

ANDRÉ T. BAHBOUTH

PROVADOR VIRTUAL

Texto apresentado à Escola Politécnica da Universidade de São Paulo como requisito para a conclusão do curso de graduação em Engenharia Elétrica, junto ao Departamento da Engenharia de Sistemas Eletrônicos (PSI).

São Paulo
2013

ANDRÉ T. BAHBOUTH

PROVADOR VIRTUAL

Texto apresentado à Escola Politécnica da Universidade de São Paulo como requisito para a conclusão do curso de graduação em Engenharia Elétrica, junto ao Departamento da Engenharia de Sistemas Eletrônicos (PSI).

Área de Concentração:

Sistemas Eletrônicos

Orientador:

Hae Yong Kim

São Paulo
2013

FICHA CATALOGRÁFICA

Bahbouth, André Tibiriçá

Provador Virtual / A. T. Bahbouth. – São Paulo, 2013.
90 p.

Monografia (Graduação em Engenharia Elétrica) — Escola
Politécnica da Universidade de São Paulo. Departamento da En-
genharia de Sistemas Eletrônicos (PSI).

1. Visão Computacional 2. Modelagem 3D 3. Antropometria
I. Universidade de São Paulo. Escola Politécnica. Departamento
da Engenharia de Sistemas Eletrônicos (PSI). II. Departamento
de Engenharia de Sistemas Eletrônicos t.

AGRADECIMENTOS

Agradecimentos aos professores Marcos Barretto, Romero Tori e Fátima Nunes e aos alunos Darlan Junior, Tomaz Fogaza, Victor Velloce e Victor Sivieri pela ajuda fornecida no desenvolvimento desse projeto. A Maria Christina Tibiriçá Bahbouth escritora literária e revisora do texto do projeto. Principalmente a meu orientador Hae Yong Kim e a família pelo suporte proporcionado.

RESUMO

Este trabalho desenvolveu um sistema de provador virtual de roupas. A partir de fotos tiradas de uma pessoa com uma câmera digital simples, ou até mesmo com uma webcam, será possível aferir as medidas do usuário. Com estas medidas corporais e determinada a roupa a ser provada pelo cliente, será viável sugerir a ele o tamanho da roupa. Em seguida, o sistema irá gerar um modelo em 3D com proporções antropométricas fiéis às do usuário, permitindo vesti-lo com as roupas por ele escolhidas. Cria-se, assim, um manequim sob medida, que poderá ser usado, por exemplo, por um fabricante de vestuário para criar roupas customizadas, para conhecer as medidas corporais dos seus clientes ou como parâmetro de análise intermediário no processo de fabricação de roupas. As aplicações são diversas, entretanto o principal escopo desta ferramenta é facilitar e propiciar as aquisições de vestuário pela internet, que proporcionará uma experiência nova ao usuário e, também, diminuirá a alta taxa de devolução das roupas adquiridas por tal meio.

SUMÁRIO

Lista de Ilustrações

Lista de Abreviaturas e Siglas

1	Introdução	12
1.1	Objetivos	12
1.2	Motivação	13
1.2.1	Expansão das Vendas Online no Brasil	13
1.2.2	Inovação tecnológica	14
1.3	Justificativa	17
2	Levantamento e Análise de Dados	18
2.1	Processo de Fabricação de Roupas	18
2.1.1	Desenho Estilístico	18
2.1.2	Modelagem da Roupas	19
2.1.3	Produção das Roupas	20
2.2	Análise do Processo de Fabricação	20
3	Alternativas para Solução do Problema	23
3.1	Análise e Processamento de Imagem	24
3.1.1	Tecnologias disponíveis	24

3.1.2	Alternativas de Projeto para a Análise e Processamento de Imagem	24
3.2	Computação Gráfica	25
3.2.1	Tecnologias Disponíveis	26
3.2.2	Alternativas de Projeto para a Computação Gráfica	28
3.2.2.1	Alternativas para a Obtenção de Modelos 3D	29
3.2.2.1.1	Geração de Modelos 3D	29
3.2.2.1.2	Customização de um <i>Avatar</i>	30
3.2.2.1.3	Seleção de um Modelo	30
3.2.2.2	Simulação de Tecido e Montagem da Roupas	30
3.2.2.2.1	Auxílio da Ferramenta <i>Blender</i>	31
3.2.2.2.2	Auxílio da Biblioteca <i>OpenGL</i>	31
3.2.2.3	Exibição dos Resultados	31
3.2.2.3.1	Baixar algumas Imagens do <i>Avatar</i>	32
3.2.2.3.2	Baixar um Vídeo do <i>Avatar</i>	32
3.2.2.3.3	Elaboração de um CAD Online	32
4	Funcionalidades, Requisitos e Restrições do Projeto	34
4.1	Aquisição de Medidas e Sugestão de Tamanho	34
4.2	Criação de um Modelo Tridimensional	35
4.3	Simulação de Tecido e Montagem da Roupas	36
4.4	Exibição dos Resultados	36

5	Metodologia	37
6	Cr�terios de Avalia��o	39
7	Escolha da Solu��o	41
7.1	<i>Camera Module</i>	42
7.2	<i>Measurer Module</i>	43
7.3	<i>Chooser Module</i>	44
7.4	<i>Clothes 2D to 3D Converter</i>	46
7.5	<i>3D Dressing Module</i>	47
7.6	<i>Display Module</i>	48
8	Arquitetura do Projeto	50
9	Plano de Gerencial	54
9.1	Introdu��o	54
9.2	Cronograma	54
9.3	An�lise dos Recursos	59
10	Plano de Testes	64
11	Resultados	71
12	Conclus��es	80
13	Refer�ncias	82

14 Bibliografia	83
Apêndice A - Modelo em Cascata	85
A.1 O Modelo	85
A.2 Uso do Modelo	85
Apêndice B - O Modelo <i>Scrum</i>	88
B.1 O <i>Scrum</i>	88
B.2 <i>Sprint</i> (corrida, tiro)	89

LISTA DE ILUSTRAÇÕES

1	Resultado final: <i>avatar</i> já vestido	13
2	Faturamento <i>e-commerce</i> no Brasil	14
3	Provador virtual de mercado	16
4	Modelagem bidimensional da roupa	19
5	Medidas utilizadas para confecção de camisetas	21
6	Tabela de medidas e tamanhos correspondentes	22
7	Silhueta do membro superior de um corpo humano	25
8	Interface gráfica do <i>Marvelous Designer</i>	27
9	Matriz de decisão: alternativas <i>online</i> e local	41
10	Matriz de decisão: alternativas <i>PHP</i> e <i>OpenCV</i>	44
11	Matriz de decisão: alternativas para obtenção do modelo 3D	45
12	Matriz de decisão: alternativas para montagem da roupa	47
13	Matriz de decisão: alternativas para exibição do modelo	49
14	Diagrama de blocos e fluxos do projeto.	51
15	Cronograma do projeto.	56
16	Duração estipulada para cada módulo	57
17	Desenvolvimento dos módulos fluindo sequencialmente como em uma cascata	58
18	Cronograma da corrida final	59

19	EAP do <i>Measurer Module</i>	61
20	EAP do <i>Camera Module</i> e <i>Display module</i>	62
21	EAP do <i>Chooser Module</i> e <i>Dressing Module</i>	62
22	EAP do <i>Clothes 2D to 3D converter</i>	63
23	Testes aplicados ao <i>Camera Module</i>	64
24	Testes aplicados ao <i>Measurer Module</i>	65
25	Desempenho do <i>Measurer Module</i>	66
26	Ficha cadastral para acurar precisão do sistema	68
27	Testes aplicados ao <i>Display Module</i>	69
28	Testes aplicados ao <i>Chooser Module</i>	70
29	UI do <i>e-commerce</i> do Proveedor Virtual	71
30	UI do <i>e-commerce</i> do Proveedor Virtual (final da página)	72
31	Primeira página do Proveedor Virtual	72
32	Visualização de fotos já carregadas	73
33	Captura de uma nova imagem	73
34	Janela de informações solicitadas ao usuário	74
35	Carregando uma nova imagem	75
36	Navegador de arquivos do usuário	75
37	Navegador de arquivos do usuário	76
38	Resultado das medidas antropométricas	77
39	<i>Avatar</i> magro e alto vestido com a roupa selecionada	78
40	<i>Avatar</i> magro e baixo vestido com a roupa selecionada	78

41	<i>Avatar</i> gordo e alto vestido com a roupa seleccionada	79
42	O Modelo em cascata estático. O andamento do processo flui de cima para baixo, como uma cascata	86
43	Processo de <i>Scrum</i> , um <i>sprint</i>	89

LISTA DE ABREVIATURAS E SIGLAS

USP	Universidade de São Paulo
CAD	Computer-Aided Design
PIL	Python Image Library
ABNT	Associação Brasileira de Normas Técnicas
ISO	International Organization for Standardization
BSD	Berkeley Software Distribution
WBS	Work Breakdown Structure
EAP	Estrutura Analítica de Projeto
OpenCV	Open Source Computer Vision Library
OpenGL	Open Source Graphics Library
GD Library	Graphics Draw Library
PHP	Personal Home Page
WebGL	Web Graphics Library
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
UI	User Interface

1 INTRODUÇÃO

1.1 Objetivos

A partir de fotos tiradas de uma pessoa com uma câmera digital simples, ou até mesmo com uma webcam, a ferramenta desenvolvida permitirá aferir as medidas do usuário. Com duas premissas definidas, quais sejam medidas corporais do usuário e uma roupa por ele escolhida, o sistema sugerirá o tamanho da roupa. Em seguida, o sistema irá gerar um modelo em 3D com proporções antropométricas fiéis às do usuário, permitindo vesti-lo com as roupas por ele escolhidas. Este modelo 3D é comumente denominado *avatar*, designação que será utilizada neste trabalho.

O sistema descrito promove a mesma sensação de um provador de vestuário real, permitindo ao usuário saber seu tamanho certo de roupa e, também, visualizar como ele ficaria vestido com determinada roupa. Destas características do sistema advém seu nome Provador Virtual.

Na figura 1 abaixo, é possível observar-se o resultado final visto pelo cliente.



Figura 1: Resultado final: *avatar* já vestido¹

1.2 Motivação

1.2.1 Expansão das Vendas Online no Brasil

O e-commerce, vendas de artigos pela internet, já se tornou uma realidade na vida dos brasileiros. As estatísticas demonstram que o brasileiro compra cada vez mais pela internet. Neste contexto, diversas empresas investiram nesse mercado e, como resultado, o faturamento anual do e-commerce brasileiro cresce exponencialmente, como observamos na figura 2 abaixo.

¹fonte: marvelousdesigner.com

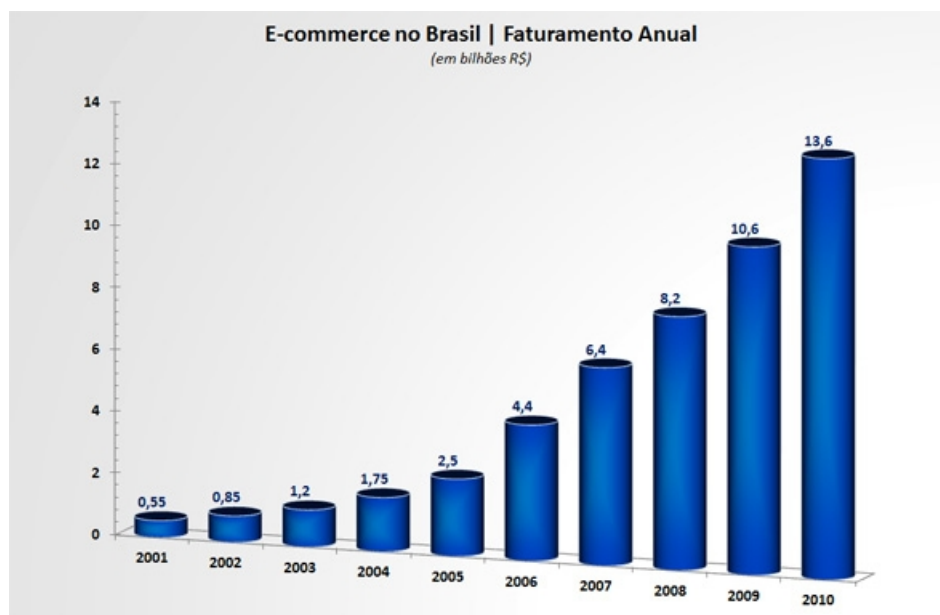


Figura 2: Faturamento *e-commerce* no Brasil ²

Como podemos observar o faturamento do e-commerce no Brasil, em 2001, era de 550 milhões de reais, praticamente desprezível frente a 13,6 bilhões de reais computados no ano de 2010. Trata-se, portanto, de um mercado em forte expansão e que está começando agora no Brasil. Em seguida, demonstraremos como as ferramentas tecnológicas não acompanharam a evolução rápida deste mercado.

1.2.2 Inovação tecnológica

Existem diversos artigos que possibilitam compras rápidas pela internet, pois eles possuem uma alta padronização. Podemos citar como exemplo, eletrodomésticos, automóveis, artigos para cozinha, material de consumo de escritório, etc.

No entanto, alguns produtos requerem uma interação maior com o cliente, pois aspectos visuais e físicos afetam mais a decisão da compra. Um exemplo

²fonte: e-Bit

que se adéqua perfeitamente a este caso são as roupas. Analisando o estado da arte para tecnologias que poderiam auxiliar na compra de vestuário pela internet, verificamos que as tecnologias ainda são muito incipientes.

Várias empresas têm utilizado soluções diferentes para este projeto, geralmente baseadas em realidade aumentada ou em virtualização da realidade.

