

FERRAMENTA WEB PARA SUPORTE À ANÁLISE POR PONTOS DE TESTE

Acadêmico: João Ricardo Rodrigues

Orientador: Everaldo Artur Grahl

Roteiro da apresentação

- Introdução
- Objetivos do trabalho
- Análise por pontos de teste
- Trabalhos correlatos
- Requisitos
- Especificação
- Operacionalidade da implementação (estudo de caso)
- Resultados obtidos
- Conclusão

Introdução

- Projetos de software devem ser muito bem planejados
- Estimativas e medições podem ajudar no planejamento
- FPA (Function Point Analysis): técnica para estimar o tempo necessário para o desenvolvimento de um projeto
- TPA (Test Point Analysis): técnica para estimar o tempo necessário para a fase de testes de um projeto
- Baseia-se em características do projeto e da equipe de testes
- Pouca disponibilidade de ferramentas com suporte a TPA
- Pouco material disponível sobre este assunto

Objetivos do trabalho

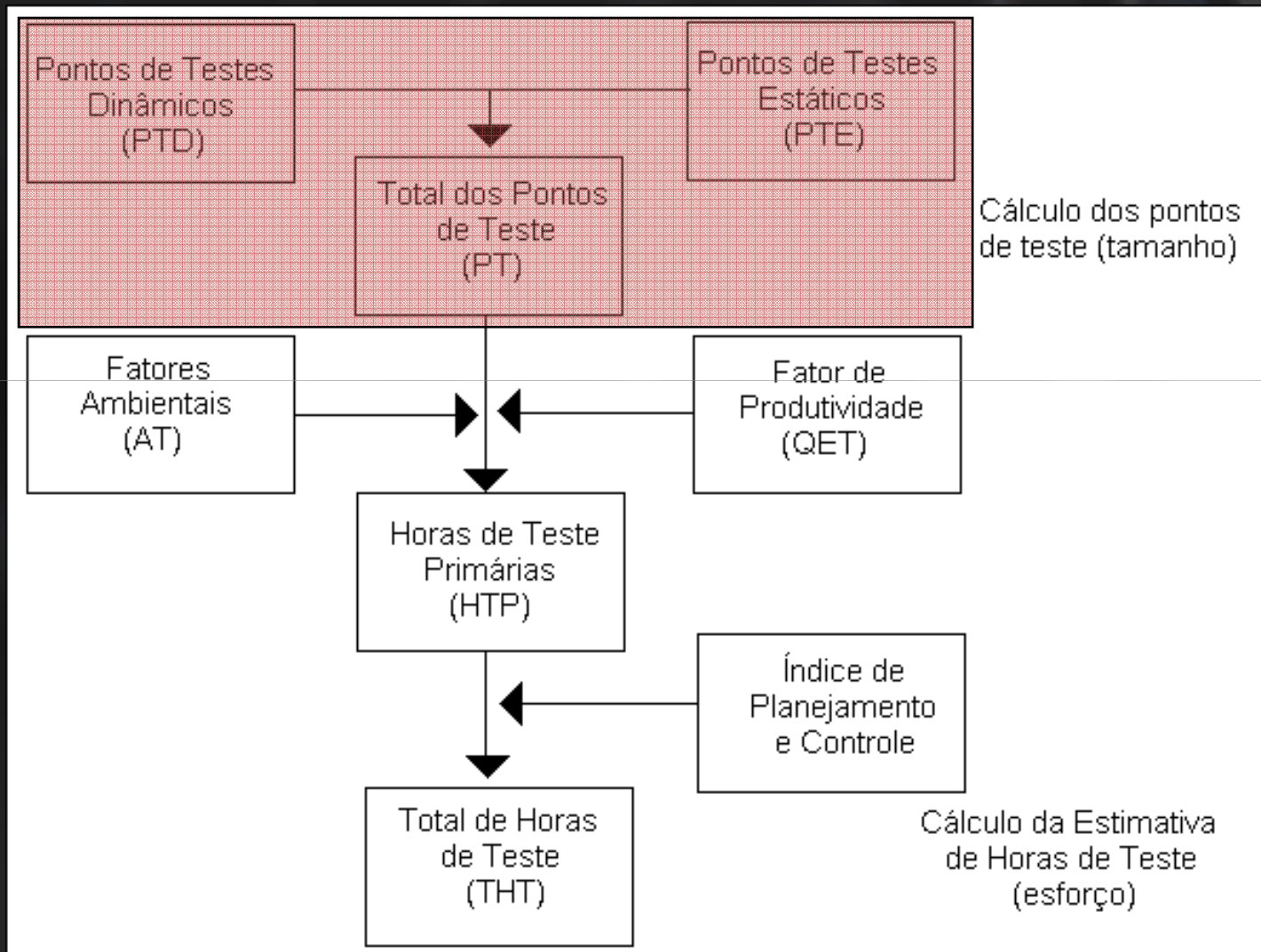
- Principal: desenvolver um protótipo de ferramenta web com suporte a técnica de análise por pontos de teste (TPA)
- Específico: aplicar a teoria de TPA em um caso real de uma software house, a fim de avaliar sua precisão

Análise por pontos de teste

Análise por pontos de teste

- Técnica para estimar o tempo exigido para a fase de testes de um projeto ou sistema
- Pode ser aplicada ao projeto como um todo ou apenas a algumas funções dele
- Pré-requisito: pontos de função (FPA) das funções a serem testadas.
- Valor mínimo para pontos de função: 500 pontos
- Considerar o histórico da equipe de testes

Análise por pontos de teste: visão geral



Cálculo dos pontos de teste

- Divide-se em pontos de testes dinâmicos e pontos de testes estáticos
- Pontos de testes dinâmicos baseiam-se principalmente nas funções do projeto, de maneira isolada
- Pontos de testes estáticos baseiam-se no projeto como um todo

Dados necessários para o cálculo (parte 1)

- Exige-se, de cada função do projeto:
 - Pontos de função (FPA)
 - Avaliação da importância do usuário
 - Avaliação da intensidade de uso
 - Avaliação da interface
 - Avaliação da complexidade
 - Avaliação da uniformidade

