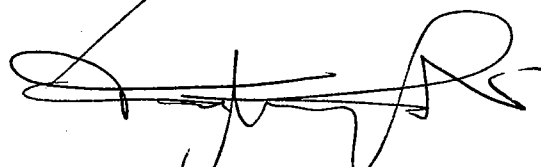


## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO


Dissertação intitulada "Abordagem de Análise do Tempo de Resposta para Teste de Desempenho em Aplicações *Web*" apresentada por Priscila Guarienti como parte dos requisitos para obtenção do grau de Mestra em Ciência da Computação, aprovada em 31/03/2014 pela Comissão Examinadora:

  
\_\_\_\_\_  
Prof. Dr. Avelino Francisco Zorzo - PPGCC/PUCRS  
Orientador

  
\_\_\_\_\_  
Prof. Dr. Duncan Dubugras Alcoba Ruiz - PPGCC/PUCRS

  
\_\_\_\_\_  
Prof. Dr. Fabian Luis Vargas - PPGEE/PUCRS

Homologada em 10/07/2014, conforme Ata No. 013 pela Comissão Coordenadora.

  
\_\_\_\_\_  
Prof. Dr. Luiz Gustavo Leão Fernandes  
Coordenador.

**PUCRS**

**Campus Central**

Av. Ipiranga, 6681 - P32- sala 507 - CEP: 90619-900  
Fone: (51) 3320-3611 - Fax (51) 3320-3621  
E-mail: [ppgcc@pucrs.br](mailto:ppgcc@pucrs.br)  
[www.pucrs.br/facin/pos](http://www.pucrs.br/facin/pos)



*"Você vê coisas e diz: Por quê?  
Mas eu sonho coisas que nunca existiram e digo: Por que não?"*  
George Bernard Shaw





## AGRADECIMENTOS

Os dois anos aqui dedicados foram resultado não só do esforço e dedicação próprios, mas também de tempos, espaços e pessoas que não puderam ser atendidos da forma como se desejava. Por isso um especial agradecimento:

À minha amada família...

Meus queridos pais, por todo apoio, em todos os momentos. Pela compreensão imensurável, pelo incentivo, pelo colo, por vibrar comigo, por acreditar nos meus sonhos, mas também por me lembrar os valores da vida. Em especial a minha mãe por decifrar meu tom de voz ao telefone e dizer as palavras que eu tanto precisava ouvir. Obrigada, esta conquista é muito mais do que minha, pois hoje sou aquilo que vocês me ensinaram a ser. Amo vocês!

Minha irmã Camila, por todo companheirismo, carinho e preocupação. Obrigada por sempre estar do meu lado e por me ouvir tantas vezes falar sobre minha pesquisa.

Ao meu orientador Avelino Francisco Zorzo, por sua paciência, pelos conselhos, pelos ensinamentos, por acreditar no meu trabalho e principalmente pelas discussões e críticas. Saiba que os desafios lançados colaboraram muito com o meu crescimento, tanto pessoal quanto profissional. Muito Obrigada AZ!

Agradeço ao apoio financeiro provido pela DELL, referente à bolsa de estudo e bolsa taxas. Também agradeço a oportunidade de desenvolver minha pesquisa como integrante junto ao projeto CoC (Center of Competence in Performance Testing) relativo ao Projeto de Desenvolvimento e Tecnologia da Informação (PDTI) no convênio DELL/PUCRS.

Aos colegas do Centro de Pesquisa em Engenharia de Software (CePES) e membros do grupo de pesquisa CoC Perf: Elder, Leandro, Artur, Valéria, Tiago e Anderson, não somente pelo trabalho, como também pelos momentos de descontração durante todo o período do mestrado. Agradeço em especial ao colega Maicon pelas revisões, as quais contribuíram para o desenvolvimento deste trabalho.

Agradeço ao professor Flávio Moreira de Oliveira, pois sempre manifestou claramente sua confiança e entusiasmo no trabalho desenvolvido no projeto.

Aos professores e ao programa de Pós Graduação da PUCRS.

Aos meus amigos, que mesmo distantes fisicamente sempre se mantinham presentes. As gurias de Porto Alegre: Adriana, Ildevana e Josi, obrigada pela companhia, pelas risadas e pelas conversas, saibam que com a ajuda de vocês foi muito melhor.

A Deus por ter me iluminado nos momentos de cansaço e me dado forças para chegar até aqui. Como diz o escritor Antoine de Saint-Exupéry: Não podemos conhecer as borboletas sem que, durante o percurso, tenhamos que suportar algumas lagartas.

A todos que durante este período de alguma forma foram importantes para a concretização de mais esta etapa.

Muito Obrigada!



# ABORDAGEM DE ANÁLISE DO TEMPO DE RESPOSTA PARA TESTE DE DESEMPENHO EM APLICAÇÕES WEB

## RESUMO

Com o avanço da tecnologia, os sistemas se tornaram cada vez maiores e mais complexos, e, conseqüentemente, mais suscetíveis a defeitos. Aliado a isso, estão presentes dois fatores cruciais no processo de desenvolvimento de *software*: os prazos de entrega dos projetos e a qualidade do produto, uma vez que a existência de pequenas falhas pode levar a prejuízos incalculáveis. Nesse aspecto, se torna indispensável incluir a atividade de teste como parte do projeto de *software*, a fim de garantir certo nível de confiabilidade. Porém, a realização da atividade de teste é geralmente bastante onerosa durante o desenvolvimento de um sistema de *software*. Isso ocorre porque boa parte dessas atividades são executadas manualmente. Nesse sentido, a abordagem de Teste Baseado em Modelos (*Model Based Testing - MBT*) é uma técnica que consiste na geração automática dos artefatos de teste com base em informações extraídas dos modelos de *software*. Isso tem sido uma alternativa para mitigar estes custos e tornar o processo de teste mais eficiente, reduzindo o tempo de geração e execução dos casos de teste. Neste contexto, a principal contribuição deste estudo é investigar a abordagem de MBT para teste de desempenho, relacionando estimativas de tempo aos modelos UML (*Unified Modeling Language*) e com base nessas estimativas medir o tempo de execução das sequências de teste. Outro fator é a definição dos Acordos de Nível de Serviço - SLAs, que tem como objetivo definir indicadores a serem atingidos que possam de fato serem medidos e assegurar que um serviço seja disponibilizado de forma correta e adequada. Diante disso, a partir de um cenário de teste é possível medir e monitorar o tempo de execução das sequências de teste avaliando se estão em conformidade com o SLA definido. Após a realização dos testes é possível coletar e analisar os resultados obtidos, além de comparar com os requisitos de tempo inicialmente estimados nos modelos. Nós aplicamos a nossa abordagem em duas em duas aplicações: TPC-W e Skills, e medimos seus tempos de execução usando a ferramenta LoadRunner. Além disso, aplicamos nossa abordagem também para um estudo de caso completo, usando a aplicação ChangePoint, e discutimos alguns dos resultados.

**Palavras-chave:** Teste Baseado em Modelos; Teste de Desempenho; Acordo de Nível de Serviço, Teste de Software.



# APPROACH TO ANALYSIS OF RESPONSE TIME PERFORMANCE TEST IN WEB APPLICATIONS

## ABSTRACT

Technology advances have allowed software systems to become more common in our society solving different types of problems we face everyday. This widespread use of software systems brought about some bigger and more complex solutions, which resulted in more systems that can fail. Furthermore, two important aspects in the software development process are being considered in every project: time to delivery and product quality. Since products have to be delivered faster, some faults may reside in the product and that could cause unforeseeable costs. In this scenario it is fundamental that the testing activity become one of the most important parts of the software development process to guarantee the quality of the final product. However, the testing activity is usually very expensive and time consuming, since most of the time it is executed manually. To improve that, Model Based Testing (MBT) has been used lately to generate testing artifacts automatically. These artifacts are usually extracted from the system model. This has helped to reduce the cost of testing, since the time to generate and execute tests can be reduced. Therefore, the main contribution of this work is to investigate how MBT can be applied for performance testing, i.e. how to include timing constraints in UML (Unified Modeling Language) diagrams and based on those constraints to execute the tests and verify whether the constraints are met or not. Another contribution of this work, is the setting of Service Level Agreements, i.e. set parameters to verify whether a system delivery answers based on pre-determined values. We applied our strategy to two sample applications, i.e. TPC-W and Skills, and measured their execution time using the LoadRunner Tool. Furthermore, we applied our whole strategy also to a complete case study, i.e. ChangePoint, and discuss some of our results.

**Keywords:** Model Based Testing; Performance Testing; Service Level Accord; Software Testing.



## LISTA DE FIGURAS

Figura 2.1	Modelo para falha, erro e defeito [1] . . . . .	28
Figura 2.2	Atividades do processo de teste baseado em modelos (adaptado de [2]) . . . . .	35
Figura 2.3	Modelo de Características da PLeTs [3] . . . . .	37
Figura 2.4	Etapas do processo de geração e execução de testes de desempenho na ferramenta LoadRunner [4] . . . . .	38
Figura 2.5	Estimativa de Tempo . . . . .	39
Figura 2.6	Ciclo de vida de um Acordo de Nível de Serviço [5] . . . . .	41
Figura 3.1	Processo para a realização de Teste de Desempenho . . . . .	44
Figura 3.2	Estereótipos e rótulos anotados no diagrama de casos de uso da aplicação <i>Skills</i> . . . . .	46
Figura 3.3	Estereótipos e rótulos anotados no diagrama de atividades da aplicação <i>Skills</i> . . . . .	47
Figura 3.4	Estrutura do arquivo XML Tempo Esperado . . . . .	51
Figura 3.5	Processo de geração dos casos de teste . . . . .	52
Figura 3.6	Diagrama de Casos de Uso da aplicação TPC-W . . . . .	54
Figura 3.7	Diagrama de atividades do caso de uso <i>Shopping</i> . . . . .	56
Figura 3.8	Cenário de teste gerado para o LoadRunner - TPC-W . . . . .	60
Figura 3.9	<i>Script</i> de teste gerado para o LoadRunner - TPC-W . . . . .	61
Figura 3.10	Gráfico de análise do tempo de resposta das transações <i>Search_Products</i> e <i>Buy_Products</i> do diagrama de atividades da Figura 3.7 . . . . .	62
Figura 3.11	Diagrama de Casos de Uso da aplicação <i>Skills</i> . . . . .	64
Figura 3.12	Diagrama de atividades do caso de uso Gerenciar Certificações . . . . .	66
Figura 3.13	Cenário de Teste gerado para o LoadRunner - <i>Skills</i> . . . . .	69
Figura 3.14	<i>Script</i> de teste gerado para o LoadRunner - <i>Skills</i> . . . . .	70
Figura 3.15	Gráfico de análise do tempo de resposta da transação <i>Editar_Certificações</i> do diagrama de atividades da Figura 3.12 . . . . .	70
Figura 4.1	Ambiente de teste de desempenho aplicação ChangePoint . . . . .	73
Figura 4.2	Diagrama de Casos de Uso da aplicação ChangePoint . . . . .	74
Figura 4.3	Diagrama de atividades do caso de uso Informar Horas Trabalhadas . . . . .	75
Figura 4.4	Subdiagrama de atividades da atividade <i>Login</i> . . . . .	76
Figura 4.5	Subdiagrama de atividades da atividade Informar Horas Trabalhadas . . . . .	77
Figura 4.6	Subdiagrama de Atividades da atividade <i>Logout</i> . . . . .	77
Figura 4.7	Cenário de teste gerado para o LoadRunner - ChangePoint . . . . .	81
Figura 4.8	<i>Script</i> de teste gerado para o LoadRunner - ChangePoint . . . . .	82
Figura 4.9	Gráfico de análise do tempo de resposta da transação Informar Horas Trabalhadas do diagrama de atividades da Figura 4.5 . . . . .	83

Figura A.1	Diagrama de atividades do caso de uso Associar/Desassociar Ordem aos Projetos . . . . .	95
Figura A.2	Subdiagrama de atividades da atividade Associar Ordem aos Projetos . . . . .	95
Figura A.3	Subdiagrama de atividades da atividade Desassociar Ordem aos Projetos . . . . .	95
Figura A.4	Diagrama de atividades do caso de uso Revisar Documentos . . . . .	96
Figura A.5	Subdiagrama de atividades da atividade Revisar Documentos . . . . .	96
Figura A.6	Diagrama de atividades do caso de uso Aprovar Recursos Humanos . . . . .	96
Figura A.7	Subdiagrama de atividades da atividade Aprovar Recursos Humanos . . . . .	97
Figura A.8	Diagrama de atividades do caso de uso Aprovar Custos do Projeto . . . . .	97
Figura A.9	Subdiagrama de atividades da atividade Aprovar Custos do Projeto . . . . .	98
Figura A.10	Diagrama de atividades do caso de uso Aprovar Projeto . . . . .	98
Figura A.11	Subdiagrama de atividades da atividade Aprovar Projeto . . . . .	99
Figura B.1	Diagrama de atividades do caso de uso Browsing . . . . .	101
Figura B.2	Diagrama de atividades do caso de uso Ordering . . . . .	101
Figura C.1	Diagrama de atividades do caso de uso Gerenciar Habilidades . . . . .	103
Figura C.2	Diagrama de atividades do caso de uso Gerenciar Experiências . . . . .	104
Figura C.3	Diagrama de atividades do caso de uso Alterar Senha . . . . .	104



## LISTA DE TABELAS

Tabela 3.1	Características e Estereótipos . . . . .	49
Tabela 3.2	Requisitos funcionais da aplicação TPC-W . . . . .	53
Tabela 3.3	Cenário de Teste da aplicação TPC-W . . . . .	55
Tabela 3.4	Requisitos funcionais da aplicação <i>Skills</i> . . . . .	63
Tabela 3.5	Cenário de teste da aplicação <i>Skills</i> . . . . .	65
Tabela 4.1	Distribuição da carga no cenário de teste de desempenho da aplicação . . . . .	75



## LISTA DE SIGLAS

AC	<i>Activity Diagram</i>
CePES	<i>Centro de Pesquisa em Engenharia de Software</i>
FSM	<i>Finite State Machine</i>
GUI	<i>Graphical User Interface</i>
HSI	<i>Harmonized State Identification</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
KSLOC	<i>Kilo Source Lines of Code</i>
MBT	<i>Model-Based Testing</i>
MEF	<i>Máquina de Estados Finitos</i>
MTTF	<i>Mean Time To Failure</i>
MTTR	<i>Mean Time To Recovery</i>
PERT	<i>Program Evaluation and Review Technique</i>
PLeTs	<i>Product Line for Model Based-testing tools</i>
PReWeb	<i>Prototype Research Web application in Netbeans platform</i>
QoS	<i>Quality of Service</i>
SEI	<i>Software Engineering Institute</i>
SGBD	<i>Sistema Gerenciador de Banco de Dados</i>
SLA	<i>Service Level Agreement</i>
SLG	<i>Service Level Guarantee</i>
SKILLS	<i>Workforce Planning: Skill Management Prototype Tool</i>
SPE	<i>Software Performance Engineering</i>
SPL	<i>Software Product Line</i>
SPT	<i>Schedulability, Performance and Time</i>
SUT	<i>System Under Test</i>
TPC-W	<i>Transactional Web e-Commerce Benchmark</i>
TPS	<i>Transações por Segundo</i>
UC	<i>Use Case Diagram</i>
UIO	<i>Unique Input Output</i>
UML	<i>Unified Modeling Language</i>
VV&T	<i>Validação, Verificação e Teste</i>

XMI  
XML

*XML Metadata Interchange*  
*Extensible Markup Language*

# SUMÁRIO

1. INTRODUÇÃO	23
1.1 Teste de <i>Software</i>	23
1.2 Objetivos	25
1.3 Organização	25
2. FUNDAMENTAÇÃO TEÓRICA	27
2.1 Teste de <i>Software</i>	27
2.2 Técnicas de Teste	29
2.3 Fases de Teste	30
2.4 Teste de Desempenho	32
2.5 Teste Baseado em Modelos	33
2.5.1 Processo MBT	34
2.5.2 Linha de Produto de <i>Software</i> para MBT	35
2.6 HP LoadRunner	37
2.7 Programa de Avaliação e Técnica de Revisão	39
2.8 Acordo de Nível de Serviço	41
2.9 Resumo	42
3. ABORDAGEM	43
3.1 Modelagem UML para Teste de Desempenho	45
3.2 Automação de SLAs para Teste de Desempenho	49
3.3 Processo de Geração dos Casos de Teste	51
3.4 Exemplo de Uso	53
3.4.1 Aplicação TPC-W	53
3.4.2 Aplicação <i>Skills</i>	63
3.5 Trabalhos Relacionados	71
3.6 Resumo	72
4. ESTUDO DE CASO	73
4.1 Configuração do Ambiente	73
4.2 Modelagem UML para Teste de Desempenho	74
4.3 Geração dos Cenários e <i>Scripts</i> de Teste de Desempenho	77
4.4 Análise e Interpretação dos Resultados	82
4.5 Resumo	84

5. CONCLUSÃO	85
5.1 Resumo . . . . .	85
5.2 Lições Aprendidas . . . . .	86
5.3 Limitações do Estudo . . . . .	86
5.4 Contribuições e Trabalhos Futuros . . . . .	86
5.5 Contribuições Finais . . . . .	87
REFERÊNCIAS	89
A. MODELOS DAS APLICAÇÃO CHANGEPOIT	95
B. MODELOS DA APLICAÇÃO TPCW	101
C. MODELOS DA APLICAÇÃO SKILLS	103
D. DADOS GERADOS NA EXECUÇÃO DOS TESTES	105

# 1. INTRODUÇÃO

## 1.1 Teste de *Software*

*"Man is, above all, he who creates.  
but it is hopelessly inadequate for showing their absence."*

Antoine de Saint-Exupéry

Cada vez mais as pessoas e as empresas fazem uso de programas de computadores para automatizar seus processos. Assim como cresce a necessidade de automatização dos processos e a complexidade dos sistemas computacionais, cresce também a exigência por sistemas e serviços confiáveis. Dessa forma, torna-se importante que durante o processo de desenvolvimento de *software* sejam aplicadas técnicas capazes de garantir que o sistema forneça um serviço confiável. A habilidade de entregar um serviço que pode, justificadamente, ser confiável é conhecida como dependabilidade (*dependability*) [1]. Os principais atributos que fazem parte da dependabilidade são definidos como: disponibilidade, confiabilidade, segurança, integridade e manutenibilidade. Conforme descrito em [1], muitas técnicas têm sido desenvolvidas com o intuito de atingir estes atributos e são representadas por quatro categorias principais: 1) Prevenção de Falhas (*Fault Prevention*) - prevenir a ocorrência ou introdução de falhas; 2) Tolerância a Falhas (*Fault Tolerance*) - evitar a ocorrência de defeitos nos serviços na presença de falhas; 3) Remoção de Falhas (*Fault Removal*) - reduzir o número de falhas e gravidade das falhas; 4) Previsão de Falhas (*Fault Forecasting*) - estimar o número atual, a incidência futura, e as prováveis consequências das falhas. Embora as quatro técnicas sejam usadas para atingir dependabilidade, o teste de *software* apresenta papel fundamental, pois é uma das técnicas mais utilizadas pela indústria de desenvolvimento de *software* na remoção de falhas.

No entanto, devido à evolução e o aumento da complexidade dos sistemas computacionais, a etapa de teste de *software* tem se tornado uma atividade tão ou mais complexa que a própria etapa de desenvolvimento em si, além de ser a mais custosa. Por este motivo torna-se indispensável à adoção de métodos, técnicas e ferramentas que possibilitem a realização das atividades de teste de maneira automatizada, aumentando a qualidade e a produtividade e diminuindo os custos [6]. Além disso, é importante que a etapa de teste não seja realizada apenas no final do processo de desenvolvimento do *software*, quando os requisitos já foram implementados, mas sim durante toda a etapa de seu desenvolvimento, desde a concepção até a entrega final do produto de *software*.

Uma das técnicas que auxiliam o processo de teste de *software* é o teste baseado em modelos (*Model Based Testing - MBT*) [7], que tem o intuito de criar artefatos de teste que descrevem o comportamento do sistema, incluindo em sua especificação as características do *software* que serão testadas [2], com o objetivo de automatizar o processo de teste de *software*. Esta técnica consiste na geração dos casos e *scripts* de teste baseado nos modelos do *software*. Além disso, o uso de MBT facilita as atividades desempenhadas pelos testadores, proporcionando maior eficácia e qualidade no

processo de teste, reduzindo a probabilidade de má interpretação dos requisitos do sistema como também o tempo de execução dos testes.

Outro fator importante relacionado com a qualidade de *software* é o desempenho. Do ponto de vista do usuário final, um *software* possui bom desempenho quando apresenta um tempo de resposta aceitável mesmo quando submetido a um volume de processamento próximo de situações reais [8]. Com o intuito de melhorar a qualidade do produto de *software*, a Engenharia de Desempenho de Software (*Software Performance Engineering - SPE*) [9] aplica seus esforços para melhorar o desempenho de duas formas: no ciclo inicial baseado em modelos preditivos ou no ciclo final baseado em medições. Além disso, o teste de desempenho abrange atividades como: configurar o ambiente e o cenário de teste; identificar critérios de aceitação de desempenho; planejar e projetar os testes; executar os testes e analisar os resultados [8].

Neste contexto, a abordagem proposta pelo presente trabalho enfatiza o teste baseado em modelos para desempenho, como uma forma uniforme e formal de associar informações de tempo aos modelos UML. Posteriormente, com a execução dos casos de teste de desempenho será possível coletar e analisar os resultados obtidos, além de comparar com os requisitos de tempo inicialmente estimados nos diagramas UML. Isso possibilita ter estimativas de tempo mais concretas e mensuráveis de acordo com o cenário em que o teste está sendo submetido, além de verificar a aceitabilidade do usuário em relação aos critérios de tempo estabelecidos nas SLAs.

Além das contribuições nas áreas de teste baseado em modelos e teste de desempenho, a proposta deste trabalho busca contribuir no desenvolvimento de uma Linha de Produto de *Software* (*Software Product Line - SPL*) [10] para geração de ferramentas de teste que usam MBT, chamada PLeTs (*Product Line for Model Based-testing tools*) [3] [11] [12] [13].

Uma SPL é uma família de sistemas de *software* que compartilham um conjunto de características comuns. O modelo tradicional de *software* é o desenvolvimento de um sistema único, enquanto a abordagem de linha de produto de *software* estende este modelo para uma família de sistemas de *softwares*. Para projetar uma SPL é importante identificar quais são as características comuns entre os sistemas que podem ser derivados a partir da linha. Essa arquitetura é composta pelas seguintes características [14]:

- *Comuns*: requeridas por todos os membros da família;
- *Opcionais*: referenciadas por apenas alguns membros da família, e;
- *Variantes*: (alternativos) que determinam sistemas com versões diferentes dos demais sistemas da família.

De acordo com [15], SPL é importante e aplicável ao desenvolvimento de *software*, pois proporciona diversos benefícios durante o processo de desenvolvimento, tais como: menor custo, melhor qualidade, maior produtividade e maior velocidade de entrada em novos mercados. Isso é possível devido a flexibilidade que os componentes de *software* possuem na arquitetura da família possibilitando assim a reusabilidade.



Desta forma, este trabalho teve como intuito auxiliar durante a fase de teste de *software*, prevendo situações e cenários que possam comprometer o desempenho de determinada aplicação e permitir a análise do tempo de resposta dos casos de teste. Para descrever estes aspectos foram definidos alguns estereótipos UML representando informações de tempo e que serão utilizados no processo de modelagem, geração e execução dos *scripts* de teste de desempenho utilizando a PLeTsPERF [12] que é uma ferramenta derivada da linha de produto PLeTs.

## 1.2 Objetivos

Esta dissertação tem como objetivo geral investigar métodos de geração de sequências de teste por meio de MEF, que possibilitem a geração automática de casos e *scripts* de teste de desempenho, aplicando a abordagem MBT em SPL. A pesquisa se concentrou na análise do tempo de resposta em aplicações *Web*. Portanto, este trabalho teve como base os seguintes objetivos:

- Identificar as principais características e requisitos relacionados ao Teste Baseado em Modelos e Teste de Desempenho.
- Utilizar modelos para apoiar o Teste de Desempenho.
- Prover um conjunto de características para que sejam gerados os casos de teste e *scripts* de teste de desempenho.
- Implementar ou adaptar o método de geração de sequências de teste HSI para a extração de informações e geração de casos de teste de desempenho baseado em modelos.
- Realizar um estudo de caso para validar a abordagem.
- Analisar os resultados obtidos com o estudo de caso.

## 1.3 Organização

Esta dissertação está organizada em cinco capítulos estruturados da seguinte maneira: o Capítulo 1 introduziu uma visão geral sobre o trabalho, a motivação e os objetivos desta pesquisa. o Capítulo 2 apresenta uma contextualização sobre Teste Baseado em Modelos, Teste de Desempenho, Linha de Produto de *Software* para MBT, HP LoadRunner, Técnica de Avaliação e Revisão de Programas e Acordo de Nível de Serviço. No Capítulo 3 é apresentada a abordagem proposta na dissertação, além de exemplos de uso e trabalhos relacionados. No Capítulo 4 é apresentado um estudo de caso com uma aplicação específica, onde posteriormente serão analisados e interpretados os resultados. E, finalizando, no Capítulo 5 são apresentadas as considerações finais, descrevendo as lições aprendidas as limitações do estudo e as perspectivas para trabalhos futuros.



## 2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo descrever os aspectos teóricos que facilitem a compreensão de alguns conceitos posteriormente abordados no desenvolvimento deste trabalho e que são fundamentais na área de teste de *software*.

### 2.1 Teste de *Software*

*"Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence."*

Edsger Dijkstra

Teste de *software* é um processo que visa, intencionalmente, encontrar defeitos de um programa ou sistema durante sua execução [16]. Pode ser definido ainda, como um processo que envolve toda atividade que tem por objetivo avaliar um requisito de um programa ou sistema e determinar que ele atenda os resultados esperados [6]. Teste de *software* é o processo de execução do sistema, ou de algum componente do sistema, aplicado em um ambiente controlado, observando ou registrando o comportamento do mesmo, a fim de analisar determinados aspectos do sistema ou do componente [17].

Além disso, o processo de teste de *software* é visto efetivamente como destrutivo, pois tem o objetivo de encontrar defeitos. Um caso de teste bem sucedido é aquele que na sua execução é capaz de fazer com que ocorram defeitos no programa e/ou sistema caso eles existam.

Outra definição de teste de *software* está diretamente relacionada com a mensuração da qualidade do produto de *software* [18]. Testar o produto é inerente ao processo de desenvolvimento e se faz altamente necessário para a garantia da qualidade do produto de *software* e para a identificação de defeitos, diminuindo assim custos de manutenção e mitigando o risco do usuário encontrar defeitos. Qualidade se traduz no cumprimento de requisitos, ou seja, é o nível em que um sistema, componente ou processo satisfaz os requisitos especificados pelas necessidades ou expectativas de clientes, usuários ou *stakeholders* [17] [19].

Atualmente existem algumas divergências entre autores no que se refere à definição dos conceitos de alguns princípios básicos que permeiam esta área do conhecimento, tais como: Falha (*Fault*), Erro (*Error*), Defeito (*Failure*) [1]. Com intuito de facilitar o entendimento desses conceitos será utilizada uma nomenclatura definida por autores da área de confiabilidade de sistemas, e que serve de referência para a comunidade científica:

- *Falha* é a causa hipotética de um erro. Está diretamente relacionada com a aplicação, ou seja, quando o sistema não está em conformidade com o que foi descrito e especificado para ele

nas etapas iniciais do projeto. Isto pode ocorrer, e.g., quando o sistema não atende alguma de suas especificações funcionais.

- *Erro* é o desvio que faz com que o serviço entregue seja diferente do especificado; um estado do sistema em que algum processamento a partir deste estado pode ocasionar um defeito. Isto acontece quando ele alcança algum estado externo do sistema.
- *Defeito* por sua vez é gerado a partir de erros do sistema, ou seja, o defeito é caracterizado por um estado inesperado do serviço ou sistema conforme a especificação definida pelo usuário. Alguns erros podem nunca serem gerados, no entanto, um defeito sempre é causado por um ou mais erros.

A Figura 2.1 a seguir mostra a relação existente entre os conceitos de falha, erro e defeito.

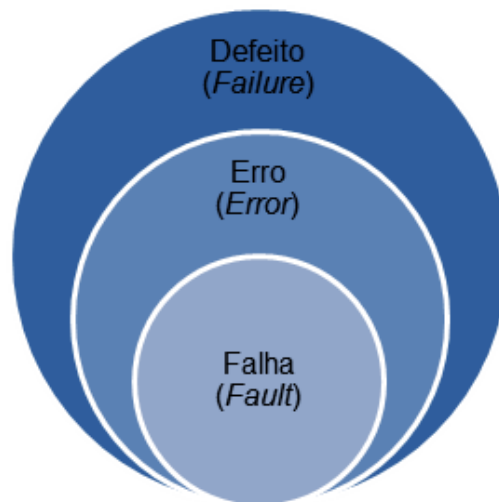


Figura 2.1: Modelo para falha, erro e defeito [1]

Diante disso, a construção de *software* com qualidade requer a combinação de atividades relacionadas ao planejamento e acompanhamento do projeto ao longo do desenvolvimento do produto de *software*, para isso existe uma série de artefatos que formalizam a realização do processo de teste de *software*. Algumas definições que conceituam estes artefatos são abordados em [17] [18] [20].

- *Plano de Teste* descreve o planejamento da atividade de teste, definindo o escopo, as técnicas que serão implementadas, o cronograma das atividades e os recursos necessários para realizar cada uma delas, bem como os responsáveis pela execução; envolve também o gerenciamento de riscos que requer um plano de contingência.
- *Cenário de Teste* descreve o conjunto de casos e/ou *scripts* de teste e a sequência que estes devem ser executados. Especifica um objetivo a ser alcançado com a execução do teste. Para cenários de teste de desempenho há uma distribuição da carga aos diferentes casos e/ou *scripts* de teste.

- *Caso de Teste* tem como pressuposto avaliar o cumprimento de um requisito de *software*, isto é, um conjunto de entradas que combinadas com determinadas condições e procedimentos são executadas pelo testador, a fim de determinar um critério de sucesso ou defeito de acordo com a especificação.
- *Especificação de Teste* é o documento que contém a especificação de um requisito ou caso de uso que deve ser satisfeito por um ou vários casos de testes.
- *Projeto de Teste* é o documento que define a especificação dos testes a serem realizados. Contém a descrição do conjunto de testes, detalhando os aspectos de execução do plano de teste.

Esses artefatos são desenvolvidos nas diferentes fases do projeto de *software*, com o intuito de registrar e acompanhar a evolução do projeto nas suas diversas etapas (iniciação, planejamento, controle e monitoração, execução e encerramento), verificando assim se os resultados alcançados estão de acordo com o que foi especificado.

Na próxima seção, será detalhado o processo de teste de *software* apresentando as diferentes técnicas de teste utilizadas.

## 2.2 Técnicas de Teste

Um teste de *software* pode ser aplicado a um sistema por diversas razões: identificar falhas, verificar a eficiência e o desempenho, além de representar a revisão final da especificação do projeto. Embora os paradigmas de desenvolvimento de *software* tenham evoluído, algumas técnicas de teste no qual um sistema computacional pode ser submetido persistem até hoje e são descritas a seguir:

- *Teste Funcional*: o teste funcional, conhecido também como teste caixa preta (*black box*) [20], ou ainda *data-driven*, *input/output driven* ou teste baseado em especificação é uma técnica que deriva casos de teste a partir da especificação do programa. Esta técnica busca avaliar o comportamento externo do programa e não detalhes internos de programação, tendo como principal objetivo avaliar o conjunto de saídas a partir de um conjunto de entradas, verificando se o resultado obtido corresponde ao resultado esperado. O teste funcional é capaz de identificar qualquer defeito de um programa ou aplicação, no entanto, para que isso aconteça é necessário aplicar todas as entradas possíveis, neste caso, o teste é chamado de teste exaustivo. Entretanto, devido à complexidade e o tamanho dos sistemas, o conjunto de entradas pode ser amplo, inviabilizando o processo de teste. Para que o processo se tornasse mais sistemático, foram definidos alguns critérios para contornar esta limitação, e.g., particionamento por equivalência, análise de valor limite e grafo causa-efeito [16] [20].
- *Teste Estrutural*: o teste estrutural também conhecido como teste caixa branca (*white box*) [20], ou caixa de vidro (*glass-box*), tem por objetivo testar a estrutura interna de partes ou de

componentes do sistema, estando intrinsecamente vinculado aos detalhes do código fonte, colocando a prova elementos de controle de fluxo, tais como: condições, repetições, variáveis e caminhos. No entanto, apesar de testados os possíveis caminhos mapeados de execuções do programa, algum caminho pode não ter sido identificado o que não garante que o programa pode vir a falhar [16] [21]. O teste estrutural é relevante para fatores de qualidade de *software* como manutenibilidade, estrutura e confiança, pois inclui casos de teste que não são avaliados por meio de testes funcionais. O teste estrutural é classificado em três categorias principais [21]: 1) Critérios baseados em fluxo de controle: para gerar casos de teste utiliza análise do fluxo de controle, avaliando laços, desvios e condições, ou outros elementos do fluxo de controle do *software*; 2) Critérios baseados em fluxos de dados: para gerar os casos de teste, utiliza análise do fluxo de dados do programa, esses critérios baseiam-se nas associações entre a definição de uma variável e o seu uso para a derivação de casos de testes; 3) Critérios baseados na complexidade: para levantamento dos requisitos de teste fundamentam-se em informações e critérios de complexidade do programa.

### 2.3 Fases de Teste

Diversos fatores podem estar relacionados a problemas encontrados em um *software*. Porém a maioria deles tem como origem o erro humano. Para atingir a dependabilidade e identificar estes possíveis erros antes da entrega do produto final, o *software* deve passar por inúmeros processos de Validação, Verificação e Teste (VV&T) [21]. Processos VV&T devem estar presentes desde a concepção do projeto, identificando os defeitos e garantindo a consistência do produto gerado. Devido a isso, o processo de teste é composto por cinco fases principais, descritas a seguir. Quando todas essas fases são executadas, diz-se que o objetivo de validação e verificação do *software* foi atingido.