A solução por realidade aumentada consiste em adicionar elementos virtuais a uma imagem real, ou seja, adicionar um modelo 3D de uma peça de roupa à imagem do cliente em tempo real. O principal problema desta solução é que, geralmente, o usuário está vestido com roupas que cobrem uma porção considerável do seu corpo. Desta forma, ao adicionar novas roupas à imagem não se obtém o efeito desejado.

Outra dificuldade refere-se à tecnologia gráfica atual, que não permite a criação de visuais que se assemelhem suficientemente aos objetos em tempo real. Assim, quando o usuário se movimenta, a posição das roupas sai de sincronia, gerando uma sensação de estranheza.

Provedores virtuais já comercializados na internet que usam a realidade aumentada fazem a sobreposição da roupa na imagem da pessoa capturada pela webcam do computador, como pode ser visto pela figura 3 abaixo.



Figura 3: Provador virtual de mercado³

Analisando a figura 3, percebemos a existência de alguns botões para aumentar e diminuir o tamanho da roupa, bem como, alguns modelos de roupas para serem provados. Este sistema disponível no mercado é extremamente limitado, pois ele não faculta ao usuário saber o tamanho apropriado de sua roupa e nem permite verificar se aquele tipo de roupa é adequado ao seu tipo físico. Neste contexto, o projeto ora apresentado é inovador, na medida em que lança um novo paradigma para provador virtual.

A solução por virtualização da realidade, proposta por este trabalho, consiste em tirar as medidas do usuário, objetivando a geração de um modelo tridimensional, que será vestido virtualmente, criando-se animações mais fiéis e realistas. É importante salientar que outros trabalhos relacionados com a virtualização da realidade utilizam câmeras de profundidade, principalmente o dispositivo *Kinect* [4] criado pela Microsoft. No entanto, neste trabalho como consideramos que, no Brasil, o *Kinect* não é um dispositivo de fácil acesso não o usaremos, pois a sua aplicação teria um alcance limitado.

³fonte: zugara.com

1.3 Justificativa

Os provadores virtuais existentes não são populares. Esta realidade indica a viabilidade de inserção de um novo produto no mercado, que apresente uma utilidade prática, qual seja a possibilidade do usuário verificar se a roupa a ser adquirida está adequada ao seu tipo físico. Ademais, esta ferramenta permitirá o aprimoramento e o aumento das vendas online de roupas, assim como diminuirá a taxa de retorno de peças cujos tamanhos ou corte não estão adequados ao cliente.

Portanto, parece-nos viável a criação de um novo produto que será competitivo, na medida em que atenderá a esta demanda do mercado.

Durante o seu desenvolvimento, o projeto proporcionará oportunidades de estudo de várias áreas interessantes, tais como computação gráfica, processamento, síntese e análise de imagens, arquitetura de sistemas, simulação de física mecânica, tecnologias web, entre outras, o que o torna uma escolha engrandecedora, também do ponto de vista acadêmico.

2 LEVANTAMENTO E ANÁLISE DE DADOS

2.1 Processo de Fabricação de Roupas

Para elaborar um sistema que permitirá ao usuário experimentar as roupas foi fundamental analisar como elas são fabricadas.

A partir de uma visita à fábrica de roupas TDB Têxtil S/A., detentora da marca Tip-Top, foi possível aferir o processo de fabricação das roupas, que está estruturado basicamente em três etapas, quais sejam, desenho estilístico, modelagem e fabricação, a serem detalhadas nos itens seguintes.

2.1.1 Desenho Estilístico

O desenho inicial da roupa é feito por um estilista. Até alguns anos atrás, este desenho era feito manualmente em papel e dependia da arte do estilista em inventar uma roupa nova.

Hoje existem sistemas conhecidos como CADs (do inglês, *Computer Aided Design*), que já possuem desenhos bidimensionais da roupa padrão. Esta ferramenta permite ao estilista modificar a estampa, bordar, verificar o caimento da roupa a fim de chegar à concepção de um novo produto.

Este desenho inicial é um desenho artístico e, portanto, não possui dimensionamento algum. Devemos imaginá-lo como se fosse uma foto de uma roupa pronta. Apesar de não ter dimensionamento o estilista já possui a especificação

dos tipos de tecido, das cores e das estampas que serão utilizados na fabricação da roupa.

2.1.2 Modelagem da Roupa

Na segunda fase da etapa de produção, o desenho será entregue para a equipe de modelagem.

Neste estágio, será gerado um modelo bidimensional da roupa, enfatizando onde será feito o corte no tecido. Esta modelagem também é feita em uma ferramenta CAD, similar àquela a que nos referimos anteriormente. A principal diferença com o desenho da etapa anterior é que este já possui o dimensionamento da roupa, para os diferentes tamanhos. Como a roupa normalmente é simétrica desenha-se no CAD apenas uma metade da roupa. Podemos ver na figura 4 abaixo o resultado final da etapa da modelagem.

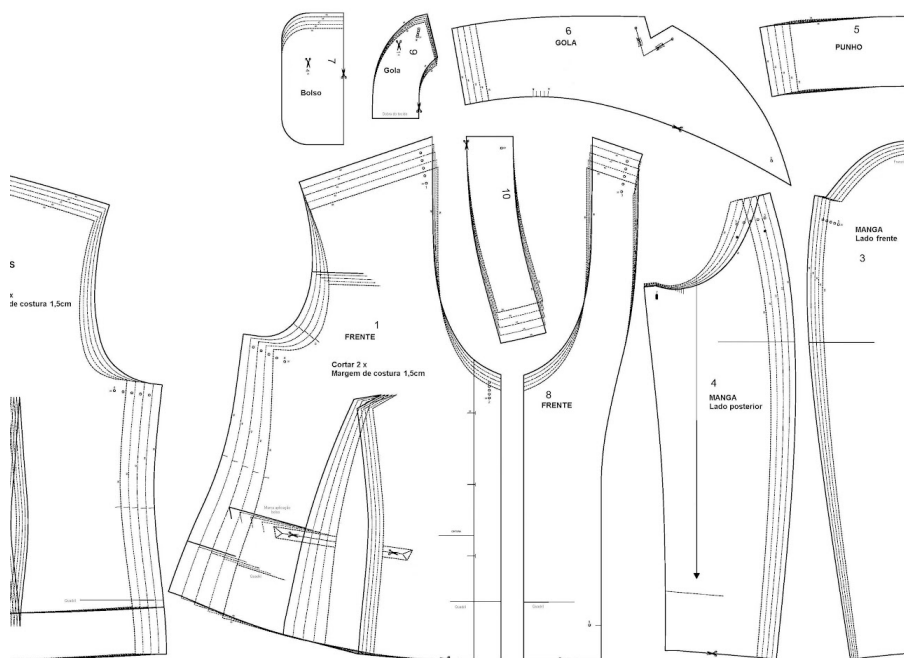


Figura 4: Modelagem bidimensional da roupa ¹

¹fonte: Tip-Top

Pela imagem da figura 4, se verifica que a roupa é constituída por diversas peças de diferentes tamanhos (manga, punho, gola, bolso) e na modelagem as medidas de cada roupa já estão definidas. Assim, cada roupa possui a sua própria modelagem que é salva em um arquivo específico do CAD utilizado. É importante salientar, que a ferramenta CAD utilizada não gera uma imagem 3D da roupa, somente uma imagem bidimensional, conforme a figura 4. O Provador Virtual mostrará uma visão tridimensional das roupas.

2.1.3 Produção das Roupas

Para conceber um protótipo, o desenho da etapa de modelagem é impresso em escala real, sobreposto ao tecido, cortado e costurado, gerando uma amostra da roupa.

No processo já industrial, o próprio modelo é lido por uma cortadora automática, que gera os cortes de tecidos que, em seguida, serão costurados manualmente.

Denote-se que o CAD não gera uma pré-visualização da roupa em 3D vestida em um *avatar*, cabendo ao estilista imaginar o visual da roupa pronta em uma pessoa, sendo necessária a concepção prévia de um protótipo. Parece-nos, portanto, que um sistema capaz de gerar uma pré-visualização da roupa seria de extrema utilidade para o estilista.

2.2 Análise do Processo de Fabricação

Dentro das fases do processo de fabricação de uma roupa a de maior interesse para este projeto é o desenvolvimento da modelagem bidimensional da roupa.

Como enfatizamos no 2.1.2 este processo é integralmente feito pela ferra-

menta CAD. Teoricamente o Proveedor Virtual poderia ter uma interface com o CAD, que executa a modelagem. Esta interface deveria ser capaz de retirar as medidas e os tamanhos de cada roupa. Assim, cruzando as medidas das roupas e dos indivíduos, poder-se-ia sugerir um tamanho para uma pessoa.

Entretanto, esta integração com o CAD é inviável, considerando a existência de diferentes softwares de modelagem de roupas, que são incompatíveis entre si.

Ainda assim, é possível obter as medidas das roupas, a partir de tabelas fornecidas pelo fabricante. Podemos verificar, por exemplo, em uma camiseta quais são os parâmetros e as correspondentes medidas que definem o seu tamanho nas figuras 5 e 6 abaixo.

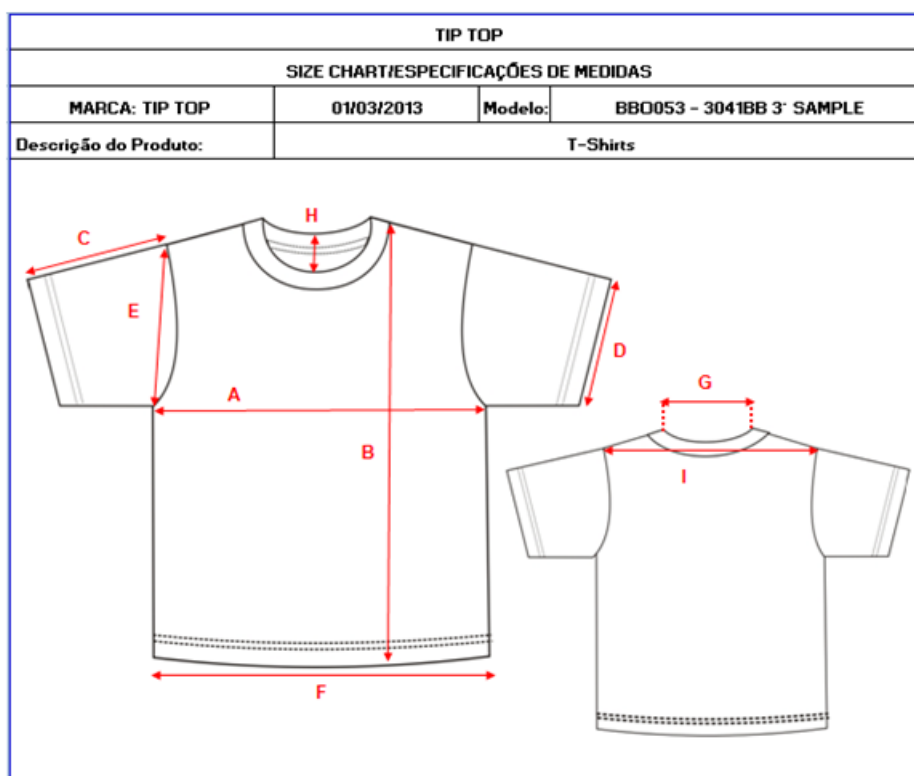


Figura 5: Medidas utilizadas para confecção de camisetas²

²fonte: Tip-Top

		TAMANHOS (s)					
	DESCRIÇÃO DA MEDIDA	P	M	G	E	SAMPLE M	TOL:
A	Chest (<i>Peito</i>)	25,5	26,5	27,5	28,5	OK	± 1
B	Total Length (<i>Comprimento total</i>)	29,0	30,5	32,0	33,5	OK	± 1
C	Sleeve length (<i>Comprimento manga</i>)	8,5	9,0	9,5	10,0	OK	± 0,5
D	Sleeve opening (<i>Abertura manga</i>)	8,0	8,5	9,0	9,5	OK	± 0,5
E	Armhole (<i>Cava</i>)	11,5	12	12,5	13	OK	± 0,5
F	Bottom (<i>Barra</i>)	25,5	26,5	27,5	28,5	OK	± 1
G	Neck Width - (<i>Largura do pescoço</i>)	10,0	10,5	11,0	11,5	9	± 0,5
H	Front Neck Drop (<i>Abertura decote frente</i>)	4,0	4,0	4,5	5,0	OK	± 0,5
I	Back shoulder to shoulder (<i>Costas de ombro a ombro</i>)	21,0	22,0	23,0	24,0	OK	± 1
	All over stretched neck (<i>Decote esticado</i>)	25,0	25,0	26,0	26,0	OK	0
Observações :							
Responsável :							

Figura 6: Tabela de medidas e tamanhos correspondentes³

³fonte: Tip-Top

3 ALTERNATIVAS PARA SOLUÇÃO DO PROBLEMA

Nesse capítulo, iremos apresentar as alternativas do projeto com um todo, além das opções tecnológicas e de projeto para as respectivas áreas do conhecimento, que serão necessárias para o desenvolvimento do sistema.

A decisão que impacta todo o sistema é a possível disponibilização do Provedor Virtual online. Existem duas alternativas. Uma delas é desenvolver um sistema que é executado cem por cento localmente na máquina do usuário; a outra é utilizar um sistema que possui uma arquitetura cliente e servidor e, portanto, pode ser exibido remotamente.

Essas alternativas impactam não somente o quesito técnico do projeto, mas também influenciam sob o ponto de vista comercial. Caso o Provedor Virtual seja executado localmente, o usuário deverá baixar o *software* para utilizá-lo. Na segunda opção, no caso online isto não será necessário. Assim, considerando que o objetivo do projeto é aumentar as vendas de vestuário pela internet, a obrigatoriedade do usuário baixar um *software* para efetuar a compra é um fator que poderá inibi-la ou dificultá-la.

A seguir, iremos apresentar as tecnologias atuais disponíveis para o desenvolvimento do sistema e, também, as alternativas de projeto relativas a cada área do seu conhecimento. As duas principais áreas de conhecimento do projeto são a análise e processamento de imagens e a computação gráfica, que se refere à

síntese de imagens.