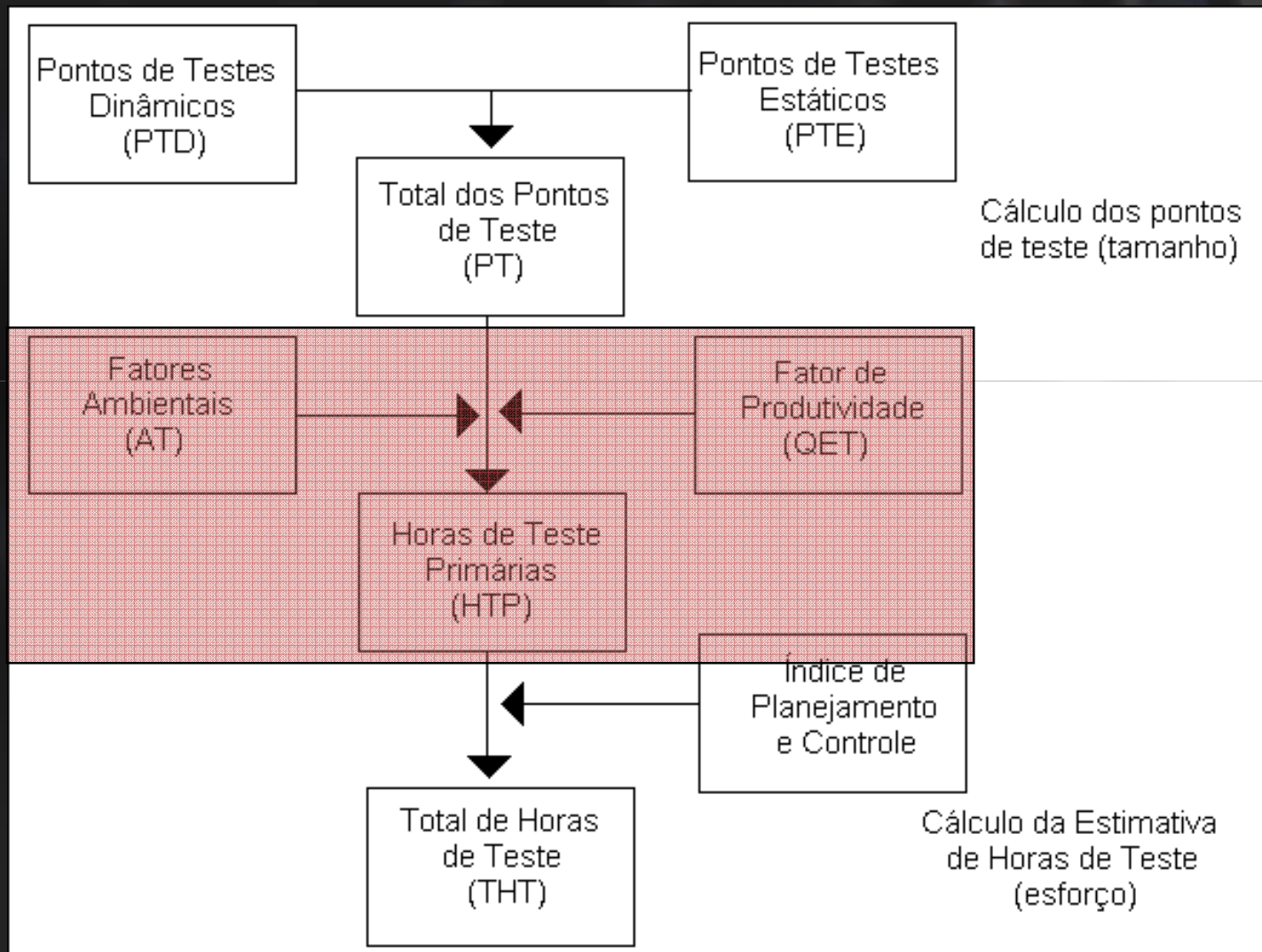
Estes dados constituem parte
dos pontos de teste dinâmicos.

Dados necessários para o cálculo (parte 2)

- Características a serem avaliadas:
 - Funcionalidade
 - Performance
 - Segurança
 - Aderência e efetividade
- Exige-se, de cada uma destas características:
 - Importância dos requisitos
 - Quantas delas possuem coleta de dados para medições
 - Quantas delas são avaliadas com checklists

Estes dados constituem parte dos pontos de testes dinâmicos e os pontos de testes estáticos.