- *Teste Unitário*: o teste unitário ou teste de módulo tem o objetivo de verificar cada módulo do sistema, testando cada unidade isoladamente e com isso garantir a correta funcionalidade dos componentes elementares de um programa, como métodos, classes e funções. O teste unitário é um dos primeiros testes a serem implementados no processo de desenvolvimento de *software* e na maioria das vezes são automatizados pela equipe de testes por meio da própria ferramenta de desenvolvimento, identificando pequenas falhas dos módulos do sistema, e com isso evitando que essas falhas se propaguem para outros níveis ou em testes subsequentes. Normalmente o defeito que o teste unitário busca detectar está relacionado a falhas em algoritmos e falhas de lógica [21]. Outro aspecto relevante é a facilidade de depuração de uma falha, pelo fato dela encontrar-se isolada em um determinado módulo, além da possibilidade do teste ser realizado em paralelo para múltiplos módulos do sistema [16].
- *Teste de Integração*: tem como objetivo encontrar erros derivados da integração dos módulos internos do *software*. Este processo de integração envolve construir e testar o sistema

resultante quanto a problemas que surjam a partir das interações dos componentes. Ou seja, verifica a compatibilidade e a consistência entre os módulos a fim de testar a operação do sistema como um todo [6] [16] [21]. O Teste de integração pode ser implementado aplicando duas abordagens: não incremental ou '*Big-Bang*' e incremental ou iterativa. Na primeira é testado cada módulo do sistema de forma independente e posteriormente testando suas combinações. Na segunda deve-se testar o novo módulo do sistema combinando com os demais módulos já existentes e que foram previamente testados. A abordagem de teste incremental possibilita que defeitos entre as interfaces sejam detectados de forma mais rápida, além de avaliar a complexidade das interações entre os módulos, uma vez que a complexidade cresce à medida que novos módulos são incorporados ao sistema [16] [20].

- *Teste de Sistema*: é o processo de verificação da consistência entre o *software* e sua especificação, avaliando as implicações dos requisitos em conformidade com o seu comportamento. Este é um processo vital, uma vez que a severidade dos defeitos e a propensão de suas ocorrências são avaliadas somente nesta etapa [16] [20] [21]. Nesta fase um conjunto de testes com diferentes características são executados em condições de teste idênticas ou próximas ao ambiente real. Alguns exemplos de teste de sistema são [6] [16]: 1) Teste de desempenho - visa avaliar se o programa satisfaz as especificações de desempenho (taxas de transferências e tempo de resposta das operações); 6) Teste de configuração - visa avaliar os requisitos necessários para que o *software* possa ser executado (configurações de *hardware*, sistemas operacionais, bancos de dados); 5) Teste de armazenamento - visa avaliar a quantidade de memória (principal e secundária) e identificar ocorrências de *overflow*. 9) Teste de confiabilidade - visa avaliar a capacidade do sistema completar suas operações em um intervalo determinado de tempo sem a ocorrência de falhas (*Mean Time To Failure - MTTF*); 10) Teste de recuperação - visa minimizar o tempo médio de recuperação do sistema (*Mean Time To Recovery - MTTR*) simulando defeitos a fim de avaliar sua reação.
- *Teste de Validação*: como o próprio nome especifica, visa validar a aceitabilidade do produto de *software*, de acordo com o que foi definido na etapa de especificação dos requisitos [16]. Possui três estratégias básicas que podem ser implementadas [20]: 1) Teste de aceitação formal - é um processo planejado, aplicado em um ambiente altamente controlado, onde são testados subconjuntos de casos de teste na etapa de teste do sistema, que normalmente são executados pelo usuário final; 2) Teste Alfa - são testes executados pelos usuários no ambiente de desenvolvimento, visam à identificação das funções e regras de negócio exploradas; Teste Beta - efetuado no próprio ambiente do usuário, que tem a liberdade de definir os critérios de rejeição ou aceitação do *software*.
- *Teste de Regressão*: consiste em refazer todos os testes que foram aplicados em versões ou ciclos de testes anteriores. Ocorre quando é realizada alguma alteração no sistema, e.g., quando é acrescentada ou retirada determinada funcionalidade, ou quando o sistema migra para outra plataforma. Quando o sistema não mantém a corretude de suas funcionalidades

em sua nova versão, diz-se que o sistema regrediu. O teste de regressão tem como objetivo evitar a não regressão das novas versões do sistema a fim de garantir que os atuais módulos e componentes funcionem corretamente e continuem válidos. Diante disso a automatização do processo de teste torna-se essencial, principalmente se o sistema está constantemente em processo de manutenção [16] [20] [21].

## 2.4 Teste de Desempenho

*"I just want people to focus on the performance"*

Janet McTeer

O teste de desempenho (*Performance Testing*) [8], conforme mencionado na Seção 2.3, é um teste de sistema que tem por objetivo avaliar o comportamento do sistema submetido a uma determinada carga em um ambiente de teste específico. Ele fornece um indicador utilizado para avaliar o quanto um sistema ou o componente de um sistema é capaz de cumprir com os requisitos de desempenho, tais como tempo de resposta ou *throughput*, além de identificar os possíveis gargalos que podem estar degradando o desempenho do sistema [22]. De acordo com a SPE - (*Software Performance Engineering*), o teste de desempenho é uma das atividades responsáveis por realizar testes de parte ou de todo o sistema sob carga normal e/ou carga de estresse. O teste de desempenho pode ainda ser dividido em: teste de carga (*Load Testing*) que verifica se o sistema atende aos requisitos de desempenho especificados; e, Teste de estresse (*Stress Testing*) que determina o comportamento do sistema quando ele é submetido além das condições de carga, ou seja, avalia sob quais condições uma aplicação vai falhar [8].

Para [23], os principais problemas encontrados nos projetos de *software* após sua entrega não são falhas no sistema, mas sim a degradação do desempenho. Isso pode ocorrer devido ao fato de que mesmo que na maioria das vezes o *software* tenha passado por testes funcionais, ele nunca foi realmente testado para avaliar o comportamento esperado sobre determinada carga de trabalho (*workload*). Os autores também indicam que as falhas de desempenho podem ser classificadas em três categorias: 1) Falta de estimativas de desempenho; 2) Falta de ter propostas de planos para a coleta de dados e; 3) Falta de um orçamento de desempenho.

Diante disso, a abordagem proposta pelo presente trabalho propõe a associação de tempo aos modelos UML com o intuito de medir e analisar o tempo de resposta (*response time*) na execução dos casos de teste e assim obter informações de tempo mais próximas e reais que permitem avaliar a disponibilidade e o comportamento de determinada aplicação quando submetida a diferentes cenários de teste, visando garantir que o *software* não apresente defeitos ou indisponibilidade. Outro aspecto, é a possibilidade de planejar melhorias no ambiente visando atender às demandas atuais e futuras e identificar de forma antecipada a ocorrência de defeitos da aplicação, determinando a probabilidade do sistema atender ou não os SLAs estabelecidas.



Neste contexto, diversas ferramentas foram criadas com o intuito de auxiliar na execução dos casos de teste, aumentando a eficiência no processo de geração dos testes de desempenho, tais como: Apache JMeter [24], HP LoadRunner [25], IBM Rational Performance Tester [26], Borland SilkPerformer [27] e Microsoft Visual Studio [28].

Além dos diferentes formalismos que podem ser utilizadas para modelar o teste de desempenho, formalizando os requisitos não funcionais e possibilitando a automatização dos testes por meio da abordagem MBT, tais como: Máquinas de Estados Finitos (MEF) [29], Cadeias de Markow [30], Redes de automatos estocásticos [31] Redes de Filas [32] e Redes de Petri [33]. É importante ressaltar ainda os modelos que são amplamente utilizados pela indústria, dentre eles, UML SPT (*Schedulability, Performance and Time*) [8] [34], onde são adicionados estereótipos e rótulos a fim de representar as informações de teste, como o número de usuários acessando a aplicação e o tempo de duração do teste.

## 2.5 Teste Baseado em Modelos

O desenvolvimento de *software* não é uma tarefa simples, podendo tornar-se bastante complexa dependendo das características e dimensões do sistema a ser criado. Por isso, está sujeito a diversos tipos de problemas que podem resultar em um produto diferente do esperado [6]. Nesse aspecto o teste de *software* torna-se uma atividade essencial, pois contribui para aumentar a qualidade do produto de *software* que está sendo desenvolvido.

Entretanto, embora importante, sabe-se que a atividade de teste tem como pressuposto ser uma das etapas mais onerosas e caras do desenvolvimento de *software* [21]. Nesse contexto, o Teste Baseado em Modelos (*Model Based Testing - MBT*) é uma abordagem que visa mitigar estes problemas, automatizando a geração de casos de teste, com o intuito de reduzir o tempo e o custo do processo de teste. O objetivo de MBT é a criação de artefatos de teste que descrevam os requisitos e o comportamento do sistema, visando à automatização da atividade de teste [7], reaproveitando, assim, os modelos construídos nas etapas de análise e projeto.

Por meio dos modelos de teste, também é possível extrair informações para a geração de novos artefatos de teste, como casos ou *scripts* de teste [35], bem como capturar e compartilhar o conhecimento do sistema, aumentar a qualidade da especificação e reusar os modelos desenvolvidos à medida que o sistema evolui. Além de rastrear as alterações na especificação do *software* [7], *e.g.*, quando o produto tem características novas, que podem ser gradualmente acrescentadas ao modelo, como também se há pessoas novas na equipe.

Modelos formais são usados com o intuito de compreender, especificar e desenvolver um sistema [7]. A criação de modelos formais baseados nos requisitos exige dos engenheiros e analistas de testes identificarem informações que muitas vezes estão implícitas em documentos tradicionais de especificação de requisitos, *e.g.*, anotações e estereótipos que quando incluídos aos modelos enriquecem a qualidade da especificação, além de ser implementada em vários contextos, pois a abordagem de MBT se aplica a diversas técnicas e/ou fases do processo de teste de *software*.

Existem diversos modelos formais que podem ser integrados com a abordagem MBT, tais como: Cadeias de Markov [30], Redes de Autômatos Estocásticos [31], Redes de Filas [32] e Redes de Petri [33]. Entre os mais importantes e usualmente aplicados na técnica baseada em modelos estão as Máquinas de Estados Finitos - MEF (*Finite State Machine - FSM*) [36], também conhecidas como máquinas de estados, ou ainda autômato finito, que se caracteriza como uma máquina hipotética composta por um conjunto de estados finitos e de transições entre estes estados.

A técnica de teste baseado em MEFs torna-se importante para saber o que é essencial no sistema, permitindo realizar a simulação do modelo e avaliando os diversos fluxos e comportamentos que o mesmo possa assumir, a fim de definir as sequências de entrada para execução dos testes [37].

Máquinas de estados finitos são uma alternativa para projetar e testar os componentes de *software*. É um modelo adequado para elaborar diversas sequências de entrada, servindo como dados para o processo de teste de maneira automatizada [38]. Uma sequência de teste é uma sequência de símbolos de entrada derivada da MEF que são submetidos à implementação correspondente para verificar a conformidade da saída gerada pela implementação com a saída especificada na sequência de teste [21].

O teste realizado a partir de uma MEF se caracteriza por um conjunto de entradas que produz um conjunto de saídas. Estas saídas quando comparadas com um oráculo de teste (*Test Oracle*), processo capaz de definir se o caso de teste foi realizado com sucesso ou não, permite automatizar a execução dos testes, delegando a tarefa de decisão de quais casos de testes produziram ou não a saída esperada. É importante ressaltar que atualmente existem diversos modelos e notações capazes de abstrair outras informações, sendo possível aplicar a técnica MBT, para a realização de outros tipos de teste, inclusive para o teste de desempenho [39].

### 2.5.1 Processo MBT

O processo da abordagem MBT exige a realização de atividades de testes específicas diferenciadas da atividade de teste de *software* habitual pois ela requer da equipe uma adaptação do seu processo de teste. As principais atividades que definem o processo de MBT são ilustradas pela Figura 2.2 e são descritas em [2]:

- Construir o modelo: construção de um modelo de teste baseado na especificação dos requisitos do sistema, que define a estrutura e o comportamento do sistema sob teste;
- Gerar entradas esperadas: utilização do modelo formal desenvolvido para geração dos casos de teste ou *scripts* de teste, a fim de obter as entradas e as saídas esperadas na execução dos testes, ou seja, prover uma ferramenta para a geração dos casos e *scripts* de teste;
- Gerar resultados esperados: geração de um mecanismo que determine se os resultados de uma execução do teste realizado estão corretos ou não. Este mecanismo é o Oráculo de Teste (*Test Oracle*), que determina se o resultado obtido é igual ao resultado esperado, ou seja, extraem-se os resultados esperados para a etapa de análise;



Engenharia de *Software* [40]<sup>1</sup> da PUCRS. Os produtos derivados desta linha [12] [13] são utilizados para automatizar a geração e execução de casos e/ou *scripts* de teste. A Figura 2.3 apresenta o modelo de características da PLeTs que representa o escopo da linha de produto. No primeiro nível do modelo existem quatro características principais, identificadas como:

- *Parser*: representa uma das principais atividades de MBT, a construção e interpretação do modelo. É uma característica obrigatória e tem duas características dependentes: *UML* e *Text*, que são usadas para extrair informações a partir de modelos UML ou de um arquivo texto.
- *Test Case Generator*: representa a etapa de geração dos casos de teste. É uma característica obrigatória e tem duas características dependentes: *SequenceGenerator* e *AbstractTestCaseGenerator*. A primeira tem três características: *FiniteStateMachine* (com os métodos HSI e UIO), *PetriNets*, *Random-DateGenerator*. A segunda tem duas características *Performance-Testing* e *Structural Testing*, uma delas deverá ser selecionada em cada variante do *software*. Estas duas características recebem a sequência de teste gerada e criam a sequência de teste abstrato para cada nível de teste.
- *Script Generator*: representa a etapa de geração dos *scripts* de teste por meio de tecnologias que usam *scripts* para realizar a execução do teste, sendo elas: *Jmeter* [24], *LoadRunner* [25], *Visual Studio* [28], *JabutiScript* [40], *EmmaScript* [41].
- *Executor*: representa a etapa de execução dos *scripts* de teste. Este recurso tem duas características: *ProgramInterface* e *Parameterization*. O primeiro define a interface do produto e é composto de outras duas características: *GUI* (*Graphical User Interface*) para interface gráfica e de console para execução de linha de comando. O segundo tem cinco características: *LoadRunnerParams*, *VisualStudioParams*, *EmmaParams*, *Jabuti-Params* e *JMeterParams*. Essas características são implementadas para iniciar a ferramenta de teste e executar os *scripts* de teste. Depois disso, uma ferramenta de teste (e.g., *LoadRunner*) coleta os resultados produzidos durante a execução do teste com o intuito de compará-los com o oráculo de teste.

Como pode-se observar na Figura 2.3 existem certas dependências entre algumas características. Caso em uma configuração do *software* seja selecionada a característica *LoadRunnerParameters*, deve também ser selecionada a característica *LoadRunnerScript* na etapa seguinte, pois a ferramenta não seria capaz de executar os testes sem um *script* de teste compatível. Outro aspecto importante é que o modelo de características pode ser estendido para suportar novas técnicas de teste ou de ferramentas, adicionando novas características dependentes das quatro principais características. Exemplificando, caso seja necessário adicionar novas características para trabalhar com a ferramenta *SilkPerformer* [27], novas características dependentes para as características principais (*Script Generation*) e (*Execution*) também devem ser incluídas.

<sup>1</sup>Homepage CePES: <http://www.cepes.pucrs.br>

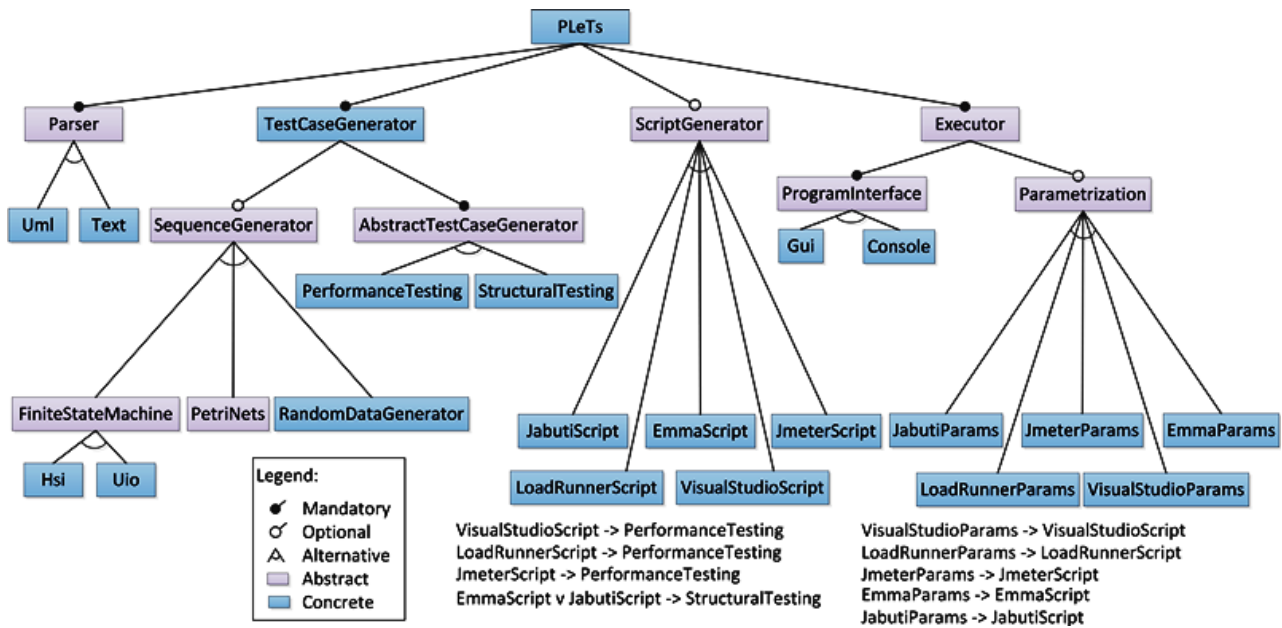


Figura 2.3: Modelo de Características da PLeTs [3]

Apesar das diversas etapas existentes no processo de teste baseado em modelos, o principal objetivo dessa pesquisa está relacionado com a característica *Test Case Generator* na PLeTs.

É importante ressaltar que já foi realizado um estudo relacionado com esta abordagem de pesquisa [42], na qual investigou-se o processo de geração de casos de teste baseados em MEFs, detalhando sua implementação por meio do método HSI [36] [43]. Entretanto, os aspectos de tempo não foram incluídos nos modelos UML, o que seria uma contribuição relevante na abordagem de teste de desempenho. Sendo assim, por meio desta pesquisa pretende-se incluir informações de tempo nos modelos UML para mensurar o tempo de resposta na execução dos casos de teste de desempenho gerados e compará-los com os valores estimados.

## 2.6 HP LoadRunner

Devido à importância do teste de desempenho e a evolução dos sistemas computacionais existentes, diferentes conjuntos de ferramentas de automação foram criadas para auxiliar na execução de cenários de teste de desempenho. Dentre elas citam-se: Apache JMeter [24], HP LoadRunner [25], IBM Rational Performance Tester [26], Borland SilkPerformer [27] e Microsoft Visual Studio [28].

Para o presente estudo será utilizada o HP LoadRunner, uma ferramenta de teste de desempenho amplamente utilizada pela indústria para avaliar o desempenho de sistemas antes da produção, com o intuito de prever possíveis defeitos gerados por baixo desempenho. HP LoadRunner [25], é uma das ferramentas inicialmente criada pela Mercury Interactive e que atualmente integra o conjunto de ferramentas da HP Quality Center. O foco da ferramenta LoadRunner está na geração e execução de testes de desempenho. Esta ferramenta utiliza a técnica de *RecordandPlayback*, conhecida também como *CaptureandReplay* [44] para geração de seus *scripts* de teste, os quais podem ser gerados nas linguagens Java e C. Conforme representado pela Figura 2.4, o processo de geração e execução

de testes no LoadRunner é composto por cinco etapas, definidas como: planejamento, criação dos *scripts* de teste, definição dos cenários, execução dos cenários e análise dos resultados [4].

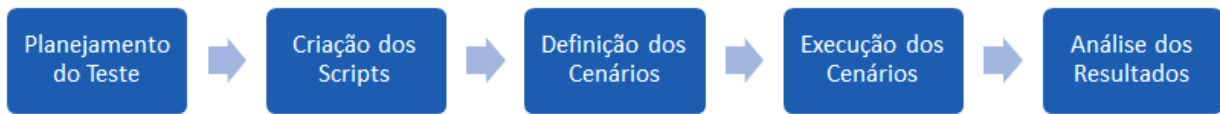


Figura 2.4: Etapas do processo de geração e execução de testes de desempenho na ferramenta LoadRunner [4]

- *Planejamento do Teste*: são definidos os requisitos para o teste de desempenho, e.g., número de usuários acessando a aplicação em um determinado intervalo de tempo;
- *Criação dos Scripts*: capturar e gravar as atividades executadas pelos usuários na aplicação que será testada e gerar automaticamente os *scripts* de teste de desempenho;
- *Definição dos Cenários*: configurar o ambiente de teste utilizando o *Controller* do LoadRunner;
- *Execução dos Cenários*: gerenciar e monitorar o teste de carga utilizando o *Controller* do LoadRunner;
- *Análise dos resultados*: gerar gráficos e relatórios com os resultados para avaliar o desempenho utilizando o *Analyser* do LoadRunner.

Para auxiliar no processo descrito, três recursos compõem a arquitetura do LoadRunner: *Virtual User Generator*, *Controller* e *Analyser*. Eles são utilizados respectivamente para a geração, execução e análise dos resultados dos casos de teste, apresentando as seguintes funcionalidades:

- *Virtual User Generator*: é responsável por capturar as informações referentes às interações do usuário e criar um teste de desempenho automático. Posteriormente, estas informações são gravadas em um *script* de teste, também conhecido como *script* de usuário virtual;
- *Controller*: é responsável pela configuração das informações de carga de trabalho, além de definir e monitorar os contadores para avaliar o comportamento do sistema que será testado. Permite visualização de alguns parâmetros de desempenho (*server*, *network*, usuários, etc.) durante o teste em tempo real;
- *Analyser*: coleta os dados e gera relatórios com as informações do teste executado, apresentando gráficos e os valores correspondentes às métricas e contadores definidos e monitorados no *Controller*.

Uma das razões pela escolha da ferramenta LoadRunner é que ela está integrada com a PLeTs na geração automatizada dos casos e *scripts* de teste e por meio dos recursos disponíveis na ferramenta

é possível simular a execução de cenários de teste a fim de analisar o desempenho da aplicação. Outro recurso presente na ferramenta LoadRunner é a possibilidade de se definir Acordos de Nível de Serviço (*Service Level Agreement - SLA*), que permitem verificar se as metas de desempenho estabelecidas para cada transação foram atingidas com a execução dos *scripts* de teste, conforme será descrito na Seção 2.8.

## 2.7 Programa de Avaliação e Técnica de Revisão

A técnica de Avaliação e Revisão de Programas (*Program Evaluation and Review Technique - PERT*) [45] começou a ser utilizada pela Marinha dos EUA no início de 1959 com o intuito de controlar seus projetos para terminá-los no prazo previsto. É muito utilizada para planejar, avaliar e controlar o tempo de execução de programas e projetos. De acordo com [45] [46] a precisão das estimativas do tempo de execução de determinadas atividades podem ser aperfeiçoadas considerando as incertezas das estimativas e os riscos. Neste contexto, esta técnica usa três estimativas de tempo para definir uma faixa aproximada para a execução de determinada atividade, conforme demonstra a Figura 2.5.

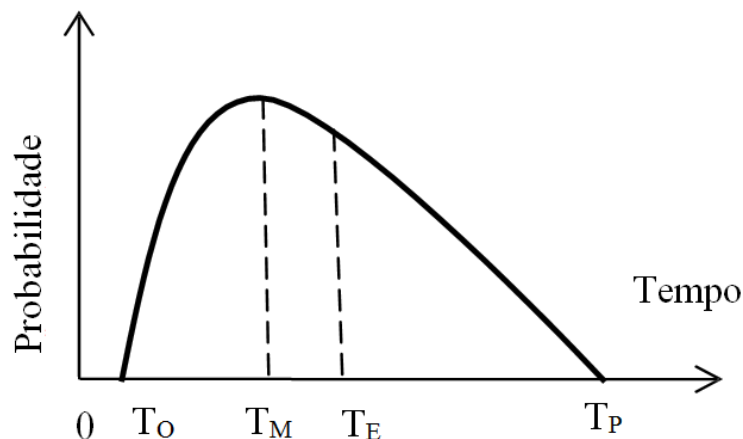


Figura 2.5: Estimativa de Tempo

- *Mais Provável* ( $T_M$ ) - a duração da atividade, dados os prováveis recursos a serem designados, sua produtividade, expectativas realistas de disponibilidade para executar a atividade, dependências de outros participantes e interrupções.
- *Otimista* ( $T_O$ ) - a duração da atividade é baseada na análise do melhor cenário para a atividade.
- *Pessimista* ( $T_P$ ) - a duração da atividade é baseada na análise do pior cenário para a atividade.

Com as estimativas de tempo estabelecidas a etapa seguinte consiste em calcular o tempo esperado ( $T_E$ ) de execução de determinada atividade ou sequência de atividades. Este cálculo é obtido por meio de uma média ponderada por meio da seguinte fórmula:

$$T_E = \frac{(Pessimista + (4 * Mais Provavel) + Otimista)}{6}$$

Essa técnica é aplicada para medir e controlar diferentes tipos de projetos, desde o planejamento de uma construção civil até a pesquisa e desenvolvimento de um novo produto. Em [47] com o intuito de melhorar os investimentos e a gestão de projetos de construção civil a técnica é usada com o objetivo de otimizar a alocação dos recursos necessários para o projeto, determinando a velocidade de execução das atividades, considerando a incerteza de tempo e custo requerido pelas mesmas. Já em [48] a técnica é aplicada com o intuito de realizar cálculos probabilísticos para projetos de construções repetitivos dentro de um período específico, identificando principalmente atividades idênticas e que são repetidas ao longo da execução de um projeto de construção. Para auxiliar neste processo é utilizado o RPERT, um *software* desenvolvido com a linguagem de programação Java que fornece uma série de recursos, com o intuito de verificar a probabilidade de conclusão de um projeto de acordo com o conjunto dos prazos previstos por cada atividade repetida, diante das três perspectivas (otimista, mais provável e pessimista).

No gerenciamento de projetos durante o desenvolvimento de *software*, dois aspectos são cruciais: a distribuição de recursos e o planejamento das atividades. Considerando a distribuição dos recursos em relação as atividades desempenhadas em um ambiente de multiprojetos, PERT é uma das primeiras técnicas propostas [49].

Como é possível observar a técnica PERT mostra-se eficiente para o controle e o planejamento em diferentes áreas do conhecimento, pois permite avaliar as interdependências das atividades envolvidas considerando os diferentes níveis de incertezas existentes [46], antecipando o tempo que cada atividade pode ter de atraso.

No processo de desenvolvimento de *software*, PERT também é utilizado em cálculos de estimativa de esforço. A técnica KSLOC (*Kilo Source Lines of Code*) [50], consiste em estimar o número de linhas que um programa deverá ter, normalmente com base na opinião de especialistas e no histórico de projetos passados, classificando em três situações principais: KSLOC otimista, mais provável e pessimista. O valor mínimo de linhas quando as condições de desenvolvimento forem favoráveis é denominado como KSLOC otimista. Para o desenvolvimento em condições de dificuldades, o valor máximo de linhas, ou seja, o KSLOC pessimista. E o KSLOC esperado, quando se espera produzir determinado número de linhas em condições de desenvolvimento normais.

É importante ressaltar que estimar tempo e esforço é ainda uma das atividades mais difíceis durante a fase de planejamento, isso inclui também as atividades de teste de *software*. No entanto, de alguma forma é necessário medir o esforço despendido para a realização dos casos de teste de desempenho realizados e avaliar se o resultado obtido está de acordo com o que foi estimado. Neste contexto e diante da dificuldade das equipes em estabelecer estimativas de tempo para a execução de testes de desempenho, os três pontos presentes na técnica PERT possibilitam uma faixa de variabilidade, sendo possível trabalhar com intervalos de tempo. É por meio desta técnica de estimativas que serão definidos os Acordos de Nível de Serviço também conhecidos como SLAs.



## 2.8 Acordo de Nível de Serviço

Acordo de Nível de Serviço (*Service Level Agreement - SLA*), também denominado como Acordo de Nível de Garantia (*Service Level Guarante - SLG*), surgiu por volta da década de 90, advindo da tecnologia da informação, dedicado inicialmente a ambientes computacionais, para medir e gerenciar a qualidade de um serviço (*Quality of Service - QoS*) [5].

Caracteriza-se como um conjunto de procedimentos e objetivos formalmente acordados entre as entidades envolvidas com a finalidade de manter a qualidade de um serviço especificada e garantir que a sua execução está sendo monitorada e avaliada. Devem incluir informações detalhadas sobre as partes envolvidas e a relação que existe entre elas, onde são definidas as condições específicas de quando e como os serviços devem ser executados ou entregues e as responsabilidades das partes. Além de estipular métricas associadas para que o desempenho dos serviços e das aplicações possam ser medidos e avaliados [51].

Estes procedimentos e objetivos são descritos para uma aplicação ou serviço específico quantificando critérios como: disponibilidade, desempenho e tempo de reparo, fazendo com que a qualidade do nível de serviço contratado seja satisfatória e atenda às necessidades do usuário. De acordo com [5] o ciclo de vida de um acordo de nível de serviço compreende cinco fases que são representadas pela Figura 2.6:

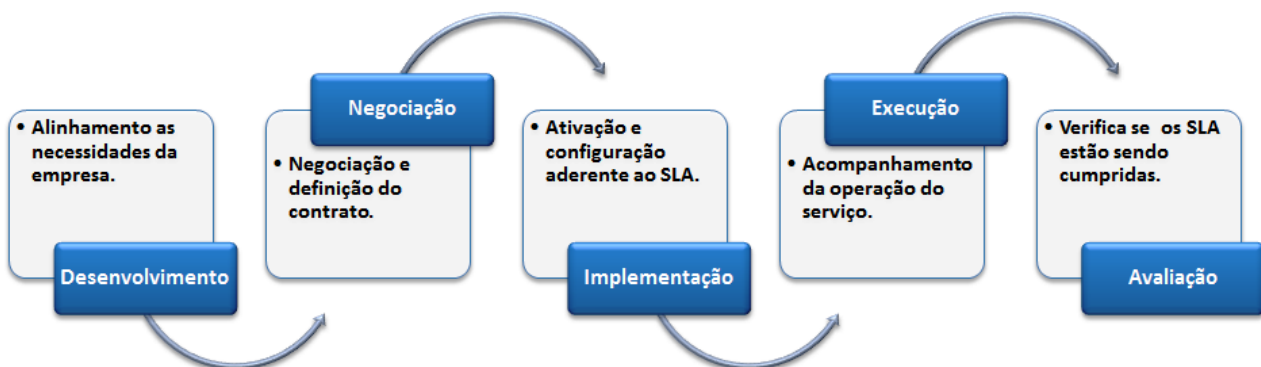


Figura 2.6: Ciclo de vida de um Acordo de Nível de Serviço [5]

- *Desenvolvimento*: é a criação propriamente dita do SLA, alinhado às necessidades específicas da empresa;
- *Negociação*: é realizada a negociação do acordo de nível de serviço entre as partes envolvidas e definido o contrato;
- *Implementação*: também chamada de provisionamento, onde os serviços são solicitados, ativados e configurados de forma aderente ao SLA;
- *Execução*: é o acompanhamento da operação normal do serviço. Isto inclui a medição dos parâmetros e emissão de relatórios com os resultados obtidos;

- *Avaliação*: deve ser periódica e aderente às necessidades do cliente. Pode ser do ponto de vista do cliente e/ou do provedor de serviços desde que ambos estejam alinhados.

## 2.9 Resumo

Os métodos, técnicas e ferramentas são fundamentais para a elaboração e execução de um projeto de teste de *software* confiável. No entanto, com o aumento da complexidade dos sistemas, a atividade de teste tornou-se umas das etapas mais onerosas do processo de desenvolvimento de *software*. Devido a isso, o Teste Baseado em Modelos torna-se fundamental para permitir a abstração da complexidade do sistema e automatizar o processo de teste na tentativa de reduzir o esforço e o custo. O estudo proposto consiste em incluir informações de tempo nos modelos UML por meio da técnica PERT, e com base nessas estimativas medir e comparar o tempo de resposta das sequências de teste. Outro aspecto, é a definição de Acordos de nível de serviço - SLAs, que tem como objetivo definir índices a serem atingidos, monitorando se o tempo de resposta está em conformidade com o que foi estabelecido no SLA.

### 3. ABORDAGEM

*"Research is what I'm doing when I don't know what I'm doing."*

Wernher von Brau

Este capítulo apresenta a abordagem do estudo realizado, sintetizando o processo de construção dos modelos e notações aplicadas ao teste de desempenho em MBT e que serão necessárias para a geração dos *scripts* de teste que serão executados pela ferramenta LoadRunner. Além disso, compreende as informações para a definição das transações que serão utilizadas para a automação dos SLAs, tendo como referência o modelo construído. Para isto, será utilizado como exemplo de uso o *benchmark* TPC-W [52], a qual tem como objetivo simular as principais operações realizadas pelo usuário em uma aplicação *e-commerce* e a aplicação *Skills* [12] que tem por objetivo gerenciar os perfis profissionais de colaboradores de uma empresa.

Devido ao esforço consumido pela atividade de teste durante o processo de desenvolvimento de *software*, estudos têm sido realizados com o intuito de identificar técnicas e métodos que tornem a atividade de teste menos custosa e mais viável. Nesta abordagem se tem o MBT, que conforme descrito na Seção 2.5 possibilita a redução do custo de geração dos casos de teste, sendo possível gerar conjuntos de teste de maneira automatizada. No entanto, nos produtos derivados da linha de produto PLeTs os casos de teste são gerados sem considerar o tempo de resposta. Assim neste trabalho é proposto uma abordagem onde o escopo da aplicação é estendido para um novo domínio, teste de desempenho, incluindo tempo na geração e execução dos casos de teste.

A abordagem proposta para a realização de teste de desempenho é apresentada na Figura 3.1. Ela tem início com a divisão das ações que cada participante ou ferramenta envolvida realizará, *i.e.* Tester, PLeTs e LoadRunner. Este processo começa com a atividade Modelar os Diagramas UML, desenvolvidos pelo Tester na especificação dos cenários de teste de desempenho, representado por diagramas de casos de uso (*Use Case Diagram - UC*) e por diagramas de atividades (*Activity Diagram - AD*). Com base no perfil UML SPT (*Schedulability, Performance and Time*) [8] são adicionadas aos modelos informações de teste relacionadas aos cenários de desempenho, cargas de trabalho (*workload*) e perfis dos usuários, representada pela atividade Adicionar Dados de Teste de Desempenho.