3.1 Análise e Processamento de Imagem

O sistema deve ter a função de aferir as medidas de um ser humano, a partir de poucas fotos do indivíduo. Sendo assim, faz-se necessário o uso de um software capaz de tratar e de processar imagens, extraíndo os dados desejados.

3.1.1 Tecnologias disponíveis

Para a criação deste *software* existem bibliotecas para o processamento de imagens, que podem ser utilizadas para auxiliar no desenvolvimento de tal função. Podemos citar algumas como *OpenCV* [1], *GD*, *PIL*, entre outras.

O *OpenCV*, acrônimo do inglês *Open Source Computer Vision Library*, é uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial.

A *GD* é uma biblioteca para tratamentos de imagens, nativa da linguagem *PHP* e seu uso facilitaria o trabalho para programação em sites na web.

A *PIL*, acrônimo do inglês de *Python Imaging Library* é uma biblioteca nativa da linguagem *Python*. Assim como o *PHP*, o *Python* é uma linguagem que permite a programação para sites na internet.

3.1.2 Alternativas de Projeto para a Análise e Processamento de Imagem

Para aferirmos as medidas do usuário podemos captar suas principais medidas antropométricas ou aferir a sua silhueta por completo. Esta ideia fica mais clara, ao analisarmos a figura 7 abaixo, que consiste em uma foto do contorno de uma pessoa, extraída a partir de uma foto fornecida pelo usuário.



Figura 7: Silhueta do membro superior de um corpo humano

Como vimos no capítulo relativo à análise e levantamento de dados, para os fabricantes de vestuário apenas algumas medidas são necessárias para definir o tamanho da roupa. Portanto, para sugerir o tamanho da roupa para o cliente, basta obter algumas medidas corporais, não sendo necessário utilizar todas as medidas obtidas a partir da silhueta completa.

No entanto, a obtenção da silhueta é necessária para a geração de um modelo tridimensional e, também, para se inferir algumas medidas corporais. Logo, a única opção para este projeto no tocante à análise e processamento de imagens é a obtenção da silhueta do usuário.

3.2 Computação Gráfica

O sistema deve permitir a exibição do *avatar* vestido com a roupa selecionada pelo usuário. Sendo assim, faz-se necessário o uso de um software de computação gráfica capaz auxiliar no desenvolvimento de tal função.

3.2.1 Tecnologias Disponíveis

Existem algumas tecnologias que podem auxiliar na exibição e criação de um modelo 3D. É possível utilizar-se de alguns *softwares*, bibliotecas ou ambientes de programação (*frameworks*) capazes de gerar a imagem de um modelo tridimensional.

Atualmente, diversos *softwares* de modelagem tridimensional são aplicados para a elaboração de desenhos técnicos. Entretanto, a maior parte deles não é projetado para representar modelos humanoides. Além disso, muitos deles são pagos, com valor impeditivo para a execução deste projeto.

Softwares usados para a representação de humanos são muito utilizados na área de criação de jogos, os famosos *games*. Nesta área, eles são chamados de *engine*. Alguns *engines* disponíveis no mercado são: *3D Rad*, *Blender* [2], *Core3D*, *Ardor 3D*, entre outros. Os *engines* são classificados pelas suas características (*features*), e dentre estas, aquelas que mais nos interessam referem-se à geração de modelos 3D e a viabilidade de serem programáveis (*scriptable*).

Para este projeto, além das capacidades dos *engines* consideraremos também o tipo de licença, pois as pagas aumentariam os custos do projeto.

Alguns *engines* são capazes de simular a física mecânica. São capazes, por exemplo, de simular o caimento de uma roupa em um *avatar*. Um software de mercado capaz de executar tal tarefa chama-se *Marvelous Designer*. Este software é uma poderosa ferramenta, na qual é possível serem gerados *avatares*, a partir de alguns parâmetros e vesti-los usando a modelagem bidimensional de uma roupa. A interface gráfica do *Marvelous Designer* pode ser vista na figura 8.

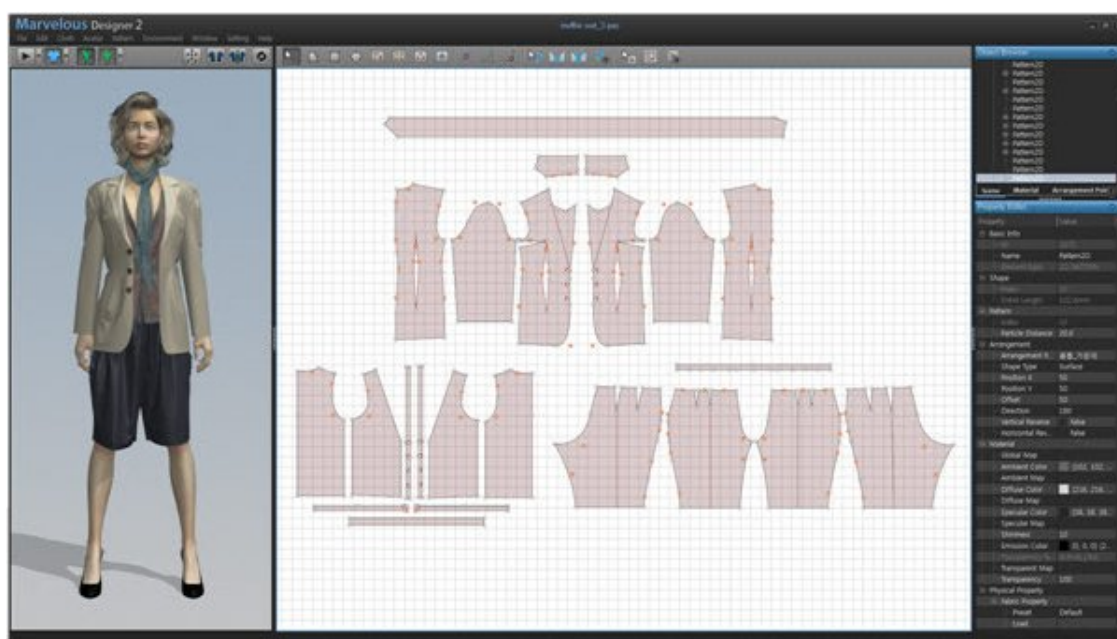


Figura 8: Interface gráfica do *Marvelous Designer*¹

Na figura 8, podemos verificar à direita que a modelagem bidimensional da roupa já está pronta para o corte. Nesta modelagem os pontos em vermelho indicam onde a costura deve ser feita. O *Marvelous Designer* [3] é capaz de modelar tridimensionalmente a roupa a partir de sua modelagem bidimensional. Em seguida, com a modelagem tridimensional a roupa pode ser vestida no *avatar*, conforme se verifica no lado esquerdo da figura.

A maior parte dos *engines* foram concebidos utilizando a biblioteca de computação gráfica chamada *OpenGL* [5]. Uma possível alternativa seria não utilizar os *engines* e programar um software específico para a exibição de modelos tridimensionais com o apoio do *OpenGL*. Esta é uma solução que tornaria o sistema mais customizado, porém exigiria maior esforço de projeto. As dificuldades enfrentadas seriam principalmente relacionadas à geração de animações com os *avatares* e, também, na simulação de tecidos das roupas, considerando que os *engines* já

¹fonte: *Marvelous Designer*

possuem suporte a este tipo de função.

Existe uma versão do *OpenGL* para desenvolvimento *web* que se chama *WebGL*. O *WebGL* possibilita a utilização da biblioteca *OpenGL* em *JavaScript*, mas não é suportado por todos navegadores, principalmente os mais antigos.

3.2.2 Alternativas de Projeto para a Computação Gráfica

Um modelo 3D de uma pessoa representa uma coleção de pontos no espaço, conectados por várias entidades geométricas, como triângulos, linhas, superfícies curvas, etc². Em computação gráfica, a coleção de pontos é chamada de vértices e as entidades geométricas são chamadas de faces. Todo o conteúdo referente à modelagem tridimensional aplica-se não só a modelagem do *avatar*, mas também a modelagem da roupa. A seguir, analisaremos a modelagem do *avatar*, salientando que as mesmas alternativas podem ser aplicadas à modelagem da roupa.

Existem algumas alternativas para a obtenção de um modelo tridimensional: (i) pode ser gerado do zero; (ii) ajustado a partir de um existente; (iii) selecionado a partir de uma base de modelos. Estas alternativas serão analisadas nos tópicos seguintes.

Obtido o modelo tridimensional do *avatar* e da roupa, esta deve ser vestida no *avatar*. Como explicaremos no capítulo relativo aos requisitos do projeto, este encaixe deve ser feito por uma simulação da física mecânica, levando em consideração o tecido e suas características (peso, maleabilidade, elasticidade, dentre outras). Essa funcionalidade pode ser implementada utilizando algum *engine* ou programando com auxílio da biblioteca *OpenGL* como explicitamos no tópico referente às tecnologias disponíveis. Para os *engines* poderíamos utilizar o *Blender* ou o *Marvelous Designer*. Porém, para essa funcionalidade o *Marvelous Designer* foi descartado, pois ele não é programável (*scriptable*) e a integração do sistema

²fonte: Wikipédia

se tornaria inviável. Esta questão será aprofundada em outros capítulos deste trabalho.

Nos próximos itens analisaremos as alternativas para a obtenção de modelos 3D, da forma de customizar o *avatar* e de simular o tecido da roupa.

3.2.2.1 Alternativas para a Obtenção de Modelos 3D

3.2.2.1.1 Geração de Modelos 3D

Para obter o conjunto de pontos de um indivíduo que formariam seu avatar seria necessário “escanear” a pessoa em diversos ângulos. Podemos fazer isso de maneira relativamente simples com um dispositivo que possua um sensor de movimento, como por exemplo, o *Kinect*, dispositivo criado pela Microsoft. Este dispositivo possui câmeras que medem a distância bidimensional de um ponto e um sensor de infravermelho, que mede a profundidade. Assim, tendo as três coordenadas espaciais de cada vértice, seria possível obter as faces e, portanto, o modelo tridimensional.

Entretanto, é possível também obter este conjunto de pontos com uma câmera. A partir de uma foto do indivíduo é possível extrair a sua silhueta, como vimos no item 3.1.2.

Se o usuário fornecer várias fotos na mesma distância da câmera, ou seja, na mesma posição espacial, porém girando o corpo um delta de graus a cada nova foto, será possível obter vários pontos espalhados no espaço. Teríamos, portanto, o mesmo resultado do que o fornecido por um *Kinect* e poderíamos criar o modelo tridimensional do indivíduo.

Como veremos no capítulo 04 sobre os requisitos do projeto, a utilização de uma simples câmera será obrigatória. No entanto, neste trabalho como consideramos que, no Brasil, o *Kinect* não é um dispositivo de fácil acesso não o usaremos,

pois a sua aplicação teria um alcance limitado.

3.2.2.1.2 Customização de um Avatar

Existe a possibilidade de usar um avatar pré-modelado e, posteriormente, customizá-lo para obter um modelo fiel ao usuário. O *Marvelous Designer*, software citado no item sobre tecnologias disponíveis de computação gráfica, faz essa customização. Ele possui um avatar pré-modelado, que permite a modificação de certas variáveis, como por exemplo, a altura do avatar, a largura da cintura, entre outros.

O mesmo efeito pode ser alcançado, com a programação em *OpenGL*, modificando-se a malha (conjunto de vértices e faces) do modelo. Nesta alternativa também poderá se obter um modelo extramente fiel ao usuário.

3.2.2.1.3 Seleção de um Modelo

Uma terceira opção, além daquela de partirmos de apenas um modelo para customizá-lo, será escolhermos vários modelos já pré-customizados, salvos em um banco de dados e selecioná-los, conforme as medidas do usuário. Escolhendo-se esta alternativa, para obtermos um avatar fiel ao cliente seria necessária uma base extensa de modelos. No entanto, sua implementação é mais simples que as propostas anteriores.

3.2.2.2 Simulação de Tecido e Montagem da Roup

A simulação de tecido pode ser feita utilizando-se a ferramenta *Blender* ou programando-se usando o *OpenGL*.

Para a montagem da roupa as mesmas alternativas podem ser usadas, mas também podemos utilizar um banco de dados com avatares já vestidos e exibi-los.

Tendo acesso a esse banco de dados a implementação não será tão complexa e, portanto, se torna mais rápida.

Entretanto, essa solução exigiria um banco de dados com alta ocupação de memória, pois deveríamos ter as combinações possíveis de roupas de todos os tamanhos com todos os *avatares* disponíveis.

3.2.2.2.1 Auxílio da Ferramenta *Blender*

O *Blender* possui um módulo de simulação de física mecânica que é um módulo de simulação de tecidos. Assim, com esta ferramenta é possível gerar uma animação do *avatar* vestindo a roupa para, posteriormente, exibi-la ao usuário. Criar uma animação desse tipo no *Blender* é relativamente simples, no entanto, automatizar esse processo via um *script* de *Python* torna esta tarefa complexa. Caso estejamos trabalhando com sistema online, a integração também se tornaria uma empreitada de alta complexidade.

3.2.2.2.2 Auxílio da Biblioteca *OpenGL*

É possível também gerar animações programando no *OpenGL*. Na verdade, o *Blender* utiliza-se dessa biblioteca de computação gráfica para gerar suas animações de maneira amigável ao usuário. Portanto, programar uma animação usando o *OpenGL*, descartando o *Blender*, exigiria maior dedicação e tempo, mas facilitaria a integração dos diversos módulos.

3.2.2.3 Exibição dos Resultados

Após o *avatar* estar vestido e o tamanho da roupa já ter sido definido pelo sistema, estes resultados devem ser mostrados ao usuário. Existem diversas formas de exibi-los.