Análise por pontos de teste: visão geral



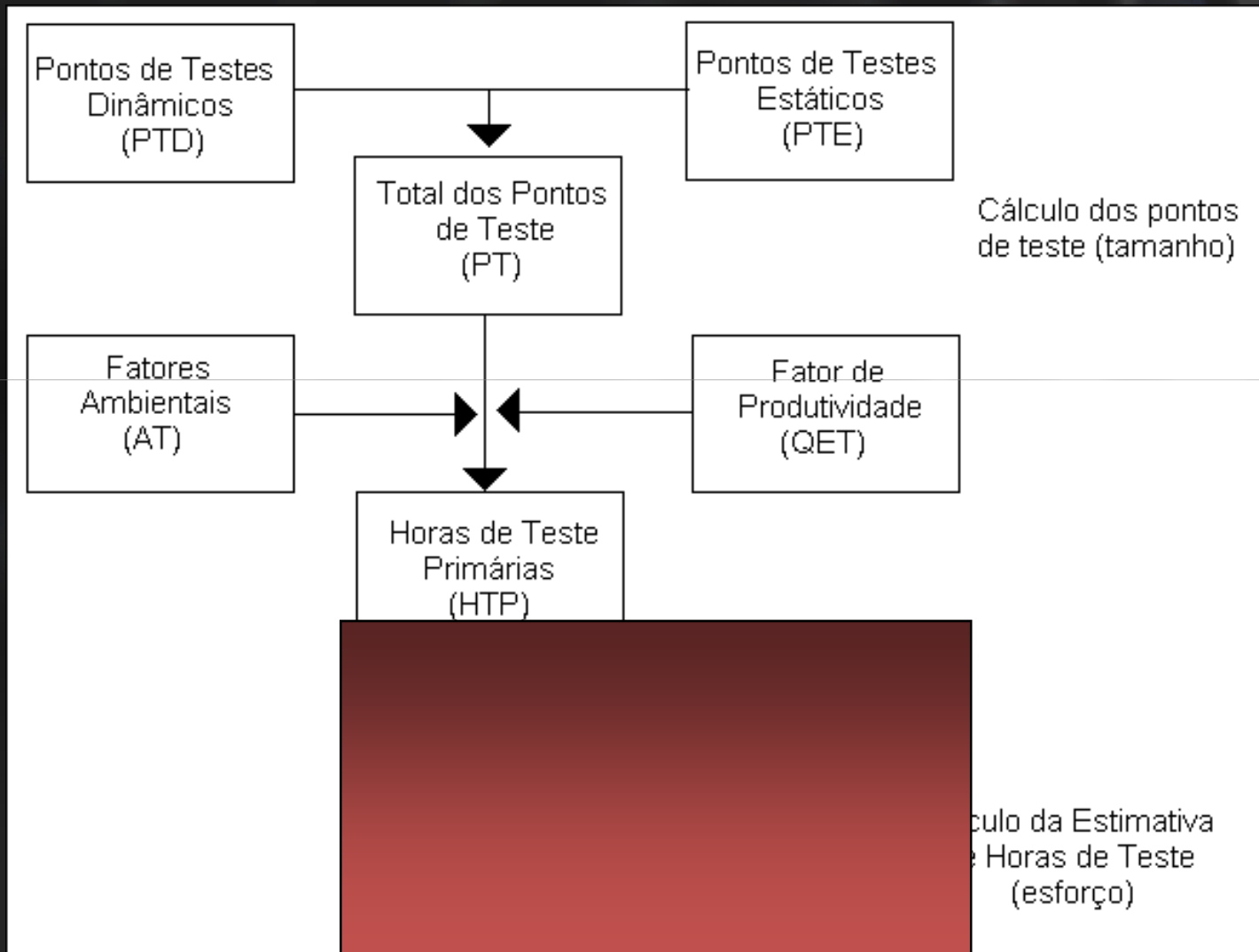
Cálculo das horas de teste primárias

- O tempo de teste do projeto (em horas) pode ser estimado após a obtenção de seus pontos de teste
- Multiplica-se a quantidade de pontos de teste por um valor de ajuste
- Este valor se divide em dois itens:
 - Fator de produtividade
 - Fatores ambientais

Fator de ajuste para horas de teste primárias

- Fator de produtividade: representa a qualificação da equipe de teste
- Fatores ambientais: avaliação dos recursos da equipe de teste
 - Disponibilidade de ferramentas de teste automatizado
 - Disponibilidade de um plano de testes de precedência
 - Disponibilidade de um padrão de documentação de testes
 - Geração da linguagem utilizada no desenvolvimento
 - Histórico de utilização do ambiente de testes
 - Disponibilidade de testware (material de testes)

Análise por pontos de teste: visão geral



Cálculo das horas de teste totais

- Após a obtenção das horas de teste primárias, pode-se estimar o tempo total de testes
- Multiplica-se a quantidade de horas de teste primárias por um índice de planejamento e controle
- Consiste na avaliação de dois itens a respeito da equipe de teste
 - Número de técnicos presentes na equipe
 - Disponibilidade de ferramentas de gerenciamento

Com a obtenção deste índice, pode-se calcular a quantidade total de horas de teste do projeto.



Trabalhos correlatos

Trabalhos correlatos

SISTEMA	CE - F	CE - P	CE - S	CE - E	CI
▶ Radar Empresarial	6	5	6	5	0
▶ Sistema Médio	6	4	4	4	0

Principal | CE - Funcionalidade | CE - Performance | CE - Segurança | CE - Efetividade | Implícitas

Sistema
Radar Empresarial

IMPORTANTE

As características de qualidade dinâmica (QRd) medem como a qualidade dos requisitos pode afetar a qualidade dos testes. Para esta medição são consideradas medidas explícitas e medidas implícitas. Portanto responda a todas as questões que estão nas próximas páginas.

Incluir Alterar Excluir Fechar

Test Manager

- Ferramenta desktop baseada em TPA
- Pode-se cadastrar e armazenar os dados
- Suporta emissão de relatórios

Trabalhos correlatos

Testimation :: Step Three

1 2 3 4 5

<< Back Start Over Help Next >>

Incidents	
33	% Test Case Failure Rate
1.00	Average Daily Incident Removal Rate

Development	
1.00	Avg. Design Time per Test Requirement (hrs)
1.00	Avg. Development Time per Test Case (hrs)

Execution	
1.00	Avg. Test Set Up Time per Test Requirement (hrs)
0.50	Avg. Execution Time per Test Case (hrs)
0.25	Avg. Execution Overhead per Test Case (hrs)

Regression	
10	% Avg. Test Case Rework per Cycle
50	% Avg. Test Case Retest per Cycle

Productivity	
6.50	Avg. Productive Testing Time per Day (hrs)
5.00	No. Working Days per Week
7.50	No. Hours per Working Day (hrs)
0	% Test Analyst Diminishing Return Overhead
95	% Test Analyst Complement
0	% General Contingency Factor

Costs	
\$ 75.00	Avg. Cost per Test Analyst (/hr)
\$ 100.00	Avg. Cost Overhead (/hr)

Concluído Internet 100%

Testimation

- Ferramenta web baseada em uma métrica própria
- Os dados informados são descartados
- Suporta emissão de relatórios

Trabalhos correlatos: aspectos não explorados

- Ferramenta web com suporte à TPA
- Suporte à emissão de gráficos

Desenvolvimento do protótipo

Requisitos principais

- Permitir ao usuário cadastrar projetos e suas funções
- Ter uma interface para que o usuário possa informar os dados exigidos para o cálculo dos pontos e horas de teste
- Calcular os pontos e horas de teste baseado nos valores cadastrados no sistema e informados pelo usuário
- Permitir a emissão de relatórios contendo os resultados
- Permitir a emissão de gráficos contendo os resultados
- Realizar controle de autenticação de usuários
- Mostrar uma descrição dos campos a serem preenchidos fornecendo informações sobre os mesmos para o usuário
- Possuir uma documentação de ajuda sobre a TPA

Especificação: ferramentas utilizadas

- Elaboração dos diagramas: Enterprise Architect
- Elaboração do MER: Power Designer