Visando à automação dos cenários e *scripts* de teste de desempenho pela PLeTs, o modelo UML gerado é exportado em um arquivo XMI (*XML Metadata Interchange*) que por sua vez é submetido ao *parser* da PLeTs, a fim de Interpretar Dados de Teste de Desempenho necessárias para a geração de uma estrutura de dados abstrata em memória (*Estrutura Intermediária*) que foram adicionadas por meio dos estereótipos.

Com o cenário de teste de desempenho especificado, cada caso de uso é decomposto em um AD, o qual demonstra o fluxo de ações executadas pelo usuário para realizar determinada tarefa no

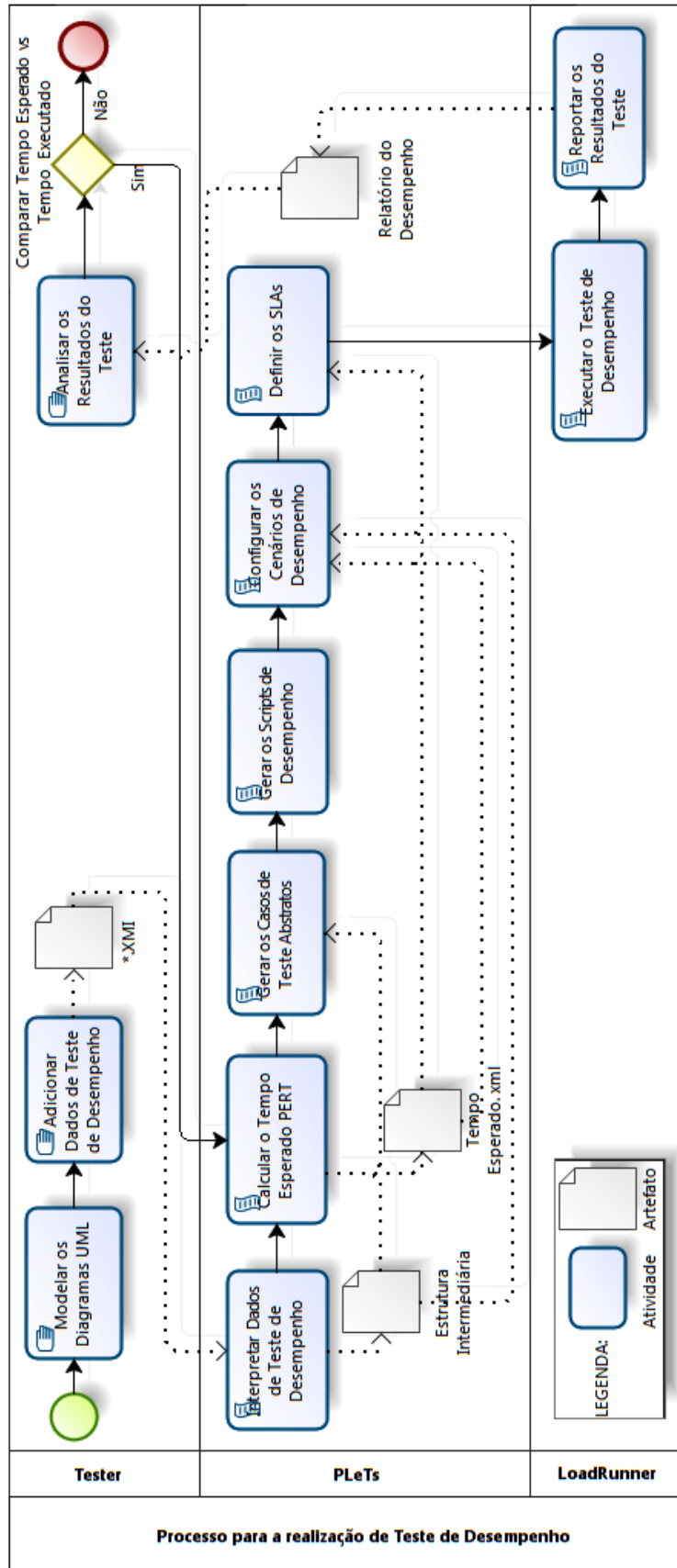


Figura 3.1: Processo para a realização de Teste de Desempenho

sistema. É nesta etapa que são definidas as transações que controlam as atividades e que serão necessárias para definir os SLAs e avaliar o tempo de resposta dos *scripts* de teste de desempenho. É importante salientar que estimar tempo e esforço é ainda uma das atividades mais difíceis durante a fase de planejamento, isso inclui também as atividades de teste de *software*.

No entanto, de alguma forma é necessário medir o esforço despendido para a realização dos casos de teste e avaliar se o resultado obtido está de acordo com o que foi estimado. Neste contexto e diante da dificuldade das equipes em estabelecer estimativas de tempo para a execução de testes de desempenho, os três pontos presentes na técnica PERT [45](ver Seção 2.7), possibilitam uma faixa de variabilidade, sendo possível trabalhar com intervalos de tempo. A estimativa dos tempos diante das três perspectivas (otimista, mais provável e pessimista) estabelecidas pelo PERT é definida inicialmente a partir da execução de um *smoke test*<sup>2</sup>. Com base nesta técnica, a atividade seguinte é responsável por Calcular o Tempo Esperado PERT para cada uma das transações mapeados nos diagramas UML interpretados, esta informação é armazenada no arquivo XML Tempo Esperado.

Em seguida, os diagramas UML com as informações de teste são convertidos em uma MEF (Máquina de Estados Finitos) e por meio da aplicação do método HSI (*Harmonized State Identification*) [36], são geradas as sequências de teste a serem executadas, representado pela atividade Gerar os Casos de Teste Abstratos.

As informações referentes à tecnologia que será usada são fornecidas nas etapas Gerar Scripts de Desempenho e Configurar os Cenários de Desempenho, quando o usuário deve fornecer todos os dados necessários para gerar os *scripts* e configurar os cenários de teste de desempenho usando uma ferramenta de teste específica, *i.e.* gerador de carga de trabalho. Na etapa a seguir é responsável por Definir os SLAs de acordo com as transações pré-definidas nos *scripts* de teste de desempenho gerados. Os cenários de teste de desempenho gerados serão usados posteriormente na atividade Executar o Teste de Desempenho pela ferramenta LoadRunner [25].

Durante a execução dos *scripts* de teste de desempenho, o LoadRunner coleta os dados a fim de (Reportar os Resultados do Teste), que são gravadas em um arquivo XML denominado (Relatório do Desempenho) e reportados para a PLeTs. Desta forma é possível o Tester verificar as transações que cumpriram o SLA e as transações que não cumpriram, além de Analisar os Resultados do Teste, comparando o tempo mínimo, médio e máximo de resposta de cada transação, *i.e.* os tempos executados contra os tempos esperados.

### 3.1 Modelagem UML para Teste de Desempenho

Os diagramas UML fornecem uma visão sobre aspectos de estrutura e comportamento dos sistemas. Com o intuito de representar as informações e características de teste relacionadas aos modelos UML e implementadas no processo de geração dos *scripts* de teste, esta seção descreve as definições estabelecidas no processo de construção dos diagramas de casos de uso e atividades.

---

<sup>2</sup>Processo no qual um produto de *software* é submetido, para verificar sua funcionalidade básica. Neste caso, será realizado um *smoke test* para coletar as estimativas de tempo que irão popular o modelo de teste.

Com base no modelo UML SPT [8] [22] são adicionadas ao diagrama de casos de uso (*Use Case Diagram - UC*), informações relacionadas aos cenários de teste, cargas de trabalho (*workload*) e aos perfis dos usuários. Posteriormente, cada caso de uso é decomposto em um diagrama de atividades (*Activity Diagram - AD*), o qual demonstra o fluxo de atividades executadas pelo usuário para realizar determinada tarefa no sistema. Essas informações são inseridas nos diagramas de casos de uso e atividades por meio de estereótipos e rótulos (*tags*), que são necessários para geração dos *scripts* de teste, bem como para avaliar a escalabilidade da aplicação em cenários com maiores solicitações e concorrência. Neste contexto foram utilizados os estereótipos de desempenho conforme [11] [42] [53] [54] e que estão representados nas Figuras 3.2: (Caso de Uso da aplicação *Skills*) e 3.3: (Diagrama de atividades da aplicação *Skills*)

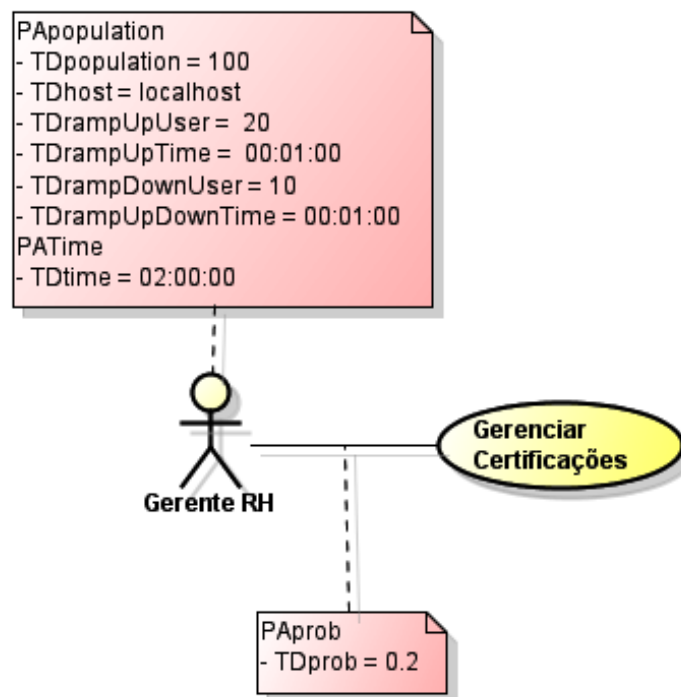


Figura 3.2: Estereótipos e rótulos anotados no diagrama de casos de uso da aplicação *Skills*

- «PApopulation» - esta informação é definida no ator do diagrama de casos de uso e possui os seguintes rótulos:
  - TDpopulation - representa o número de usuários virtuais que irão executar a aplicação;
  - TDhost - define o endereço (caminho) para executar a aplicação;
  - TDrampUpUser - determina o número de usuários virtuais que acessam o sistema a cada intervalo de tempo (TDrampUpTime) durante a inicialização do teste;
  - TDrampUpTime - representa o intervalo de tempo (segundos) de cada ciclo de acesso a novos usuários ao sistema durante a inicialização do teste;
  - TDrampDownUser - determina o número de usuários virtuais que finalizam seu acesso ao sistema a cada intervalo de tempo (TDrampDownTime) durante a finalização do teste;

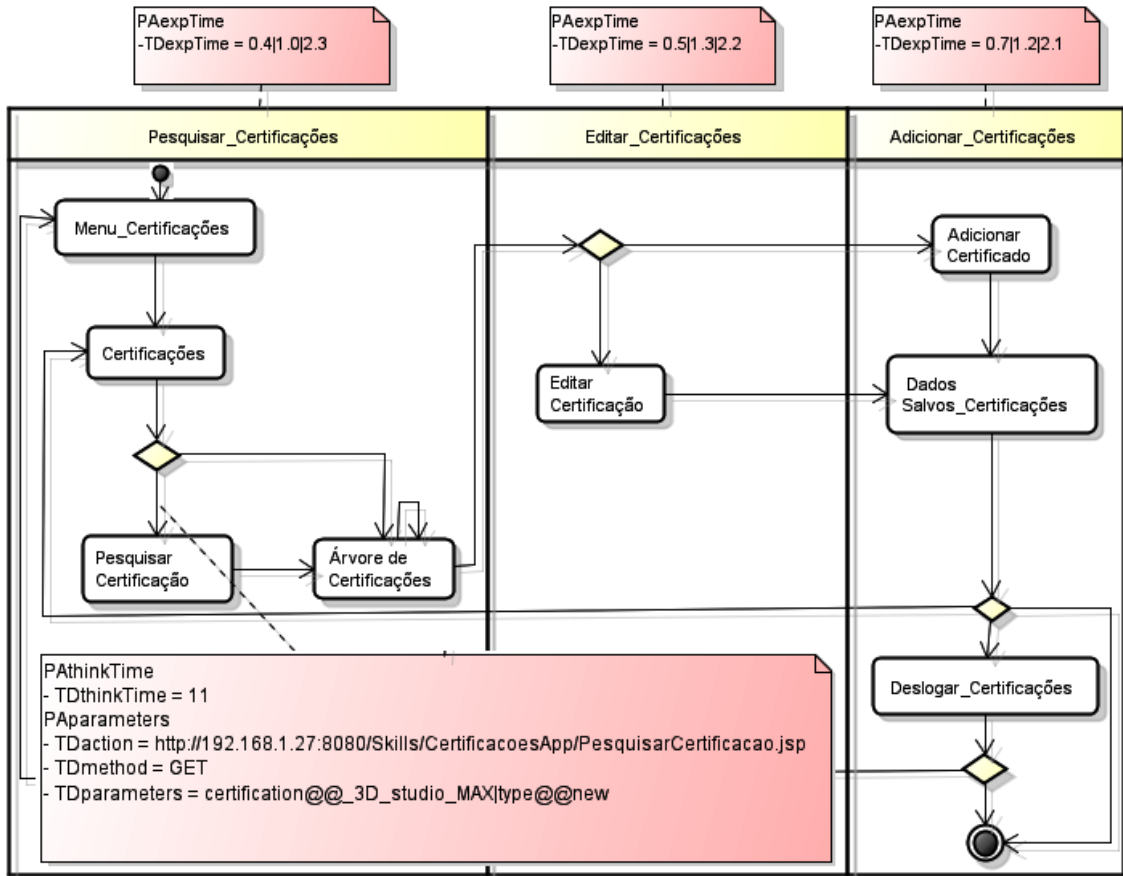


Figura 3.3: Estereótipos e rótulos anotados no diagrama de atividades da aplicação Skills

- TDdrampDownTime - representa o intervalo de tempo (segundos) de cada ciclo de finalizações de acessos dos usuários ao sistema durante a finalização do teste.

- «PAprob» - diferente dos rótulos citados acima que são anotadas no(s) ator(es) presente(s) no diagrama de casos de uso, este rótulo é anotado na transição existente entre o ator(es) e o(s) respectivo(s) caso(s) de uso com que ele interage. É por meio desta informação que é definida a probabilidade de distribuição dos usuários virtuais em cada caso de uso. Esta informação deve estar presente em todas as associações *Actor-UseCase* e a soma de todos os rótulos TDprob anotados para um mesmo ator devem ser igual a 1 (ou seja, 100%).

Na ausência do rótulo TDprob o valor total (100%) será distribuído entre todos os casos de uso envolvidos. Por exemplo, em um diagrama de casos de uso com um ator relacionado a quatro casos de uso, para cada associação será atribuído TDprob = 0,25. Do contrário todas as associações deverão ser anotadas, garantindo que a soma seja igual a um.

É importante ressaltar que o rótulo TDprob pode também ser informado nas transições existentes nos diagramas de atividades com o objetivo de distribuir os usuários na execução de determinadas atividades da aplicação, como por exemplo em situações de paralelismo, representadas no diagrama de atividades por um losango.

- «PAtime» - determina o tempo total de execução do teste. Esta informação é representada nos casos de uso do diagrama de casos de uso pelo rótulo TDtime.
- «PAtinkTime» - denota o tempo existente entre a disponibilidade de execução da atividade até o início de sua real execução pelo usuário. Pode ser exemplificado pelo preenchimento de um formulário até sua submissão. É definido nas transições das atividades presentes no diagrama de atividades, representado pelo rótulo TDthinkTime.
- «PAparameters» - define os parâmetros necessários para a execução dinâmica de determinadas atividades. Este estereótipo é formado por três rótulos:
  - TDaction - define o endereço do *link* acessado pelo ator;
  - TDmethod - define o método de requisição HTTP utilizado pelo servidor;
  - TDparameters - indica os parâmetros enviados ao servidor para processar a requisição HTTP. É formado pela concatenação de duas informações: nome do objeto (*name*) e valor do objeto (*value*), separadas pelo delimitador @@. Quando há mais de um parâmetro a anotação é representada da mesma forma, porém separando cada parâmetro pelo delimitador |.

O principal objetivo dos estereótipos é prover informações ao modelo que possibilitem a geração dos *scripts* de teste para a realização dos testes de desempenho. Referente aos rótulos TDdrampUpUser, TDdrampUpTime, TDdrampDownUser, TDdrampDownTime, caso nos modelos não sejam anotados os respectivos valores, por padrão é considerado que todos os usuários irão acessar e sair da aplicação ao mesmo tempo.

Como a maior contribuição deste estudo é gerar *scripts* de teste de desempenho considerando o tempo de resposta usando modelos UML, foi adicionado um novo estereótipo (PAexpTime), representado pelo rótulo TDexpTime. Este rótulo está presente nas transições do diagrama de atividades, e nele são informadas as estimativas de tempo referente à execução de cada transação, diante de três perspectivas: otimista, mais provável e pessimista. Caso esta informação não esteja presente no modelo, é executado um *smoke test* com o intuito de coletar os dados de teste de desempenho e gravar no arquivo XML Tempo Esperado (ver Figura 3.4).

De acordo com o mapeamento do conjunto de característica para teste de desempenho realizado em [42], na Tabela 3.1, é apresentado os estereótipos e os rótulos que serão utilizadas nos exemplos de uso descritos na Seção 3.4. A coluna "Característica" representa a informação que será incluída no modelo para executar o teste; a coluna "Estereótipo" lista os estereótipos implementados no modelo UML SPT para representar o modelo de teste; a coluna "Rótulo" descreve o nome de cada um dos rótulos que são vinculados aos estereótipos. Por meio desses rótulos é que são incluídas nos modelos as informações do teste. E a coluna "Diagrama" faz referência ao diagrama UML (*UC - Use Case Diagram ou AD - Activity Diagram*) em que foi adicionada a informação correspondente.

Além dos estereótipos, é importante ressaltar conforme descrito anteriormente que no processo de modelagem inicialmente é criado o cenário de teste, representado pelo diagrama de casos de uso



Tabela 3.1: Características e Estereótipos

Característica	Estereótipo	Rótulo ( <i>tag</i> )	Diagrama
Parâmetro	PAparameters	TDparameters	AD
Probabilidade	PAprob	TDprob	UC, AD
Requisição	PAparameters	TDaction e TDmethod	AD
SUT	PApopulation	TDhost	UC
Tempo de Espera	PAthinkTime	TDthinkTime	AD
Tempo de Execução	PAtime	TDtime	UC
Tempo de Inicialização	PApopulation	TDrampUpTime	UC
Tempo de Finalização	PApopulation	TDrampDownTime	UC
Usuários de Inicialização	PAparameters	TDrampUpUser	UC
Usuários de Finalização	PAparameters	TDrampDownUser	UC
Usuários Virtuais	PApopulation	TDpopulation	UC
Estimativas de Tempo de Resposta	PAexpTime	TDexpTime	AD

e em seguida cada caso de uso é expandido em um diagrama de atividades, os quais descrevem as ações que serão executadas pelo usuário para realizar determinada tarefa no sistema.

Já a modelagem dos diagramas de atividades é composta por duas etapas principais. A primeira etapa consiste na criação de um diagrama de atividades de alto nível, que na segunda etapa será expandido em outros diagramas de atividades mais detalhados. Isso possibilita ter uma visão macro do processo de execução, além de permitir referenciar a um mesmo diagrama secundário várias vezes, possibilitando, assim, o reuso dos modelos e dos *scripts* na execução dos testes.

É importante ressaltar que o teste de desempenho também é composto por transações (*Transaction*) e requisições (*Request*). Cada transação especifica um conjunto de requisições contidas em um *script* de teste e são definidas com o intuito de isolar o processamento em diferentes pontos, facilitando a identificação dos gargalos do sistema, além de mensurar as Transações por Segundo - TPS e verificar se estas satisfazem os critérios de aceitação de acordo com as métricas estabelecidas no SLA.

Neste contexto, nos diagramas de atividades as transações são representadas por sequências, onde o nome dado a cada sequência é igual ao nome da transação (*transaction*) que será gerada no *script* de teste. Dentro de cada sequência devem ser adicionadas as atividades (requisições) que fazem parte da transação. Por padrão, cada atividade atômica é considerada uma subtransação que faz parte de uma transação. Vale ressaltar ainda que cada *script* gerado a partir dos diagramas de atividades possuem uma transação genérica gerada automaticamente, a qual compreende todas as atividades da sequência do teste gerado.

### 3.2 Automação de SLAs para Teste de Desempenho

Além das informações de teste, os modelos possuem informações de tempo que são necessárias para a realização de teste de desempenho, permitindo medir de maneira automatizada o tempo de resposta das sequências de teste, como também na definição dos SLAs. Neste contexto, as

informações relacionadas aos diagramas servem como entrada para as ferramentas de teste baseado em modelos, que são geradas a partir da linha de produto - PLeTs.

O processo para definição dos SLAs inicia na criação dos modelos UML, em que são estabelecidas as transações e anotadas as estimativas de tempo para sua execução por meio do estereótipo TDexpTime, conforme mostrado na Figura 3.3. A partir do modelo UML criado um arquivo XMI é exportado. Em seguida, este arquivo XMI é interpretado e calculado o tempo de resposta esperado para execução de cada transação, por meio da técnica PERT. As informações do teste de desempenho, referente a criação do SLA, como nome da transação e estimativas do tempo (otimista, mais provável e pessimista) ficam armazenadas em um arquivo XML, chamado Tempo Esperado, bem como o valor referente ao tempo de resposta esperado. Após calcular o valor esperado, tendo como referência os valores que estão presentes no arquivo XML, são gerados e executados os *scripts* de teste de desempenho, incluindo a criação dos SLAs no LoadRunner.

Com os resultados obtidos, é realizada a etapa de análise, onde é possível identificar, de acordo com o cenário criado e a definição dos SLAs quais as transações que passaram e quais as transações que falharam no teste de desempenho. É possível também fazer uma análise comparando o tempo mínimo, médio e máximo de resposta de cada transação, aferindo nos critérios de tempo caso seja necessário calibrar o modelo. É importante ressaltar que essa iteração poderá ser efetuada diretamente na PLeTs. A Figura 3.4 apresenta a estrutura do arquivo XML, onde estão presentes as transações, e em cada transação as estimativas e o tempo esperado para sua execução, tanto para um teste de carga normal como de *stress*.

```
<workloads>
  <workload ="normal">
    <transacoes>
      <nome da transacao="Pesquisar_Certificacoes">
        <Otimista>0.400</Otimista>
        <MaisProvavel>1.000</MaisProvavel>
        <Pessimista>2.300</Pessimista>
        <TempoEsperado>1.116</TempoEsperado>
      </transacao>
      <nome da transacao="Editar_Certificacoes">
        <Otimista>0.500</Otimista>
        <MaisProvavel>1.300</MaisProvavel>
        <Pessimista>2.200</Pessimista>
        <TempoEsperado>1.316</TempoEsperado>
      </transacao>
      <nome da transacao="Adicionar_Certificacoes">
        <Otimista>0.700</Otimista>
        <MaisProvavel>1.200</MaisProvavel>
        <Pessimista>2.100</Pessimista>
        <TempoEsperado>1.266</TempoEsperado>
      </transacao>
    </transacoes>
  </workload>
```

```

<workload ="stress">
  <transacoes>
    <nome da transacao="Pesquisar_Certificacoes">
      <Otimista>3.000</Otimista>
      <MaisProvavel>5.500</MaisProvavel>
      <Pessimista>7.900</Pessimista>
      <TempoEsperado>5.483</TempoEsperado>
    </transacao>
    <nome da transacao="Editar_Certificacoes">
      <Otimista>2.500</Otimista>
      <MaisProvavel>3.400</MaisProvavel>
      <Pessimista>4.600</Pessimista>
      <TempoEsperado>3.450</TempoEsperado>
    </transacao>
    <nome da Transacao ="Adicionar_Certificacoes">
      <Otimista>3.000</Otimista>
      <MaisProvavel>4.500</MaisProvavel>
      <Pessimista>6.000</Pessimista>
      <TempoEsperado>4.500</TempoEsperado>
    </transacao>
  </transacoes>
</workload>
</workloads>

```

Figura 3.4: Estrutura do arquivo XML Tempo Esperado

Neste arquivo XML estão presentes as transações modeladas no diagrama de atividades representado pela Figura 3.3. Para a transação *Pesquisar\_Certificações* por exemplo foram informados os seguintes tempos de resposta em um teste de carga (*workload*) normal: Otimista = 0.400, Mais Provável = 1.000 e Pessimista = 2.300 resultando em um Tempo Esperado de 1.116 segundos. Já para um teste de stress foram informados os seguintes tempos de resposta: Otimista = 3.000, Mais Provável = 5.500 e Pessimista = 7.900 resultando em um Tempo Esperado de 5.483 segundos.

### 3.3 Processo de Geração dos Casos de Teste

Conforme pode ser observado na Figura 3.5, o processo de geração dos cenários, casos e *scripts* de teste inicia com a modelagem dos diagramas UML. Nesta etapa são incluídas informações do cenário de teste e das interações do usuário com o sistema, além das transações e das estimativas de tempo referente a execução de cada uma delas.

Após a construção dos modelos, com o cenário de teste definido e as informações incluídas, é gerado um arquivo XML correspondente. Neste arquivo é descrita a estrutura do sistema, contemplando as informações referentes ao diagrama de casos de uso e diagrama de atividades (transações e estereótipos).

Na etapa seguinte, essas informações são submetidas a PLeTs a fim de convertê-las em uma estrutura de dados intermediária armazenada em memória, que visa implementar o conjunto de

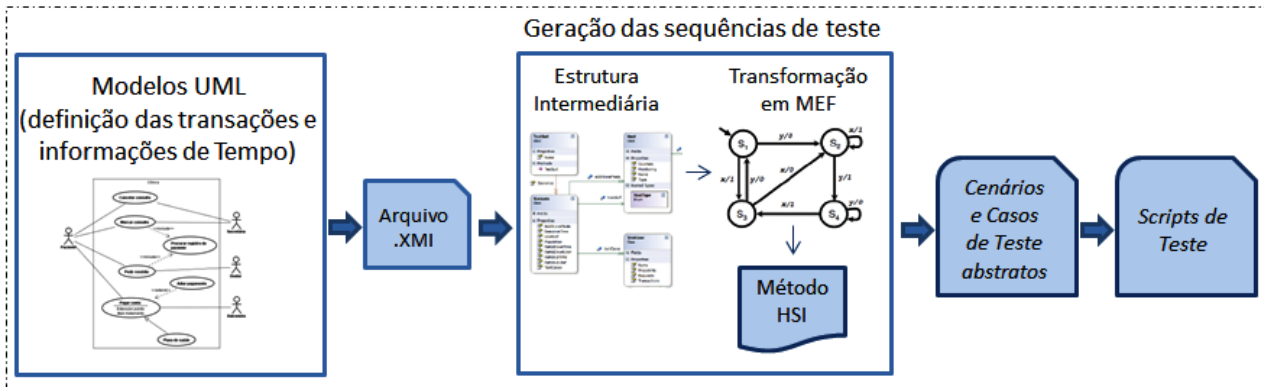


Figura 3.5: Processo de geração dos casos de teste

características para teste de desempenho. É esta estrutura intermediária que os diferentes modelos formais que podem ser usados, (e.g.,MEF) terão como entrada. Isso possibilita que novos produtos derivados da PLeTs para MBT possam ter diferentes modelos de entrada, não se resumindo apenas a UML. É importante ressaltar, que nesta etapa são interpretadas as informações referente as estimativas de tempo (otimista, mais provável e pessimista) de cada transação e que é calculado o valor esperado que definirá o SLA no cenário de teste abstrato.

Posteriormente, esta estrutura intermediária é convertida em uma MEF, e por meio da aplicação do método de geração de seqüências de teste HSI são gerados cenários e casos de desempenho que serão executados pelo LoadRunner. O método HSI (*Harmonized State Identification*) [36] [43] [42] utiliza uma máquina de estados finitos com entradas e saídas para identificar os possíveis caminhos de execução. Este método foi aplicado, uma vez que ele atende as especificações definidas e faz parte da linha de produto PLeTs. A partir do conjunto de seqüências de teste obtidos com a aplicação do método HSI, a ferramenta permite concretizá-los em casos e cenários de testes abstratos.

Os cenários de teste abstratos possuem as informações relacionadas ao contexto do teste e a distribuição dos usuários virtuais propagados para cada caso de teste que compõem o cenário, de acordo com a probabilidade informada por meio do estereótipo TDprob além dos SLAs, referente a cada transação que foi especificada nos diagrama de atividades juntamente com o tempo esperado, calculado com base nos valores informados por meio do rótulo TDexpTime. Enquanto que os casos de teste abstratos contém as informações necessárias das tarefas a serem realizadas pelo usuário.

Uma das vantagens na geração de cenários e casos de teste abstratos é a possibilidade de converter estas informações em *scripts* que serão interpretados por geradores de carga, tais como o LoadRunner ou qualquer outra ferramenta que utilize *scripts* para automação do teste. Em outras palavras, permite criar *scripts* de teste, independentemente da ferramenta ou tecnologia, sendo necessário apenas implementar um novo *plug-in* [55] para a ferramenta PLeTs a fim de converter os casos de teste abstratos em instâncias de teste executáveis para uma ferramenta de teste de desempenho específica.

É importante salientar que o processo de transformação do modelo UML em MEFs, limitou-se aos diagramas de atividades, uma vez que estes possuem as informações relacionadas as interações dos

usuários com o SUT. Desta forma, as informações relacionadas ao teste de desempenho anotadas são mantidas nos modelos de entrada e armazenadas na estrutura intermediária. Isso possibilita flexibilidade na escolha da ferramenta ou tecnologia utilizada na etapa de execução dos casos de testes, de acordo *e.g.*, com o projeto de *software* que está sendo desenvolvido, ou ainda de acordo com o conhecimento da equipe de testes.

### 3.4 Exemplo de Uso

Com o intuito de validar as informações de tempo relacionadas aos modelos UML e implementadas no processo de geração de casos de teste da PLeTs, esta seção exemplifica o processo de construção dos diagramas de casos de uso e atividades com informações de tempo relacionadas, bem como a geração dos casos de teste baseados nos modelos e sua instrumentalização em *scripts* de teste para a ferramenta LoadRunner. O objetivo é compreender as premissas e definições necessárias para elaborar os modelos que irão apoiar as atividades MBT na realização dos testes de desempenho.

Conforme descrito na Seção 2.5, o processo de geração dos casos de teste tem início com a criação de um cenário de teste, representado pelo diagrama de casos de uso. Em seguida, cada caso de uso é decomposto em um diagrama de atividades onde as transações são especificadas e as informações de tempo são incluídas. Estas informações são definidas com base na técnica PERT e são adicionadas nas transições do diagrama de atividades, juntamente com os demais estereótipos necessários para a geração e execução dos casos de teste de desempenho. Para isso, serão utilizados o *benchmark* TPC-W e a aplicação *Skills* como exemplos de uso.

#### 3.4.1 Aplicação TPC-W

O TPC-W (*Transactional web e-commerce benchmark*) [52] é um *software* desenvolvido para realizar testes de desempenho em infraestruturas. Ele pode ser utilizado, por exemplo, para verificar o número de usuários que um ambiente pode suportar, ou se este dispõe da quantidade de recursos necessários para prover um serviço. Para desempenhar esta função, ele simula uma série de ações do usuário, reproduzindo as principais operações realizadas durante a visita a uma aplicação de *e-commerce*. A Tabela 3.2 aborda os principais requisitos funcionais da aplicação.

Tabela 3.2: Requisitos funcionais da aplicação TPC-W

ID	Funcionalidade	Requisito
RF01	Browsing	O sistema deve possibilitar ao usuário pesquisar pelos produtos com base nos cadastros existentes.
RF02	Ordering	O sistema deve permitir o usuário selecionar os produtos que deseja adquirir e adicionar ao carinho de compras.
RF03	Shopping	O sistema deve permitir o usuário realizar a compra dos produtos selecionados.

Baseado nos requisitos funcionais da aplicação TCP-W definidos na Tabela 3.2, o diagrama de casos de uso da Figura 3.6 representa estas funcionalidades divididas em três casos de uso, sendo eles: *Browsing*, *Shopping*, *Ordering*. O modelo demonstra o comportamento de iteração de dois perfis de usuários com o sistema: *Customer* e *New customer*. Em termos de funcionalidades os perfis diferem especificamente na execução da atividade *shopping*, atribuída somente ao ator *Customer*, uma vez que é preciso estar cadastrado no sistema para realizar alguma compra.

Os estereótipos PApopulation, PAtime e PApob, foram atribuídos nos atores definidos no modelo, sendo que eles basicamente diferem nos seguintes aspectos:

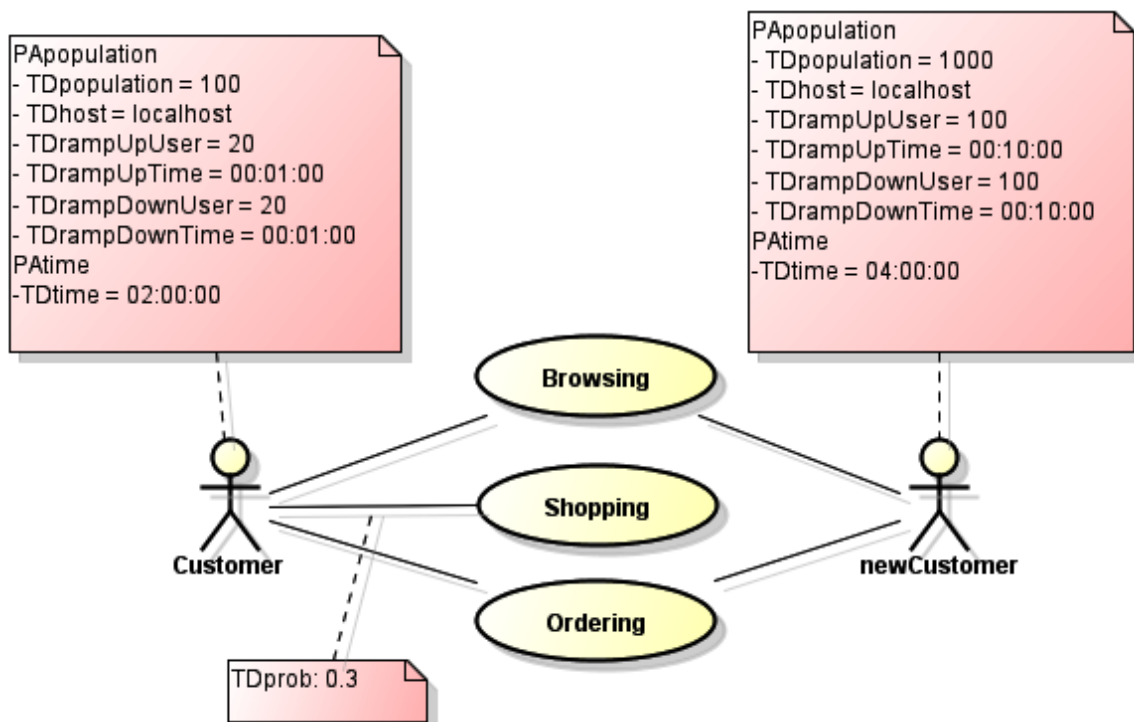


Figura 3.6: Diagrama de Casos de Uso da aplicação TPC-W

1.“Customer“- foi modelado um cenário fictício de 100 usuários virtuais. Durante a inicialização do teste são acrescentados 20 usuários (TDrampUpUser) a cada minuto (TDrampUpTime), totalizando 5 minutos para a inicialização do teste. Para a finalização do teste (TDrampDownUser e TDrampDownTime) foram configurados os mesmos valores da inicialização. Assim, subtraindo o total de 10 minutos entre a inicialização e finalização do teste, resultando em 1h50min de execução do teste com 100 usuários virtuais concorrentes, para uma duração total da execução do teste de 2h (TDtime).