Se o Provedor Virtual for um *software* distribuído e rodado localmente, a exibição dos resultados será processada e exibida no computador do usuário.

Caso o sistema seja disponibilizado online, por um site, por exemplo, é necessário organizar-se uma arquitetura do tipo cliente-servidor. Nesta arquitetura, o resultado do que é processado no servidor deve ser exportado para o cliente. Existem algumas formas de executar tal tarefa.

3.2.2.3.1 Baixar algumas Imagens do Avatar

O *avatar* poderá ser exibido ao usuário por fotos em diversas posições. Desta maneira, as imagens do *avatar* serão disponibilizadas para o cliente, que poderá baixá-las. Esta opção é de implementação relativamente simples, porém o usuário não terá uma visão completa do seu avatar.

3.2.2.3.2 Baixar um Vídeo do Avatar

Outra alternativa seria gerar uma animação do *avatar* vestido desfilando, por exemplo. Esta animação seria reindenizada na forma de um vídeo, que seria disponibilizado para o usuário baixá-lo. Esta opção promove uma experiência satisfatória para o usuário, por permitir a ele sua visão com a roupa em movimento em diversos ângulos. Porém, a complexidade de implementação desta solução tem forte impacto no *time-to-market* deste projeto. Além disto, a reindenização pode levar alguns minutos, o que pode levar o cliente a desistir de prosseguir na compra online.

3.2.2.3.3 Elaboração de um CAD Online

A última opção, dentre todas a mais elegante, seria disponibilizar o modelo tridimensional para o usuário. Isto significa que o usuário terá controle sobre o seu

avatar, podendo girá-lo e movimentá-lo com simples cliques. Esta solução proporciona a melhor experiência ao usuário, porém sua implementação depende de *softwares* ou bibliotecas que deem suporte online para a exibição do *avatar*, como o *WebGL*, por exemplo. É preciso ressaltar que isto tornaria o desenvolvimento do sistema mais complexo.

4 FUNCIONALIDADES, REQUISITOS E RESTRIÇÕES DO PROJETO

O sistema deve ser capaz de cumprir as seguintes macro-funcionalidades: (i) aquisição de medidas e sugestão de tamanho; (ii) criação do modelo tridimensional; (iii) simulação de tecido e montagem da roupa; (iv) exibição dos resultados. Cada funcionalidade para ser executada deverá cumprir determinados requisitos, a seguir descritos.

4.1 Aquisição de Medidas e Sugestão de Tamanho

A aquisição de medidas deve ser feita de forma automatizada, com o mínimo possível de esforço para o usuário.

A utilização de uma câmera simples disponível em telefones celulares e computadores portáteis facilitará a disseminação do Provador Virtual. A aquisição das fotos deverá ser feita em uma posição específica da pessoa e uma referência como altura, por exemplo, deverá ser dada como entrada pelo usuário. É possível, também, exigir-se que o usuário segure algum objeto conhecido como, por exemplo, um CD ou uma régua, ou ainda que ele indique algum objeto de tamanho conhecido na imagem. Será recomendado ao cliente que utilize roupas justas para melhor aferição de medidas.

A partir das fotos e de uma referência dimensional da pessoa serão aferidas suas medidas básicas.

É necessário ressaltar, que as medidas básicas de um indivíduo são normatizadas pela ABNT ISO 7250-1:2008. Para este projeto, levamos em consideração somente as medidas antropométricas da parte superior do indivíduo, pois o trabalho está restrito ao vestuário relativo ao tronco (camisas e camisetas). Assim, para cumprimento deste requisito as medidas foram simplificadas em distância entre ombros, largura do pescoço e do tórax.

Para a sugestão de tamanho da roupa ao usuário teremos como entrada o modelo bidimensional das roupas fornecido pelo fabricante e, também, às aferições das medidas. Assim, com uma simples comparação entre as medidas do modelo bidimensional fornecido pelo fabricante e aquelas do usuário será possível sugerir a ele um tamanho de determinada roupa.

4.2 Criação de um Modelo Tridimensional

A partir das medidas antropométricas obtidas nos termos do item 4.1 será gerado um modelo tridimensional, que represente da forma mais exata possível às dimensões do usuário. Isto garantirá que na próxima etapa, seja possível a realização de uma simulação para verificar como ficaria a roupa no usuário.

Denote-se que a sugestão do tamanho da roupa e a criação do modelo tridimensional dependem da aferição das medidas do indivíduo, mas são funcionalidades independentes entre si. Tais funcionalidades são de extrema importância para o usuário aprovar a roupa escolhida. A primeira (sugestão do tamanho) possibilita que a roupa tenha as dimensões corretas do usuário e, a segunda (criação do modelo tridimensional vestido), garante que o vestuário esteja adequado ao corpo do cliente.

4.3 Simulação de Tecido e Montagem da Roup

Estas funcionalidades se iniciam quando o *avatar* foi gerado, foram obtidas as medidas da roupa por intermédio do sistema e, também, já possuímos a modelagem tridimensional da roupa. A partir daí, a roupa deve ser encaixada no *avatar*. Esse encaixe é feito por uma simulação da física mecânica, levando em consideração o tecido e suas características (peso, maleabilidade, elasticidade, dentre outras).

Assim, poderemos visualizar na etapa seguinte como o usuário ficaria vestido com aquela roupa, caracterizando assim o Provador Virtual.

4.4 Exibição dos Resultados

A exibição dos resultados consiste na sugestão de um tamanho para a roupa selecionada, bem como, na demonstração do *avatar* vestido com ela. As proporções antropométricas do *avatar* devem ser fieis às do cliente. A medida da roupa sugerida, também deve se enquadrar dentro de uma faixa aceitável.

5 METODOLOGIA

Para controle das atividades o projeto foi dividido em seis módulos:

1. *Camera Module*;
2. *Measurer Module*;
3. *Chooser Module*;
4. *Clothes 2D to 3D Converter*;
5. *3D Dressing Module*;
6. *Display Module*;

A descrição de cada módulo será dada no capítulo 07 que explica a solução escolhida para o projeto. Nesse capítulo vamos nos ater a metodologia de desenvolvimento do projeto.

A implementação de funcionalidades de cada módulo pode ser levada em paralelo, apesar da saída de um módulo fornecer a entrada para o outro. Para facilitar o entendimento vamos tomar o caso do *Measurer Module*. Esse módulo faz a aferição de medidas do cliente, baseado nas imagens fornecidas. Portanto, ele necessita como entrada as imagens capturadas pelo *Camera Module*, o que faz parecer o que seu desenvolvimento necessita ser feito previamente ao do *Measurer Module*. No entanto, uma imagem pode facilmente ser fornecida manualmente

para o *Measurer Module*, enquanto o *Camera Module* ainda não é capaz de executar essa funcionalidade.

De maneira análoga, o *Measurer Module* fornece as medidas da pessoa para os demais módulos processarem. Porém, mesmo um proporcionando entrada para outros módulos, é possível fornecer manualmente medidas teóricas para os demais módulos.

Objetivamos ter a qualquer instante do projeto um protótipo com funcionalidades mínimas. Assim, a cada pequeno conjunto de novas funcionalidades agregadas ao projeto faz-se necessário uma integração das mesmas.

Portanto, o método adotado para o desenvolvimento do Provedor Virtual foi o *Scrum*. O princípio básico do *Scrum* consiste em entregar em um curto espaço de tempo (um *sprint*) um produto com poucas *features*¹, mas completo. Assim, ao final de cada *sprint* entrega-se um produto, que vai sendo aperfeiçoado e incorporado de forma incremental. Para mais informações sobre o processo do *Scrum*, vide o apêndice B.

Mesmo assim, como todo projeto tem prazos, teremos um cronograma que levará em consideração uma produção sequencial dos módulos. O encadeamento sequencial de atividades de um projeto é conhecido como modelo em cascata. Assim, no cronograma teremos o início do desenvolvimento de um módulo após o término de um módulo precedente. Para maiores detalhes sobre o modelo em cascata vide o apêndice A.

O cronograma, portanto é uma *baseline*² com datas de início e término de cada módulo e suas subatividades. No capítulo relativo ao plano gerencial entraremos em mais detalhes sobre o cronograma do projeto.

¹ Funcionalidades ou características do sistema.

² A tradução literal do inglês é “linha como base”, esse termo é utilizado em gerência de projetos. No contexto, o termo é usado para enfatizar que o cronograma é uma base para mensurar o tempo das atividades do projeto.

6 CRITÉRIOS DE AVALIAÇÃO

Para cada funcionalidade do projeto, diversas alternativas foram estudadas. Objetivando a seleção destas alternativas foram eleitos cinco critérios de avaliação, que refletem as prioridades adotadas no projeto. São eles, por ordem de prioridade:

1. *time-to-market*;
2. qualidade do produto;
3. integração;
4. custo;
5. desempenho;

Devido ao curto espaço de tempo do projeto (*time-to-market*), não é possível incluir no seu escopo soluções cujo período de implementação seja longo. Nesse sentido, a facilidade de implantação, que pressupõe a agilidade do projeto é o critério de avaliação de maior peso no escopo deste trabalho.

Por outro lado, em que pese à necessidade de rapidez em sua implantação, o produto deve ter as funcionalidades indispensáveis para realizar todas as suas funções e de maneira agradável ao usuário. Assim, a qualidade do produto ocupa o segundo lugar no ranking dos critérios de avaliação.

Não podemos esquecer que, em todo e qualquer projeto, o custo é de extrema importância. No caso do Provisor Virtual este critério aparece apenas na terceira posição, pois em um projeto de *software* o custo é principalmente impactado pela mão de obra e pelo custo dos *softwares* auxiliares para o desenvolvimento. Portanto, priorizamos o uso de *softwares* livres a fim de baratear o projeto.

O desempenho refere-se ao tempo que o sistema demora em executar suas funcionalidades, denominado de “tempo de execução do sistema”. Este critério foi listado em último lugar, pois certamente o usuário não será exigente com relação a pequenas variações no tempo de execução do sistema. Entretanto, esperas por volta de um minuto muitas vezes não são toleradas. Assim, o desempenho ainda aparece como um critério de avaliação, mas é menos prioritário.

7 ESCOLHA DA SOLUÇÃO

Considerando as alternativas estudadas e descritas no capítulo 03 e as funcionalidades, requisitos e restrições aventadas no capítulo 04, concluímos a solução completa do projeto neste tópico. Começaremos a análise definindo as decisões do projeto como um todo. Em seguida, elegeremos as opções de implementação de cada módulo específico. Importante frisar, que para a escolha das soluções utilizaremos os critérios de avaliação definidos no capítulo anterior.

Como explicitamos no capítulo 03, a grande decisão que impacta todo o projeto é a definição se o Proveedor Virtual será disponibilizado online. O intuito do projeto é de aumentar as vendas pela internet. No entanto, a execução de um projeto com cliente e servidor torna-se mais complexa. Para definirmos a solução utilizamos a matriz de decisão que segue na figura 9 abaixo.

Critérios	Pesos	Alternativas	
		Online	Local
<i>Time-to-market</i>	30%	4	9,5
Qualidade do produto	30%	10	3
Integração	20%	6	9,5
Custo	10%	10	10
Desempenho	10%	9,5	4
Total	100%	7,35	7,05

Figura 9: Matriz de decisão: alternativas *online* e local

A matriz de decisão foi elaborada com base nos critérios de avaliação do capítulo 06. Devido à complexidade de implementação de um software que exibe

avatares online, o impacto no tempo do desenvolvimento desta ferramenta é alto, por isso a nota é baixa para o critério *time-to-market*. A qualidade do produto nesta hipótese é extremamente superior, já que apenas um produto *online* atenderá aos clientes remotamente.

Por outro lado, a integração é feita de maneira mais fácil se o sistema é executado localmente, pois evita transferências de dados entre o cliente e o servidor. O desempenho, no caso de todo processamento ser executado localmente, é inferior. Há duas razões para esta fragilidade do desempenho: os PCs são inferiores aos servidores e a arquitetura cliente servidor possibilita a distribuição do processamento entre ambas as partes.

Neste contexto, percebemos que apesar da alternativa *online* apresentar um tempo de projeto maior devido a sua complexidade, ela oferece um produto extremamente superior, considerando que o objetivo do projeto é aumentar o número de vendas de vestuário pela internet. Portanto, escolhemos um Provedor Virtual que será disponibilizado *online*.

Partindo da premissa de um *software* que será disponibilizado *online*, seguimos com a divisão modular do projeto. Explicitaremos quais alternativas foram utilizadas na execução deste projeto em cada um dos seis módulos.

7.1 *Camera Module*

Esse módulo é responsável pela captação da imagem do usuário e também da roupa por ele escolhida. O usuário poderá fornecer as imagens de duas maneiras, pela webcam do computador ou carregando uma imagem na memória do computador. Além disso, o *Camera Module* também é a interface gráfica inicial do Provedor Virtual.

Como optou-se nesse projeto pelo desenvolvimento de um *software online*,

o *Camera Module* deve ser executado no computador do cliente, para conseguir captar as imagens do usuário. Assim, ele deverá ser desenvolvido nas linguagens feitas para o desenvolvimento de páginas web que são: *HTML5*, *CSS3*, *JavaScript* e *Flash*. Essas linguagens de programação são largamente utilizadas para programação web. O *JavaScript* e o *Flash* são muito utilizados para criar páginas dinâmicas e animadas, que serão necessárias na criação desses módulos.

7.2 **Measurer Module**

Esse módulo deve ser capaz de processar as imagens obtidas do usuário e extrair suas medidas antropométricas. Além disso, tendo as medidas corporais do usuário e as medidas da roupa ele deve ser capaz de sugerir um tamanho de roupa para o usuário.