Diagrama de casos de uso

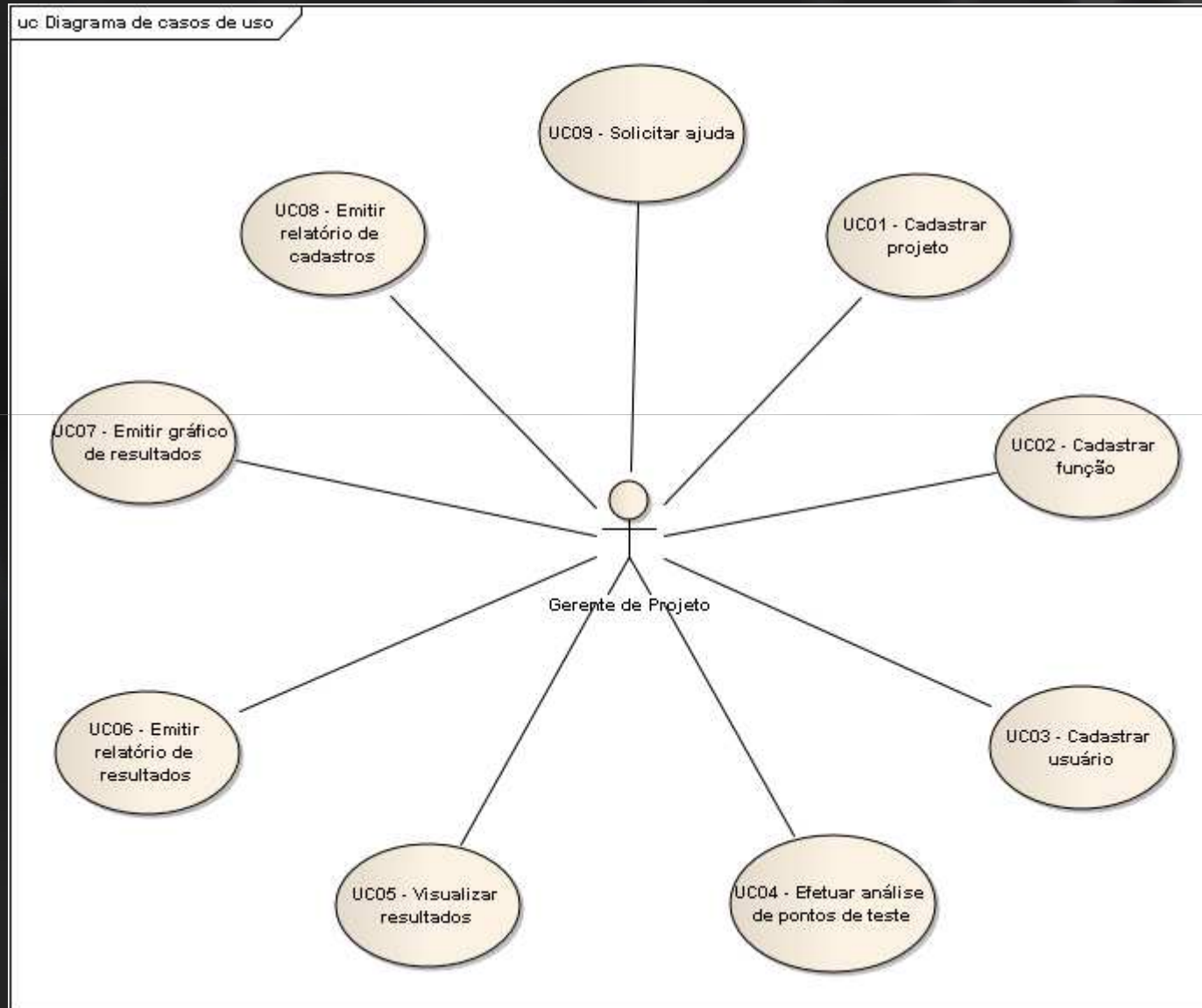


Diagrama de classes