2.“New Customer“- também foi modelado um cenário fictício composto por 1000 usuários virtuais, sendo que durante a inicialização do teste são acrescentados 100 usuários (TDrampUpUser) a cada 10 minutos (TDrampUpTime), totalizando 1h40min para a inicialização total do teste, assim como a finalização do teste (TDrampDownUser e TDrampDownTime) também foram configurados com os mesmos parâmetros. Totalizando 3h20min entre a inicialização e finalização do teste, resultando em 40min de execução do teste com os 1000 usuários virtuais concorrentes para uma duração

total de execução do teste de 4h (TDtime).

Outra característica importante presente no modelo é o parâmetro TDprob. Esta informação define a probabilidade de distribuição dos usuários virtuais na associação entre ator e casos de uso. Conforme demonstra a Figura 3.6, a probabilidade do ator *Customer* executar o caso de uso *Shopping* é 30% (TDprob), representando 30 usuários virtuais dos 100 concorrentes. Os valores apresentados na Tabela 3.3 representam as demais probabilidades de distribuição dos atores em relação aos casos de uso com que eles interagem. É importante ressaltar que por meio deste estereótipo também é possível simular casos em que 100% dos usuários virtuais executem determinado caso de uso, isso implicaria no pior caso de utilização do sistema.

Tabela 3.3: Cenário de Teste da aplicação TPC-W

Ator	Caso de Uso	Probabilidade	Usuários Virtuais
Customer	Browsing	40%	40
	Shopping	30%	30
	Ordering	30%	30
			100
New Customer	Browsing	55%	550
	Ordering	45%	450
			1000

Após definido o cenário de teste, a próxima etapa da elaboração do modelo é decompor cada caso de uso em diagramas de atividades, os quais descrevem as ações que o usuário irá executar para realizar determinada tarefa no sistema.

Foi desenvolvido um diagrama de atividades representado pela Figura 3.7, correspondente a um dos casos de uso referente à aplicação TPC-W com o intuito de representar as anotações das informações por meio dos estereótipos, além do comportamento de iteração do usuário com a aplicação. Os retângulos representam os estados e/ou ações e as setas determinam as transições entre eles. É importante considerar que no modelo, é possível aplicar desvios no fluxo como também modelar atividades paralelas.

A Figura 3.7 demonstra o diagrama de atividades do caso de uso *Shopping*, representando . Os diagramas de atividades referente aos demais casos de uso apresentados na Figura 3.6 podem ser encontrados no Apêndice A.

A primeira iteração é representada pela atividade *Home Page*, que habilita as opções de escolha do sistema após seu acesso. Em seguida um novo produto poderá ser cadastrado, por meio da atividade *New Products*, como também poderá ser realizada a busca por um produto já existente (*Search Request*). Essa busca poderá ser realizada por exemplo pelo Título do livro. Clicando no resultado selecionado o usuário será redirecionado para uma página com os detalhes do produto (*Product Detail*). Após verificação dos detalhes, para realizar a compra (*Shopping Cart*) será necessário preencher uma página referente ao cadastro das informações, tanto do usuário como também do endereço de entrega do produto e a forma de pagamento (*Customer Registration*), para em seguida, efetuar o pedido (*Buy Request*) e confirmar a compra (*Buy Confirm*). Conforme pode-

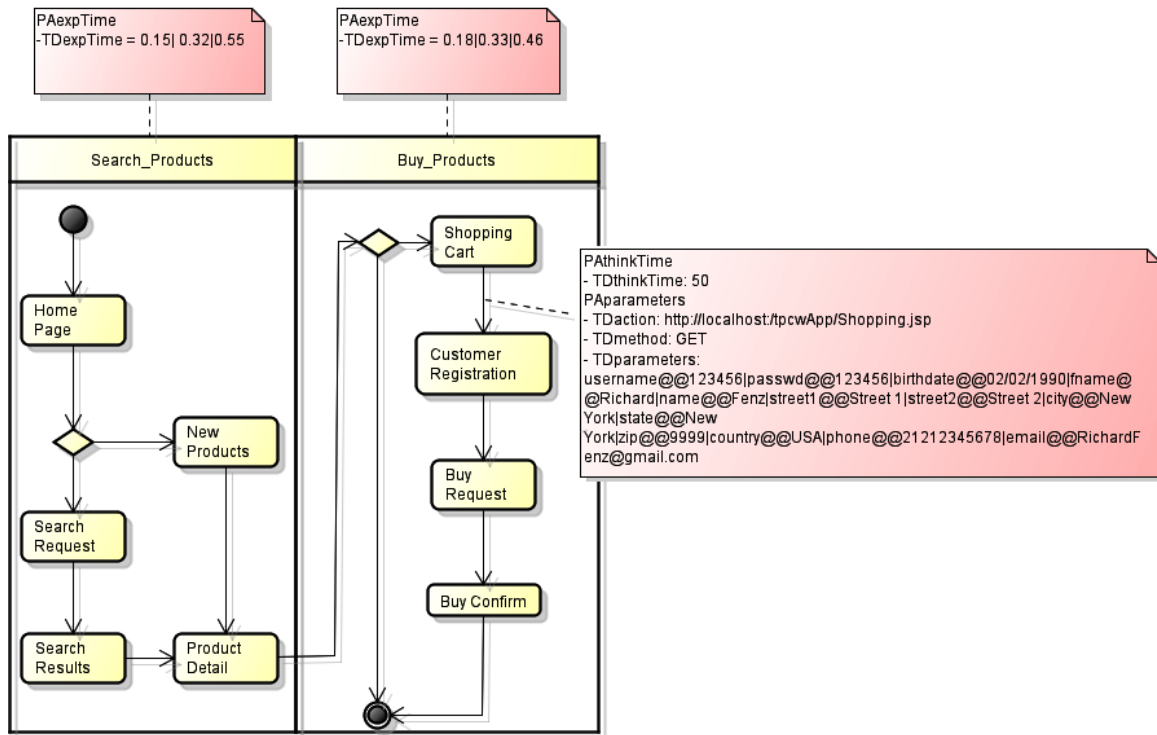


Figura 3.7: Diagrama de atividades do caso de uso *Shopping*

se observar na Figura 3.7 essas atividades fazem parte de duas transações principais identificadas como: *Search\_Products* e *Buy\_Products*.

Com relação as informações de teste de desempenho, no diagrama de atividades é exemplificado uma transição com os seguintes estereótipos anotados: *PAparameters* e *PAtthinkTime*. O *TDthinkTime* define um tempo estimado de 0.50 segundos para o preenchimento dos dados pelo usuário para a submissão da requisição. Estes dados são submetidos para o endereço *http://localhost:tpcwApp/Shopping.jsp*, anotado no rótulo *TDaction* usando o método *GET* definido no rótulo *TDmethod*. Já o rótulo *TDparameters* é formado pela concatenação das seguintes informações: usuário, senha, data de aniversário, nome, sobrenome, endereço do cliente, endereço de entrega, cidade, estado, cep, país e número para contato.

Estes parâmetros são representados no modelo pelo nome do objeto (*name*) e pelo valor do objeto (*value*), separadas pelo delimitador "@@". Quando há mais de um parâmetro a anotação é representada da mesma maneira, porém separando cada parâmetro pelo delimitador "|", anotados da seguinte forma: "username@@123456|passwd@@123456|birthdate@@02021990|fname@@Richard|name@@Fenz|street1@@Street 1|street2@@Street 2|city@@New York|state@@New York|zip@@9999|country@@USA|phone@@21212345678|email@@RichardFenz@gmail.com".

Como a maior contribuição deste estudo é gerar sequências de teste para teste de desempenho, considerando o tempo de resposta das sequências de teste, nos modelos UML, foi adicionado um novo estereótipo (*PAexpTime*), representado pelo rótulo *TDexpTime*. Este rótulo ao contrário dos demais está presente nas transações do diagrama de atividades, e nele são informadas as estimativas de tempo referente à execução de cada transação, representada na Figura 3.7 pelos valores 0.15,



0.32, e 0.55 na transação *Search\_Products* e 0.18, 0.33 e 0.46 na transação *Buy\_Products*. É importante ressaltar que estes valores serão utilizados pela ferramenta LoadRunner na definição do SLA. Caso esta informação não for adicionada ao modelo, estes valores são coletados por meio da realização de um *smoke test*, conforme descrito anteriormente.

Os diagramas UML modelados são representados por um arquivo XML com as informações de teste. Posteriormente, esses modelos UML são convertidos em MEF, onde por meio da aplicação do método de geração de sequência de teste HSI são gerados os casos e os cenários de teste com as informações (requisições, parâmetros e transações) que foram incluídas nos modelos por meio dos estereótipos, conforme descrito na Seção 3.1.

Baseado nas sequências de teste geradas pelo HSI a estrutura abaixo representa o caso de teste abstrato da funcionalidade *Shopping*.

- #Caso de Teste: Shopping

1. Home Page

«TDmethod:POST»

«TDaction:http://192.165.1.27:8080/TpcwApp/Homepage.jsp»

«TDthinkTime: 5»

«TDtransaction:[Home Page | Search\_Products]»

2. Search Request

«TDmethod:POST»

«TDaction:http://192.165.1.27:8080/TpcwApp/SearchRequest.jsp»

«TDthinkTime:5»

«TDtransaction:[Search Request | Search\_Products]»

3. Search Results

«TDmethod:POST»

«TDaction:http://192.165.1.27:8080/TpcwApp/SearchResults.jsp»

«TDparameters:[Title@@PerformanceTest]»

«TDthinkTime:5»

«TDtransaction:[Search Results | Search\_Products]»

4. Product Detail

«TDmethod:POST»

«TDaction:http://192.165.1.27:8080/TpcwApp/ProductDetail.jsp»

«TDthinkTime:15»

«TDtransaction:[Product Detail | Search\_Products]»

5. Shopping Cart

«TDmethod:GET»

«TDaction:http://192.165.1.27:8080/TpcwApp/Product/ShoppingCart.jsp»

«TDthinkTime:5»

«TDtransaction:[Shopping Cart | Buy\_Products]»

6. Customer Registration

```

«TDmethod:GET»
«TDaction:http://192.165.1.27:8080/TpcwApp/Product/CustomerRegistration.jsp»
«TDparameters:[id@@123456|password@@123456|BirthDate@@02/02/1990|FirstName@@
Richard|LastName@@Fenz|Address1@@Street1|Address2@@Street2|City@@NewYork|Sta
te@@NewYork|Zip@@USA|Country@@USA|Phone@@12345678@E-mail@@RichardFenz@gmail»
.com]>»
«TDthinkTime:5»
«TDtransaction:[Customer Registration | Buy_Products]>»
7. Buy Request
«TDmethod:GET»
«TDaction:http://192.165.1.27:8080/TpcwApp/Product/BuyRequest.jsp»
«TDparameters:[CreditCardType@@Visa|NameCreditCard@@RichardFenz|CreditCard
Number@@1457784587457845|CreditCardExpirationDate@@02/02/2015|ShippingMeth»
od@@AIR]>»
«TDthinkTime:4»
«TDtransaction:[Buy Request | Buy_Products]>»
8. Buy Confirm
«TDmethod:GET»
«TDaction:http://192.165.1.27:8080/TpcwApp/Product/BuyConfirm.jsp»
«TDthinkTime:4»
«TDtransaction:[Buy Confirm | Buy_Products]>»
9.Exit
«TDaction:exit»

```

Os casos de teste abstratos fazem uso de uma abordagem hierárquica, onde as atividades são enumeradas e estruturadas de acordo com a dependência entre as atividades do AD. Outro detalhe que é possível observar na descrição dos casos de teste abstratos é a variação dos valores adicionados aos parâmetros de cada atividade, demonstrando a flexibilidade de configuração dos modelos.

Esta mesma abordagem poderia ser estendida para transmitir o paralelismo e a sincronização modelados no diagrama de atividades por meio dos elementos UML *Fork* e *Join*. Desta forma, se uma atividade pertence a um determinado nível, ele deve cumprir todos os requisitos antes de proceder para a próxima atividade.

Com o intuito de definir os SLAs, no rótulo TDtransaction é anotada a atividade e a qual transação ela pertence. Por exemplo, no rótulo TDtransaction da atividade 1.Home Page, é anotado o nome da atividade Home Page e a transação *Search\_Products*.

Já a quantidade de cenários de testes gerados a partir de um modelo UML está relacionado diretamente a quantidade de atores adicionados ao modelo. Neste caso, para o exemplo de uso da aplicação *Skills* foram gerados dois cenários de teste abstratos, um deles representado pelo cenário de teste *Customer*.

- Nome do Cenário de Teste: Customer
  - ## Configuração do Teste
  - Usuários Virtuais : «TDpopulation:100»
  - Host do SUT : «TDhost:localhost»
  - Tempo de Execução : «TDtime:7200»
  - Tempo de Inicialização : «TDrampUpTime:60»
  - Usuários de Inicialização : «TDrampUpUser:20»
  - Tempo de Finalização : «TDrampDownTime:60»
  - Usuários de Finalização : «TDrampDownUser:20»
  - ## Distribuição dos Casos de Teste:
  - «TDprob:0.40» «TDpopulation:40»
  - 1. Browsing
  - «TDprob:0.30» «TDpopulation:30»
  - 2. Shopping
  - «TDprob:0.30» «TDpopulation:30»
  - 3. Ordering
  - ## Contadores de Desempenho:
  - Transações por Segundo : «TDtps:Yes»
  - Tempo de Resposta : «TDresponse:Yes»
  - Requisições por Segundo : «TDrequest:Yes»
  - Vazão: «TDthroughput:Yes»
  - Utilização de Recursos : «TDresource:Yes»
  - ## SLAs:
  - Search\_Products: «TDexpTime: 0.330»
  - Buy\_Products: «TDexpTime: 0.327 »

Conforme pode ser observado, um cenário de teste de desempenho agrega informações relacionadas ao contexto do teste e o conjunto dos casos de testes que devem ser testados, incluindo a distribuição do número de usuários virtuais para cada caso de teste. Desta forma, a estrutura do cenário está dividida em quatro blocos: 1) Configuração - carrega as características genéricas que são aplicadas a todo contexto do teste, basicamente, informações oriundas do modelo UML de diagrama de casos de uso. 2) Distribuição - são vinculados as diferentes sequências de testes geradas pelas MEFs. Observa-se que no cabeçalho de cada caso de teste abstrato constam as informações de probabilidade e sua respectiva quantidades de usuários virtuais propagadas. 3) Contadores - constam os dados de quais contadores de desempenho devem ser mensurados para o cenário de teste abstrato. 4) SLAs - formados pelas transações presentes no diagrama de atividade com os respectivos tempos de resposta esperados. Estes tempos são calculados por meio da fórmula presente na técnica PERT, tendo como base as estimativas de tempo incluídas em cada transação por meio do rótulo TDexpTime.

Baseados nos cenários e casos de teste abstratos apresentados, a próxima etapa tem por finalidade gerar as instâncias destes cenários e casos de teste abstratos. Estas instâncias são chamadas de casos de teste concretos ou executáveis, pois dependem da escolha da tecnologia utilizada para a geração dos cenários e *scripts*. Conforme descrito, para o presente trabalho será utilizada a ferramenta HP LoadRunner [25].

Na Figura 3.8, são apresentadas algumas das informações utilizadas para a configuração do cenário de teste de desempenho gerado pela PLETs para o LoadRunner. Nesta figura estão destacados as seguintes informações: *Vusers*, que corresponde ao número total de usuários configurados para executar o teste *RunFor*, refere-se ao tempo total de execução do teste em milissegundos; *TotalVusersNumber*, que corresponde à quantidade de usuários que irá executar as atividades descritas em determinado *script*; *Count*, que corresponde ao número de usuários que iniciarão o teste (Quantidade de Usuários da Rampa de Subida) ou irão deixar de realizar suas requisições (Quantidade de Usuários da Rampa de Descida) e *Interval*, que define o tempo que levará para um conjunto de usuários iniciar o teste (Tempo de Rampa de Subida) ou terminá-lo (Tempo de Rampa de Descida).

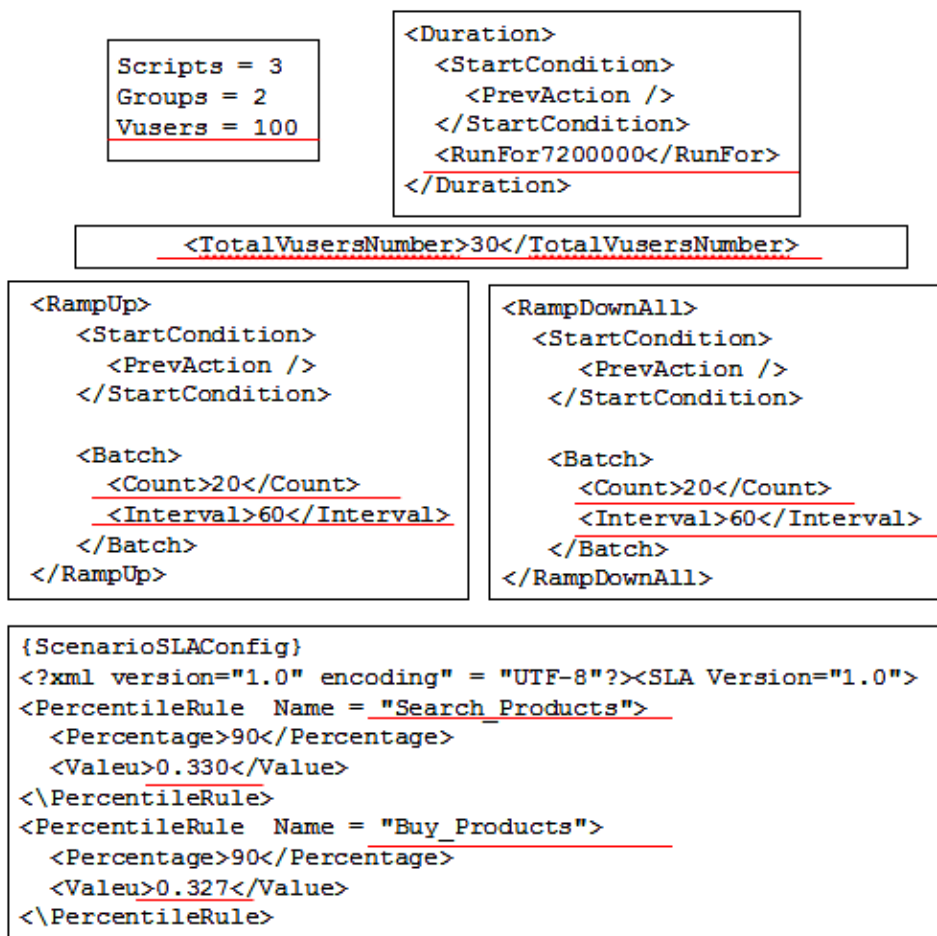


Figura 3.8: Cenário de teste gerado para o LoadRunner - TPC-W

Outra informação presente no cenário de teste são os SLAs de cada transação, que foram calculados tendo como base os tempos estimados informados nos modelos. Por exemplo, a transação

*Search\_Products* tem o SLA igual a 0.330 segundos. Por limitações de espaço foram apresentados somente os SLAs das transações presentes no diagrama de atividades *Shopping* do ator *Customer*, no entanto, no cenário estão presentes todas as transações definidas nos demais diagramas de atividades, assim como há um outro cenário para o ator *New Customer*.

Já a Figura 3.9, apresenta um trecho do código XML, referente ao *script* de teste de desempenho gerado pela PLeTs para o LoadRunner. Esse *script* de teste foi instanciado a partir do caso de teste abstrato *Shopping*. Conforme pode-se observar, ele é composto pelas diversas requisições (requests) presentes na transação *Search\_Request*, efetuadas pelo usuário na pesquisa de um livro.

```

lr_start_transaction("Search_Products")
    lr_start_sub_transaction("Home Page")
        web_url("Home Page ",
            "URL=http://192.165.1.27:8080/TpcwApp/Homepage.jsp",
            "Method=POST",
            "Resource=0",
            "Referer=http://192.165.1.27:8080/TpcwApp/Homepage.jsp",
            "Mode=HTTP",
            lr_end_sub_transaction("Home Page",LR_AUTO)
lr_think_time(5);
    lr_start_sub_transaction("Search Request")
        web_url("Home Page ",
            "URL=http://192.165.1.27:8080/TpcwApp/SearchRequest.jsp»,
            "Method=POST",
            "Resource=0",
            "Referer=http://192.165.1.27:8080/TpcwApp/SearchRequest.jsp»,
            "Mode=HTTP",
            lr_end_sub_transaction("Search Request",LR_AUTO)
lr_think_time(5);
    lr_start_sub_transaction("Search Results")
        web_url("Home Page ",
            "URL=http://192.165.1.27:8080/TpcwApp/SearchResults.jsp",
            "Method=POST",
            "Resource=0",
            "Referer=http://192.165.1.27:8080/TpcwApp/SearchResults.jsp",
            "Mode=HTTP",
            ITEMDATA,
                "Name=Title", "Value=PerformanceTest", ENDITEM,
            LAST);
        lr_end_sub_transaction("Search Results",LR_AUTO)
lr_think_time(5);
    lr_start_sub_transaction("Product Detail")
        web_url("Home Page ",
            "URL=http://192.165.1.27:8080/TpcwApp/ProductDetail.jsp",
            "Method=POST",
            "Resource=0",
            "Referer=http://192.165.1.27:8080/TpcwApp/ProductDetail.jsp",
            "Mode=HTTP",
            lr_end_sub_transaction("Product Detail",LR_AUTO)
lr_think_time(5);
lr_end_transaction("Folha Ponto Tarefa",LR_AUTO);

```

Figura 3.9: *Script* de teste gerado para o LoadRunner - TPC-W

Ao término da execução dos *scripts*, o LoadRunner apresenta um resumo, demonstrando os tempos de resposta de cada transação, além do status de quais as transações que passaram no teste

e quais as transações que falharam de acordo com o SLA definido, possibilitando assim, comparar e analisar os resultados obtidos com a realização do teste.

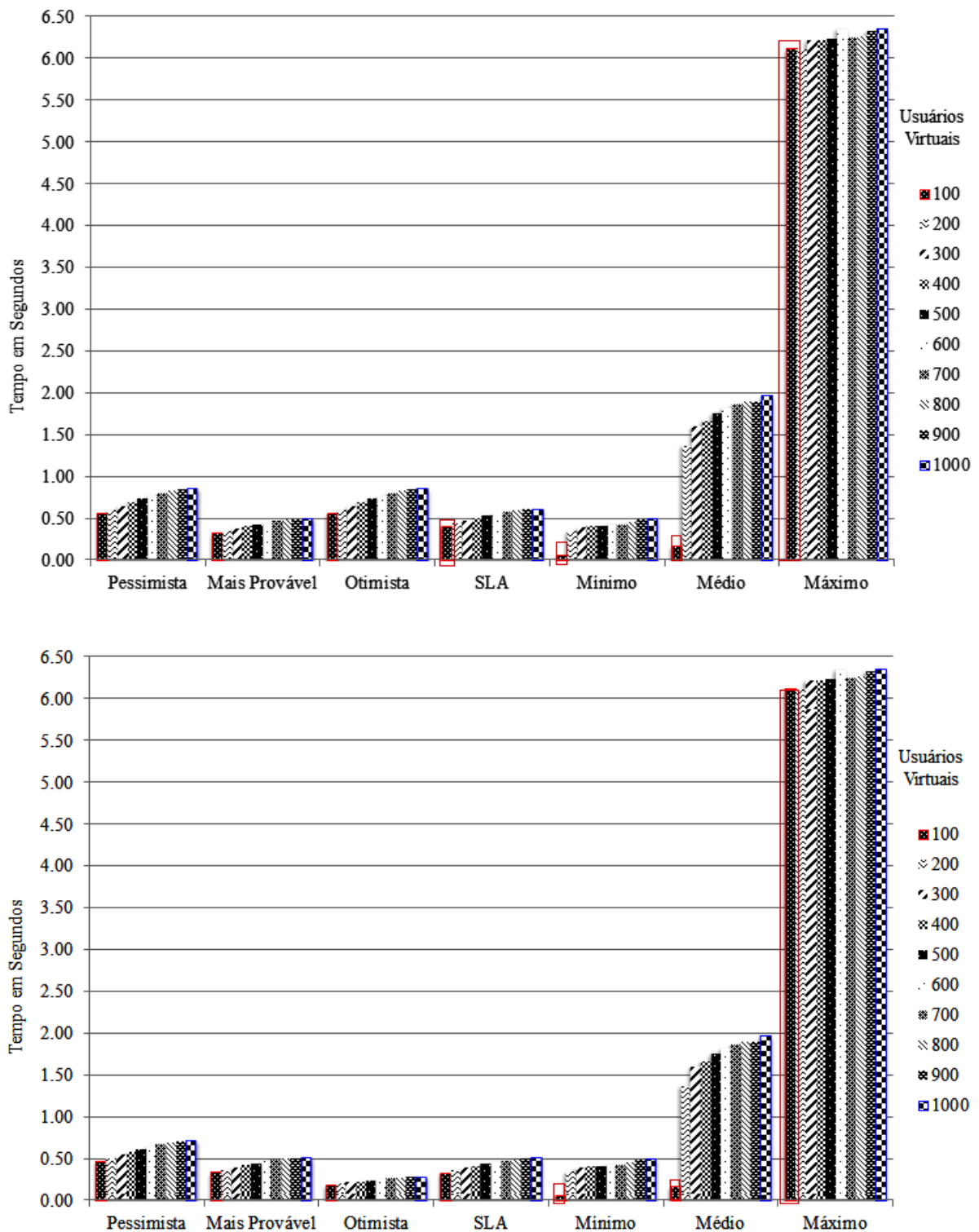


Figura 3.10: Gráfico de análise do tempo de resposta das transações *Search\_Products* e *Buy\_Products* do diagrama de atividades da Figura 3.7

Por meio dos gráficos presentes na Figura 3.10 é possível observar que ambas as transações tiveram um tempo médio de execução maior que o tempo definido pelo SLA. Isso só não aconteceu

quando a primeira rodada do teste foi executada. Nota-se um pequeno aumento do tempo em relação aos SLAs, no entanto, este aumento não está de acordo com o aumento do tempo referente a execução dos testes. Neste caso é importante que as estimativas de tempo que servem como base para a definição dos SLAs sejam revistas desde as rodadas iniciais do teste e que os valores estimados tomem como base os próprios resultados da execução, uma vez que o tempo médio de execução teve um aumento considerável já na segunda rodada do teste. Isso possibilitaria que as transações atendessem os SLAs em um tempo médio de execução e não somente em um tempo mínimo.

É importante ressaltar que as informações referente a execução dos testes, bem como as transações e estimativas de tempo também são armazenadas em um arquivo XML chamado de Tempo Esperado, conforme descrito na Seção 3 e são reportadas para a PLeTs. Sendo assim, o usuário poderá além de analisar os resultados, interagir com a aplicação calibrando o modelos de acordo com o cenário e a carga que deseja testar, tendo como base os próprios resultados de execução dos testes.

### 3.4.2 Aplicação *Skills*

Além do *benchmark* TPCW, foi utilizada a aplicação *web Workforce Planning: Skill Management Prototype Tool*, também conhecida como *Skills* [12]. A aplicação tem por objetivo controlar o conhecimento dos colaboradores de uma empresa em termos de habilidades, certificações e experiências, para assim gerenciar a alocação dos recursos disponíveis no desenvolvimento dos projetos. Este *software* foi desenvolvido em linguagem de programação Java, utiliza o Sistema Gerenciador de Banco de Dados (SGBD) MySQL [56] para a persistência dos dados e o TomCat [57] como servidor da aplicação. A Tabela 3.4 aborda os principais requisitos funcionais que fazem parte da aplicação *Skills*.

Tabela 3.4: Requisitos funcionais da aplicação *Skills*

ID	Funcionalidade	Requisito
RF01	Gerenciar Habilidades	O sistema deve permitir ao usuário adicionar e/ou editar suas habilidades. Isso inclui tecnologias, <i>hardware</i> , <i>software</i> , técnicas, métodos e línguas, informando o nível de proficiência e o ano em que adquiriu a habilidade e quando ela foi implementada.
RF02	Gerenciar Experiências	O sistema deve permitir o usuário adicionar e/ou editar as experiências adquiridas.
RF03	Gerenciar Certificações	O sistema deve permitir ao usuário alterar e/ou editar suas certificações tanto profissionais quanto acadêmicas.
RF04	Alterar Senha	O sistema deve permitir ao usuário alterar sua senha.

Baseado nos requisitos funcionais da aplicação *Skills* definidos na Tabela 3.4, o diagrama de casos de uso da Figura 3.11 representa estas funcionalidades divididas em quatro casos de uso, sendo eles: gerenciar habilidades, gerenciar experiências, gerenciar certificações e alterar senha. Por

meio do diagrama é possível observar o comportamento de dois perfis de usuários que interagem com a aplicação, sendo eles: gerente RH e empregado. Basicamente, o que difere os perfis em termos de funcionalidades é o caso de uso alterar senha que é executado somente pelo ator Gerente RH. Os estereótipos PApopulation, PAtime e PApob, foram atribuídos nos autores definidos no modelo, sendo que eles apresentam as seguintes características:

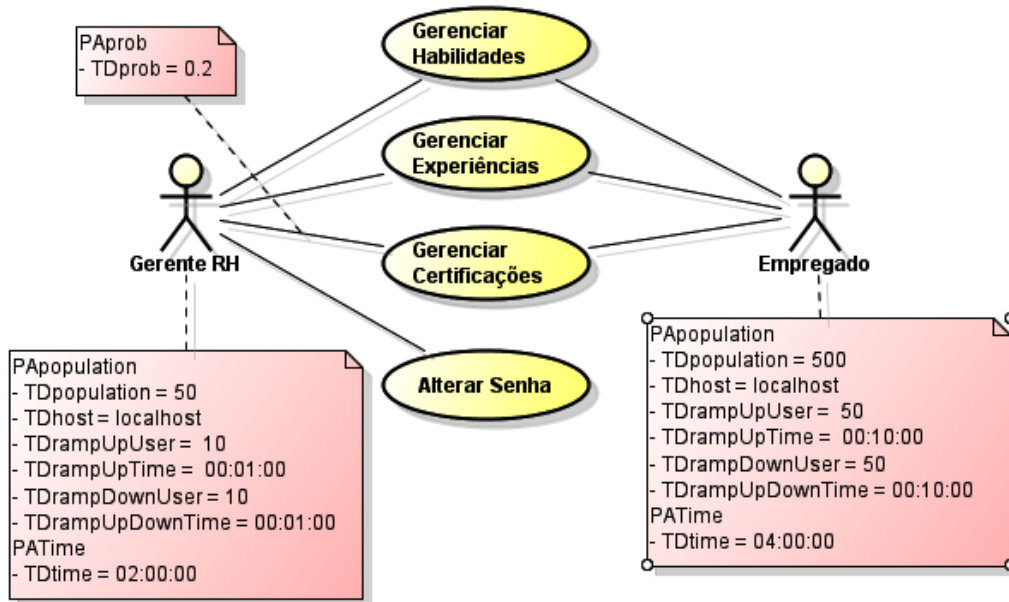


Figura 3.11: Diagrama de Casos de Uso da aplicação *Skills*

1.“Gerente RH“- foi modelado um cenário fictício de 50 usuários virtuais. Durante a inicialização do teste são acrescentados 10 usuários (TDramUpUser) a cada minuto (TDdrampUpTime), totalizando 5 minutos para a inicialização do teste, para a finalização do teste (TDramDownUser e TDramDownTime) foram configurados os mesmos valores da inicialização. Assim, subtraindo o total de 10 minutos entre a inicialização e finalização do teste, resultando em 1h50min de execução do teste com 50 usuários virtuais concorrentes, para uma duração total da execução do teste de 2h (TDtime).

2.“Empregado“- também foi modelado um cenário fictício composto por 500 usuários virtuais, sendo que durante a inicialização do teste são acrescentados 50 usuários (TDramUpUser) a cada 10 minutos (TDramUpTime), totalizando 1h40min para a inicialização total do teste, assim como a finalização do teste (TDramDownUser e TDramDownTime) também foram configurados com os mesmos parâmetros. Totalizando 3h20min entre a inicialização e finalização do teste, resultando em 40min de execução do teste com os 500 usuários virtuais concorrentes para uma duração total de execução do teste de 4h (TDtime).

Em relação a distribuição dos usuários virtuais por meio do rótulo TDprob, conforme mostrado na Figura 3.11, a probabilidade do ator Gerente RH executar o caso de uso Gerenciar Certificações é 20% (TDprob), representando 15 usuários virtuais dos 50 concorrentes. Os valores apresentados na Tabela 3.5 representam as demais probabilidades de distribuição dos atores Gerente RH e Empregado em relação aos casos de uso com que eles interagem.