É no Measurer Module que ocorre o processamento de imagem, atividade fundamental do projeto. Portanto, a escolha da alternativa certa deve ser feita detalhadamente. As alternativas eram trabalhar com o *OpenCV*, *GD* (biblioteca do *PHP*) ou *PIL* (biblioteca do *Python*). O *Python* e o *PHP* são linguagens de programação voltadas para o desenvolvimento web e, portanto, possuem características similares. Assim, a disputa pela solução ideal seria entre linguagens de programação web e o *OpenCV*.

No entanto, o *Python* foi deixado de lado, pois o desenvolvedor do projeto já possui conhecimento prévio em *PHP*. Logo, a escolha se restringirá ao *PHP* e o *OpenCV*. Vale ressaltar que independentemente dessa escolha, o *PHP* continuará fazendo parte do projeto para o desenvolvimento do site, apenas não será utilizado também para o processamento de imagem. Para essa tomada de decisão elaboramos outra matriz de decisão, que pode ser vista na figura 10 abaixo.

Critérios	Pesos	Alternativas	
		PHP	OpenCV
<i>Time-to-market</i>	30%	10	9
Qualidade do produto	30%	8	10
Integração	20%	10	7
Custo	10%	10	10
Desempenho	10%	5	10
Total	100%	8,9	9,1

Figura 10: Matriz de decisão: alternativas *PHP* e *OpenCV*

O *PHP* por ser uma linguagem de programação para web, exclui a necessidade de integração e, portanto, também reduz o tempo de projeto. No entanto, o *OpenCV* é uma biblioteca voltada para o rápido processamento de imagens. Portanto, ela possui um altíssimo desempenho e produz resultados superiores ao *GD* do *PHP*. Pela matriz restou comprovado que a implementação com o *OpenCV* é mais vantajosa.

7.3 Chooser Module

O *Chooser Module* é o módulo responsável pela geração dos modelos 3D. Saliente-se que a solução escolhida para geração de modelos é uma pseudo-geração. Isso significa que, na realidade, os modelos não serão gerados. Eles serão selecionados de uma base de modelos e exibidos ao usuário dando a impressão ao usuário que o modelo foi feito exclusivamente para si. Daí advém o nome *Chooser Module* e não *Model Generator Module*, por exemplo.

Como vimos no capítulo 03, existem três opções para a criação dos modelos: a geração de um modelo do zero, a customização de um avatar pré-modelado e a seleção de um avatar pronto. O próprio nome deste módulo (*Chooser Module*) indica que optamos pela seleção de módulos. Justificaremos esta escolha, por intermédio da matriz de decisão a seguir apresentada.

Critérios	Pesos	Alternativas		
		Geração	Customização	Seleção
<i>Time-to-market</i>	30%	2	5	10
Qualidade do produto	30%	10	10	8
Integração	20%	2	2	8
Custo	10%	5	5	7
Desempenho	10%	4	9	10
Total	100%	4,9	6,3	8,7

Figura 11: Matriz de decisão: alternativas para obtenção do modelo 3D

O principal critério de escolha para a seleção de um avatar já pré-modelado foi o *time-to-market*, ou seja, o exíguo tempo para a elaboração do projeto. A utilização de avatares pré-modelados torna a execução do projeto menos complexa e, portanto, muito mais rápida do que as demais.

Denote-se que a ferramenta *Marvelous Designer* auxilia nos ajustes dos modelos, fazendo com que a construção da base de modelos diversificada se torne uma tarefa simples. Assim, mesmo modelando o avatar conseguimos um produto de boa qualidade. O desempenho da seleção de modelos é superior às demais alternativas, pois basta selecionar e carregar o modelo, não sendo necessário proceder nenhuma alteração nele.

A integração também se torna mais fácil, considerando que para proceder aos ajustes no modelo seria necessário chamar um CAD, como *Blender*, por exemplo, para executar tal tarefa. Isso tornaria mais complexa a integração dos diversos módulos. Um ponto negativo a ser destacado é que o projeto fica dependente da ferramenta *Marvelous Designer*, pois ela foi a “geradora” dos modelos. Portanto, ainda que este processo seja utilizado uma única vez, é provável que haja necessidade de pagamento da licença do software, o que aumenta o custo de projeto.

7.4 *Clothes 2D to 3D Converter*

Como vimos no capítulo 2, relativo ao levantamento e análise de dados, o fabricante possui as roupas já pré-modeladas em CAD bidimensional. Porém, para exibi-las vestidas em um avatar deve ser gerado um modelo tridimensional de roupas. Este módulo automatiza a conversão de roupas bidimensionais para roupas tridimensionais.

Considerando que os fabricantes de vestuário utilizam diferentes CADs para modelar suas roupas, haverá a necessidade de se escolher um CAD específico para tentar compatibilizar o sistema. Para melhor explicitar este tema, vamos utilizar como exemplo o CAD utilizado pela Tip-Top, fábrica de vestuário visitada para levantamento de dados. O CAD por eles utilizado chama-se Audaces. Os modelos bidimensionais salvos pelo Audaces são salvos em arquivos com extensão.ads, um formato proprietário. Portanto, para conseguir extrair as medidas desse arquivo, é necessário conseguir interpretá-lo corretamente.

Uma vez extraída as medidas do arquivo, seria possível dá-las como entrada para o *Blender*. O *Blender*, como afirmado anteriormente é um software programável (*scriptable*) em *Python*. Assim, com um *script* em *Python* poderíamos enviar comandos ao *Blender* que, de forma automática, formaria o modelo tridimensional da roupa.

Saliente-se que a tentativa de automatizar a modelagem das roupas é complexa e, também limitada, pois o sistema funcionaria apenas para um fabricante específico. Por isso, concluímos que neste projeto, a modelagem das roupas será feita manualmente no *Blender*.

O módulo *Clothes 2D to 3D Converter* é o módulo de modelagem de roupas no *Blender*, que será feita manualmente.

Lembramos que as medidas das roupas não necessitam ser extraídas do arquivo com extensão *.ads*, por exemplo. Elas são fornecidas em planilhas como mostramos anteriormente na figura 5.

7.5 3D Dressing Module

Escolhemos selecionar um avatar pré-modelado de uma base. Optamos, também, por selecionar uma roupa pré-modelada de uma base.

A partir do modelo tridimensional da roupa e da figura do usuário, ambos deverão ser superpostos, de maneira que o usuário seja vestido com a roupa. Sendo assim, o *3D Dressing Module* será o responsável pelo alinhamento dos modelos, pela simulação da gravidade e pelo encaixe da roupa no modelo.

Como afirmamos anteriormente, podemos executar essa simulação utilizando a ferramenta *Blender* ou programando em *OpenGL*. Para a montagem da roupa podemos exibir *avatars* já vestidos selecionados em um banco de dados. Entretanto, para escolhermos a solução mais adequada para o projeto, com relação à simulação da gravidade e ao encaixe da roupa no modelo, apresentamos uma nova matriz de decisão pode ser vista na figura 12 abaixo.

		Alternativas		
CrITÉRIOS	Pesos	Blender	OpenGL	Modelos prontos
<i>Time-to-market</i>	30%	5	5	10
Qualidade do produto	30%	9	10	5
Integração	20%	5	9,5	9,5
Custo	10%	10	10	10
Desempenho	10%	4	9	9
Total	100%	6,6	8,3	8,3

Figura 12: Matriz de decisão: alternativas para montagem da roupa

Sem dúvida, o desenvolvimento de um sistema por completo em *OpenGL*,

geraria um produto de qualidade superior a qualquer outra alternativa, já que o produto seria 100% (cem por cento) customizado para o Proveedor Virtual. No entanto, considerando que o *time-to-market* é o primeiro critério de avaliação para a escolha das alternativas deste projeto, não seria possível atender aos prazos em um sistema totalmente customizado.

A solução pelo *Blender* dificultaria muito a integração e, por isso, ficou abaixo das demais. Apesar do empate entre o uso da ferramenta *OpenGL* e a utilização de modelos prontos, vamos optar pela utilização de um banco de dados com *avatars* já vestidos, para garantirmos a exequibilidade do projeto.

7.6 *Display Module*

O *Display Module* é responsável pela exibição de todos os dados que foram processados pelos módulos anteriores. Portanto, ele é a interface gráfica da etapa final do projeto. Assim, este módulo deverá ser capaz de exibir o *avatar* vestido. Como vimos anteriormente, existe uma série de opções sobre como exibir um modelo online, que podem ser assim resumidas:

- Baixar fotos do *avatar*.
- Baixar uma animação do *avatar*.
- Transformar o navegador em um CAD e carregar o modelo no navegador.

Elaboramos uma nova matriz de decisão para explicitar e valorar as alternativas que podem ser utilizadas neste projeto, com relação ao *Display Module*, conforme se verifica na Figura 13 abaixo.

Critérios	Pesos	Alternativas		
		Fotos	Animação	CAD
<i>Time-to-market</i>	30%	9	9	8
Qualidade do produto	30%	5	8	10
Integração	20%	10	10	9
Custo	10%	10	10	10
Desempenho	10%	9	9	9
Total	100%	8,1	9	9,1

Figura 13: Matriz de decisão: alternativas para exibição do modelo

Não há dúvidas que a alternativa de transformar o navegador em um CAD e carregar o modelo no navegador é a mais complexa. Entretanto, como ao utilizá-la conseguiremos obter um produto de alta qualidade, optamos por escolhê-la. A complexidade de sua implementação não prolongará demasiadamente os prazos do projeto, pois o *WebGL* já é integrado nos navegadores mais recentes.

8 ARQUITETURA DO PROJETO

Nos capítulos anteriores observamos uma tendência da sociedade em comprar cada vez mais produtos *online*. Por outro lado, verificamos que as compras de roupas pela internet oferecem maior dificuldade, em relação aos demais artigos, considerando que o vestuário exige maior interação com o usuário.

Em razão desta situação fática, entendemos que a elaboração de um software, com as características do Proveedor Virtual, poderia estimular as vendas de roupas pela internet. No capítulo 03 abordamos as alternativas técnicas existentes para executarmos o projeto e no capítulo 07 determinamos as soluções mais adequadas, em face de critérios abordados e definidos.

Neste momento, integraremos as diversas soluções selecionadas a fim de gerar um protótipo único, capaz de executar todos os requerimentos listados anteriormente. Um diagrama de blocos e fluxos contendo os módulos e o fluxo de dados do projeto encontra-se na figura 14 abaixo.

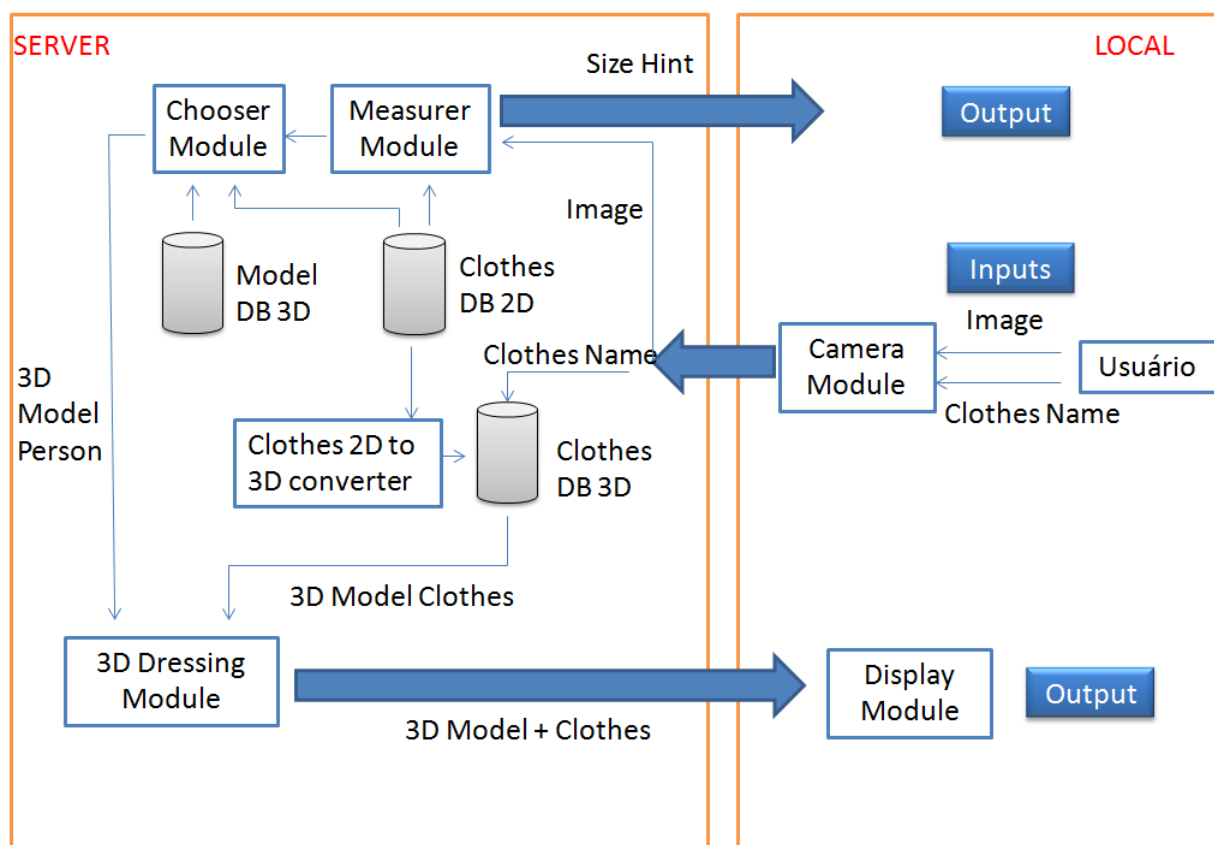


Figura 14: Diagrama de blocos e fluxos do projeto.

Neste diagrama de blocos e fluxos temos uma visão clara do projeto e podemos perceber que o software será disponibilizado em uma plataforma de e-commerce, sendo que parte do processamento será executada em servidores e parte no PC dos clientes.