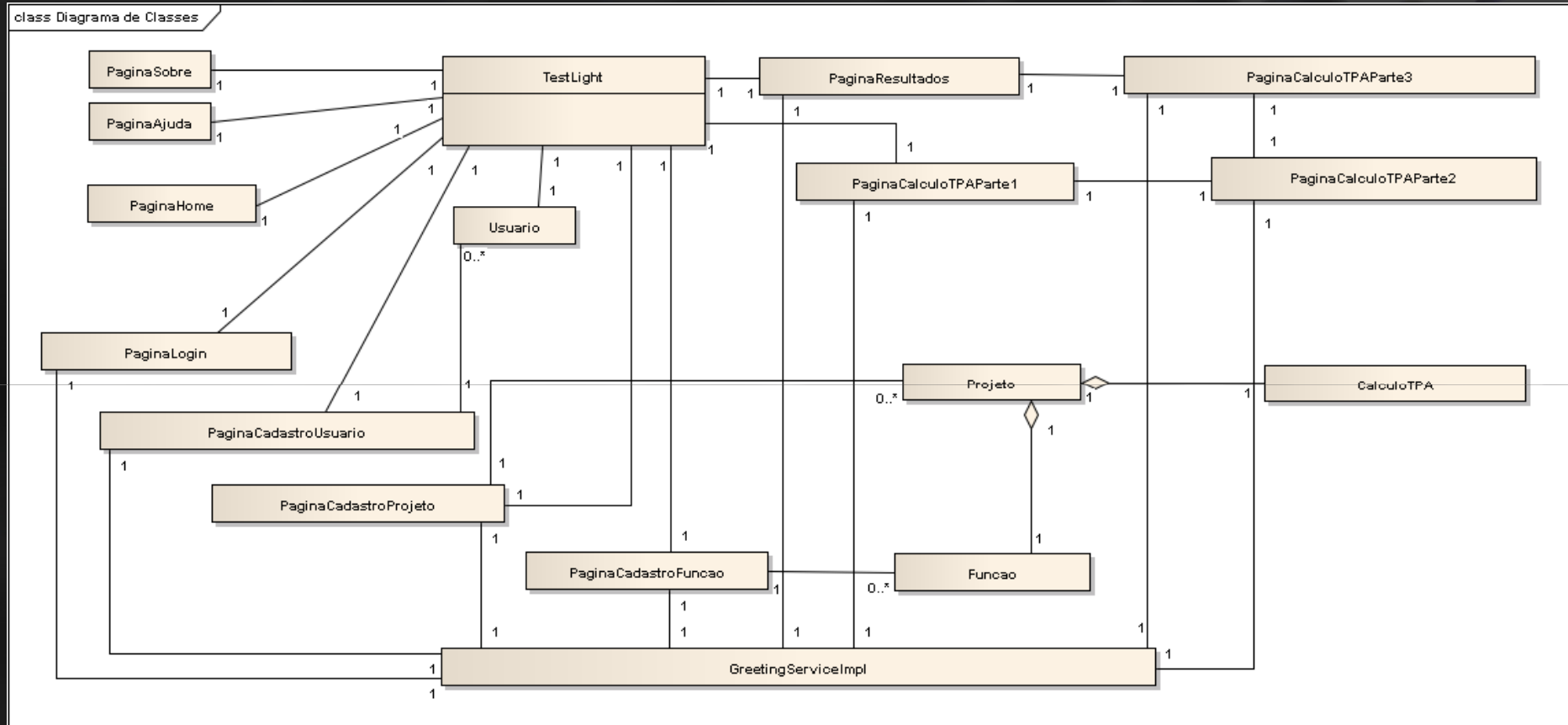
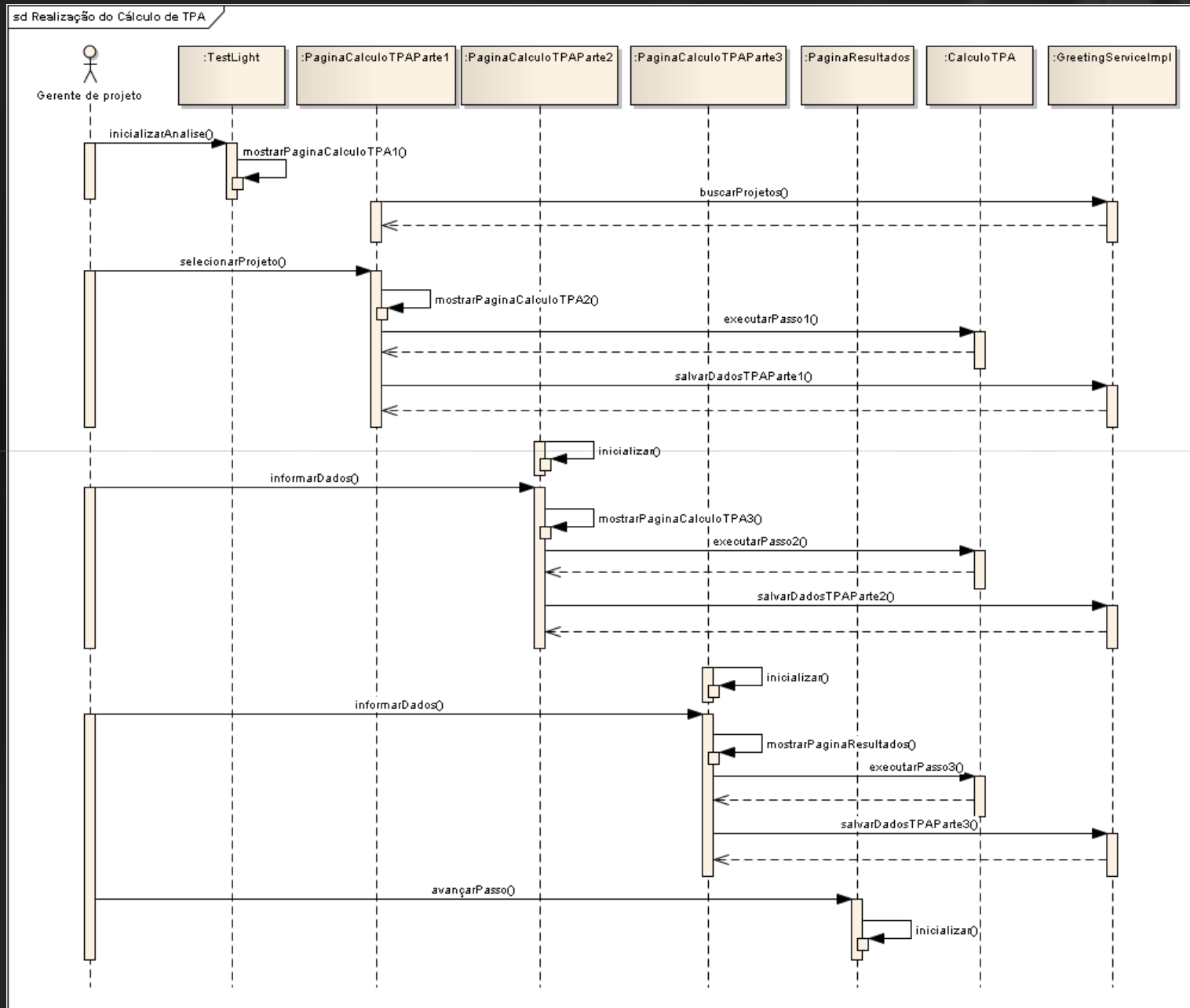
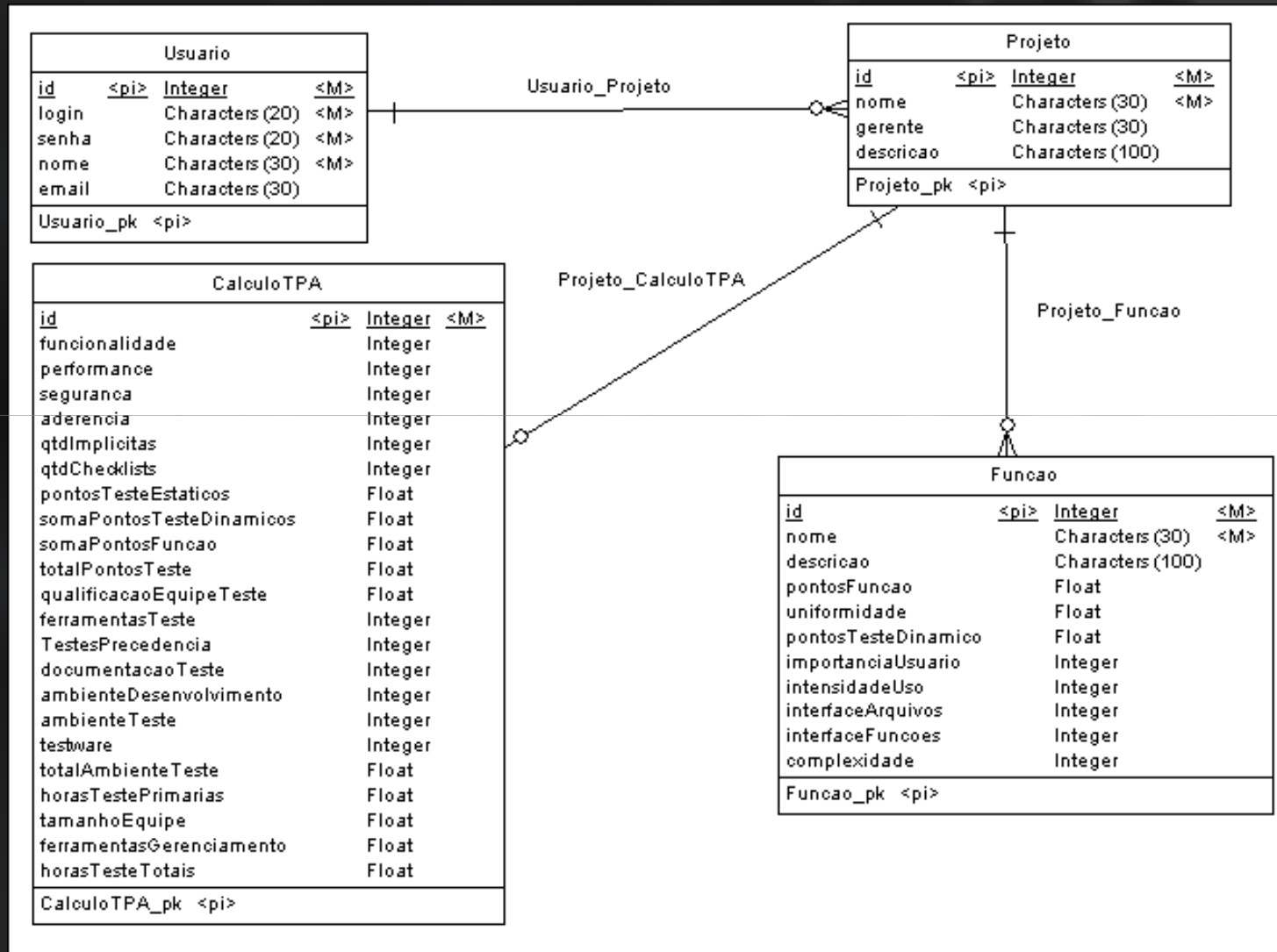