Tabela 3.5: Cenário de teste da aplicação *Skills*

Ator	Caso de Uso	Probabilidade	Usuários Virtuais
Gerente RH	Gerenciar Habilidades	40%	20
	Gerenciar Certificações	20%	15
	Gerenciar Experiências	30%	10
	Alterar Senha	10%	5
			50
Empregado	Gerenciar Habilidades	40%	200
	Gerenciar Certificações	25%	175
	Gerenciar Experiências	35%	125
			500

Após criar o cenário de teste, a próxima etapa consiste em decompor cada caso de uso em diagramas de atividades, representando as ações que o usuário irá executar. Com o intuito de representar a anotação das informações por meio dos estereótipos é apresentado o diagrama de atividades referente ao caso de uso Gerenciar Certificações. Os retângulos representam os estados e/ou ações e as setas determinam as transições entre eles. É importante considerar que conforme pode ser observado no modelo, é possível aplicar desvios no fluxo, que são representados por um losango.

A Figura 3.12 apresenta o exemplo do cadastro de certificações, demonstrando o comportamento de interação do usuário com a aplicação *Skills* referente ao caso de uso Gerenciar Certificações. Os diagramas de atividades referente aos demais casos de uso apresentados na Figura 3.11 podem ser encontrados no Apêndice B.

A primeira iteração é representada pela atividade Menu Certificações que habilita as opções de escolha do sistema após seu acesso. Posteriormente, será selecionada a opção Certificações no menu do sistema, e em seguida por meio da atividade Pesquisar Certificação é possível realizar o filtro de determinada certificação pela opção de pesquisa. Outra maneira de realizar este filtro é pela iteração da atividade Árvore de Certificações que possibilita o usuário navegar pelas diferentes certificações listadas na árvore. Depois de selecionado um certificado, independente do caminho, o usuário poderá executar a atividade Adicionar Certificado, caso ela ainda não tenha sido adicionada ou Editar Certificado para editar uma certificação já existente. E por fim, a iteração Deslogar para o usuário sair do sistema.

As anotações das informações do teste de desempenho nos diagramas de atividades são rotuladas com os seguintes estereótipos: PParameters, PThinkTime e PExpTime. Conforme pode-se observar na Figura 3.12 essas atividades fazem parte de três transações principais, identificadas como: *Pesquisar\_Certificações*, *Editar\_Certificações* e *Adicionar\_Certificações*.

Na Figura 3.12 é exemplificada uma transição com os estereótipos anotados. O TDthinkTime define um tempo estimado de 0.11 segundos para o preenchimento dos dados pelo usuário para a submissão da requisição. Estes dados são submetidos para o endereço <http://192.168.1.27:8080/Skills/CertificacoesApp/PesquisarCertificacao.jsp> anotado no rótulo TDaction usando o método

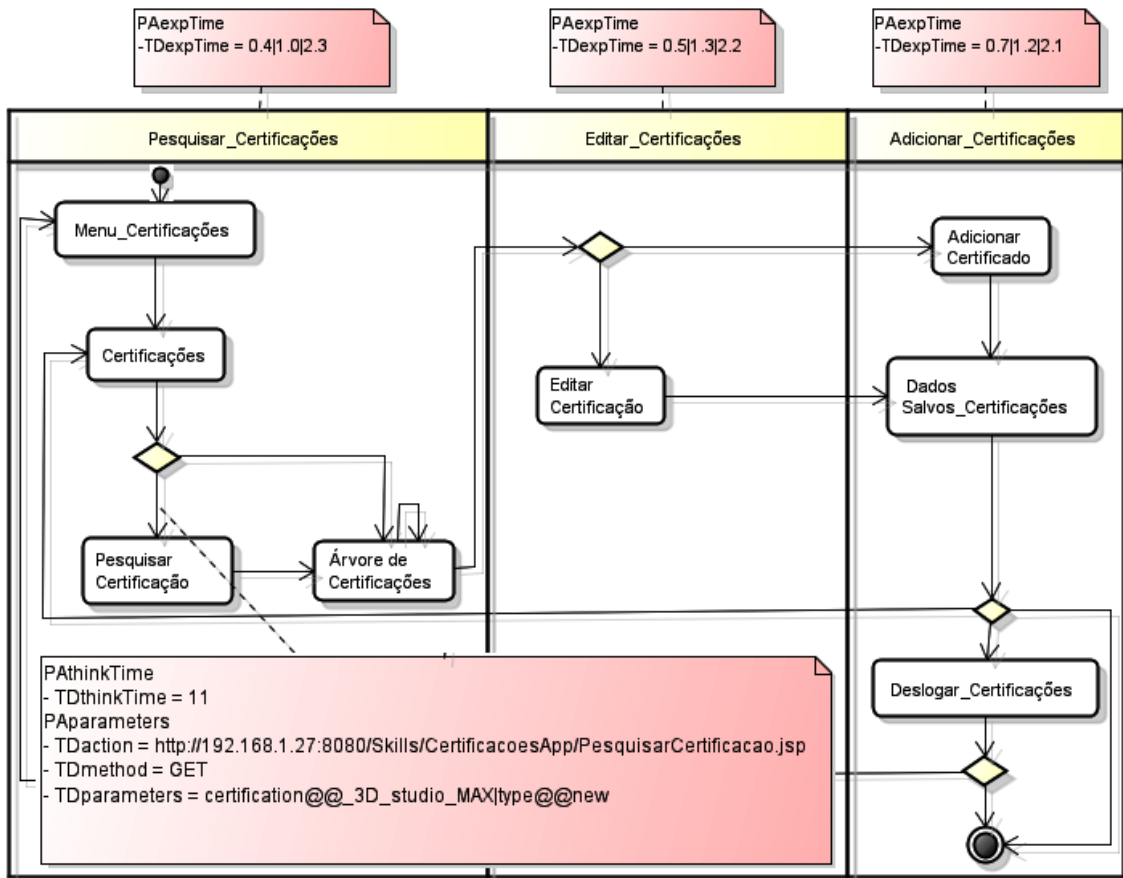


Figura 3.12: Diagrama de atividades do caso de uso Gerenciar Certificações

GET definido no rótulo TDmethod. O rótulo TDparameters é formada pela concatenação de duas informações: certification@@\_3D\_Studio\_MAX|type@@new, nome do objeto (*name*) e o valor do objeto (*value*), separadas pelo delimitador @@, informando a certificação obtida.

É possível observar também o rótulo TDexpTime, onde são anotadas as estimativas de tempo de resposta referente à execução das transações diante das três perspectivas: otimista, mais provável e pessimista com o intuito de calcular o valor esperado. Na transação *Pesquisar\_Certificações* estão anotados os valores 0.4, 1.0 e 2.3, na transação *Editar\_Certificações* estão anotados os valores 0.5, 1.3 e 2.2 e na transação *Adicionar\_Certificações* estão anotados os valores 0.7, 1.2 e 2.1.

Após criados os diagramas com as informações de tempo conforme descrito anteriormente, estes modelos UML são convertidos em MEF e por meio da aplicação do método de geração de sequência de teste HSI são gerados os casos e os cenários de teste abstratos com as informações de teste que foram incluídas nos modelos.

As estruturas apresentadas abaixo representam o caso de teste abstrato da funcionalidade Gerenciar Certificações e o cenário de teste abstrato do ator Gerente RH.

- #Caso de Teste: Gerenciar Certificações
  1. Menu Certificações
    - «TDmethod:POST»

```

«TDaction:http://192.168.1.27:8080/SkillsApp/Login.jsp»
«TDparameters:[user@@admin|pass@@admin]»
«TDthinkTime: 5»
«TDtransaction:[Menu Certificações | Pesquisar_Certificações ]»
2. Certificações
«TDmethod:POST»
«TDaction:http://192.168.1.27:8080/Skills/CertificacoesApp.jsp»
«TDthinkTime:5»
«TDparameters:[link@@Certificacoes»
«TDtransaction:[Certificações | Pesquisar_Certificações ]»
3. Pesquisar Certificação
«TDmethod:POST»
«TDaction:http://192.168.1.27:8080/Skills/CertificacoesApp/PesquisarCertifi»
cacao.jsp
«TDparameters:[certification@@_3D_studio_MAX|type@@new]»
«TDthinkTime:5»
«TDtransaction:[Pesquisar Certificação | Pesquisar_Certificações ]»
4. Arvore de Certificação
«TDmethod:POST»
«TDaction:http://192.168.1.27:8080/Skills/CertificacoesApp/ServletArvore.jsp»
«TDparameters:[nomeArvore@@certificacoes|funcao@@getNodosPai|modo@@showAll]»
«TDthinkTime:15»
«TDtransaction:[Arvore de Certificações | Pesquisar_Certificações ]»
5. Editar Certificação
«TDmethod:GET»
«TDaction:http://192.168.1.27:8080//Skills/CertificacoesApp/EditarCertifica»
cao.jsp?cert=Governance&type=new»»
«TDparameters:[Proficiency_level_field@@Expert|Acquired_Date_field@@2000|La»
st_time_used_field@@2008|postType@@add|edit_fields@@Proficiency_level_field,»
Acquired_Date_field, Last_time_used_field]»
«TDthinkTime:5»
«TDtransaction : [Editar Certificação | Editar_Certificacoes]»
6. Dados salvos certificações
«TDmethod:GET»
«TDaction:http://192.168.1.27:8080/SkillsApp/Product/CustomerRegistration.jsp»
«TDparameters:[Proficiency_level_field@@Expert|Acquired_Date_field@@2000|»
Last_time_used_field@@2008|postType@@add|edit_fields@@Proficiency_level_field,»
Acquired_Date_field, Last_time_used_field]»
«TDthinkTime:5»

```

```

«TDtransaction:[Dados salvos certificações | Adicionar_Certificações]»
7. Deslogar Certificações
«TDmethod:GET»
«TDaction:http://192.168.1.27:8080/SkillsApp/Product/BuyRequest.jsp»
«TDthinkTime:4»
«TDtransaction:[Deslogar Certificações | Adicionar_Certificações]»
8. Exit
«TDaction:exit»

```

- Nome do Cenário de Teste: Gerente RH
  - ## Configuração do Teste
  - Usuários Virtuais : «TDpopulation:500»
  - Host do SUT : «TDhost:localhost»
  - Tempo de Execução : «TDtime14400»
  - Tempo de Inicialização : «TDrampUpTime:600»
  - Usuários de Inicialização : «TDrampUpUser:50»
  - Tempo de Finalização : «TDrampDownTime:600»
  - Usuários de Finalização : «TDrampDownUser:50»
  - ## Distribuição dos Casos de Teste:
  - «TDprob:0.40» «TDpopulation:20»
  - 1. Gerenciar Habilidades
  - «TDprob:0.20» «TDpopulation:15»
  - 2. Gerenciar Experiências
  - «TDprob:0.30» «TDpopulation:10»
  - 3. Gerenciar Certificações
  - «TDprob:0.10» «TDpopulation:5»
  - 4. Alterar Senha
  - ## Contadores de Desempenho:
  - Transações por Segundo : «TDtps:Yes»
  - Tempo de Resposta : «TDresponse:Yes»
  - Requisições por Segundo : «TDrequest:Yes»
  - Vazão: «TDthroughput:Yes»
  - Utilização de Recursos : «TDresource:Yes»
  - ## SLAs:
  - Pesquisar\_Certificações: «TDexpTime: 1.117»
  - Editar\_Certificações: «TDexpTime: 1.317 »
  - Adicionar\_Certificações: «TDexpTime: 0.267 »

Após gerados os cenários e os casos de teste abstratos, são gerados pela PLEts os cenários e os *scripts* para a ferramenta LoadRunner executar os testes. A Figura 3.13, apresenta as informações utilizadas para a configuração do cenário de teste de desempenho gerado, referente ao caso de teste abstrato Gerenciar Certificações. Nela estão presentes as seguintes informações: estão presentes as seguintes informações: `Vusers=500`; `TotalVusersNumber=15`; `Count=50` e `Interval=600`.

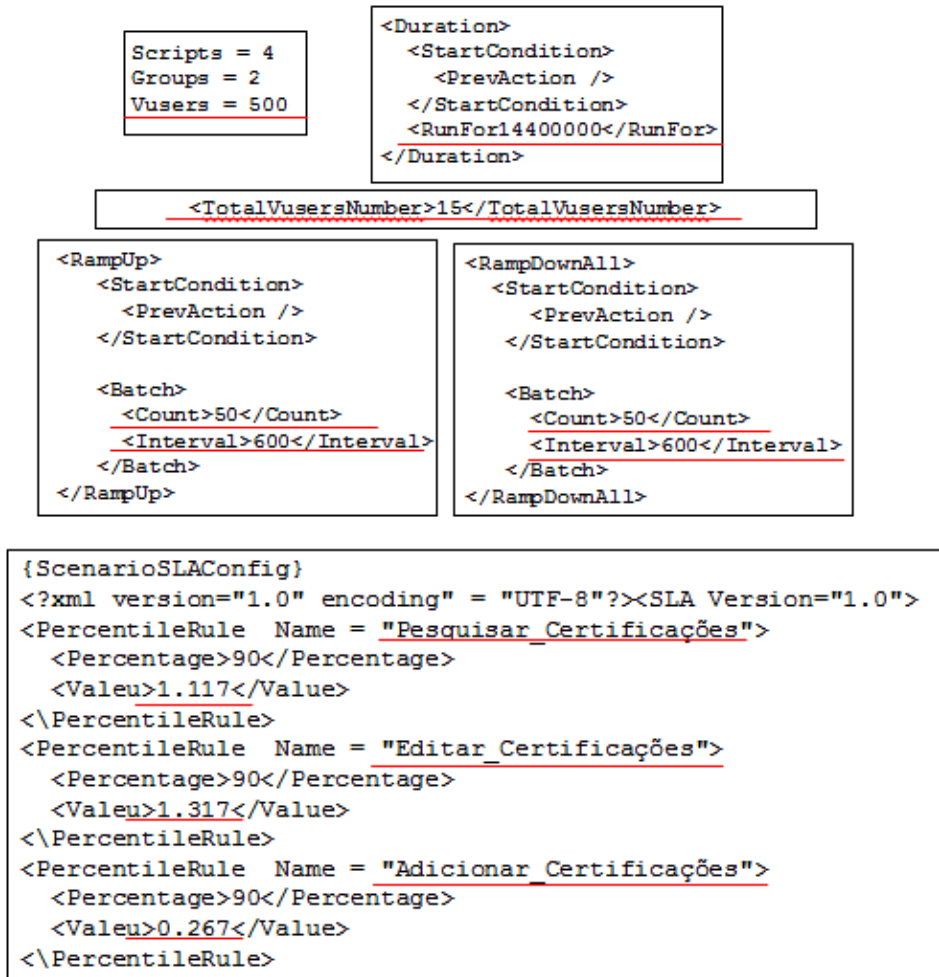


Figura 3.13: Cenário de Teste gerado para o LoadRunner - *Skills*

Já a Figura 3.14 apresenta um trecho do código XML, referente ao *script* de teste de desempenho gerado pela PLEts para o LoadRunner. Esse *script* de teste foi instanciado a partir do caso de teste abstrato *Shopping*. Conforme pode-se observar, ele é composto pelas diversas requisições (requests) presentes na transação *Editar Certificações*, efetuadas pelo usuário no Gerenciamento de Certificações.

```
lr_start_transaction("Editar_Certificacoes")
lr_start_sub_transaction("Editar Certificação")
web_url("Home Page ",
  "URL=http://192.168.1.27:8080//Skills/CertificacoesApp/
  EditarCertificacao.jsp?cert=Governance\&type=new",
```

```

"Method=GET",
"Resource=0",
"Referer=http://192.168.1.27:8080//Skills/Certificacoes
App/EditarCertificacao.jsp?cert=Governance\&type=new",
"Mode=HTTP",
ITEMDATA,
"Name=Proficiency\_level\_field", "Value=Expert",ENDITEM,
"Name=Acquired\_Date\_field", "Value=2000", ENDITEM,
"Name=Acquired\_Date\_field", "Value=2000", ENDITEM,
"Name=Last\_time\_used\_field", "Value=2008", ENDITEM,
"Name=postType","Proficiency_level_field,Acquired
_Date_field, Last_time_used_field",ENDITEM,LAST);
lr_end_sub_transaction("Editar Certificação",LR_AUTO)
lr_think_time(5);
lr_end_transaction("Editar_Certificacoes",LR_AUTO);

```

Figura 3.14: *Script* de teste gerado para o LoadRunner - *Skills*

Após a execução dos *scripts* de teste de desempenho, o gráfico representado pela Figura ??, apresenta o tempo de resposta da transação *Editar\_Certificação* presente no diagrama de atividades Gerenciar Certificações (3.12). Por meio da execução dos testes é possível observar que os tempos médios de resposta foram equivalentes aos tempos definidos pelos SLAs, exceto na primeira rodada, onde o tempo definido no SLA é superior quando comparado ao tempo médio de resposta. No entanto, quando comparado ao tempo máximo de resposta obtido com a execução dos testes seria necessário rever os SLAs, uma vez que o tempo de resposta foi significativamente maior aos tempos definidos nos SLAs.

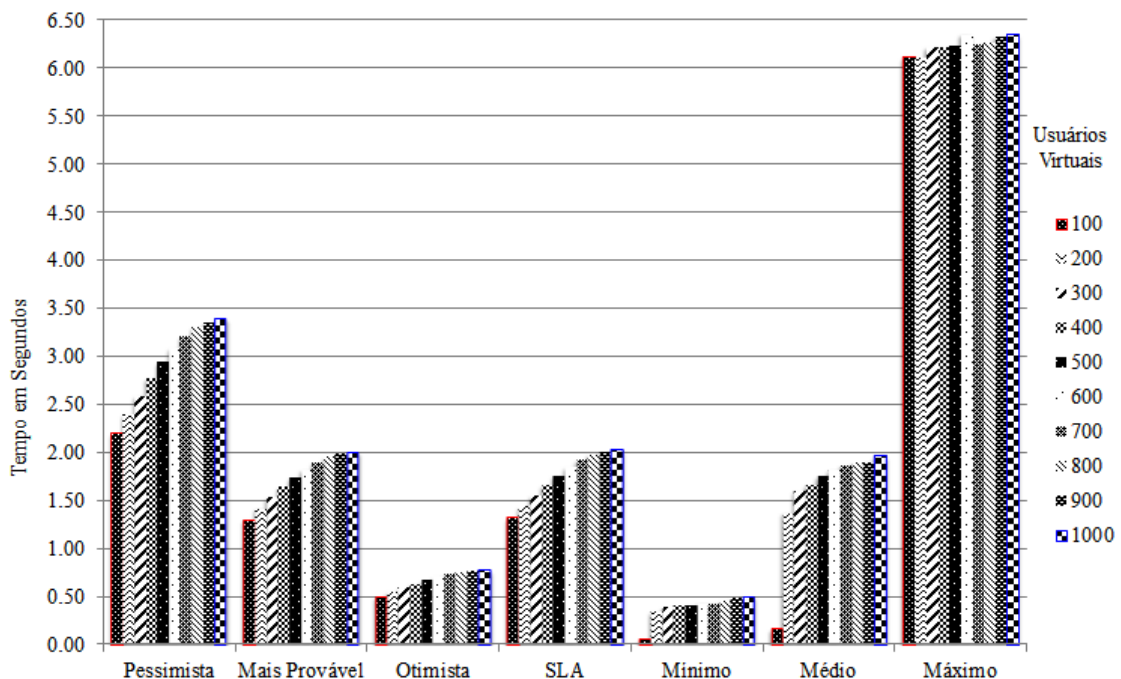


Figura 3.15: Gráfico de análise do tempo de resposta da transação *Editar\_Certificações* do diagrama de atividades da Figura 3.12

É importante ressaltar que as outras transações que fazem parte do diagrama de atividades Gerenciar Certificações (*Pesquisar\_Certificações* e *Adicionar\_Certificações*) tiveram um tempo de resposta similar.

### 3.5 Trabalhos Relacionados

Atualmente, há um interesse crescente relacionado à área de teste de desempenho. Neste contexto, diversos autores têm apresentado abordagens e metodologias para testar o desempenho das aplicações utilizando técnicas de teste aplicadas no desenvolvimento de *software*. Em particular, teste de desempenho com modelos UML é um dos tópicos mais abordados pelos pesquisadores. Diversos trabalhos apresentam abordagens e metodologias que derivam cenários e *scripts* de teste de desempenho a partir de modelos UML adaptados para representar informações de desempenho, e.g., UCs, ADs, etc. Alguns dos principais trabalhos nesta área são apresentados em [58], [59], [60] e [61].

Na abordagem proposta em [58], é apresentada uma ferramenta chamada PReWebN (*Prototype Research Web application in Netbeans platform*) com o objetivo de avaliar o desempenho de aplicações com base em tecnologia Java. A ferramenta realiza uma análise estatística a partir dos dados coletados pela ferramenta de geração de carga de trabalho para diferentes métricas de desempenho, entre elas o tempo de resposta (*response time*). No entanto, a abordagem proposta pela autora não se baseia na abordagem MBT. Portanto, tal abordagem não se beneficia das vantagens da adoção de MBT, como a validação de requisitos de desempenho por meio de modelos de teste, geração sistemática e automação dos SLAs.

Em [59] um *framework* híbrido para um modelo de processo de predição de desempenho é apresentado, a fim de avaliar o desempenho de aplicações já na fase de estudo de viabilidade do projeto de *software*. Os autores propõem uma abordagem de engenharia de desempenho com base em diagramas de casos de uso UML, oferecendo flexibilidade para integrar o processo de previsão de desempenho com processos de engenharia de *software*. Apesar do estudo ser pautado na abordagem MBT, o *framework* proposto não leva em consideração a geração automática e análise dos SLAs.

Um *framework* usando uma heurística sensível ao tempo de resposta é apresentado em [60]. Este *framework* serve como base para suportar a alocação dinâmica de servidores em ambiente de computação em nuvem, dada a carga de processamento exigido pela aplicação. Apesar do estudo levar em consideração a automação dos SLAs, os mesmos não são previamente definidos com base em modelos (MBT) e integrados a ferramentas de desempenho, tal como a abordagem proposta neste artigo, mas sim avaliados por meio da integração de ferramentas de gerenciamento da escalabilidade em ambientes de computação em nuvem. Por fim, em [61] uma abordagem para avaliação sistemática de desempenho com base em aplicações de referência (*benchmark*) adaptadas é descrita. Assim como os demais trabalhos apresentados nesta seção, os autores também não contemplam em sua abordagem a automação dos SLAs como proposto pelo presente trabalho.

Embora estes trabalhos apresentem alternativas que contribuem para minimizar o esforço na

atividade de teste de desempenho, a maioria foca somente em avaliar as métricas de desempenho. Em geral, a maioria destas pesquisas deixam muitas lacunas quando se referem a automação dos SLAs de desempenho. Por outro lado, a abordagem proposta neste trabalho também busca minimizar este esforço por meio da abordagem MBT para geração e avaliação sistemática do tempo de resposta usando SLAs.

### 3.6 Resumo

Este capítulo apresentou a abordagem proposta referente ao estudo realizado, demonstrando o processo de construção dos modelos UML para teste de desempenho, incluindo informações que são necessárias para medir de maneira automatizada o tempo de resposta das sequências de teste. Outro aspecto importante foi o processo de automação de SLAs, tendo como base as transações definidas ainda no processo de construção dos modelos. Finalizando, foram apresentados dois exemplos de uso referente as aplicações TPC-W e *Skills* com o intuito de exemplificar as premissas e notações necessárias para a criação e execução dos *scripts* de teste de desempenho, que vierem também a contribuir no planejamento e realização do estudo de caso que será apresentado no capítulo seguinte. Além de apresentar alguns trabalhos relacionados com a abordagem proposta.



## 4. ESTUDO DE CASO

"Example is not the main thing in influencing others. It is the only thing."

Albert Schweizer

Com o intuito de avaliar a abordagem proposta para teste de desempenho na perspectiva de inclusão de tempo nos modelos UML, este capítulo detalha a implementação de um estudo de caso, apresentando o processo de modelagem, a geração dos cenários, casos e *scripts* de desempenho e sua instrumentalização para a ferramenta LoadRunner, juntamente com a definição dos SLAs. Além de contribuir com uma linha de produto de *software* de ferramentas MBT. Ao final, pretende-se realizar uma análise, apresentando os resultados obtidos com a execução dos testes. Para isto, será utilizado uma aplicação denominada ChangePoint [62], a qual tem por objetivo a gerência de portfólio de produtos e projetos multidisciplinares. Partes deste capítulo que descreve o estudo de caso foram publicados no artigo "Abordagem de análise do tempo de resposta para teste de desempenho em aplicações web", apresentado no XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) - WTF, 2014<sup>3</sup>

### 4.1 Configuração do Ambiente

Conforme representado pela Figura 4.1, o cenário de testes de desempenho da aplicação ChangePoint é composto por quatro servidores da aplicação (*App Servers*), dois servidores *web* (*Web Servers*), e três servidores de banco de dados (*Database Servers*). É importante ressaltar que neste ambiente foram executados os testes de desempenho que serão descritos nas seções seguintes. Por meio da figura apresentada, também é possível verificar os aspectos de configuração das máquinas, tais como: CPUs, memória e sistema operacional.

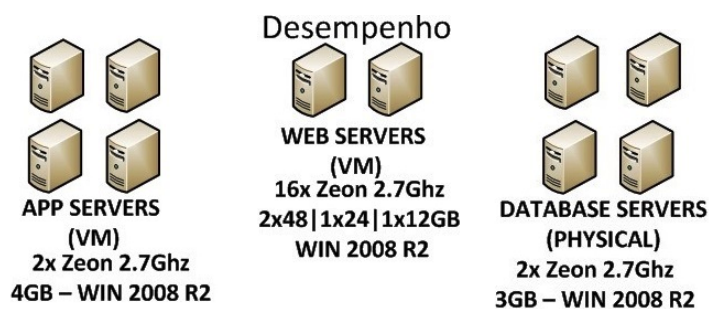


Figura 4.1: Ambiente de teste de desempenho aplicação ChangePoint

<sup>3</sup>O SBRC (Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos) foi realizado de 05 a 09 de maio de 2014 na cidade de Florianópolis - Santa Catarina, Brasil.

## 4.2 Modelagem UML para Teste de Desempenho

O processo de modelagem tem início com a criação do cenário de teste representado pelo diagrama de casos de uso. Neste diagrama é demonstrada a iteração do ator com os casos de uso que ele está associado, tendo como base as principais funcionalidades da aplicação e que são utilizadas no planejamento, criação e controle dos projetos. O diagrama de casos de uso da Figura 4.2 representa estas funcionalidades divididas em seis casos de uso, sendo eles: Associar/Desassociar Ordem aos Projetos, Revisar Documentos, Aprovar Recursos Humanos, Informar Horas Trabalhadas, Aprovar Custos do Projeto, Aprovar Projeto. Os estereótipos PApopulation e PAtime foram atribuídos ao ator definido no modelo, sendo que eles representam as informações do cenário para a execução dos testes.

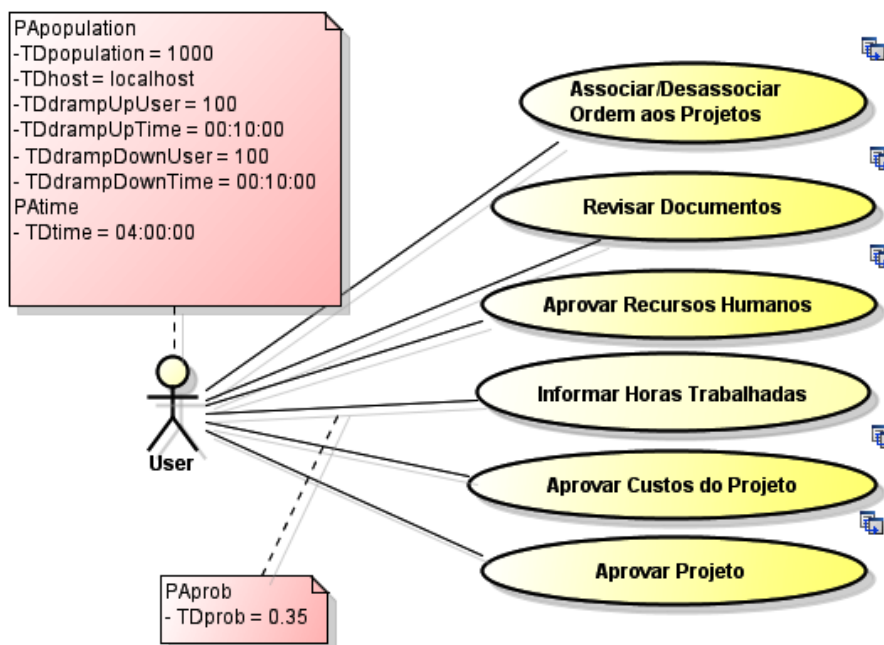


Figura 4.2: Diagrama de Casos de Uso da aplicação ChangePoint

Para a execução dos testes foi modelado um cenário de carga de trabalho composto por 1000 usuários virtuais, sendo que durante a inicialização do teste são acrescentados 100 usuários (TDdrampUpUser) a cada 10 minutos (TDdrampUpTime), totalizando 1h40min para a inicialização total do teste, assim como a finalização do teste (TDdrampDownUser e TDdrampDownTime) também foram configurados os mesmos parâmetros. Totalizando 3h20min entre a inicialização e finalização do teste, resultando em 40min de execução do teste com os 1000 usuários virtuais concorrentes para uma duração total de execução do teste de 4h (TDtime). Outra característica importante neste modelo é o parâmetro TDprob, definido em cada associação entre o ator e os casos de uso. Esta informação define a probabilidade de distribuição dos usuários virtuais em cada caso de uso relacionado a ele.

A Tabela 4.1 apresenta a distribuição dos usuários virtuais em relação aos casos de uso. Conforme demonstrado na Figura 4.2, a probabilidade do ator *User* executar o caso de uso Informar Horas

Trabalhadas é igual a 35% (TDprob), o qual representa 350 usuários virtuais dos 1000 concorrentes. É importante considerar que esta probabilidade pode também ser propagada aos casos de teste gerados para os diagramas de atividades em situações de paralelismo, onde os usuários virtuais podem executar atividades diferentes ao mesmo tempo. Por limitação de espaço e tamanho da imagem não foram comentadas todas as associações entre o ator e os casos de uso apresentados, no entanto estes valores foram informados no diagrama por meio do rótulo TDprob.

Tabela 4.1: Distribuição da carga no cenário de teste de desempenho da aplicação

Ator	Caso de Uso	Probabilidade	Usuários Virtuais
User	Associar/Desassociar Ordem aos Projetos	20%	200
	Informar Horas Trabalhadas	35%	350
	Revisar Documentos	20%	200
	Aprovar Recursos Humanos	10%	100
	Aprovar Custos do Projeto	10%	100
	Aprovar Projeto	5%	50
			<b>1000</b>

Após definido o cenário de teste, a próxima etapa de elaboração do modelo de teste é decompor cada caso de uso em um diagrama de atividades, demonstrando o fluxo das atividades executadas pelo usuário para realizar determinada tarefa no sistema. O diagrama de atividades (*Activity Diagram - AD*) ilustra a sequência de atividades ou etapas que determinam um processo (*workflow*) do sistema. Um AD demonstra o Fluxo de Controle de um componente do sistema, no qual os estados são as atividades que representam a execução de operações e as transições são executadas pelo resultado dessas operações [63].

Sendo assim, foram construídos diagramas de atividades correspondentes aos casos de uso apresentados na Figura 4.2, representando as itereções do usuário com o sistema. Para representar este processo será exemplificado o caso de uso Informar Horas Trabalhadas. É importante destacar que todos os demais casos de uso presentes no diagrama também foram expandidos, no entanto, por limitação de espaço será apresentado apenas um deles. Os demais diagramas referente a aplicação ChangePoint podem ser consultados no Apêndice C.

Na modelagem dos diagramas de atividades, inicialmente é criado um modelo alto nível, identificando todas as atividades que serão necessárias para a criação e execução dos casos e *scripts* de desempenho. Isso permite ter uma visão geral do processo, além de reusar os diagramas em diferentes casos de uso. Conforme observa-se na Figura 4.3 o diagrama de atividades é composto por três atividades principais, sendo que a primeira e a terceira são referentes as atividades de *login* e *logout*, e a do meio é referente a atividade Informar Horas Trabalhadas.

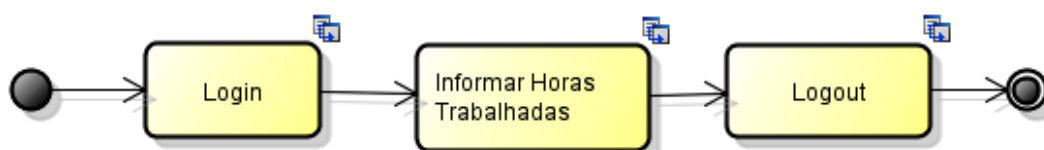


Figura 4.3: Diagrama de atividades do caso de uso Informar Horas Trabalhadas

É importante considerar que neste diagrama não existem estereótipos e informações nas atividades e nas transições, ele serve apenas para demonstrar a sequência de execução dos demais diagramas que compõem o cenário de teste e auxiliar no gerenciamento e reuso dos modelos.

Em seguida, por meio de um *hyperlink*, cada uma das três atividades (*Login*, *Informar Horas Trabalhadas* e *Logout*), são expandidas em novos diagramas que possuirão as informações de teste, conforme representados respectivamente pelas Figuras 4.4, 4.5 e 4.6.

O diagrama de atividades representado pela Figura 4.4, representa as iterações do usuário no momento em que o mesmo irá logar no sistema. Este processo é composto por duas transações principais: *Carregar Página de Login* e *Submeter Dados*. A primeira transação é composta pelas atividades: *Página Inicial*, *Carregar Página de Login* e *Informar Dados*, que equivalem respectivamente as ações de acesso a página principal do sistema, carregar página de *Login* e informar os dados do usuário (usuário e senha). Já a segunda transação controla as atividades de validação e submissão dos dados dos usuários ao sistema.