O usuário que utiliza o sistema pela primeira vez deverá entrar com uma foto do seu corpo inteiro e com a roupa escolhida para provar. É necessário ressaltar, que uma vez cadastradas suas medidas, o usuário não necessitará mais fornecer uma foto como entrada. Estes dados são captados pelo *Camera Module* e a imagem e o nome da roupa são enviados para processamento no servidor. Nesse momento, o nome da roupa escolhida e a imagem são separados, pois serão processados separadamente. Posteriormente, estes dados já processados se re-

encontrarão no *3D Dressing Module*.

A imagem passa pelo *Measurer Module*, módulo de aferição de medidas. Nesse módulo ocorrerá o processamento da imagem, a fim de extrair as medidas antropométricas do usuário. Extraídas as medidas, o módulo faz uma busca no *Clothes 2D to 3D*, o banco de dados que contem todas as medidas de todas as roupas para todos os tamanhos. Com as medidas da roupa escolhida e do usuário, será viável sugerir o seu tamanho exato. A definição do tamanho da roupa será enviada ao usuário, que é uma das saídas do sistema.

As medidas antropométricas do cliente também são enviadas ao *Chooser Module*, módulo que busca na base de modelos o avatar com as características físicas mais similares às do indivíduo. O *avatar* selecionado é enviado para o próximo módulo designado *3D Dressing Module*.

Por outro lado, o nome da roupa é usado como índice em um banco de dados contendo todas as roupas já pré-modeladas tridimensionalmente. A roupa selecionada pelo usuário é enviada ao *3D Dressing Module*. Este módulo é responsável pela integração do avatar selecionado pelo *Chooser Module* com a roupa proveniente do *Clothes 2D to 3D Converter*.

Este módulo ainda processado no servidor executaria a simulação da gravidade: o modelo e a roupa são posicionados de maneira que a roupa em queda livre possa vestir o *avatar*. Ou seja, a roupa é posicionada acima do avatar e cai em queda livre sobre ele. Essas simulações seriam rodadas e salvas na forma de uma animação e, em seguida, elas seriam enviadas para o *Display Module*.

No entanto, como analisamos no capítulo 7, relativo à escolha de soluções, algumas simplificações foram determinadas para este projeto, considerando a exiguidade dos prazos de execução. Assim, selecionaremos avatares já vestidos de uma base determinada. No *Chooser Module* o *avatar* vestido selecionado será

enviado para o *Display Module*.

O *Display Module* carrega o modelo enviado pelo *3D Dressing Module* e executa a simulação de tecido para a roupa do *avatar*. Como a roupa não é um corpo rígido e, portanto, possui certa maleabilidade a simulação de tecido é necessária para uma boa percepção pelo cliente. A alternativa escolhida para esse módulo foi à construção de um CAD *online*, que permitirá ao usuário girar e movimentar a cena, proporcionando-lhe uma ótima experiência com o *avatar*.

Esse módulo, portanto, permite ao usuário não só verificar como ficaria o caimento da roupa, mas também ver a cena de seu modelo vestindo a roupa e, inclusive, navegar pela cena.

A arquitetura do projeto foi desenvolvida pensando no Provedor Virtual como um software que pode ser disponibilizado *online*.

9 PLANO DE GERENCIAL

9.1 Introdução

O plano gerencial do projeto consiste na organização das atividades que serão desenvolvidas ao longo do projeto. Assim, para acompanhamento dos prazos e entregas elaboramos um cronograma de projeto, que é uma *baseline* com marcos de cada grande entrega.

Para termos outra visualização das tarefas e, também estimar os custos do projeto, foi elaborado um WBS do inglês *Work Breakdown Structure*, que consiste em uma árvore orientada às entregas.

Seguindo a metodologia do projeto, tanto o cronograma como o WBS, foram baseados na estrutura modular do projeto. E assim, as atividades do projeto são o desenvolvimento dos módulos do projeto. Como alguns módulos são muito complexos, suas atividades foram divididas em sub-atividades para uma boa gerência do projeto. Nos itens seguintes, analisaremos a elaboração do cronograma, do WBS e estimaremos os custos do projeto.

9.2 Cronograma

Para elaboração do cronograma consideramos o desenvolvimento em cascata das atividades, mesmo que a metodologia adotada para o projeto tenha sido o Scrum. Como mencionamos anteriormente o cronograma é uma *baseline* e,

portanto, sua função é estabelecer prazos para as atividades.

Na figura 15 abaixo temos o cronograma do projeto.

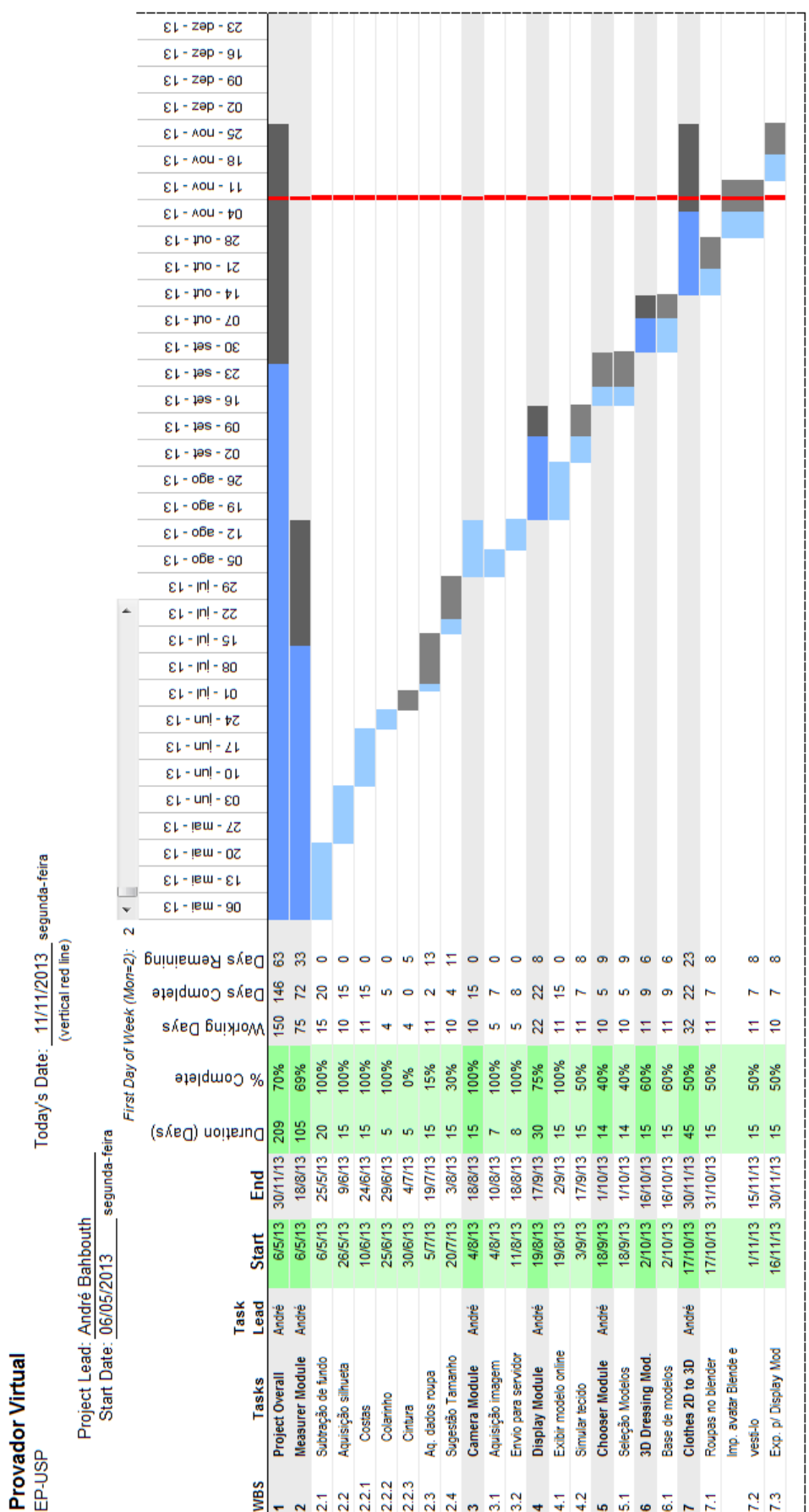


Figura 15: Cronograma do projeto.

A partir do cronograma, é possível extrair diversos dados do projeto. Na figura 16 abaixo vemos, por exemplo, a duração estipulada para o desenvolvimento de cada módulo.

Módulo	Duração (dias)
Measurer Module	105
Camera Module	15
Display Module	30
Chooser Module	14
3D Dressing Module	15
Clothes 2D to 3D converter	45
Total	224

Figura 16: Duração estipulada para cada módulo

O *Measurer Module* é o módulo que requer o maior esforço, devido à complexidade de se aferir as medidas do corpo humano corretamente, com apenas poucas fotos de entrada. O módulo de implementação mais rápida é o *Chooser Module*, considerando que optamos por utilizar modelos prontos e já vestidos e selecioná-los não é uma tarefa complexa.

Denote-se que o tempo de projeto é curto e seus prazos são exíguos. Neste sentido, trata-se de um projeto arrojado. Em que pese tais circunstâncias, garantimos o prazo de entrega do projeto que tem data final definida para 30/11/2013.

Ademais, salientamos que no cronograma existem algumas tarefas previstas para o futuro que já foram executadas e, outras que já deveriam ter sido feitas, mas ainda não foram realizadas. Como explicamos no capítulo 05, relativo à metodologia do projeto, isto ocorre em razão do cronograma se basear no método de desenvolvimento em cascata, que tem como modelo de desenvolvimento um software sequencial e que o cronograma trata-se uma *baseline* para o acompanhamento das atividades. Para maiores detalhes sobre o modelo em cascata vide o apêndice A. Portanto, pelo cronograma apenas ao final da execução de um mó-

dulo, inicia-se a execução do próximo com mostra a figura 17 abaixo.

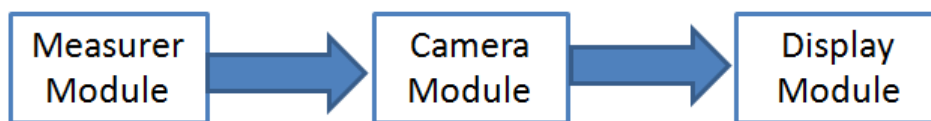


Figura 17: Desenvolvimento dos módulos fluindo sequencialmente como em uma cascata

Entretanto, como salientamos no capítulo 05, relativo à metodologia do projeto, o método adotado para o desenvolvimento do Proveedor Virtual foi o Scrum. Para mais informações sobre o processo do *Scrum*, vide o apêndice B.

Nesse sentido, o projeto do Proveedor Virtual foi desenvolvido por pedaços de cada módulo e, em seguida, foi feita uma integração e um teste, gerando um *release*¹. A cada *release*, novas *features* foram incorporadas ao produto de maneira incremental. Assim, a qualquer instante, durante o ano de elaboração do projeto, se o cliente exigisse uma demonstração do produto ou até mesmo alterasse seus requerimentos, sempre haveria um produto para mostrar. Ademais, por este método é possível serem realizadas alterações no sistema facilmente. Isso explica porque algumas *features* de módulos que já estavam previstas para serem implementadas ainda não foram e outras que seu desenvolvimento estava previsto para uma data futura já foram implementadas.

Como podemos ver a barra superior do cronograma (Figura 15) indica a porcentagem total do projeto completa que é de 70% (setenta por cento). Para o acompanhamento final das tarefas, que correspondem a 30% (trinta por cento) do projeto, elaboramos o cronograma da corrida final, apresentado na figura 18 abaixo.

¹Em gerência de projetos chama-se um *release* a liberação de um *software novo*.

Provedor Virtual
EP-USP

Today's Date: 15/11/2013 sexta-feira
(vertical red line)

Project Lead: André Bahbouth
Start Date: 06/05/2013 segunda-feira

First Day of Week (Mon=2): 2

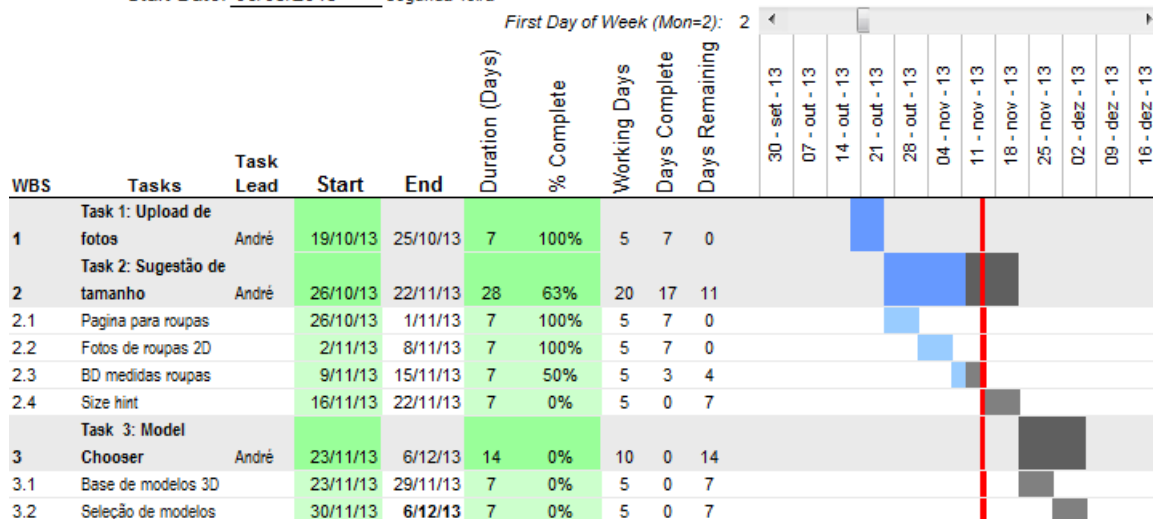


Figura 18: Cronograma da corrida final

Para elaboração do cronograma da corrida final, listamos todas as tarefas remanescentes do projeto de maneira detalhada e estipulamos as prioridades e os prazos. A tarefa prioritária era dar ao usuário a opção de carregar uma foto sua, sem precisar tirar uma foto nova. Outra prioridade do projeto se referia à funcionalidade “sugestão do tamanho” do módulo *Measurer Module*. Esta tarefa foi dividida em atividades que ainda não haviam sido concluídas.