Diagrama de seqüência



Modelo de entidade e relacionamento



Implementação: ferramentas utilizadas

- Linguagem de programação: Java
- Framework auxiliar: Google Web Toolkit
- Componente para geração dos relatórios e gráficos: Jasper Reports
- Ambiente de desenvolvimento: Eclipse
- Banco de dados: MySQL
- Servidor para simulação de ambiente cliente-servidor: Apache Tomcat

Trecho de código

- Função executarPasso2 da classe CalculoTPA, que calcula o total de pontos de teste.

```
public ArrayList<Double> executarPasso2(int funcionalidade, int performance, int segurancia, int aderencia,
                                       int implicitas, int qtdChecklists, ArrayList<Funcao> funcoes) {

    this.funcionalidade = funcionalidade;
    this.performance = performance;
    this.segurancia = segurancia;
    this.aderencia = aderencia;
    this.qtdImplicitas = implicitas;
    this.qtdChecklists = qtdChecklists;

    double caracteristicasExplicitas = acharCaracteristicasExplicitas();
    double qualidadeDinamica = caracteristicasExplicitas + (qtdImplicitas * 0.02);

    Iterator<Funcao> iterator = funcoes.iterator();
    Funcao atual = null;
    somaPontosTesteDinamicos = 0;

    while (iterator.hasNext()) {
        atual = iterator.next();
        somaPontosTesteDinamicos += atual.getPontosTesteDinamico(qualidadeDinamica);
    }

    pontosTesteEstaticos = this.qtdChecklists * 16;

    if (somaPontosFuncao >= 500) {
        totalPontosTeste = somaPontosTesteDinamicos + ((somaPontosFuncao * pontosTesteEstaticos) / 500);
    } else {
        totalPontosTeste = somaPontosTesteDinamicos + ((500 * pontosTesteEstaticos) / 500);
    }

    ArrayList<Double> retorno = new ArrayList<Double>();
    retorno.add(pontosTesteEstaticos);
    retorno.add(somaPontosTesteDinamicos);
    retorno.add(totalPontosTeste);

    return retorno;
}
```


Trecho de código

- Exemplos de utilização do Google Web Toolkit

```
public class PaginaCalculoTPAParte1 extends Composite {  
    private DialogBox dialog = new DialogBox();  
    private Label lblExplicacao = new Label();  
    private Label lblCadastrarProjeto = new Label();  
    private Label lblInformarProjeto = new Label();  
    private Label lblInformarProjeto2 = new Label();  
    private Label lblPassos = new Label();  
    private Label lblCadastradas = new Label();  
    private ListBox boxProjetos = new ListBox();  
    private Button btnCadastrarProjeto = new Button();  
    private Button btnProximoPasso = new Button();  
    private Button btnDialogOk = new Button();  
    private VerticalPanel pnlCadastroProjeto = new VerticalPanel();  
    private VerticalPanel pnlSelecaoProjeto = new VerticalPanel();  
    private VerticalPanel pnlCadastradas = new VerticalPanel();  
    private HorizontalPanel pnlBotoes = new HorizontalPanel();  
    private FlexTable tblSelecaoProjeto = new FlexTable();  
    private FlexTable tblCadastradas = new FlexTable();  
    private ArrayList<Projeto> projetos = new ArrayList<Projeto>();  
    private ArrayList<Funcao> funcoes = new ArrayList<Funcao>();
```

```
pnlCadastroProjeto.add(lblCadastrarProjeto);  
pnlCadastroProjeto.add(btnCadastrarProjeto);  
tblSelecaoProjeto.setWidget(0, 0, lblInformarProjeto2);  
tblSelecaoProjeto.setWidget(0, 1, boxProjetos);  
pnlSelecaoProjeto.add(lblInformarProjeto);  
pnlSelecaoProjeto.add(tblSelecaoProjeto);  
pnlCadastradas.add(lblCadastradas);  
pnlCadastradas.add(tblCadastradas);  
pnlBotoes.add(lblPassos);  
pnlBotoes.add(btnProximoPasso);  
pnlConteudo.add(lblExplicacao);  
pnlConteudo.add(pnlCadastroProjeto);  
pnlConteudo.add(pnlSelecaoProjeto);  
pnlConteudo.add(pnlBotoes);  
pnlConteudo.add(pnlCadastradas);  
pnlConteudo.setSpacing(20);  
lblPassos.setStyleName("passo");
```

Operacionalidade da implementação (estudo de caso)

Operacionalidade da implementação

- Estudo de caso: exemplo prático de utilização da TPA
- Projeto analisado: Projeto Dubai
- Software house: Senior Sistemas S/A
- Objetivo do projeto: integração do sistema Ronda Acesso e Segurança com um novo dispositivo

Ronda Acesso e Segurança

- Sistema integrado de segurança desenvolvido pela Senior Sistemas
- Alguns recursos:
 - Controle de acesso de pessoas
 - Controle de acesso de veículos
 - Monitoramento de alarmes
 - Gerenciamento de políticas de acesso
 - Monitoramento através de circuito fechado de televisão
- Integração com dispositivos de hardware (de leitoras de crachá a catracas) para controle de acesso

Ronda Acesso e Segurança



Projeto Dubai

- Recurso analisado: validação do acesso de pessoas
- Funções analisadas:
 - Validar acesso de pessoas
 - Validar acesso no dispositivo
 - Validar acesso no gerenciador de dados
- Colaboradores consultados para obtenção dos dados:
 - Gerente de projeto
 - Analista de sistemas
 - Analista de qualidade
 - Arquiteto de software
 - 2 programadores

TestLight: cadastro de funções

TestLight

Login	Registre-se	Sair	Bem-vindo(a), João Ricardo!				
Página inicial	Cadastrar projeto	Cadastrar função	Calcular horas de teste	Visualizar resultados	Emitir documentos	Ajuda	Sobre o TestL

Efetue através do formulário abaixo o cadastro das funções do seu projeto. Estas informações serão utilizadas para o cálculo dos pontos de teste do seu projeto. Para maiores informações sobre o que significa cada um destes valores e que valor deve ser usado, basta clicar sobre o campo.