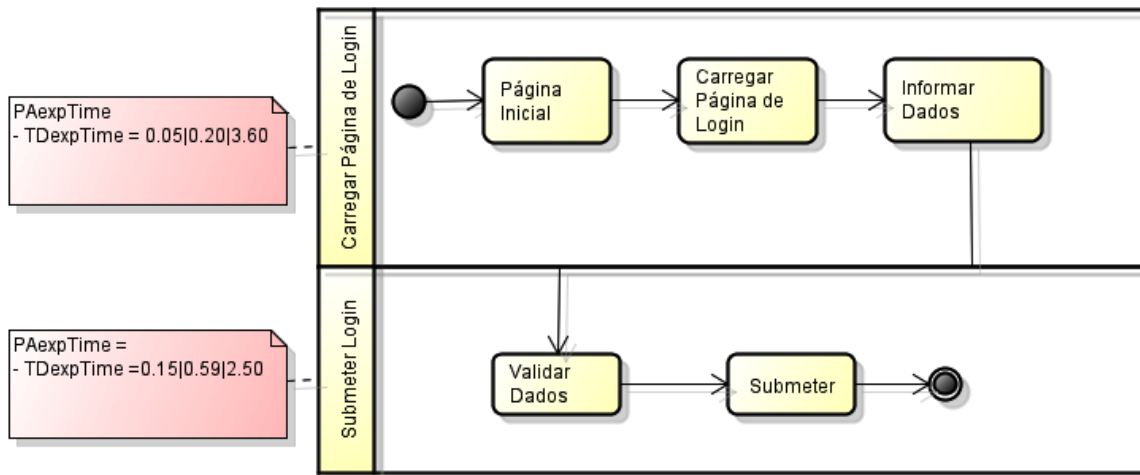


Figura 4.4: Subdiagrama de atividades da atividade *Login*

As informações de teste são anotadas nas transições existentes entre as atividades que fazem parte do diagrama por meio dos estereótipos apresentados na Seção 3.1, conforme demonstrado no exemplo de uso (3.4). Já as estimativas de tempo de resposta diante das três perspectivas (otimista, mais provável e pessimista) são anotadas nas transações existentes no diagrama. A Figura 4.4, mostra como são anotadas as estimativas de tempo de resposta nas transações *Carregar Página de Login* e *Submeter Dados*, por meio do estereótipo *TDexpTime*.

O diagrama mostrado na Figura 4.5 representa a iteração do usuário no sistema no momento em que o mesmo irá informar o tempo consumido em cada tarefa relacionada ao projeto. Esta ação é composta por três transações: *Acessar Folha Ponto*, *Folha Ponto Projeto* e *Folha Ponto Tarefa*.

Na transação *Acessar Folha Ponto* estão presentes as ações de acesso a página principal do sistema, selecionando em seguida as opções *Aplicações* e *Folha Ponto*. Na transação *Folha Ponto Projeto* são executadas as ações *Pesquisar* e *Selecionar Projeto*. Finalizando, na transação *Folha Ponto Tarefa* é incluída a tarefa no projeto, e definido as horas destinadas para sua execução,

submetendo em seguida as informações. Por meio do estereótipo *TDexptime* anotado em cada transação é possível verificar os tempos estimados para execução de cada sequência de teste. Por exemplo, na transação *Acessar Folha Ponto* foram estimados os seguintes tempos de execução: mínimo: 0.50 médio: 2.50 máximo: 3.89.

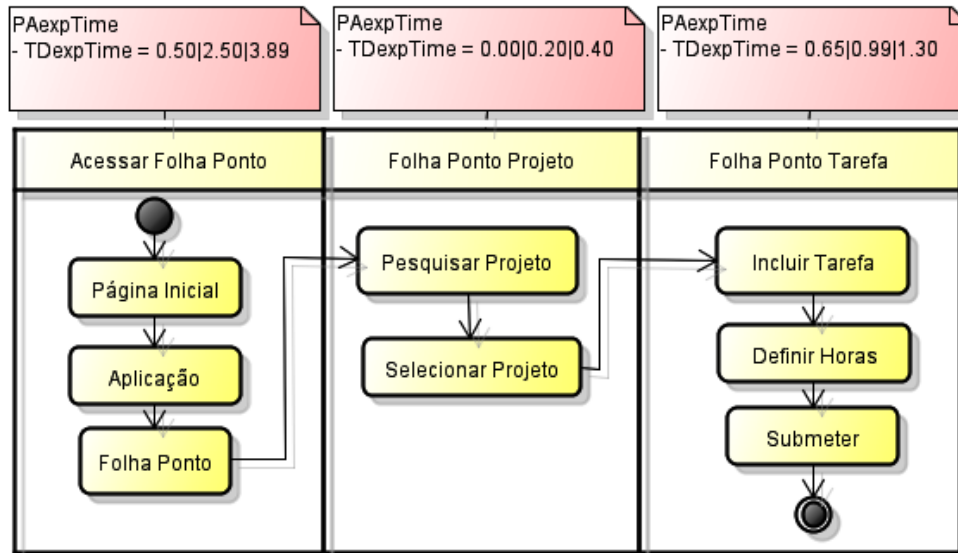


Figura 4.5: Subdiagrama de atividades da atividade Informar Horas Trabalhadas

Já o diagrama representado pela Figura 4.6, representa a ação de *Logout*, executada pelo usuário ao sair do sistema. Esta ação é representada pela transação *Logout*, cujos tempos mínimo, médio e máximo de execução estimados são respectivamente: 2.01, 2.78, 3.12.

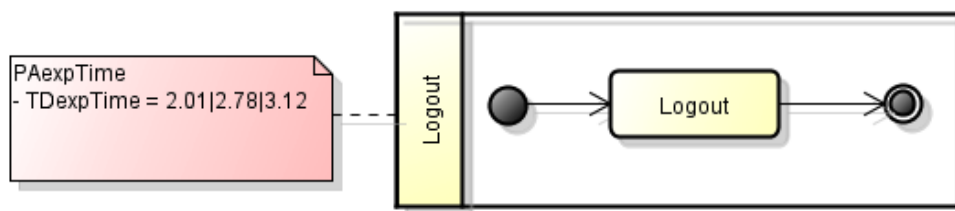


Figura 4.6: Subdiagrama de Atividades da atividade Logout

Por meio das transações definidas nos diagramas de atividades apresentados e das estimativas de tempo de resposta informadas é que serão definidos os SLAs para cada transação. Posteriormente estas transações serão interpretadas pela ferramenta LoadRunner na execução dos cenários e *scripts* de teste de desempenho, conforme descrito na Seção 4.3.

### 4.3 Geração dos Cenários e *Scripts* de Teste de Desempenho

Conforme demonstrado nos exemplos de uso apresentados na Seção 3.4, após criar os modelos UML com as informações de tempo, os mesmos são convertidos em MEFs e por meio da aplicação do método de geração de sequência de teste HSI são gerados os casos e os cenários de teste abstratos.

A estrutura apresentada abaixo representa o caso de teste abstrato da funcionalidade Informar Horas Trabalhadas.

- #Caso de Teste: Informar Horas Trabalhadas
  1. Pagina Inicial
    - «TDmethod:POST»
    - «TDaction:http://192.168.1.27:8080/ChangePointApp/Homepage.jsp»
    - «TDthinkTime : 5»
    - «TDtransaction:[Pagina Inicial | Acessar Folha Ponto]»
  2. Aplicação
    - «TDmethod:POST»
    - «TDaction:http://192.168.1.27:8080/ChangePointApp/Application.jsp»
    - «TDthinkTime:5»
    - «TDtransaction:[Aplicação | Acessar Folha Ponto]»
  3. Folha Ponto
    - «TDmethod:POST»
    - «TDaction:http://192.168.1.27:8080/ChangePointApp/Application/TimeSheet.jsp»
    - «TDthinkTime:5»
    - «TDtransaction:[Folha Ponto | Acessar Folha Ponto]»
  4. Pesquisar Projeto
    - «TDmethod:POST»
    - «TDaction:http://192.168.1.27:8080/ChangePointApp/Application/SearchProject.jsp»
    - «TDparameters:[nomeProject@@Project|TypeProject@@Type]»
    - «TDthinkTime:15»
    - «TDtransaction:[Pesquisar Projeto | Folha Ponto Projeto]»
  5. Selecionar Projeto
    - «TDmethod:GET»
    - «TDaction:http://192.168.1.27:8080/ChangePointApp/Application/selectproject.jsp»
    - «TDparameters:[nomeProject@@Project|funcao@@getProject]»
    - «TDthinkTime:5»
    - «TDtransaction : [Selecionar Projeto | Folha Ponto Projeto]»
  6. Incluir Tarefa
    - «TDmethod:GET»
    - «TDaction:http://192.168.1.27:8080/ChangePointApp/Application/Task.jsp»
    - «TDparameters:[Project@@Project1|Task@@Task1|BusinessUnit@@3BSDR]»
    - «TDthinkTime:5»
    - «TDtransaction:[Incluir Tarefa | Folha Ponto Tarefa]»
  7. Definir Horas
    - «TDmethod:GET»
    - «TDaction:http://192.168.1.27:8080/ChangePointApp/Project/sethours.jsp»

```

«TDparameters:[hidDescription@@|hidCurDate@@08/3/2012|hidRegularHours@@8|hid
OverTimeaHours@@0.00|hidTaskId@@taskid_taskid|hidTimeZone@@|hidTimeType@@t|
hidAdjustedRegular@@hidAdjusteOverTime@@|hidStartHour@@hidStartMinute@@]»
«TDthinkTime:4»
«TDtransaction:[Definir Horas | Folha Ponto Tarefa]»
8. Submeter
«TDmethod:GET»
«TDaction:http://192.168.1.27:8080/ChangePointApp/Project/submitt.jsp»
«TDthinkTime:4»
«TDtransaction:[Submeter | Folha Ponto Tarefa]»
9. Exit
«TDaction:exit»

```

Nesta estrutura estão presentes as atividades e transações, além das informações que foram incluídas nos modelos UML. Por exemplo, na atividade Pesquisar Projeto foram informados dados referente ao nome e ao tipo do projeto (nomeProject@@Project|TypeProject@@Type) para o endereço `http://192.168.1.27:8080/` usando o método POST. O thinkTime definido foi de cinco segundos para o preenchimento e submissão dos dados e esta atividades faz parte da transação Folha Ponto Projeto.

Já no cenário de teste estão presentes as informações referente ao cenário de teste, tais como: usuários virtuais, host do SUT, tempo de execução, tempo de inicialização, tempo de execução, tempo de finalização e distribuição dos casos de teste em relação ao número de usuários, conforme informadas no diagrama de casos de uso.

Além disso, no cenário de teste também é possível observar os SLAs referente a cada transação, e.g, o SLA da transação Acessar Folha Ponto é igual a 2.398. Por limitação de espaço, na estrutura apresentada no cenário de teste *User* estão presentes apenas os SLAs referentes ao caso de uso Informar Horas Trabalhadas, no entanto fazem parte deste cenário as demais transações presentes nos diagramas de atividades da aplicação ChangePoint.

- Nome do Cenário de Teste : User
- ## Configuração do Teste
- Usuários Virtuais : «TDpopulation:1000»
- Host do SUT : «TDhost:localhost»
- Tempo de Execução : «TDtime:14400»
- Tempo de Inicialização : «TDrampUpTime:100»
- Usuários de Inicialização : «TDrampUpUser:10»
- Tempo de Finalização : «TDrampDownTime:100»
- Usuários de Finalização : «TDrampDownUser:10»
- ## Distribuição dos Casos de Teste:

```

«TDprob:0.20» «TDpopulation:200»
1. Associar/Desassociar Horas Trabalhadas
«TDprob:0.35» «TDpopulation:350»
2. Revisar Documentos
«TDprob:0.20» «TDpopulation:200»
3. Aprovar Recursos Humanos
«TDprob:0.10» «TDpopulation:100»
4. Informar Horas Trabalhadas
«TDprob:0.10» «TDpopulation:100»
5. Aprovar Custos do Projeto
«TDprob:0.05» «TDpopulation:50»
6. Aprovar Projeto
## Contadores de Desempenho:
Transações por Segundo : «TDtps:Yes»
Tempo de Resposta : «TDresponse:Yes»
Requisições por Segundo : «TDrequest:Yes»
Vazão: «TDthroughput:Yes»
Utilização de Recursos : «TDresource:Yes»
## SLAs:
Carregar Página Login: «TDexpTime: 0.742»
Submeter Dados: «TDexpTime: 0.835 »
Acessar Folha Ponto : «TDexpTime:2.398»
Folha Ponto Projeto : «TDexpTime:0.200»
Folha Ponto Tarefa : «TDexpTime:0.985»
Logout: «TDexpTime: 2.708»

```

Com base nos cenários e casos de teste abstratos apresentados, são gerados pela PLeTs os casos de teste concretos ou executáveis. Na Figura 4.7, são apresentadas algumas das informações utilizadas para a configuração do cenário de teste de desempenho da aplicação ChangePoint que foram informados por meio dos estereótipos, tais como: número total de usuários configurados para executar o teste  $Vusers = 1000$ , tempo total (em milissegundos) de execução do teste  $RunFor = 14400000$ , quantidade de usuários que irá executar as atividades descritas em determinado *script*  $TotalVusersNumber = 350$ , número de usuários que iniciarão o teste (Quantidade de Usuários da Rampa de Subida) ou que irão deixar de realizar suas requisições (Quantidade de Usuários da Rampa de Descida)  $Count = 10$  e tempo que levará para um conjunto de usuários iniciar o teste (Tempo de Rampa de Subida) ou terminá-lo (Tempo de Rampa de Descida)  $Interval = 100$ .

Outra informação presente no cenário de teste são os SLAs de cada transação. Por limitações de espaço foram apresentados somente os SLAs das transações que representam a iteração do usuário no sistema no momento em que o mesmo irá informar o tempo consumido em determinada tarefa



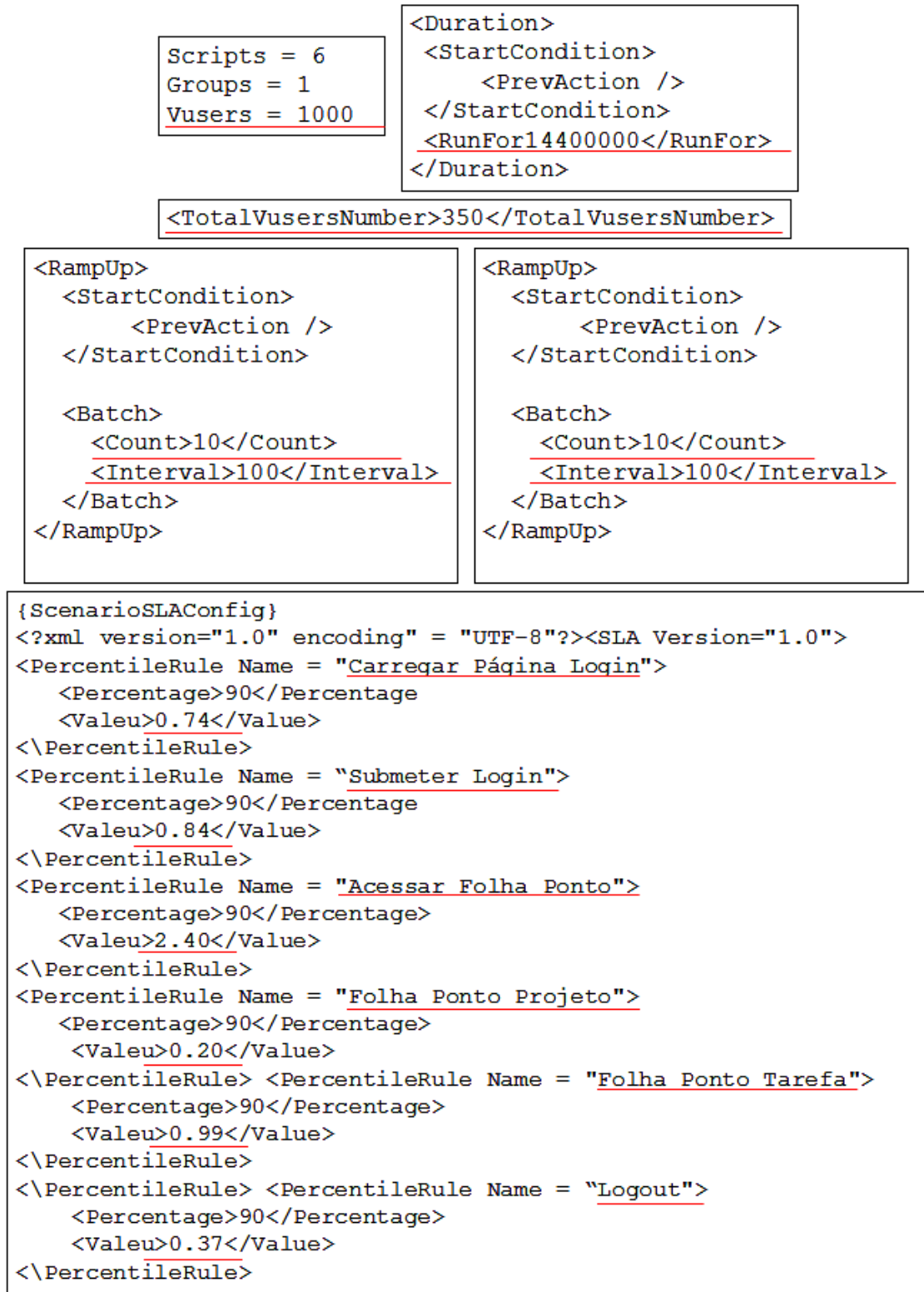


Figura 4.7: Cenário de teste gerado para o LoadRunner - ChangePoint

relacionada ao projeto (Informar Horas Trabalhadas). Esta ação é composta por três atividades: Acessar Folha Ponto, Folha Ponto Projeto e Folha Ponto Tarefa. Incluindo também as transações de *Login*: Carregar Página de *Login* e Submeter *Login*, e a transação de *Logout*.

Já a Figura 4.8, apresenta um trecho do código XML, referente ao *script* de teste de desempenho gerado pela PLeTs para o LoadRunner. Esse *script* de teste foi instanciado a partir do caso de teste abstrato *Informar Horas Trabalhadas*.

```
lr_start_transaction("Folha Ponto Tarefa")
  lr_start_sub_transaction("Incluir Tarefas")
    web_url("Incluir Tarefas",
      "http://192.168.1.27:8080/ChangePointApp/
      Application/Task.jsp",
      "Method=GET",
      "Resource=0",
      "Referer=http://192.168.1.27:8080/ChangePointApp/
      Application/Task.jsp",
      "Mode=HTTP",
      ITEMDATA,
      "Name=Project ", "Value= Project1",ENDITEM,
      "Name=Task", "Value=Task1",ENDITEM,
      "Name=BusinessUnit", "Value=3BSDR",
      ENDITEM,
      LAST);
  lr_end_sub_transaction("Incluir Tarefas",LR_AUTO)
  lr_think_time(5);
lr_end_transaction("Folha Ponto Tarefa",LR_AUTO);
```

Figura 4.8: *Script* de teste gerado para o LoadRunner - ChangePoint

Conforme pode-se observar, ele é composto pelas diversas requisições (requests) presentes na transação Incluir Tarefas, efetuadas pelo usuário no momento de informar uma tarefa em determinado projeto.

#### 4.4 Análise e Interpretação dos Resultados

Tendo apresentado as etapas do processo de modelagem dos diagramas em que são definidas as transações e informadas as estimativas de tempo utilizadas na criação dos SLAs e na execução dos casos, cenários e *scripts* de teste de desempenho, conforme descrito na Seção 4.3. Esta seção apresenta a análise e avaliação dos resultados obtidos após a execução dos casos de teste, tendo como base os dados gerados pela ferramenta LoadRunner.

De acordo com o cenário de desempenho apresentado foram avaliados dez diferentes cargas de trabalho, em que em cada uma delas foram acrescentados 100 usuários. Após coletar e analisar os resultados do teste de desempenho, foi possível realizar algumas considerações referente ao teste realizado.

Conforme é possível observar no gráfico apresentado na Figura 4.9 em que destaca a evolução do tempo de resposta em relação a cada uma das cargas de trabalho submetidas, essa transação compreende o conjunto de atividades apresentados na Figura 4.5 que faz referência ao caso de uso

Informar Horas Trabalhadas. A figura apresenta os tempos de resposta para cada carga de trabalho, os tempos estimados de acordo com a técnica PERT (otimista, mais provável e pessimista), bem como o SLA definido pelo tempo esperado, e ainda os tempos executados do teste de desempenho, destacando o tempo Mínimo, Médio e Máximo coletados pelo LoadRunner.

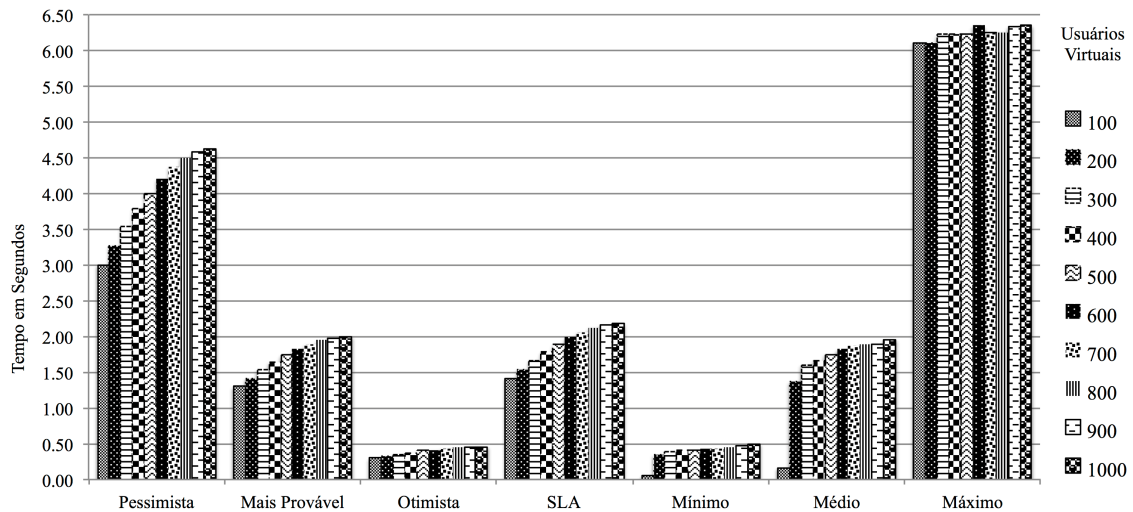


Figura 4.9: Gráfico de análise do tempo de resposta da transação Informar Horas Trabalhadas do diagrama de atividades da Figura 4.5

Analisando o gráfico da Figura 4.9 pode-se evidenciar que a carga de 100 usuários obteve os tempos médios e mínimos muito inferiores às demais cargas, inclusive com um desvio superior ao SLA estimado. Entretanto, as demais cargas obtiveram tempos médios próximos aos SLAs definidos. Vale destacar que a carga de 300 usuários virtuais em que o tempo do SLA estimado foi de 1.67s contra o tempo médio executado pelo teste de desempenho de 1.60s, essa foi a menor diferença entre as estimativas de tempo (SLA) e o tempo executado, variando apenas 4.2%.

Com a realização do teste de desempenho, pode-se observar também que dependendo do aumento dos usuários virtuais que acessam o sistema, o SLA deverá ser revisto. Pois, à medida que mais usuários acessavam a aplicação, o tempo de resposta das sequências de teste tendem a aumentar e as transações que inicialmente cumpriam com o valor estabelecido no SLA passam a não mais cumprir. Outro fator que também deve ser considerado quando o SLA é definido é o ambiente em que a aplicação será executada, além dos históricos de execuções que permitam verificar o tempo de resposta das transações, permitindo assim, definir estimativas mais concretas e próximas ao ambiente real de execução dos testes de desempenho. Isso pode ser observado na Figura 4.9 apresentada, uma vez que a medida que novas execuções eram realizadas o valor estimado para o SLA ficava mais próximo do tempo médio, o que não acontece com a carga de 100 usuários, em que os tempos são estimados sem considerar execuções anteriores.

Já no que diz respeito ao tempo de resposta das transações, gerados pelo LoadRunner, de uma forma geral, é importante considerar que se a aplicação sendo testada tem uma taxa fixa de atendimentos por unidade de tempo, as transações deveriam ter um tempo de resposta significativamente semelhante. No entanto, uma das limitações identificadas é o não conhecimento da forma que o

LoadRunner calcula os tempos de resposta, uma vez que os dados analisados são disponibilizados de forma sumarizada pela ferramenta. Esse conhecimento se torna importante para considerar as diversas variáveis extrínsecas que podem influenciar no tempo de resposta das transações.

#### 4.5 Resumo

Esta seção teve como objetivo demonstrar a abordagem proposta, por meio da aplicação ChangePoint, onde foi possível representar toda a sequência de passos realizada, desde a etapa de modelagem UML até execução dos *scripts* de desempenho. Outro aspecto importante é que por meio do respectivo trabalho foi possível verificar que a abordagem MBT pode ser utilizada para diferentes tipos de teste, inclusive para desempenho. E isso não está relacionado somente a avaliação de métricas, mas também com o intuito de mitigar o esforço na realização de testes gerando e avaliando de forma sistemática o tempo de resposta das transações por meio de SLAs.

## 5. CONCLUSÃO

*"It's more fun to arrive a conclusion than to justify it."*

Malcolm Stevenson Forbes

Esta seção apresenta um resumo da dissertação bem como as considerações finais relacionadas à pesquisa. Destaca ainda, as lições aprendidas, as limitações do estudo, assim como, sua contribuição e perspectivas para trabalhos futuros a partir dos resultados obtidos.

### 5.1 Resumo

Este trabalho apresentou uma abordagem de "Análise do tempo de resposta para teste de desempenho em aplicações *Web*". Para isso destacam-se duas etapas como sendo as principais para efetivação do trabalho. Inicialmente o processo de modelagem descrito na Seção 3.1 apresenta as informações necessárias para a realização de teste de desempenho, incluindo critérios de tempo que permitem medir de maneira automatizada o tempo de resposta das sequências de teste.

Estas informações de tempo relacionadas aos diagramas de atividades servem como entrada para as ferramentas de teste baseados em modelos, que são geradas a partir da linha de produto - PLeTs. Por meio desse modelo as ferramentas MBT geram casos e *scripts* de teste de desempenho para um gerador de carga, e.g. LoadRunner. As aplicações *Skills* e TPC-W foram usadas como exemplo de uso, a fim de demonstrar as premissas e considerações necessárias para o processo de Modelagem.

No exemplo de uso apresentado na Seção 3.4, foi utilizada a ferramenta LoadRunner, onde com base nos tempos informados nos diagramas UML e por meio da técnica PERT foi definido o SLA, caracterizando assim a segunda etapa.

Para complementação do trabalho e demonstração da abordagem proposta foi desenvolvido um estudo de caso fazendo uso da aplicação ChangePoint, onde foram criados casos e *scripts* de teste para a ferramenta de teste de desempenho LoadRunner. O objetivo foi medir o tempo de resposta de cada transação em relação aos tempos definidos no SLA. Ao final foi realizada uma análise referente aos tempos que haviam sido estimados e os tempos obtidos com a realização dos testes conforme descrito na Seção 4.4.

É importante ressaltar que em cada uma das etapas existentes na PLeTs foram incluídos critérios com o intuito de medir o tempo de resposta das sequências de teste, que antes da realização deste estudo eram inexistentes. Nos modelos foi incluído um novo estereótipo TDexpTime que em seguida foi interpretado pelo Parser. Assim como no processo de geração dos cenários e casos de teste, o fator tempo também foi considerado, conforme pode ser observado nos cenários e casos de teste de desempenho gerados nos exemplos de uso como também no estudo de caso, onde estão presentes os SLAs referente a cada transação. Finalizando, com base nos cenários e casos de teste foram

gerados pela PLeTs os *scripts* de desempenho, automatizando a criação dos SLAs na ferramenta LoadRunner.

## 5.2 Lições Aprendidas

Com o advento de novas tecnologias, torna-se eminente a expansão de soluções e aplicações *web* que possuem uma grande quantidade de usuários e desta forma torna-se necessário a validação do desempenho destes sistemas. A pesquisa que originou o trabalho apresentado nesta dissertação teve início no estudo dos conceitos relacionados ao teste de *software*. Após uma visão geral, a pesquisa focou no teste de desempenho, cujo objetivo principal foi mensurar o tempo de resposta das sequências de teste.

No entanto, durante a realização deste trabalho foi pesquisado diversas questões relacionadas a modelagem de testes de desempenho. Inicialmente foi proposto a aplicação de um novo método de geração de sequências de teste, que tivesse as características necessárias para a realização de testes de desempenho. No entanto, após avaliar a ferramenta LoadRunner, identificou-se que a mesma possuía recursos que permitiam execução de *scripts* de teste com a definição de SLAs. Diante disso, foi investigado uma forma de gerar casos e *scripts* de teste com informações de tempo, que pudessem definir e automatizar os SLAs.

## 5.3 Limitações do Estudo

Uma das limitações evidenciadas com a realização deste trabalho, mas precisamente no que se refere a ferramenta utilizada para a geração e execução dos casos e *scripts* de teste de desempenho, no caso o LoadRunner, é que a mesma permite a criação de um único SLA para um cenário de teste. No entanto, um cenário de teste normalmente é formado por um conjunto de casos de uso que resultam em inúmeras transações, eventualmente usadas em diferentes diagramas de atividades. Neste caso, torna-se necessário identificar a qual diagrama de atividades a transação pertence, uma vez que os SLAs são criados de acordo com as transações definidas nos diagramas.

## 5.4 Contribuições e Trabalhos Futuros

Após criar o SLA o teste é executado, e com os resultados obtidos com a execução é realizada a etapa de análise dos resultados. Nesta etapa, é possível identificar, de acordo com o cenário criado e os tempos definidos nos SLAs quais as transações que passaram e quais as transações que falharam no teste, é possível também fazer uma análise comparando os tempos que foram estimados em relação ao tempo mínimo, médio e máximo de execução de cada transação. Outro aspecto importante a ser considerado, é que uma vez criados os modelos, os tempos estimados que são informados, podem ser ajustados de acordo com o cenário de teste que se deseja testar, tendo como base os históricos obtidos com as execuções anteriores.

Também é importante ressaltar que os *scripts* de teste gerados foram executados pelo LoadRunner. No entanto, a abordagem proposta foi desenvolvida com o objetivo de ser estendida para outras ferramentas de desempenho, como por exemplo, o VisualStudio. Da mesma forma que os produtos gerados pela PLeTs, não compreendem um módulo de análise dos resultados que até então foram analisados manualmente, sendo essa uma das atividades de trabalhos futuros, permitindo uma integração maior com as tecnologias de teste de geradores de carga de trabalho.

## 5.5 Contribuições Finais

Diante da abordagem proposta, e na realização de um estudo de caso baseado na aplicação ChangePoint, foi possível demonstrar todas as etapas realizadas para a execução de teste de desempenho. Descrevendo deste a fase de modelagem UML até a definição dos SLAs e instrumentalização dos *scripts* de teste gerados para a ferramenta LoadRunner. Além disso, outro aspecto significativo na realização do trabalho é que os dados obtidos com a execução dos testes são referentes a uma aplicação amplamente utilizada pela indústria, o que proporciona um cenário e um ambiente de teste muito similar a situações reais.





## REFERÊNCIAS

- [1] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 11–33, January 2004.
- [2] I. K. El-Far and J. A. Whittaker, *Model-based Software Testing*. New York, NY, USA: Wiley, 2001, pp. 825–837.
- [3] PLeTs, “Product Line Testing Tool,” Capturado em: <http://www.cepes.pucrs.br/plets/?a=specification>, Maio 2013.
- [4] I. C. Mercury, “Mercury LoadRunner Tutorial,” Capturado em: [http://qageek.files.wordpress.com/2007/05/loadrunner\\_tutorial.pdf](http://qageek.files.wordpress.com/2007/05/loadrunner_tutorial.pdf), Abril 2013.
- [5] J. Lee and R. Ben-Natan, *Integrating Service Level Agreements: Optimizing Your OSS for SLA Delivery*. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [6] B. Beizer, *Software Testing Techniques*, 2nd ed. New York, NY, USA: Van Nostrand Reinhold, 1990.
- [7] L. Apfelbaum and J. Doyle, “Model Based Testing,” in *Proceedings of the Software Quality Week Conference*, San Francisco, CA, USA, 1997, pp. 296–300.
- [8] J. Meier, C. Farre, P. Bansode, S. Barber, and D. Rea, *Performance testing guidance for web applications: patterns & practices*. Redmond, WA, USA: Microsoft Press, 2007.
- [9] M. Woodside, G. Franks, and D. C. Petriu, “The Future of Software Performance Engineering,” in *Proceedings of the Future of Software Engineering*, 2007, pp. 171–187.
- [10] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*. Addison-Wesley Longman Publishing, 2001.
- [11] E. M. Rodrigues, L. D. Viccari, A. F. Zorzo, and I. M. de Souza Gimenes, “PLeTs-Test Automation using Software Product Lines and Model Based Testing,” in *Proceedings of the 22th International Conference on Software Engineering and Knowledge Engineering*, 2010, pp. 483–488.
- [12] M. B. Silveira, E. M. Rodrigues, A. F. Zorzo, L. T. Costa, H. V. Vieira, and F. M. de Oliveira, “Generation of Scripts for Performance Testing Based on UML Models,” in *Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering*, jul. 2011, pp. 258–263.

- [13] L. T. Costa, R. M. Czekster, F. M. de Oliveira, E. M. Rodrigues, M. B. Silveira, and A. F. Zorzo, "Generating Performance Test Scripts and Scenarios Based on Abstract Intermediate Models," in *Proceedings of the 24th International Conference on Software Engineering and Knowledge Engineering*, jul. 2012, pp. 112–117.
- [14] H. Gomaa, *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.
- [15] S. E. I. (SEI), "Software Product Lines (SPL)," Capturado em: <http://www.sei.cmu.edu/productlines/>, May 2013.
- [16] G. J. Myers and C. Sandler, *The Art of Software Testing*. John Wiley & Sons, 2004.
- [17] IEEE, "IEEE Standard Glossary of Software Engineering Terminology: Std 610.12-1990," IEEE, Tech. Rep., 1991.
- [18] W. C. Hetzel and B. Hetzel, *The Complete Guide to Software Testing*. New York, NY, USA: John Wiley & Sons, 1991.
- [19] D. Galin, *Software Quality Assurance: From Theory to Implementation*, 1st ed. Harlow, UK: Addison-Wesley, 2004.
- [20] M. Young and M. Pezze, *Software Testing and Analysis: Process, Principles and Techniques*. New York, NY, USA: John Wiley & Sons, 2005.
- [21] M. E. Delamaro, J. C. Maldonado, and M. Jino, *Introdução ao Teste de Software*. Rio de Janeiro, RJ, Brazil: Elsevier, 2007, p. 408.
- [22] C. U. Smith, *Software Performance Engineering*. New York, NY, USA: John Wiley & Sons, 2002.
- [23] E. Weyuker and F. Vokolos, "Experience with performance testing of software systems: issues, an approach, and case study," *IEEE Transactions on Software Engineering*, vol. 26, no. 12, pp. 1147–1156, December 2000.
- [24] Y. Jing, Z. Lan, W. Hongyuan, S. Yuqiang, and C. Guizhen, "Jmeter-based aging simulation of computing system," in *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on*, vol. 5, August 2010, pp. 282–285.
- [25] Hewlett Packard - HP, "Software HP LoadRunner," Disponível em: [https://h10078.www1.hp.com/cda/hpms/display/main/hpms\\_content.jsp?zn=bto&cp=1-11-126-178\\_4000\\_100\\_\\_>](https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-126-178_4000_100__>), Setembro 2010.