O novo cronograma para a corrida final fixou como data final da entrega do projeto o dia 6/12/2013.

9.3 Análise dos Recursos

Para avaliarmos o custo do projeto, consideramos como único recurso do projeto a mão de obra do desenvolvedor do projeto. A remuneração do desenvolvedor

é igual à de um estagiário, que ganha em média R\$12,00 (doze reais) por hora de trabalho. A partir deste custo e alocando o estagiário desenvolvedor em todas as tarefas do projeto, chegamos ao WBS, descrito nas figuras 19, 20 e 21 abaixo.

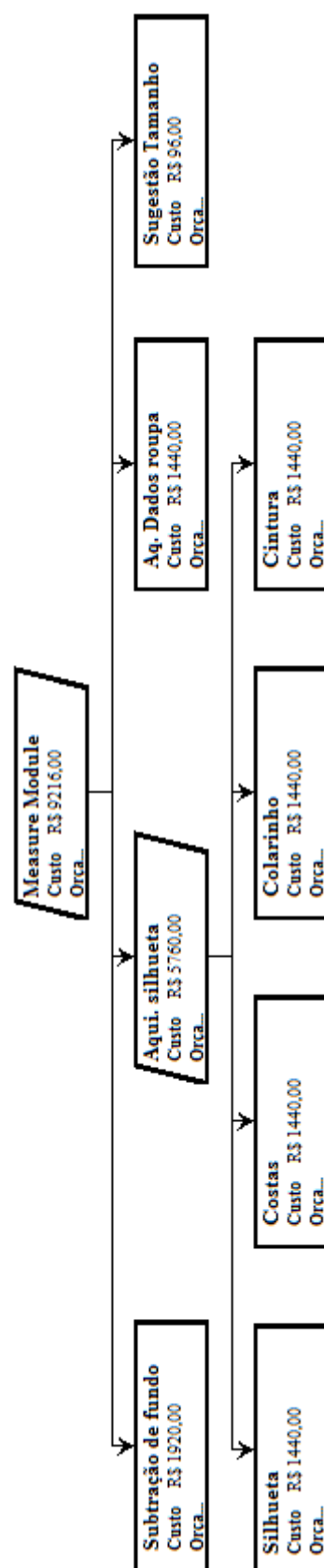


Figura 19: EAP do *Measurer Module*

Como o *Measurer Module* é o módulo que mais exigiu esforço e, portanto, maior número de homens horas, ele é o módulo mais caro do projeto com valor de R\$ 9.216,00 (nove mil, duzentos e dezesseis reais).

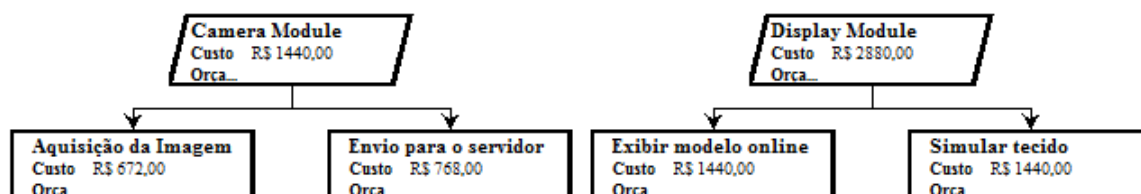


Figura 20: EAP do *Camera Module* e *Display module*

O *Camera Module* e o *Display Module* são módulos com poucas atividades, que não exigiram muitas horas de trabalho. Os custos destes módulos são R\$ 1.440,00 (um mil, quatrocentos e quarenta reais) para o *Camera Module* e R\$ 2.880,00 (dois mil, oitocentos e oitenta reais) para o *Display Module*.

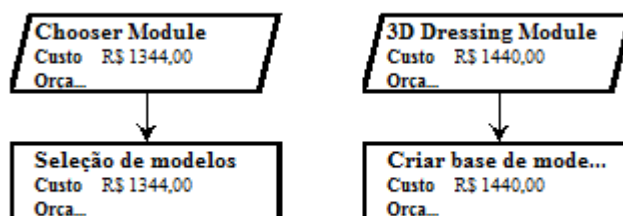


Figura 21: EAP do *Chooser Module* e *Dressing Module*

O *Chooser Module* e o *3D Dressing Module* são módulos que possuem uma única atividade. No caso do *3D Dressing Module*, salientamos que a criação da base de modelos é feita com auxílio da ferramenta *Marvelous Designer* e o valor de sua licença não foi considerada. O valor do *Chooser Module* é de R\$ 1.344,00 (um mil, trezentos e quarenta e quatro reais) e do *3D Dressing Module* é de R\$ 1.440,00 (um mil, quatrocentos e quarenta reais).

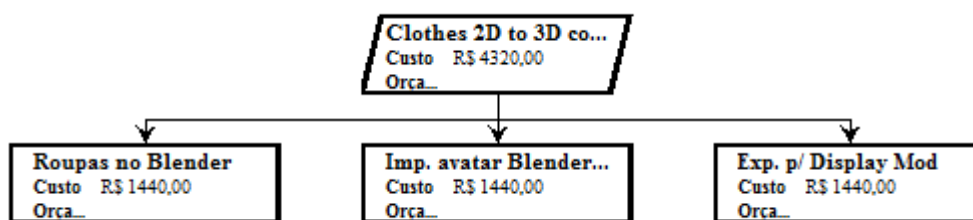


Figura 22: EAP do *Clothes 2D to 3D converter*

O *Clothes 2D to 3D Converter* é o módulo de modelagem das roupas no *Blender*, que é feito manualmente. Esse trabalho pode ser feito por um modelista. Porém, como sua execução limita-se a apenas um desenvolvedor, o tempo para sua implementação é longo. Assim, o *Clothes 2D to 3D converter* tem custo estimado de R\$ 4.320,00 (quatro mil, trezentos e vinte reais).

Com estes dados podemos estimar o custo deste projeto, somando-se o custo de todos os seus módulos, na importância de R\$ 20.640,00 (vinte mil, seiscentos e quarenta reais). Trata-se de um projeto econômico e extremamente atraente, considerando o seu potencial de mercado.

A viabilidade do projeto está demonstrada em face do cumprimento de seu prazo de entrega para dezembro e do EAP, que comprova o seu baixo custo.

10 PLANO DE TESTES

Neste capítulo vamos relatar alguns testes, a que o sistema foi submetido. Assim, analisaremos o atual estágio do projeto e avaliaremos as suas limitações. Os testes serão separados por módulos e, posteriormente, serão feitos alguns testes de performance no sistema como um todo. A começar pelo módulo de captura de imagem *Camera Module*, temos os seguintes testes e resultados.

Camera Module			
Teste	Status		Observações
	27/set	15/nov	
Abrir camera, capturar a imagem e exibi-lá para usuário.	Passou	Passou	
Enviar a imagem para o servidor e salvar imagem no banco de dados.	Passou	Passou	
Chamar o Measurer Module com a correspondente imagem como entrada para processamento e	Passou	Passou	
Resgatar imagem já salva no servidor e chamar Measurer	On going	Passou	
Exibir imagens já salvas pelo	Passou	Passou	

Figura 23: Testes aplicados ao *Camera Module*.

Os testes demonstraram que o módulo de aquisição de imagem e transporte atende a todas as suas funcionalidades. Futuramente deverão ser implementadas algumas *features*, a fim de que este módulo possibilite maior interatividade com o usuário. A seguir, passaremos a análise do módulo de medição.

Measurer Module			
Teste	Status		Observações
	27/set	15/nov	
Aquisição de foto com fundo qualquer e uma roupa qualquer	Falha	Falha	Funciona em alguns casos, nos outros ocorre erro na de subtração de fundo. O algoritmo atualmente implementado depende parcialmente da cor do usuário e do fundo para fazer a segmentação.
Aquisição de foto com fundo qualquer e com uma roupa monocromática diferente da cor do fundo.	Falha	Falha	Ajustes feitos no <i>Measurer Module</i> fizeram que o funcione na maior parte dos casos, porém eventualmente ainda ocorrem falhas.
Aquisição de foto com fundo monocromático e com uma roupa monocromática.	Passou	Passou	
Desempenho: tempo de execução < 40 s	Falha	Passou	Para aumentar o desempenho do algoritmo as imagens são redimensionadas antes de serem processadas.

Figura 24: Testes aplicados ao *Measurer Module*.

Durante a execução do projeto percebemos que existem algumas dificuldades com o tratamento da foto enviada pelo usuário ao sistema. Neste projeto desejamos garantir que o algoritmo de segmentação funcione corretamente, quando o usuário está vestido com uma roupa monocromática em um ambiente de fundo monocromático, porém diferente da cor da sua roupa. Após vários ajustes, o *Measurer Module* funciona na maior parte dos casos, quando temos um fundo qualquer e uma roupa monocromática. Porém, como o sistema ainda é passível de falhas nesse caso, não é possível reportar que o módulo passou por esse teste.

Como afirmamos no Capítulo 04, para a aferição das medidas é necessário que o usuário use roupas justas, caso contrário suas medidas serão aumentadas. Várias melhorias já foram implementadas neste módulo. Respeitando-se o crité-

rio para o correto funcionamento da segmentação, o usuário pode tirar uma foto de corpo inteiro ou somente da parte superior do corpo que o algoritmo é funcional. O algoritmo também suporta diversos formatos e resolução de imagens. Para reduzir a diferença de desempenho do algoritmo com relação à imagem de entrada, todas as imagens antes de serem processadas são proporcionalmente redimensionadas. Dessa maneira conseguimos diminuir drasticamente o tempo de processamento do módulo e também equaliza-lo. Na figura 25 abaixo, vemos o desempenho com algumas fotos de entrada.

Imagem	Resolução	Tempo de processamento
JPEG (compressão = 0%)	2432 x 4320	1 min 12 s
JPEG (compressão = 90%)	2433 x 4320	53 s
JPEG (compressão = 0%)	1216 x 2160	48 s
JPEG (compressão = 0%)	608 x 1080	4 s
JPEG (compressão = 0%)	640 x 480	Falha

Figura 25: Desempenho do *Measurer Module*.

Para medirmos o desempenho deste módulo, partimos de uma mesma imagem *jpeg* e aplicamos compressões e redimensionamentos na mesma. Tendo como entrada a imagem original, o módulo demora mais de um minuto para processá-la, o que torna inviável utilizar a imagem tal qual. Aplicando uma compressão e processando novamente, percebemos uma queda no tempo de processamento. Porém, além de deixar as medidas extremamente imprecisas devido à alta distorção da imagem, o tempo de processamento é ainda muito elevado. Assim, percebe-se que a compressão da imagem não é uma boa solução. Optamos, portanto, pelo seu redimensionamento. Dividimos a imagem por dois e obtivemos uma redução para 48 (quarenta e oito) segundos. Em seguida, dividimos por dois novamente a imagem chegando em um tempo de processamento de 4 (quatro) segundos. No último caso, tentado redimensionar a imagem sempre

para uma resolução fixa o algoritmo chega a falhar. Portanto, a melhor solução é redimensionar a imagem de maneira que ela se aproxime a uma resolução 600 x 1000 *pixels*, mas não impondo um redimensionamento fixo que pode distorcer a imagem e fazer com que o *Measurer Module* falhe.

Nos casos em que as diversas etapas de processamento funcionam com sucesso, é possível se obter as medidas do colarinho, a distância entre os ombros e o tamanho do tronco do indivíduo. Para obtenção das medidas de cintura seria necessária uma foto lateral do usuário que desde a elaboração do cronograma da corrida final, não está sendo solicitado pelo sistema. Para a obtenção das medidas em centímetros é usado um CD como referência ou é solicitado ao usuário, que indique onde está a ponta de sua cabeça ou de seus pés.

Ainda não foram efetuados testes suficientes com uma amostra representativa para acurar a precisão do sistema. Uma ficha pedindo medidas aferidas pelo módulo está sendo distribuída para que se obtenha uma base representativa como teste do módulo. Um exemplo segue na figura 26 abaixo.

	Nome:	Lucas Giacomini		Altura:	178
Superior	REF.	DESCRIÇÃO DA MEDIDA	Distorção	S. Distorção	Contorno
	A	Chest (<i>Peito</i>)	35	32	98
	B	Total Length (<i>Comprimento total</i>)	53,5	53,5	NA
	E	Armhole (<i>Cava</i>)	22	22	NA
	F	Bottom (<i>Barra</i>)	37,5	34,5	
	G	Neck Width - (<i>Largura do pescoço</i>)	13	13	40
	H	Front Neck Drop (<i>Abertura decote frente</i>)	10,5	11	NA
	I	Back shoulder to shoulder	48,5	45	NA
Inferior	A	Total Length (<i>Comprimento total</i>)	99	99	NA
	C	STRETCHED WAIST (<i>cintura tencionada</i>)	38	28	88
		STRETCHED WAIST L. (<i>cintura tencionada</i>)	26	23	NA
	E	Hip (<i>Quadril</i>)	38	32	99
		Hip L. (<i>Quadril</i>)	25	27	NA

Figura 26: Ficha cadastral para acurar precisão do sistema

Os testes aplicados no *Display Module*, foram transcritos na figura 27 abaixo.