Cada função deve estar associada a um projeto. Selecione na lista abaixo o projeto que contém esta função. Se você ainda não cadastrou seu projeto, utilize o botão "Cadastrar projeto".

Projeto:

Nome:	<input type="text" value="Validar acesso de pessoas"/>
Descrição:	<input type="text" value="Validar acesso de pessoa"/>
FPA:	<input type="text" value="12.21"/>
Imp. Usuário:	<input type="text" value="12"/>
Int. Uso:	<input type="text" value="8"/>
Interface (arquivos):	<input type="text" value="4"/>
Interface (funções):	<input type="text" value="2"/>
Complexidade:	<input type="text" value="12"/>
Uniformidade:	<input type="text" value="1.0"/>

Informe neste campo o nome desta função. Este nome será usado para identificar a função dentro do projeto.

Lista de funções deste projeto:

NOME	DESCRIÇÃO	P. FUNÇÃO	IMP. USUÁRIO	INT. USO	INTERFACE (ARQ.)	INTERFACE (FUNC.)	COMPLEXIDADE	UNIFORMIDADE	EDITAR	EXCLUIR
Validar acesso	Validar acesso de	12.21	12	8	4	2	12	1.0	<input type="button" value="Editar"/>	<input type="button" value="Excluir"/>

TestLight: início do cálculo

TestLight

[Login](#)[Registre-se](#)[Sair](#)

Bem-vindo(a), João Ricardo!

[Página inicial](#)[Cadastrar projeto](#)[Cadastrar função](#)[Calcular horas de teste](#)[Visualizar resultados](#)[Emitir documentos](#)[Ajuda](#)[Sobre o TestLight](#)

Para realizar a análise por pontos de teste do seu projeto, siga os passos descritos abaixo.

Para iniciar a análise, é preciso que você tenha cadastrado seu projeto, assim como as funções do mesmo que serão analisadas. Ao cadastrar uma função você irá informar dados necessários para o cálculo, bem como a quantidade de pontos de função (FPA) e outros conceitos. Caso você não tenha cadastrado seu projeto, clique no botão abaixo, onde também pode se obter mais informações.

[Cadastrar projeto](#)

Selecione o projeto para o qual você deseja efetuar a análise por pontos de teste e em seguida clique no botão "Próximo Passo". Os dados cadastrados para o projeto e para suas funções serão mostrados abaixo.

Projeto: **Passo 1/4** [Próximo passo](#)

NOME	DESCRIÇÃO	P. FUNÇÃO	IMP. USUÁRIO	INT. USO	INTERFACE (ARQ.)	INTERFACE (FUNC.)	COMPLEXIDADE	UNIFORMIDADE
Validar acesso de pessoas	Validar acesso de pessoas	12.21	12	8	4	2	1.0	12
Validar acesso no gerenciador de dados	Validar acesso no gerenciador de dados	12.21	12	8	4	2	1.0	12
Validar acesso no dispositivo	Validar acesso no dispositivo	119.88	12	8	4	2	1.0	12

TestLight: cálculo das horas de teste

TestLight

Login	Registre-se	Sair	Bem-vindo(a), João Ricardo!				
Página inicial	Cadastrar projeto	Cadastrar função	Calcular horas de teste	Visualizar resultados	Emitir documentos	Ajuda	Sobre o TestLi

Agora devem ser informados os dados necessários para o cálculo dos pontos de teste do projeto.

Informe, para cada uma das características abaixo, qual é a importância da qualidade dos seus requisitos para o resultado dos testes. Para maiores informações sobre cada um dos itens, basta clicar sobre o campo correspondente.

Funcionalidade:

Performance:

Segurança:

Aderência e efetividade:

Informe no campo a seguir quantas das características acima (0 a 4) possuem dados ou indicadores coletados para futuras medicoes quanto à qualidade dos testes. Um exemplo deste tipo de dado sao registros de tempo para a realização de testes de performance.

Quantidade de características com dados coletados:

Caso a equipe de qualidade adote processos de revisão de documentacao e de códigos utilizando checklists, informe no campo a seguir quantas das características acima (0 a 4) possuem checklists para a sua avaliação.

Quantidade de checklists:

Passo 2/4

Informe neste campo, de acordo com os valores abaixo, qual a importância dos requisitos de acordo com a funcionalidade do sistema.

- 0 - A qualidade dos requisitos nao é importante para o resultado dos testes.
- 3 - A qualidade dos requisitos nao é importante, mas precisa ser considerada para o resultado dos testes.
- 4 - A qualidade dos requisitos tem importância média para o resultado dos testes. Normalmente é escolhido este índice.
- 5 - A qualidade dos requisitos é muito importante.
- 6 - A qualidade dos requisitos é extremamente importante.

TestLight: cálculo das horas de teste

Pontos de teste do projeto: 367.0646

Agora que foram calculados os pontos de teste, será efetuado o cálculo das horas de teste.

Informe abaixo os valores referentes ao ambiente de teste do projeto. Para maiores informações sobre o que significa cada um destes valores e que valor deve ser usado, basta clicar sobre o campo.

Ferramentas de teste:	<input type="text" value="1"/>
Testes de precedência:	<input type="text" value="4"/>
Documentação de teste:	<input type="text" value="12"/>
Ambiente de desenvolvimento:	<input type="text" value="8"/>
Ambiente de teste	<input type="text" value="4"/>
Testware	<input type="text" value="2"/>

Informe abaixo um valor de 0.7 a 2.0 referente a qualificação e produtividade da equipe de teste. Este valor deve ser determinado através de uma base histórica sendo que quanto menor o valor, mais experiente é a equipe. O valor 2.0 representa uma equipe completamente inexperiente com os tipos de teste que serão realizados no projeto e deve ser utilizado em caso de dúvida ou incerteza.

Qualificacao da equipe de teste:

Informe abaixo valores referentes às atividades de planejamento e controle. Para maiores informações sobre que valores podem ser informados, o que eles significam e que valor deve ser usado, basta clicar sobre o campo.