- [26] D. Chadwick, C. Davis, M. Dunn, E. Jessee, A. Kofaldt, K. Mooney, R. Nicolas, A. Patel, J. Reinstrom, K. Siefkes, P. Silva, S. Ulrich, and W. Yeung, *Using Rational Performance Tester Version 7*. Riverton, NJ, USA: IBM Redbooks, 2008.
- [27] BCM Software, "Performance Benchmarking Kit Using Incident Management with SilkPerformer," BMC Software, Tech. Rep., 2007.
- [28] Bai, Xin, "Testing the Performance of an SSAS Cube Using VSTS," in *Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations*. Washington, USA: IEEE, 2010, pp. 986–991.
- [29] T. Chow, "Testing software design modeled by finite-state machines," *Software Engineering, IEEE Transactions on*, vol. SE-4, no. 3, pp. 178–187, May 1978.
- [30] M. D. Beaudry, "Performance-related reliability measures for computing systems," *IEEE Transactions on Computers*, vol. C-27, no. 6, pp. 540–547, June 1978.
- [31] B. Plateau and K. Atif, "Stochastic automata network of modeling parallel systems," *IEEE Transactions on Software Engineering*, vol. 17, no. 10, pp. 1093–1108, October 1991.
- [32] C.-S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Transactions on Automatic Control*, vol. 39, no. 5, pp. 913–931, May 1994.
- [33] M. K. Molloy, "Performance Analysis Using Stochastic Petri Nets," *IEEE Transactions on Computers*, vol. 31, no. 9, pp. 913–917, September 1982.
- [34] C. U. Smith and L. G. Williams, *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [35] M. Veanes, C. Campbell, W. Grieskamp, W. Schulte, N. Tillmann, and L. Nachmanson, "Formal methods and testing," R. M. Hierons, J. P. Bowen, and M. Harman, Eds. Springer-Verlag, 2008, ch. Model-based testing of object-oriented reactive systems with spec explorer, pp. 39–76.
- [36] G. Luo, A. Petrenko, and G. v. Bochmann, "Selecting test sequences for partially-specified nondeterministic finite state machines," in *7th IFIP WG 6.1 International Workshop on Protocol Test Systems*. London, UK, UK: Chapman & Hall, Ltd., October 1995, pp. 95–110.
- [37] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines—a survey," *Proceedings of the IEEE*, vol. 84, no. 8, pp. 1090–1123, 1996.
- [38] K. El-Fakih, N. Yevtushenko, and G. Bochmann, "Fsm-based incremental conformance testing methods," *Software Engineering, IEEE Transactions on*, vol. 30, no. 7, pp. 425–436, 2004.

- [39] A. C. Dias Neto, R. Subramanyan, M. Vieira, and G. H. Travassos, "A survey on model-based testing approaches: A systematic review," in *Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies: Held in Conjunction with the 22Nd IEEE/ACM International Conference on Automated Software Engineering (ASE) 2007*, ser. WEASEL Tech '07. New York, NY, USA: ACM, 2007, pp. 31–36.
- [40] PLeTs, "Centro de Pesquisa em Engenharia de Sistemas," Capturado em: <http://www.cepes.pucrs.br/page>, Maio 2013.
- [41] R. V. EMMA, "EMMA: a Free Java Code Coverage Too," Capturado em: <http://emma.sourceforge.net>, Janeiro 2013.
- [42] M. B. da Silveira, "Conjunto de Características para Teste de Desempenho: Uma Visão a partir de Modelos," Master's thesis, Programa de Pós-Graduação em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, Brasil, 2012.
- [43] A. Petrenko, N. Yevtushenko, A. Lebedev, and A. Das, "Nondeterministic State Machines in Protocol Conformance Testing," in *Proceedings of the 6th International Workshop on Protocol Test Systems VI*, 1993, pp. 363–378.
- [44] O. El Ariss, D. Xu, S. Dandey, B. Vender, P. McClean, and B. Slator, "A systematic capture and replay strategy for testing complex gui based java applications," in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, April 2010, pp. 1038–1043.
- [45] G. Stilian, "Pert: un nuevo instrumento de planificación y control," Capturado em: <http://books.google.com.br/books?id=XVYSnQEACAAJ>, Novembro 1967.
- [46] E. D. Hahn, "Mixture densities for project management activity times: A robust approach to {PERT}," *European Journal of Operational Research*, vol. 188, no. 2, pp. 450 – 459, 2008.
- [47] T. A. S. Foundation, "Apache Tomcat," Capturado em: <http://dx.doi.org/10.1177/0037549711428341>, Novembro 2013.
- [48] "Rpert: Repetitive-projects evaluation and review technique," *Alexandria Engineering Journal*, vol. 53, no. 1, pp. 81–93, 2013.
- [49] A. Bertolino, E. Marchetti, and R. Mirandola, "Real-time uml-based performance engineering to aid manager's decisions in multi-project planning," in *Proceedings of the 3rd international workshop on Software and performance*, ser. WOSP '02. New York, NY, USA: ACM, 2002, pp. 251–261.
- [50] R. Wazlawick, *Engenharia de Software - Conceitos e Práticas*. Rio de Janeiro, RJ, BRASIL: Campus, 2013.

- [51] G. Blokdijk and I. Menken, *Service Level Management Best Practice Handbook: Building, Running and Managing Effective Service Level Management SLAs - Ready to Use Supporting Documents Bringing ITIL Theory into Practice*. London, UK, UK: Emereo Pty Ltd, 2008.
- [52] T. Benchmark-W, "TPC-W," Capturado em: <http://www.tpc.org/tpcw>, Agosto 2013.
- [53] F. M. de Oliveira, R. da S. Menna, H. V. Vieira, and D. D. Ruiz, "Performance Testing from UML Models with Resource Descriptions," *I Brazilian Workshop on Systematic and Automated Software Testing*, pp. 47–51, 2007.
- [54] K. de Pinho Peralta, "Especificação de Aspectos de Segurança e Geração de Casos de Teste," Master's thesis, Programa de Pós-Graduação em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, Brasil, 2009.
- [55] J. Mayer, I. Melzer, and F. Schweiggert, "Lightweight plug-in-based application development," in *Objects, Components, Architectures, Services, and Applications for a Networked World*. Berlin, Heidelberg:SpringerVerlag, 2003, pp. 87–102.
- [56] MySQL, "MySQL," Capturado em: <http://www.mysql.fr/>, Novembro 2013.
- [57] T. A. S. Foundation, "Apache Tomcat," Capturado em: <http://tomcat.apache.org/>, Novembro 2013.
- [58] M. Kalita, S. Khanikar, and T. Bezboruah, "Investigation on performance testing and evaluation of PReWebN: a java technique for implementing web application," *Institution of Engineering and Technology Software*, vol. 5, no. 5, pp. 434–444, October 2011.
- [59] D. E. Geetha, T. S. Kumar, and K. R. Kanth, "Framework for Hybrid Performance Prediction Process Model: Use Case Performance Engineering Approach," *SIGSOFT Software Engineering Notes*, vol. 36, no. 3, pp. 1–15, May 2011.
- [60] M. Vasar, S. N. Srirama, and M. Dumas, "Framework for Monitoring and Testing Web Application Scalability on the Cloud," in *Proceedings of the WICSA/ECSA Companion Volume*, 2012, pp. 53–60.
- [61] C. Weiss, D. Westermann, C. Heger, and M. Moser, "Systematic Performance Evaluation Based on Tailored Benchmark Applications," in *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, 2013, pp. 411–420.
- [62] Changepoint, "Changepoint," Capturado em: <http://www.changepoint.com>, Dezembro 2013.
- [63] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-oriented Modeling and Design*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991.



## A. MODELOS DAS APLICAÇÃO CHANGEPOIT

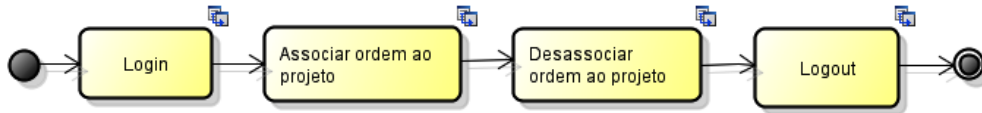


Figura A.1: Diagrama de atividades do caso de uso Associar/Desassociar Ordem aos Projetos

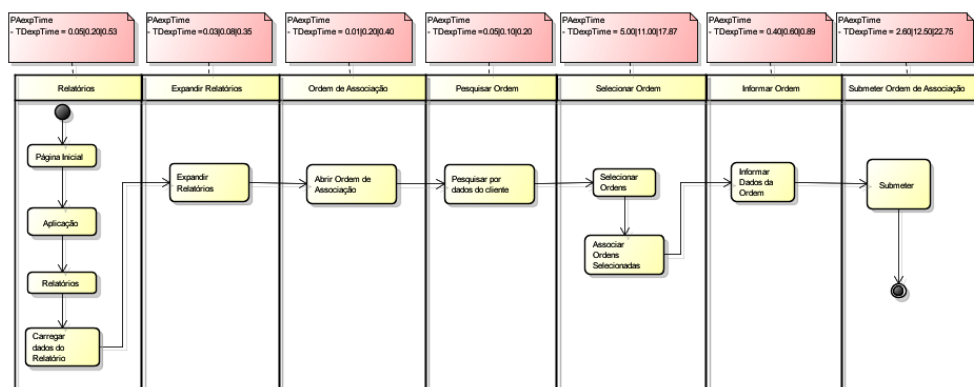


Figura A.2: Subdiagrama de atividades da atividade Associar Ordem aos Projetos

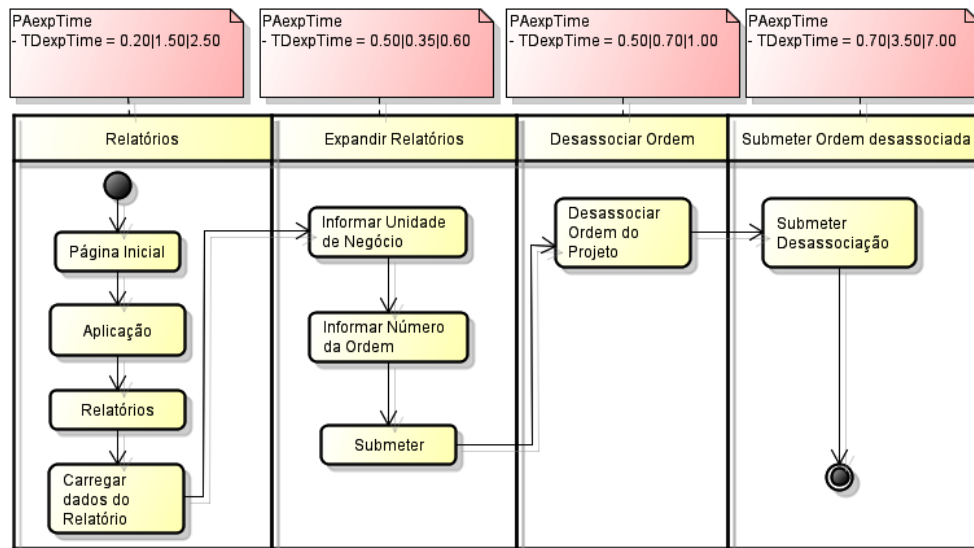


Figura A.3: Subdiagrama de atividades da atividade Desassociar Ordem aos Projetos

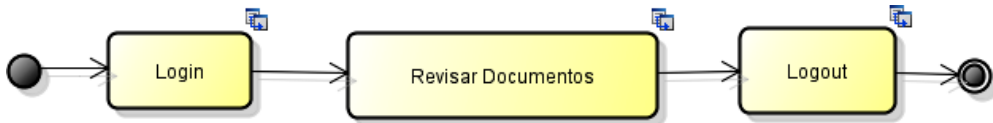


Figura A.4: Diagrama de atividades do caso de uso Revisar Documentos

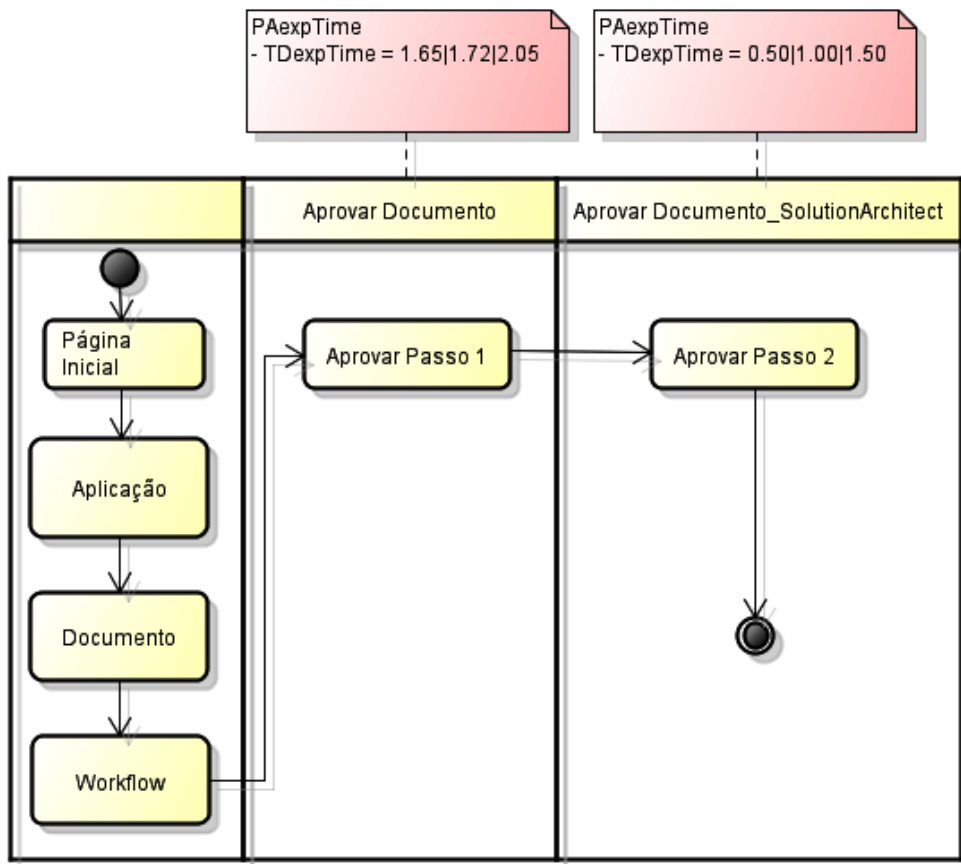


Figura A.5: Subdiagrama de atividades da atividade Revisar Documentos

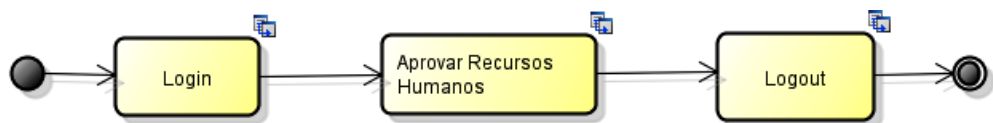


Figura A.6: Diagrama de atividades do caso de uso Aprovar Recursos Humanos



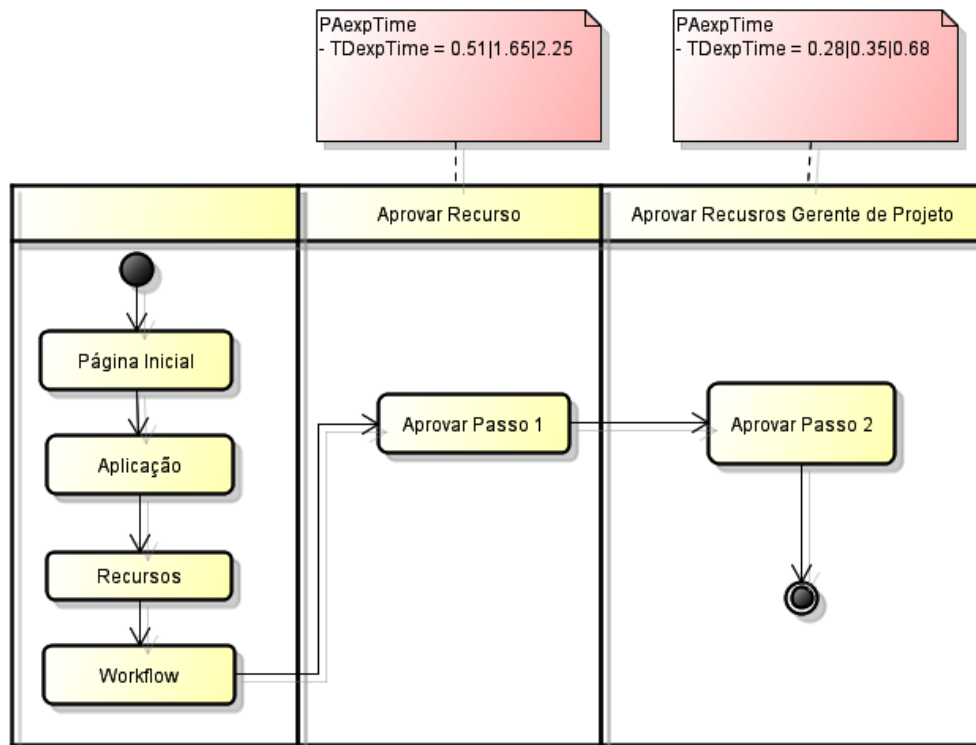


Figura A.7: Subdiagrama de atividades da atividade Aprovar Recursos Humanos

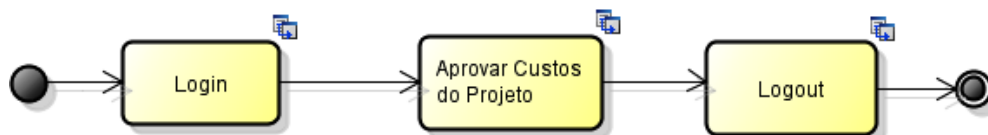


Figura A.8: Diagrama de atividades do caso de uso Aprovar Custos do Projeto

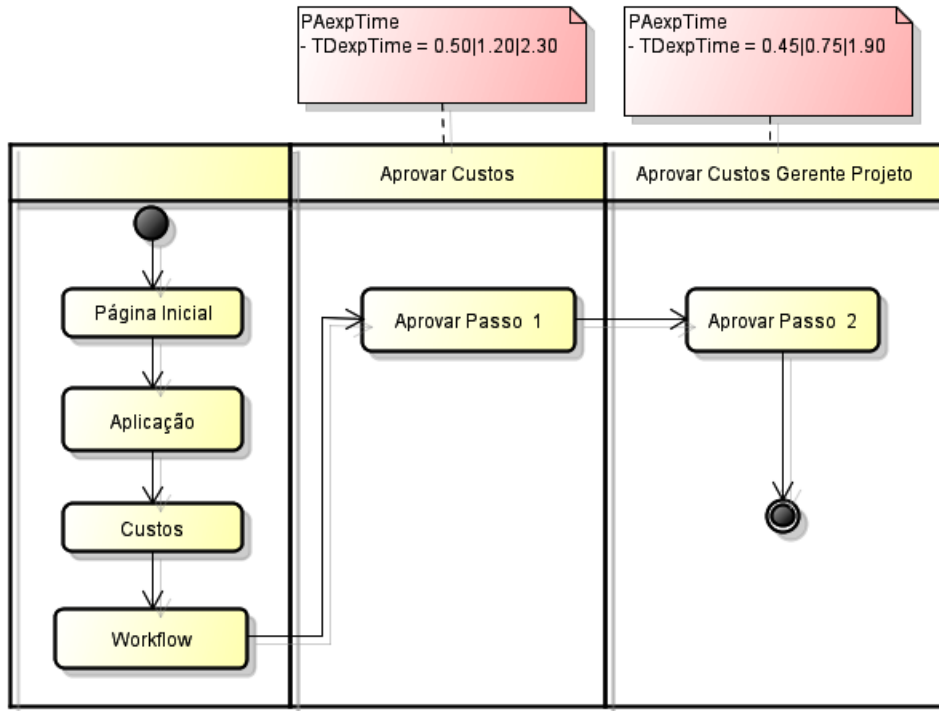


Figura A.9: Subdiagrama de atividades da atividade Aprovar Custos do Projeto

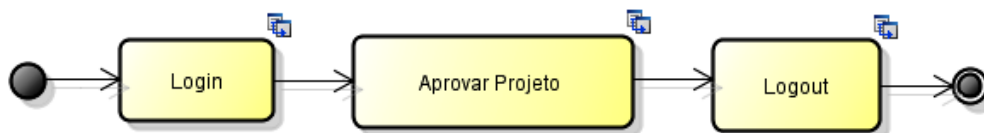


Figura A.10: Diagrama de atividades do caso de uso Aprovar Projeto

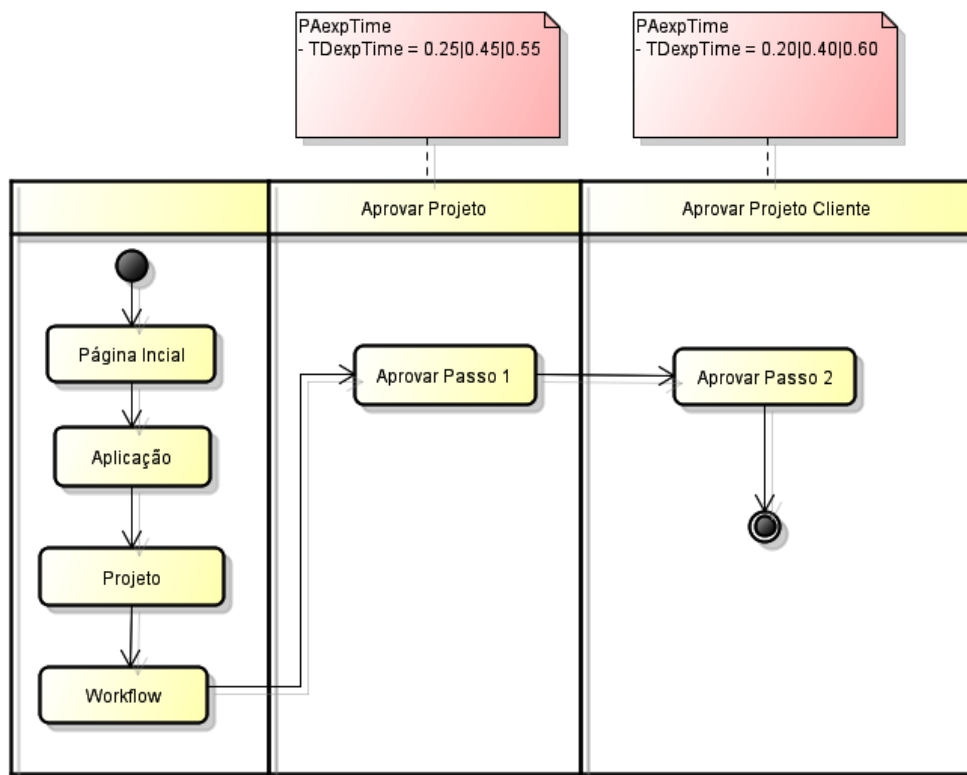


Figura A.11: Subdiagrama de atividades da atividade Aprovar Projeto



## B. MODELOS DA APLICAÇÃO TPCW

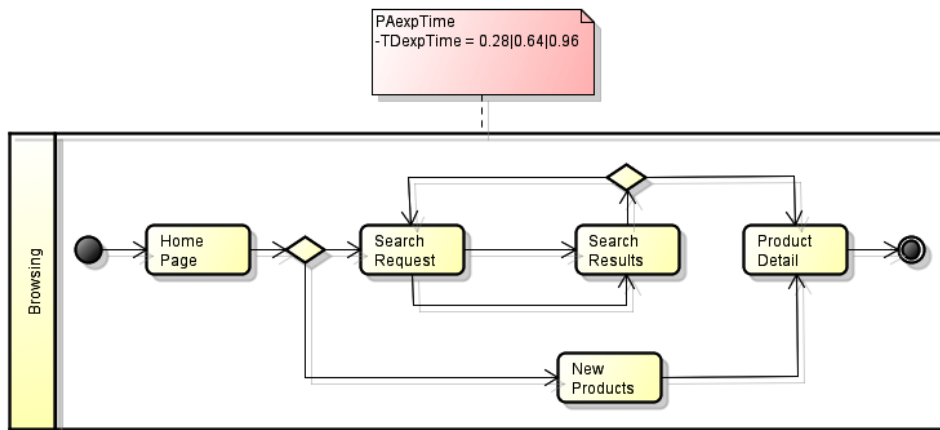


Figura B.1: Diagrama de atividades do caso de uso Browsing

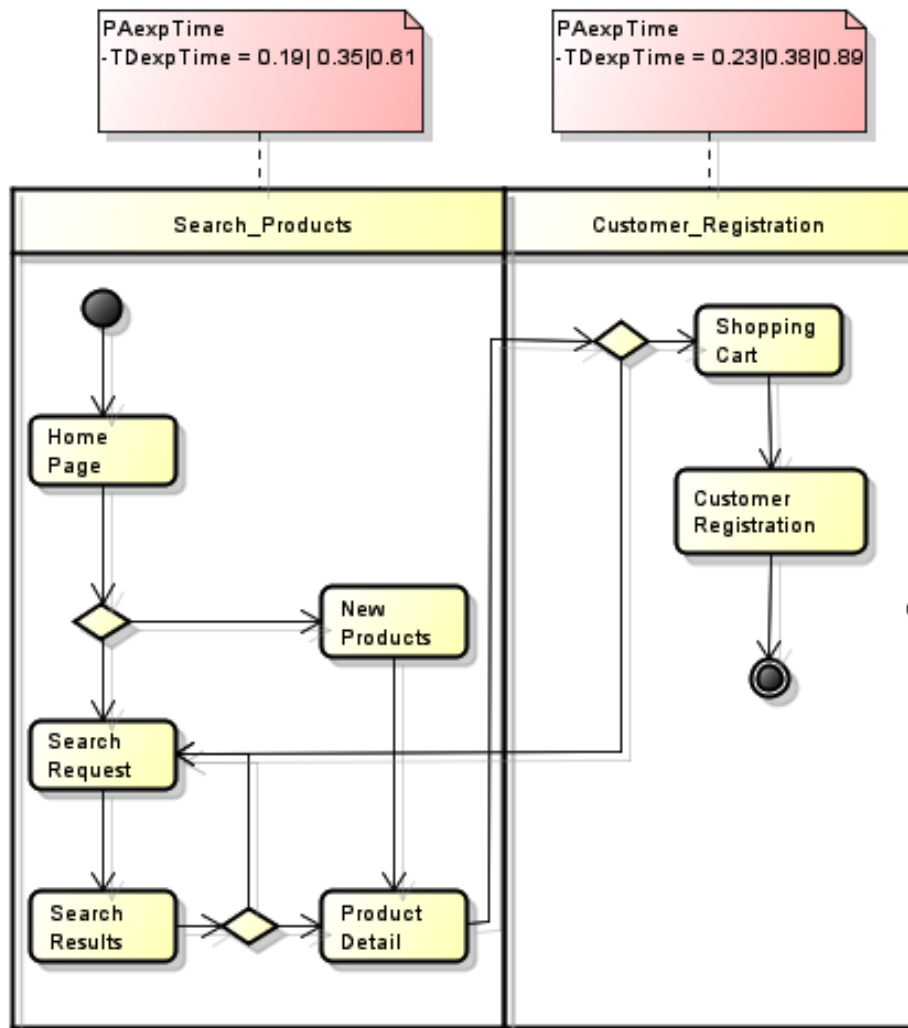


Figura B.2: Diagrama de atividades do caso de uso Ordering



## C. MODELOS DA APLICAÇÃO SKILLS

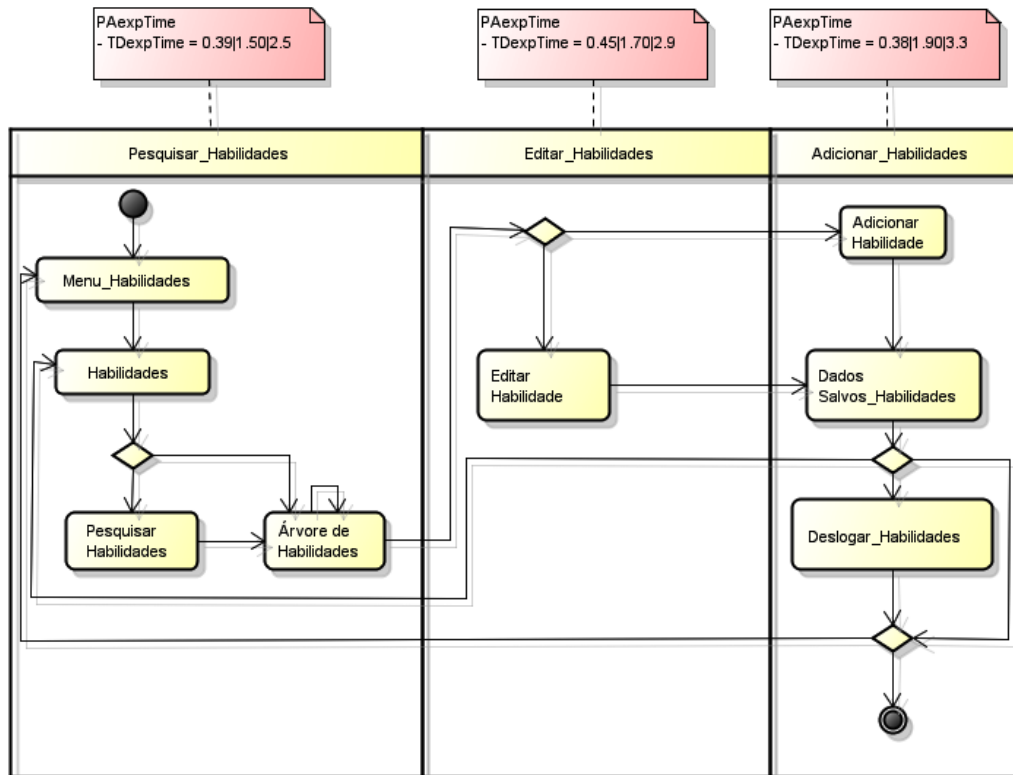


Figura C.1: Diagrama de atividades do caso de uso Gerenciar Habilidades

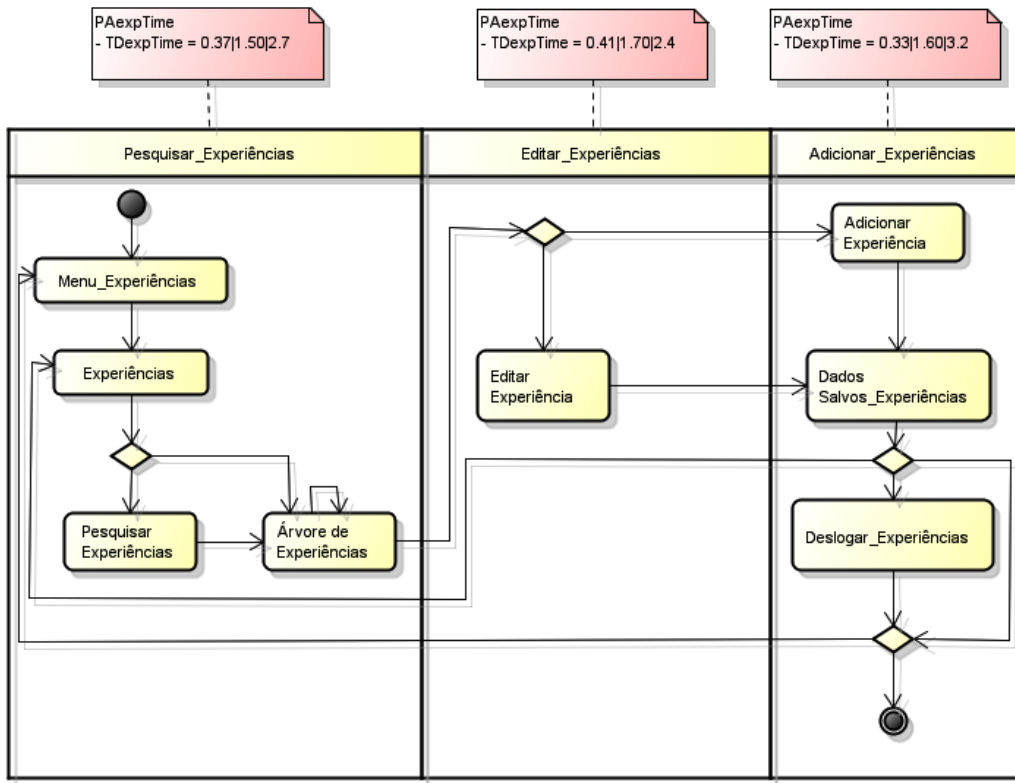


Figura C.2: Diagrama de atividades do caso de uso Gerenciar Experiências

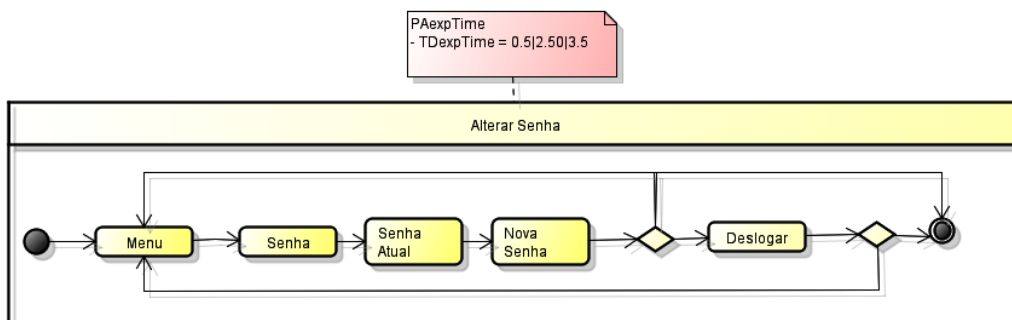


Figura C.3: Diagrama de atividades do caso de uso Alterar Senha



## D. DADOS GERADOS NA EXECUÇÃO DOS TESTES

Dados da execução dos Testes

Usuários = 100

**Associar/Desassociar Ordem aos Projetos**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.07	0.30	5.96	0.74	1.21	Falhou
	Submeter Login	0.15	0.59	2.50	0.16	0.62	2.59	0.84	0.87	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.08	0.22	1.46	1.45	0.40	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.02	0.10	0.63	0.42	0.18	Passou
	Ordem de Associação	0.50	0.70	1.00	0.05	0.11	1.05	0.62	0.26	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.17	2.40	6.98	3.62	2.79	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.08	0.22	1.46	0.23	0.40	Falhou
	Informar Ordem	0.03	0.08	0.35	0.02	0.10	0.63	0.12	0.18	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.08	0.12	0.40	0.20	0.16	Passou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.06	0.12	0.29	0.11	0.14	Falhou
	Expandir Relatórios	5.00	11.00	17.87	9.33	11.18	19.92	11.15	12.33	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.41	0.60	1.06	0.62	0.65	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.23	11.53	21.86	12.56	11.37	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.03	0.06	2.38	0.37	0.44	Falhou

**Informar Horas Trabalhadas**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.05	0.50	6.06	0.74	1.35	Falhou
	Submeter Login	0.15	0.59	2.50	0.14	0.68	1.39	0.84	0.71	Passou
<b>Informar Horas Trabalhadas</b>	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.02	0.07	3.67	2.40	0.66	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.00	0.02	0.80	0.20	0.15	Passou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.03	0.07	1.64	0.99	0.33	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.07	0.09	1.32	0.37	0.29	Passou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.03	0.60	4.06	0.74	1.08	Falhou
	Submeter Login	0.15	0.59	2.50	0.13	0.12	1.89	0.84	0.42	Passou
<b>Revisar</b>	Aprovar Documento	1.65	1.72	2.05	0.05	0.09	0.20	1.76	0.10	Passou
<b>Documentos</b>	Aprovar Documento_SA	0.50	1.00	1.50	0.47	1.00	1.83	1.00	1.05	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.05	0.04	1.96	0.37	0.36	Passou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.15	0.60	4.19	0.74	1.12	Falhou
	Submeter Login	0.15	0.59	2.50	0.18	0.52	1.59	0.84	0.64	Passou
<b>Aprovar Recursos</b>	Aprovar Recurso	0.51	1.65	2.25	0.24	0.40	2.17	1.56	0.67	Passou
<b>Humanos</b>	Aprovar Recurso_PM	0.28	0.35	0.68	0.08	0.25	1.03	0.39	0.35	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.17	0.34	1.78	0.37	0.55	Falhou

### Aprovar Custos do Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.09	0.20	3.96	0.74	0.81	Falhou
	Submeter Login	0.15	0.59	2.50	0.17	0.18	2.10	0.84	0.50	Passou
<b>Aprovar Custos do</b>	Aprovar Custos	0.50	1.20	2.30	0.11	0.22	2.06	1.27	0.51	Passou
<b>Projeto</b>	Aprovar Custos_PM	0.45	0.75	1.90	0.06	0.11	0.53	0.89	0.17	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.04	0.20	1.00	0.37	0.31	Passou

### Aprovar Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.08	0.10	2.36	0.74	0.47	Passou
	Submeter Login	0.15	0.59	2.50	0.19	0.32	2.49	0.84	0.66	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.03	0.03	0.10	0.43	0.04	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.00	0.00	0.02	0.40	0.00	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.32	0.01	1.03	0.37	0.23	Passou

Usuários = 200

### Associar/Desassociar Ordem aos Projetos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.09	0.33	6.75	0.74	1.36	Falhou
	Submeter Login	0.15	0.59	2.50	0.18	0.67	2.61	0.84	0.91	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.12	0.25	1.49	1.45	0.44	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.07	0.16	0.67	0.42	0.23	Passou
	Ordem de Associação	0.70	0.50	1.00	0.06	0.16	1.13	0.62	0.31	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.18	2.41	7.03	3.62	2.81	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.11	0.26	1.53	0.23	0.45	Falhou
	Informar Ordem	0.03	0.08	0.35	0.05	0.13	0.69	0.12	0.21	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.11	0.17	0.43	0.20	0.20	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.08	0.14	0.33	0.11	0.16	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.05	12.68	21.02	11.15	13.63	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.43	0.68	1.34	0.62	0.75	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.26	11.56	22.05	12.56	11.43	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.06	0.07	2.56	0.37	0.48	Falhou

### Informar Horas Trabalhadas

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.13	0.51	6.06	0.74	1.37	Falhou
	Submeter Login	0.15	0.59	2.50	0.17	0.89	1.39	0.84	0.85	Falhou
Informar Horas Trabalhadas	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.24	0.52	3.67	2.40	1.00	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.04	0.40	0.80	0.20	0.41	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.06	0.45	1.64	0.99	0.58	Passou
Logout	Logout	0.00	0.03	2.10	0.11	0.13	1.45	0.37	0.35	Passou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.05	0.65	4.06	0.74	1.12	Falhou
	Submeter Login	0.15	0.59	2.50	0.15	0.23	1.89	0.84	0.49	Passou
Revisar Documentos	Aprovar Documento	1.65	1.72	2.05	0.09	0.17	0.32	1.76	0.18	Passou
	Aprovar Documento_SA	0.50	1.00	1.50	0.53	1.34	1.92	1.00	1.30	Falhou
Logout	Logout	0.00	0.03	2.10	0.89	1.50	2.13	0.37	1.50	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.19	0.68	5.26	0.74	1.36	Falhou
	Submeter Login	0.15	0.59	2.50	0.24	0.67	2.35	0.84	0.88	Falhou
Aprovar Recursos Humanos	Aprovar Recurso	0.51	1.65	2.25	0.37	0.56	2.17	1.56	0.80	Passou
	Aprovar Recurso_PM	0.28	0.35	0.68	0.13	0.32	1.34	0.39	0.46	Falhou
Logout	Logout	0.00	0.03	2.10	0.23	0.41	2.89	0.37	0.79	Falhou

**Aprovar Custos do Projeto**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.12	1.35	4.01	0.74	1.59	Falhou
	Submeter Login	0.15	0.59	2.50	0.19	0.25	2.89	0.84	0.68	Passou
<b>Aprovar Custos do Projeto</b>	Aprovar Custos	0.50	1.20	2.30	0.11	0.22	2.06	1.27	0.51	Passou
	Aprovar Custos_PM	0.45	0.75	1.90	0.06	0.11	0.53	0.89	0.17	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.06	0.25	1.35	0.37	0.40	Falhou

**Aprovar Projeto**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.11	0.15	2.87	0.74	0.60	Passou
	Submeter Login	0.15	0.59	2.50	0.22	0.35	2.56	0.84	0.70	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.08	0.11	0.16	0.43	0.11	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.03	0.09	0.12	0.40	0.09	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.39	0.50	1.05	0.37	0.57	Falhou

Usuários = 300

**Associar/Desassociar Ordem aos Projetos**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.13	0.35	6.77	0.74	1.38	Falhou
	Submeter Login	0.15	0.59	2.50	0.19	0.68	2.63	0.84	0.92	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.14	0.27	1.53	1.45	0.46	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.09	0.17	0.71	0.42	0.25	Passou
	Ordem de Associação	0.70	0.50	1.00	0.08	0.19	1.15	0.62	0.33	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.19	2.43	7.06	3.62	2.83	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.13	0.28	1.56	0.23	0.47	Falhou
	Informar Ordem	0.03	0.08	0.35	0.09	0.14	0.71	0.12	0.23	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.13	0.19	0.45	0.20	0.22	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.1	0.16	0.35	0.11	0.18	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	12.65	21.05	11.15	13.62	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.45	0.69	1.35	0.62	0.76	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.27	11.59	22.07	12.56	11.45	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.08	0.11	2.58	0.37	0.52	Falhou

**Informar Horas Trabalhadas**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.14	0.53	6.08	0.74	1.39	Falhou
	Submeter Login	0.15	0.59	2.50	0.18	0.90	1.41	0.84	0.87	Falhou
<b>Informar Horas Trabalhadas</b>	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.26	0.53	3.69	2.40	1.01	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.06	0.42	0.84	0.20	0.43	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.07	0.65	1.69	0.99	0.73	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.12	0.14	1.46	0.37	0.36	Passou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.06	0.66	4.06	0.74	1.13	Falhou
	Submeter Login	0.15	0.59	2.50	0.16	0.25	1.90	0.84	0.51	Passou
<b>Revisar</b>	Aprovar Documento	1.65	1.72	2.05	0.10	0.19	0.33	1.76	0.20	Passou
<b>Documentos</b>	Aprovar Documento_SA	0.50	1.00	1.50	0.53	1.35	1.94	1.00	1.31	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.89	1.52	2.14	0.37	1.52	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.19	0.70	5.26	0.74	1.38	Falhou
	Submeter Login	0.15	0.59	2.50	0.25	0.68	2.35	0.84	0.89	Falhou
<b>Aprovar Recursos</b>	Aprovar Recurso	0.51	1.65	2.25	0.38	0.57	2.19	1.56	0.81	Passou
<b>Humanos</b>	Aprovar Recurso_PM	0.28	0.35	0.68	0.14	0.33	1.35	0.39	0.47	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.24	0.42	2.89	0.37	0.80	Falhou

### Aprovar Custos do Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.13	1.36	4.01	0.74	1.60	Falhou
	Submeter Login	0.15	0.59	2.50	0.20	0.26	2.89	0.84	0.69	Passou
<b>Aprovar Custos do</b>	Aprovar Custos	0.50	1.20	2.30	0.11	0.23	2.06	1.27	0.52	Passou
<b>Projeto</b>	Aprovar Custos_PM	0.45	0.75	1.90	0.06	0.11	0.54	0.89	0.17	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.07	0.27	1.38	0.37	0.42	Falhou



### Aprovar Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.12	0.16	2.89	0.74	0.61	Passou
	Submeter Login	0.15	0.59	2.50	0.23	0.36	2.58	0.84	0.71	Passou
Aprovar Projeto	Aprovar Projeto	0.25	0.45	0.55	0.09	0.14	0.18	0.43	0.14	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.04	0.11	0.14	0.40	0.10	Passou
Logout	Logout	0.00	0.03	2.10	0.41	0.52	1.07	0.37	0.59	Falhou

Usuários = 400

### Associar/Desassociar Ordem aos Projetos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.13	0.36	6.76	0.74	1.39	Falhou
	Submeter Login	0.15	0.59	2.50	0.2	0.71	2.63	0.84	0.95	Falhou
Associar ordem ao projeto	Relatórios	0.20	1.50	2.50	0.15	0.32	1.55	1.45	0.50	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.09	0.21	0.72	0.42	0.28	Passou
	Ordem de Associação	0.70	0.50	1.00	0.09	0.21	1.15	0.62	0.35	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.2	2.46	7.07	3.62	2.85	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.14	0.31	1.57	0.23	0.49	Falhou
	Informar Ordem	0.03	0.08	0.35	0.09	0.16	0.71	0.12	0.24	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.14	0.20	0.46	0.20	0.23	Falhou
Desassociar ordem ao projeto	Relatórios	0.05	0.10	0.20	0.3	0.18	0.36	0.11	0.23	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	12.67	21.05	11.15	13.63	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.46	0.70	1.37	0.62	0.77	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.28	11.59	22.07	12.56	11.45	Passou
Logout	Logout	0.00	0.03	2.10	0.09	0.13	2.59	0.37	0.53	Falhou

### Informar Horas Trabalhadas

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.16	0.55	6.08	0.74	1.41	Falhou
	Submeter Login	0.15	0.59	2.50	0.19	0.99	1.41	0.84	0.93	Falhou
Informar Horas Trabalhadas	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.27	0.55	3.69	2.40	1.03	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.06	0.45	0.84	0.20	0.45	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.07	0.66	1.69	0.99	0.73	Passou
Logout	Logout	0.00	0.03	2.10	0.13	0.16	1.47	0.37	0.37	Falhou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.06	0.90	4.06	0.74	1.29	Falhou
	Submeter Login	0.15	0.59	2.50	0.17	0.27	1.90	0.84	0.53	Passou
Revisar Documentos	Aprovar Documento	1.65	1.72	2.05	0.12	0.21	0.34	1.76	0.22	Passou
	Aprovar Documento_SA	0.50	1.00	1.50	0.53	1.38	1.94	1.00	1.33	Falhou
Logout	Logout	0.00	0.03	2.10	0.90	1.63	2.14	0.37	1.59	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.19	0.75	5.26	0.74	1.41	Falhou
	Submeter Login	0.15	0.59	2.50	0.25	0.72	2.35	0.84	0.91	Falhou
Aprovar Recursos Humanos	Aprovar Recurso	0.51	1.65	2.25	0.38	0.58	2.19	1.56	0.82	Passou
	Aprovar Recurso_PM	0.28	0.35	0.68	0.14	0.35	1.36	0.39	0.48	Falhou
Logout	Logout	0.00	0.03	2.10	0.25	0.43	2.89	0.37	0.81	Falhou

**Aprovar Custos do Projeto**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.14	1.38	4.03	0.74	1.62	Falhou
	Submeter Login	0.15	0.59	2.50	0.21	0.28	2.91	0.84	0.71	Passou
<b>Aprovar Custos do Projeto</b>	Aprovar Custos	0.50	1.20	2.30	0.13	0.25	2.06	1.27	0.53	Passou
	Aprovar Custos_PM	0.45	0.75	1.90	0.07	0.13	0.54	0.89	0.19	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.09	0.28	1.38	0.37	0.43	Falhou

**Aprovar Projeto**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.12	0.19	2.89	0.74	0.63	Passou
	Submeter Login	0.15	0.59	2.50	0.24	0.37	2.58	0.84	0.72	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.09	0.16	0.18	0.43	0.15	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.05	0.12	0.15	0.40	0.11	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.41	0.55	1.07	0.37	0.61	Falhou

Usuários = 500

**Associar/Desassociar Ordem aos Projetos**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.14	0.39	6.76	0.74	1.41	Falhou
	Submeter Login	0.15	0.59	2.50	0.3	0.73	2.63	0.84	0.98	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.16	0.35	1.55	1.45	0.52	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.1	0.26	0.72	0.42	0.31	Passou
	Ordem de Associação	0.70	0.50	1.00	0.09	0.25	1.15	0.62	0.37	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.5	2.99	7.07	3.62	3.26	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.14	0.35	1.57	0.23	0.52	Falhou
	Informar Ordem	0.03	0.08	0.35	0.09	0.19	0.71	0.12	0.26	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.14	0.25	0.46	0.20	0.27	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.3	0.22	0.36	0.11	0.26	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	13.99	21.05	11.15	14.51	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.46	0.74	1.37	0.62	0.80	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.28	12.02	22.07	12.56	11.74	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.09	0.16	2.59	0.37	0.55	Falhou

**Informar Horas Trabalhadas**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.16	0.57	6.08	0.74	1.42	Falhou
	Submeter Login	0.15	0.59	2.50	0.19	0.99	1.41	0.84	0.93	Falhou
<b>Informar Horas Trabalhadas</b>	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.27	0.58	3.69	2.40	1.05	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.06	0.49	0.84	0.20	0.48	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.08	0.68	1.70	0.99	0.75	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.13	0.18	1.47	0.37	0.39	Falhou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.07	1.20	4.06	0.74	1.49	Falhou
	Submeter Login	0.15	0.59	2.50	0.17	0.29	1.90	0.84	0.54	Passou
<b>Revisar Documentos</b>	Aprovar Documento	1.65	1.72	2.05	0.12	0.22	0.34	1.76	0.22	Passou
	Aprovar Documento_SA	0.50	1.00	1.50	0.53	1.41	1.94	1.00	1.35	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.90	1.63	2.14	0.37	1.59	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.20	1.05	5.26	0.74	1.61	Falhou
	Submeter Login	0.15	0.59	2.50	0.26	0.74	2.35	0.84	0.93	Falhou
<b>Aprovar Recursos Humanos</b>	Aprovar Recurso	0.51	1.65	2.25	0.38	0.61	2.19	1.56	0.84	Passou
	Aprovar Recurso_PM	0.28	0.35	0.68	0.14	0.37	1.36	0.39	0.50	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.25	0.45	2.89	0.37	0.82	Falhou

### Aprovar Custos do Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.14	1.39	4.03	0.74	1.62	Falhou
	Submeter Login	0.15	0.59	2.50	0.21	0.28	2.91	0.84	0.71	Passou
<b>Aprovar Custos do Projeto</b>	Aprovar Custos	0.50	1.20	2.30	0.13	0.29	2.06	1.27	0.56	Passou
	Aprovar Custos_PM	0.45	0.75	1.90	0.07	0.15	0.54	0.89	0.20	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.09	0.32	1.38	0.37	0.46	Falhou

### Aprovar Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.12	0.26	2.89	0.74	0.68	Passou
	Submeter Login	0.15	0.59	2.50	0.24	0.39	2.58	0.84	0.73	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.09	0.17	0.23	0.43	0.17	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.05	0.13	0.10	0.40	0.11	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.41	0.56	1.07	0.37	0.62	Falhou

Usuários = 600

### Associar/Desassociar Ordem aos Projetos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.15	0.41	6.76	0.74	1.43	Falhou
	Submeter Login	0.15	0.59	2.50	0.4	0.74	2.64	0.84	1.00	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.16	0.37	1.55	1.45	0.53	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.2	0.28	0.72	0.42	0.34	Passou
	Ordem de Associação	0.70	0.50	1.00	0.09	0.27	1.15	0.62	0.39	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.5	3.00	7.07	3.62	3.26	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.14	0.37	1.57	0.23	0.53	Falhou
	Informar Ordem	0.03	0.08	0.35	0.09	0.20	0.71	0.12	0.27	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.14	0.26	0.46	0.20	0.27	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.3	0.23	0.36	0.11	0.26	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	14.00	21.05	11.15	14.52	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.46	0.75	1.37	0.62	0.81	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.28	12.05	22.07	12.56	11.76	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.09	0.17	2.59	0.37	0.56	Falhou

### Informar Horas Trabalhadas

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.16	0.58	6.08	0.74	1.43	Falhou
	Submeter Login	0.15	0.59	2.50	0.20	1.10	1.42	0.84	1.00	Falhou
Informar Horas Trabalhadas	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.27	0.61	3.69	2.40	1.07	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.07	0.51	0.85	0.20	0.49	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.09	0.70	1.71	0.99	0.77	Passou
Logout	Logout	0.00	0.03	2.10	0.13	0.21	1.47	0.37	0.41	Falhou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.07	1.30	4.06	0.74	1.56	Falhou
	Submeter Login	0.15	0.59	2.50	0.18	0.30	1.91	0.84	0.55	Passou
Revisar Documentos	Aprovar Documento	1.65	1.72	2.05	0.12	0.24	0.35	1.76	0.24	Passou
	Aprovar Documento_SA	0.50	1.00	1.50	0.53	1.43	1.94	1.00	1.37	Falhou
Logout	Logout	0.00	0.03	2.10	0.91	1.65	2.15	0.37	1.61	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.21	1.98	5.27	0.74	2.23	Falhou
	Submeter Login	0.15	0.59	2.50	0.26	0.75	2.35	0.84	0.94	Falhou
Aprovar Recursos Humanos	Aprovar Recurso	0.51	1.65	2.25	0.39	0.62	2.19	1.56	0.84	Passou
	Aprovar Recurso_PM	0.28	0.35	0.68	0.15	0.38	1.37	0.39	0.51	Falhou
Logout	Logout	0.00	0.03	2.10	0.26	0.46	2.90	0.37	0.83	Falhou

### Aprovar Custos do Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.15	1.41	4.03	0.74	1.64	Falhou
	Submeter Login	0.15	0.59	2.50	0.21	0.29	2.91	0.84	0.71	Passou
<b>Aprovar Custos do Projeto</b>	Aprovar Custos	0.50	1.20	2.30	0.14	0.30	2.07	1.27	0.57	Passou
	Aprovar Custos_PM	0.45	0.75	1.90	0.07	0.17	0.54	0.89	0.22	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.09	0.33	1.38	0.37	0.47	Falhou

### Aprovar Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.12	0.27	2.89	0.74	0.68	Passou
	Submeter Login	0.15	0.59	2.50	0.25	0.41	2.58	0.84	0.75	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.09	0.19	0.22	0.43	0.18	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.05	0.15	0.10	0.40	0.13	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.42	0.58	1.08	0.37	0.64	Falhou



Usuários = 700

**Associar/Desassociar Ordem aos Projetos**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.15	0.43	6.77	0.74	1.44	Falhou
	Submeter Login	0.15	0.59	2.50	0.4	0.75	2.64	0.84	1.01	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.17	0.38	1.56	1.45	0.54	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.2	0.29	0.72	0.42	0.35	Passou
	Ordem de Associação	0.70	0.50	1.00	0.1	0.28	1.17	0.62	0.40	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.6	3.00	7.08	3.62	3.28	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.14	0.38	1.58	0.23	0.54	Falhou
	Informar Ordem	0.03	0.08	0.35	0.1	0.20	0.72	0.12	0.27	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.15	0.27	0.47	0.20	0.28	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.3	0.24	0.37	0.11	0.27	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	14.00	21.06	11.15	14.52	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.47	0.76	1.38	0.62	0.82	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.28	12.06	22.08	12.56	11.77	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.1	0.18	2.59	0.37	0.57	Falhou

**Informar Horas Trabalhadas**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.17	0.59	6.08	0.74	1.44	Falhou
	Submeter Login	0.15	0.59	2.50	0.20	1.11	1.43	0.84	1.01	Falhou
<b>Informar Horas Trabalhadas</b>	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.27	0.62	3.69	2.40	1.07	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.07	0.53	0.85	0.20	0.51	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.09	0.71	1.71	0.99	0.77	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.14	0.22	1.48	0.37	0.42	Falhou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.07	1.31	4.06	0.74	1.56	Falhou
	Submeter Login	0.15	0.59	2.50	0.19	0.32	1.92	0.84	0.57	Passou
<b>Revisar</b>	Aprovar Documento	1.65	1.72	2.05	0.12	0.25	0.36	1.76	0.25	Passou
<b>Documentos</b>	Aprovar Documento_SA	0.50	1.00	1.50	0.54	1.44	1.95	1.00	1.38	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.91	1.67	2.15	0.37	1.62	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.20	1.05	5.26	0.74	1.61	Falhou
	Submeter Login	0.15	0.59	2.50	0.26	0.74	2.35	0.84	0.93	Falhou
<b>Aprovar Recursos</b>	Aprovar Recurso	0.51	1.65	2.25	0.38	0.61	2.19	1.56	0.84	Passou
<b>Humanos</b>	Aprovar Recurso_PM	0.28	0.35	0.68	0.14	0.37	1.36	0.39	0.50	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.25	0.45	2.89	0.37	0.82	Falhou

### Aprovar Custos do Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.16	1.43	4.04	0.74	1.65	Falhou
	Submeter Login	0.15	0.59	2.50	0.21	0.31	2.91	0.84	0.73	Passou
<b>Aprovar Custos do</b>	Aprovar Custos	0.50	1.20	2.30	0.14	0.32	2.08	1.27	0.58	Passou
<b>Projeto</b>	Aprovar Custos_PM	0.45	0.75	1.90	0.07	0.18	0.54	0.89	0.22	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.09	0.34	1.38	0.37	0.47	Falhou

### Aprovar Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.13	0.28	2.89	0.74	0.69	Passou
	Submeter Login	0.15	0.59	2.50	0.26	0.43	2.59	0.84	0.76	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.09	0.21	0.23	0.43	0.19	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.05	0.17	0.10	0.40	0.14	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.43	0.61	1.09	0.37	0.66	Falhou

Usuários = 800

### Associar/Desassociar Ordem aos Projetos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.16	0.44	6.77	0.74	1.45	Falhou
	Submeter Login	0.15	0.59	2.50	0.5	0.76	2.64	0.84	1.03	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.18	0.39	1.56	1.45	0.55	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.2	0.30	0.73	0.42	0.36	Passou
	Ordem de Associação	0.70	0.50	1.00	0.1	0.29	1.18	0.62	0.41	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.6	3.10	7.09	3.62	3.35	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.14	0.39	1.58	0.23	0.55	Falhou
	Informar Ordem	0.03	0.08	0.35	0.2	0.21	0.73	0.12	0.30	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.15	0.28	0.48	0.20	0.29	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.4	0.25	0.37	0.11	0.30	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	14.10	21.06	11.15	14.59	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.47	0.77	1.38	0.62	0.82	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.29	12.07	22.09	12.56	11.78	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.1	0.19	2.59	0.37	0.58	Falhou

### Informar Horas Trabalhadas

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.17	0.60	6.08	0.74	1.44	Falhou
	Submeter Login	0.15	0.59	2.50	0.20	1.12	1.43	0.84	1.02	Falhou
Informar Horas Trabalhadas	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.27	0.63	3.69	2.40	1.08	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.08	0.54	0.85	0.20	0.52	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.10	0.72	1.72	0.99	0.78	Passou
Logout	Logout	0.00	0.03	2.10	0.15	0.23	1.49	0.37	0.43	Falhou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.08	1.32	4.07	0.74	1.57	Falhou
	Submeter Login	0.15	0.59	2.50	0.19	0.32	1.92	0.84	0.57	Passou
Revisar Documentos	Aprovar Documento	1.65	1.72	2.05	0.12	0.25	0.37	1.76	0.25	Passou
	Aprovar Documento_SA	0.50	1.00	1.50	0.55	1.45	1.96	1.00	1.39	Falhou
Logout	Logout	0.00	0.03	2.10	0.91	1.68	2.15	0.37	1.63	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.22	2.00	5.27	0.74	2.25	Falhou
	Submeter Login	0.15	0.59	2.50	0.26	0.77	2.35	0.84	0.95	Falhou
Aprovar Recursos Humanos	Aprovar Recurso	0.51	1.65	2.25	0.40	0.64	2.19	1.56	0.86	Passou
	Aprovar Recurso_PM	0.28	0.35	0.68	0.15	0.40	1.38	0.39	0.52	Falhou
Logout	Logout	0.00	0.03	2.10	0.27	0.48	2.91	0.37	0.85	Falhou

### Aprovar Custos do Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.16	1.44	4.04	0.74	1.66	Falhou
	Submeter Login	0.15	0.59	2.50	0.21	0.31	2.91	0.84	0.73	Passou
<b>Aprovar Custos do Projeto</b>	Aprovar Custos	0.50	1.20	2.30	0.14	0.33	2.08	1.27	0.59	Passou
	Aprovar Custos_PM	0.45	0.75	1.90	0.07	0.19	0.54	0.89	0.23	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.09	0.34	1.38	0.37	0.47	Falhou

### Aprovar Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.13	0.29	2.89	0.74	0.70	Passou
	Submeter Login	0.15	0.59	2.50	0.26	0.44	2.59	0.84	0.77	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.09	0.22	0.23	0.43	0.20	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.05	0.18	0.10	0.40	0.15	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.43	0.62	1.09	0.37	0.67	Falhou

Usuários = 900

**Associar/Desassociar Ordem aos Projetos**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.16	0.44	6.77	0.74	1.45	Falhou
	Submeter Login	0.15	0.59	2.50	0.5	0.78	2.64	0.84	1.04	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.19	0.40	1.57	1.45	0.56	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.3	0.32	0.74	0.42	0.39	Passou
	Ordem de Associação	0.70	0.50	1.00	0.1	0.30	1.19	0.62	0.42	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.6	3.11	7.09	3.62	3.36	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.14	0.39	1.58	0.23	0.55	Falhou
	Informar Ordem	0.03	0.08	0.35	0.2	0.23	0.73	0.12	0.31	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.15	0.28	0.48	0.20	0.29	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.4	0.25	0.37	0.11	0.30	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	14.13	21.06	11.15	14.61	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.47	0.79	1.38	0.62	0.84	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.29	12.09	22.09	12.56	11.79	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.2	0.20	2.59	0.37	0.60	Falhou

**Informar Horas Trabalhadas**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.18	0.60	6.09	0.74	1.45	Falhou
	Submeter Login	0.15	0.59	2.50	0.20	1.13	1.44	0.84	1.03	Falhou
<b>Informar Horas Trabalhadas</b>	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.27	0.64	3.74	2.40	1.10	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.09	0.54	0.86	0.20	0.52	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.12	0.72	1.73	0.99	0.79	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.16	0.25	1.49	0.37	0.44	Falhou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.08	1.33	4.07	0.74	1.58	Falhou
	Submeter Login	0.15	0.59	2.50	0.19	0.34	1.92	0.84	0.58	Passou
<b>Revisar Documentos</b>	Aprovar Documento	1.65	1.72	2.05	0.13	0.27	0.37	1.76	0.26	Passou
	Aprovar Documento_SA	0.50	1.00	1.50	0.56	1.46	1.97	1.00	1.40	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.91	1.68	2.16	0.37	1.63	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.23	2.10	5.27	0.74	2.32	Falhou
	Submeter Login	0.15	0.59	2.50	0.26	0.77	2.36	0.84	0.95	Falhou
<b>Aprovar Recursos Humanos</b>	Aprovar Recurso	0.51	1.65	2.25	0.40	0.65	2.19	1.56	0.87	Passou
	Aprovar Recurso_PM	0.28	0.35	0.68	0.16	0.41	1.38	0.39	0.53	Falhou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.27	0.48	2.92	0.37	0.85	Falhou

### Aprovar Custos do Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.17	1.45	4.05	0.74	1.67	Falhou
	Submeter Login	0.15	0.59	2.50	0.22	0.32	2.92	0.84	0.74	Passou
<b>Aprovar Custos do Projeto</b>	Aprovar Custos	0.50	1.20	2.30	0.15	0.35	2.10	1.27	0.61	Passou
	Aprovar Custos_PM	0.45	0.75	1.90	0.08	0.21	0.55	0.89	0.25	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.11	0.35	1.39	0.37	0.48	Falhou

### Aprovar Projeto

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.14	0.30	2.89	0.74	0.71	Passou
	Submeter Login	0.15	0.59	2.50	0.27	0.45	2.59	0.84	0.78	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.09	0.23	0.24	0.43	0.21	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.07	0.19	0.10	0.40	0.16	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.44	0.62	1.09	0.37	0.67	Falhou

Usuários = 1000

### Associar/Desassociar Ordem aos Projetos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.18	0.45	6.81	0.74	1.47	Falhou
	Submeter Login	0.15	0.59	2.50	0.5	0.78	2.65	0.84	1.05	Falhou
<b>Associar ordem ao projeto</b>	Relatórios	0.20	1.50	2.50	0.19	0.43	1.57	1.45	0.58	Passou
	Expandir Relatórios	0.50	0.35	0.60	0.3	0.33	0.75	0.42	0.40	Passou
	Ordem de Associação	0.70	0.50	1.00	0.1	0.35	1.21	0.62	0.45	Passou
	Pesquisar Ordem	0.70	3.50	7.00	0.63	3.14	7.09	3.62	3.38	Passou
	Selecionar Ordem	0.05	0.20	0.53	0.15	0.41	1.59	0.23	0.56	Falhou
	Informar Ordem	0.03	0.08	0.35	0.2	0.24	0.74	0.12	0.32	Falhou
	Submeter Ordem de Associação	0.01	0.20	0.40	0.16	0.29	0.49	0.20	0.30	Falhou
<b>Desassociar ordem ao projeto</b>	Relatórios	0.05	0.10	0.20	0.5	0.26	0.37	0.11	0.32	Falhou
	Expandir Relatórios	5.00	11.00	17.87	10.07	14.15	21.09	11.15	14.63	Falhou
	Desassociar Ordem	0.40	0.60	0.89	0.48	0.80	1.39	0.62	0.85	Falhou
	Submeter Ordem desassociada	2.60	12.50	22.75	0.29	12.12	22.09	12.56	11.81	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.2	0.20	2.59	0.37	0.60	Falhou



### Informar Horas Trabalhadas

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.18	0.63	6.09	0.74	1.47	Falhou
	Submeter Login	0.15	0.59	2.50	0.23	1.14	1.45	0.84	1.04	Falhou
Informar Horas Trabalhadas	Acessar Horas Trabalhadas	0.50	2.50	3.89	0.28	0.67	3.74	2.40	1.12	Passou
	Folha Ponto Projeto	0.00	0.20	0.40	0.09	0.56	0.88	0.20	0.54	Falhou
	Folha Ponto Tarefa	0.65	0.99	1.30	0.13	0.73	1.73	0.99	0.80	Passou
Logout	Logout	0.00	0.03	2.10	0.16	0.26	1.52	0.37	0.45	Falhou

### Revisar Documentos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.08	1.35	4.08	0.74	1.59	Falhou
	Submeter Login	0.15	0.59	2.50	0.21	0.36	1.92	0.84	0.60	Passou
Revisar Documentos	Aprovar Documento	1.65	1.72	2.05	0.13	0.27	0.37	1.76	0.26	Passou
	Aprovar Documento_SA	0.50	1.00	1.50	0.57	2.00	1.99	1.00	1.76	Falhou
Logout	Logout	0.00	0.03	2.10	0.91	1.69	2.17	0.37	1.64	Falhou

### Aprovar Recursos Humanos

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
Login	Carregar Página de Login	0.05	0.20	3.60	0.24	2.10	5.27	0.74	2.32	Falhou
	Submeter Login	0.15	0.59	2.50	0.26	0.79	2.37	0.84	0.97	Falhou
Aprovar Recursos Humanos	Aprovar Recurso	0.51	1.65	2.25	0.40	0.67	2.19	1.56	0.88	Passou
	Aprovar Recurso_PM	0.28	0.35	0.68	0.16	0.43	1.38	0.39	0.54	Falhou
Logout	Logout	0.00	0.03	2.10	0.27	0.48	2.92	0.37	0.85	Falhou

**Aprovar Custos do Projeto**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.17	1.45	4.05	0.74	1.67	Falhou
	Submeter Login	0.15	0.59	2.50	0.22	0.33	2.92	0.84	0.74	Passou
<b>Aprovar Custos do Projeto</b>	Aprovar Custos	0.50	1.20	2.30	0.15	0.36	2.10	1.27	0.62	Passou
	Aprovar Custos_PM	0.45	0.75	1.90	0.08	0.23	0.55	0.89	0.26	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.11	0.35	1.39	0.37	0.48	Falhou

**Aprovar Projeto**

Atividade	Transação	Estimado			Executado			SLA	Tempo de Resposta	Status
		MínimoE	MédioE	MáximoE	Mínimo	Médio	Máximo			
<b>Login</b>	Carregar Página de Login	0.05	0.20	3.60	0.14	0.32	2.89	0.74	0.72	Passou
	Submeter Login	0.15	0.59	2.50	0.27	0.47	2.59	0.84	0.79	Passou
<b>Aprovar Projeto</b>	Aprovar Projeto	0.25	0.45	0.55	0.09	0.34	0.24	0.43	0.28	Passou
	Aprovar Projeto_Cliente	0.20	0.40	0.60	0.07	0.20	0.10	0.40	0.16	Passou
<b>Logout</b>	Logout	0.00	0.03	2.10	0.44	0.62	1.09	0.37	0.67	Falhou