Tamanho da equipe:

Ferramentas de gerenciamento:

[Passo anterior](#) **Passo 3/4** [Proximo passo](#)

Informe neste campo um dos valores listados abaixo, de acordo com a descrição que melhor condiz com a situação das ferramentas de teste a serem utilizadas no projeto.

- 1 - Existe uma ferramenta de automação para as fases de especificação e execução dos testes.
- 2 - Existe uma ferramenta de automação para as fases de especificação ou para a fase de execução. (Em caso de dúvida ou incerteza, utilize esta opção.)
- 4 - Não existe ferramenta de automação de teste.

TestLight: visualização de resultados

TestLight

[Login](#)[Registre-se](#)[Sair](#)

Bem-vindo(a), João Ricardo!

[Página inicial](#)[Cadastrar projeto](#)[Cadastrar função](#)[Calcular horas de teste](#)[Visualizar resultados](#)[Emitir documentos](#)[Ajuda](#)[Sobre o TestLight](#)

Baseado nos dados informados, a análise por pontos de teste estimou para o projeto Dubai os seguintes valores:

Total de pontos de teste:367.0646

Total de horas de teste:1024.110234

Este valor já está salvo no seu projeto. Se desejar refazer a análise com valores diferentes, utilize o botão "Passo anterior" para voltar ao passo 3 ou o botão "Cadastrar projeto" para alterar o cadastro do projeto. Você também pode usar o botão "Emitir documentos" para emitir um relatório ou um gráfico contendo os dados da análise.

[Cadastrar projeto](#)[Emitir documentos](#)[Passo anterior](#) **Passo 4/4**

Projeto Dubai: valores obtidos

- Pontos de função
 - Validar acesso de pessoas: 12,21 PF
 - Validar acesso no gerenciador de dados: 12,21 PF
 - Validar acesso no dispositivo: 119,88 PF
- Pontos de testes dinâmicos
 - Validar acesso de pessoas: 28,35 PTD
 - Validar acesso no gerenciador de dados: 28,35 PTD
 - Validar acesso no dispositivo: 278,36 PTD
- Pontos de testes totais: 367,06 pontos de teste
- Horas de teste totais: 1024,05 horas de teste



Resultados obtidos

Resultados obtidos: implementação

- Objetivo de desenvolver uma ferramenta web de suporte à TPA foi atingido
- Requisitos foram atendidos
- Reuniu as principais funcionalidades dos dois trabalhos correlatos analisados
- Possibilidade de emissão de gráficos

CARACTERÍSTICAS	TEST MANAGER	TESTIMATION	TESTLIGHT
realiza estimativas de teste	X	X	X
cálculo de TPA	X		X
gerenciador de sistemas e seus dados	X		X
aplicação web		X	X
resultados das estimativas visíveis em formato de relatório	X	X	X

Resultados obtidos: estudo de caso

- Tempo estimado para os testes: 1024,05 horas
- Tempo real de testes: 209 horas
- Acréscimo considerável em relação ao tempo real de testes
- Segundo o analista de qualidade, o projeto necessitaria de uma quantidade maior de tempo dedicado a testes
- Durante o projeto, não houve tempo hábil para a execução completa da fase de testes
- O tempo estimado pela métrica seria suficiente para a execução de todos os testes necessários

Resultados obtidos: análise da precisão da TPA

- Alguns valores estão sujeitos a subjetividade
- Certos dados e conceitos exigidos não são bem especificados
- Valores informados podem variar dependendo do ponto de vista ou opinião pessoal de quem os fornece

O valor final da estimativa pode variar significativamente se estas possibilidades forem consideradas!

Resultados obtidos: experimentos com o projeto Dubai

- Experimento 1:
 - Alterou-se: importância do usuário, intensidade de uso
 - Horas de teste obtidas: 764,41 horas
- Experimento 2:
 - Alterou-se: importância dos requisitos de funcionalidade, performance, segurança e aderência e efetividade
 - Horas de teste obtidas: 842,92 horas
- Experimento 3:
 - Junção dos experimentos 1 e 2
 - Horas de teste obtidas: 633,49 horas

Resultados obtidos: experimentos com o projeto Dubai

Item analisado	Horas de teste	Variação (em horas)
Valor estimado pela TPA	1024,11	-
Experimento 1 (imp. usuário e int. uso alteradas)	764,41	-259,69
Experimento 2 (qualidade dinâmica alterada)	842,91	-181,19
Experimento 3 (experimento 1 + experimento 2)	633,49	-390,62
Tempo real de testes do projeto Dubai	209,00	-815,11

- Os experimentos representam as possíveis variações de resposta às quais a análise está sujeita
- Experimento 3 se aproximou mais da realidade do que a estimativa original
- A precisão da métrica não se adequou a realidade?
- O tempo de testes do projeto foi insuficiente?

Análise da TPA por outras fontes

- Campos (2009) menciona que manter uma base histórica de dados é fundamental
- Campos e Birnfeld (2010) apontam sobre a aplicação da TPA de acordo com a realidade do projeto
- Sugerem que a técnica de TPA é apenas para projetos grandes (500 PF)
- Lopes e Nelson (2008) estudaram a análise de 6 projetos
- 2 deles obtiveram mais de dez mil pontos de teste
- Meisen (2005) estudou a análise de 2 projetos
- Um deles estimou 23% a mais de tempo; o outro 34% a menos

Análise da TPA: conclusões obtidas

A técnica de análise por pontos de teste pode ser utilizada para a estimativa do tempo de testes de um projeto ou sistema se os itens a serem avaliados forem mais objetivos e não tão influenciados pela opinião pessoal ou pelo modo que cada avaliador vê algumas questões exigidas pela métrica.

Apesar destes detalhes, a técnica se mostra útil para a realização de estimativas, desde que possam ser obtidas as respostas para as perguntas feitas e as classificações para os itens que devem ser avaliados.

Conclusão

- A TPA foi criada para auxiliar as empresas desenvolvedoras de software a estimar o tempo de testes de seus projetos
- Esta estimativa permite um melhor planejamento das fases do projeto como um todo
- Como este planejamento deve considerar diversos fatores, todo auxílio pode ser relevante
- O estudo de caso realizado avaliou hipoteticamente a precisão da técnica
- Os experimentos realizados verificaram a influência da subjetividade de alguns dos itens exigidos
- Há uma escassez de material específico sobre TPA

Conclusão

- O protótipo desenvolvido torna a aplicação prática da TPA mais simples
- Esta técnica demanda tempo, esforço e conhecimento
- Limitação do protótipo: não realiza cálculo de pontos de função (pré-requisito da TPA)
- Sugestão para trabalho futuro: ferramenta que suporta pontos de função e pontos de teste, permitindo a análise completa

Obrigado!