Display Module			
Teste	Status		Observações
	27/set	15/nov	
Capturar avatar enviado pelo <i>3D Dressing Module</i> e enviar para o navegador do usuário.	Passou	Passou	
Carregar no navegador o modelo e suas texturas.	Falha	Passou	Anteriormente o modelo já era carregado, mas seu número de polígonos era reduzido extremamente de maneira que o modelo fica com aspecto estranho. Existia também uma falha ao carregar algumas texturas. Esses problemas foram resolvidos.
Prover simulação de tecido para o modelo.	On going	Passou	A partir do cronograma da corrida final, a simulação de tecido passou a ser executada no <i>Marvelous Designer</i> , e o modelo vestido salvo no Banco de dados.

Figura 27: Testes aplicados ao *Display Module*

O *Display Module* é capaz de carregar o modelo enviado pelo *3D Dressing Module* e permitir ao usuário navegar ao redor do *avatar*. A simulação de tecido foi deixada por conta do *Marvelous Designer* a partir do cronograma da corrida final. Após algumas alterações no módulo, foi corrigida a falha no carregamento do modelo e, também, de suas texturas.

Passaremos a análise dos testes aplicados ao *Chooser Module*, consubstanciados na Figura 28 abaixo.

Chooser Module			
Teste	Status		Observações
	27/set	15/nov	
Seleção do sexo correto.	On going	Passou	Foi criado um banco de dados com modelos masculinos e femininos de diversos tamanhos.
Seleção do modelo com medidas corporais adequado.	On going	Passou	

Figura 28: Testes aplicados ao *Chooser Module*

Dois testes simples foram aplicados ao *Chooser Module* a fim de medir sua capacidade de selecionar os modelos. O sexo é solicitado como entrada para o usuário e não é uma tarefa difícil de ser cumprida. A seleção de modelos foi implementada selecionando-se o *avatar* de um banco de dados com alguns modelos.

Não temos testes aplicados ao *3D Dressing Module* e ao *Clothes 2D to 3D Converter*, pois esses módulos executam tarefas de modelagem de roupas que são feitas manualmente.

11 RESULTADOS

Nesse capítulo, vamos mostrar como ficaram as UI¹e o *wireframe*²do projeto. Tratando-se de um site de vendas, começamos com uma página de um e-commerce como pode ser visto pelas figuras 29 e 30 abaixo.

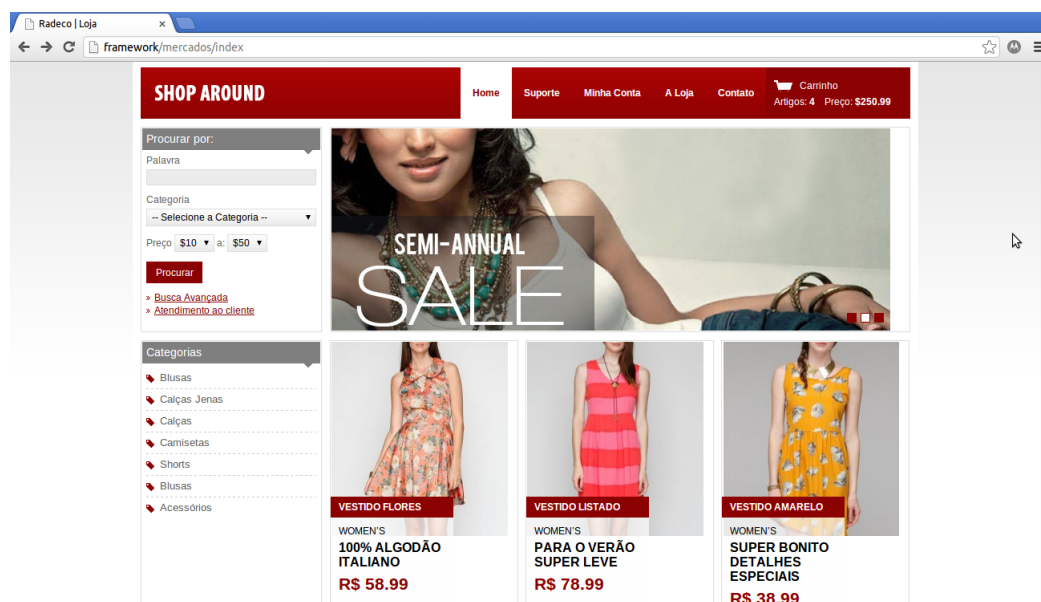


Figura 29: UI do e-commerce do Proveedor Virtual

²Do inglês *User Interface*, refere-se as interfaces gráficas do Proveedor Virtual.

²A sequência das UIs é conhecida como *wireframe*.

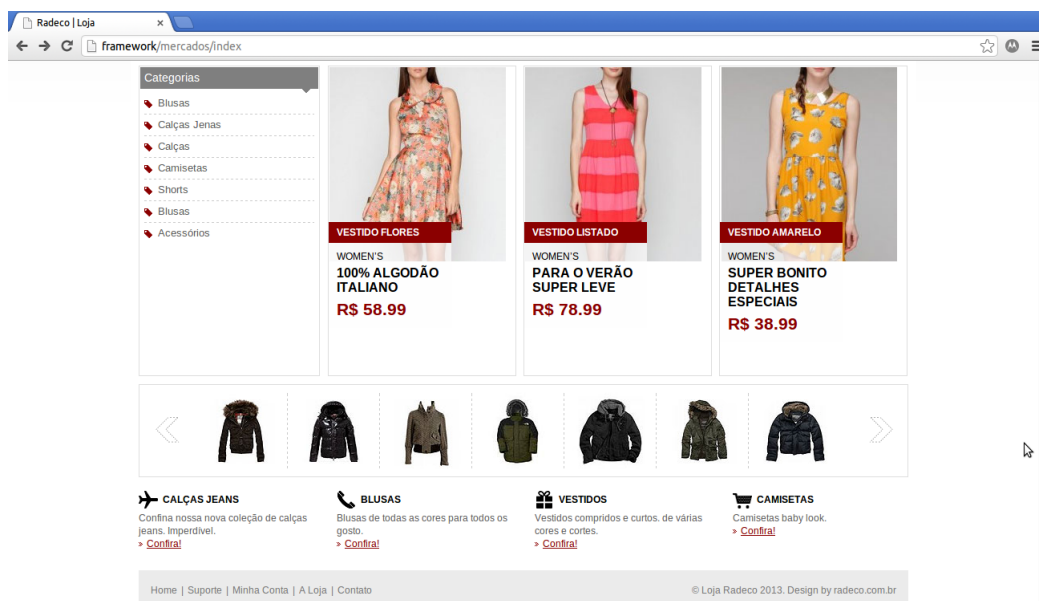


Figura 30: UI do e-commerce do Proveedor Virtual (final da página)

Selecionando umas das opções de roupas, entramos na página do Proveedor Virtual em si. A primeira pagina do Proveedor Virtual é referente ao *Camera Module* e pede uma foto como entrada, essa foto pode ser carregada da memória ou tirada uma nova. Esse página pode ser vista na figura 31 abaixo.

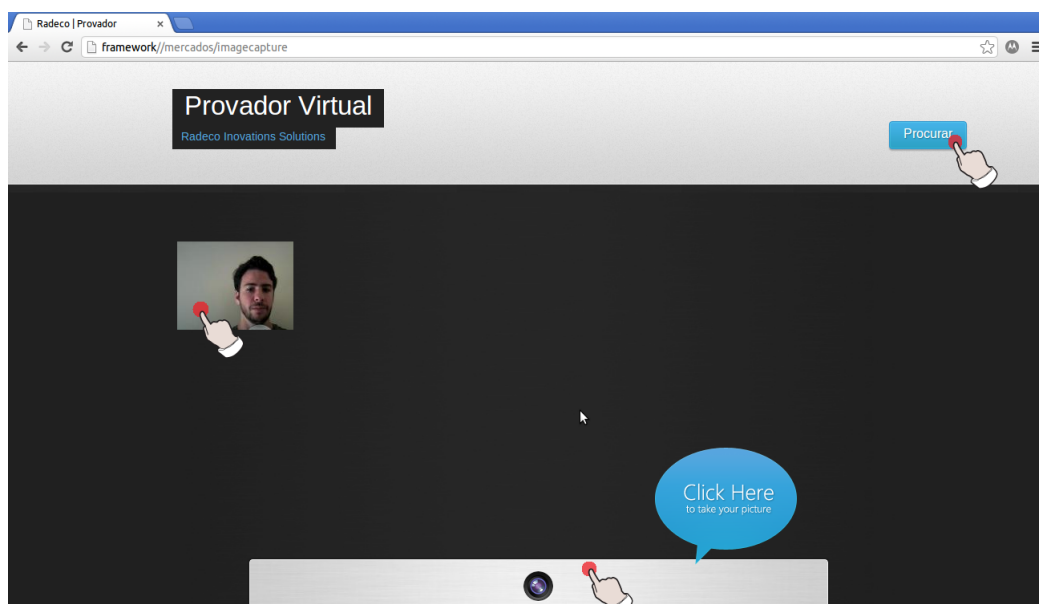


Figura 31: Primeira página do Proveedor Virtual

A mão indica os possíveis cliques que é possível fazer na página. Podemos ver as imagens que já foram carregadas anteriormente, clicando nelas, como mostra a figura 32 abaixo.

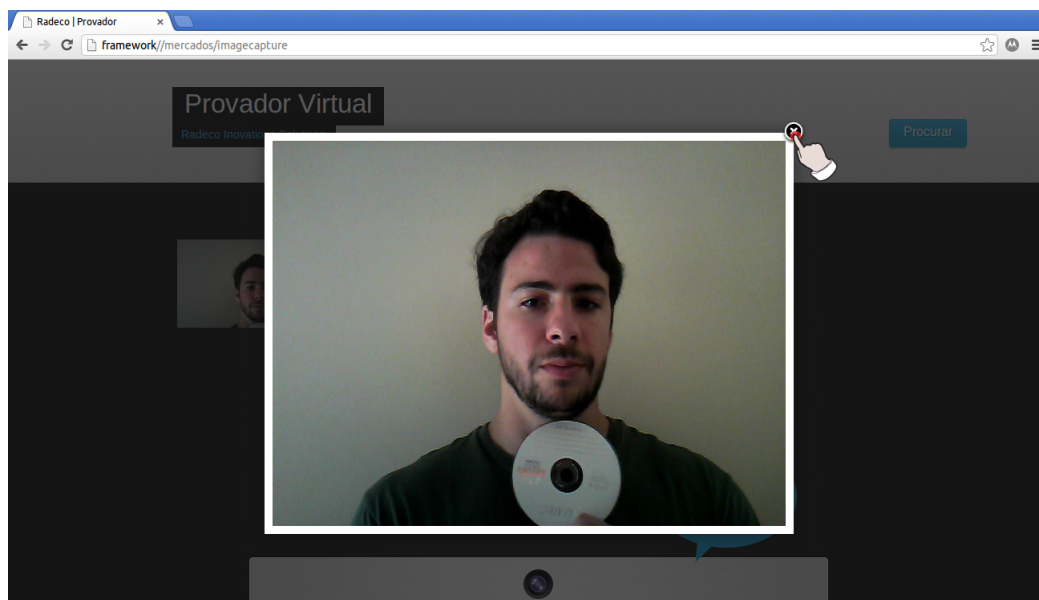


Figura 32: Visualização de fotos já carregadas

Caso o usuário deseje trocar de foto e tirar uma nova ele pode clicar na barra inferior que abrirá a webcam do computador, como mostra a figura 33 abaixo.

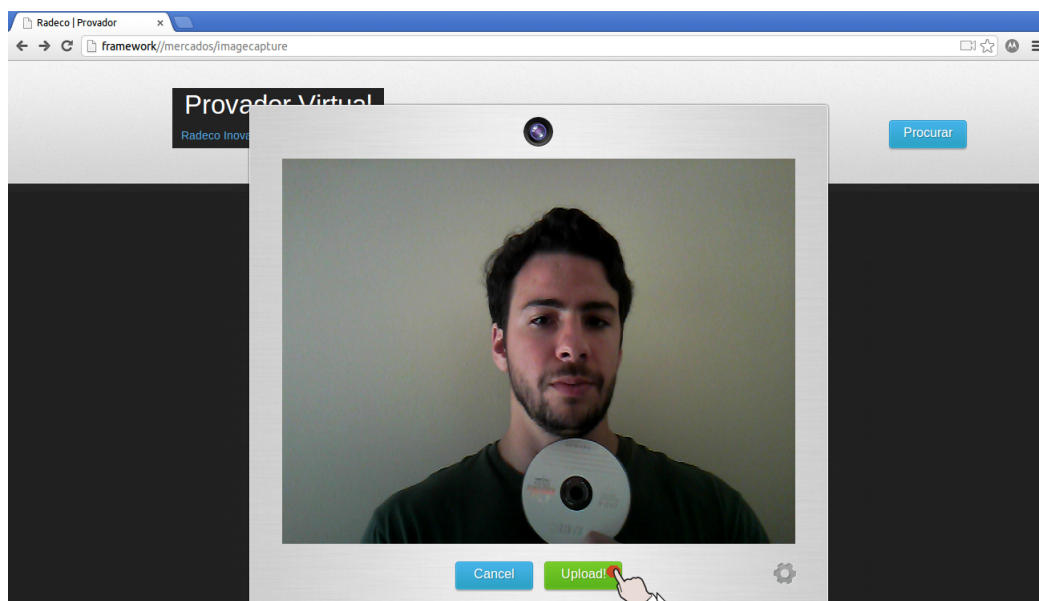


Figura 33: Captura de uma nova imagem

Como pode ser visto na imagem de uma webcam não cabe uma foto de corpo inteiro do indivíduo. Portanto, tiramos uma foto apenas da parte superior do corpo. Nesse caso é necessário que o usuário segure um CD.

Ao clicar em *Upload* , para carregar a imagem, o sistema pede o gênero e a altura do usuário, como mostra a figura 34 abaixo.

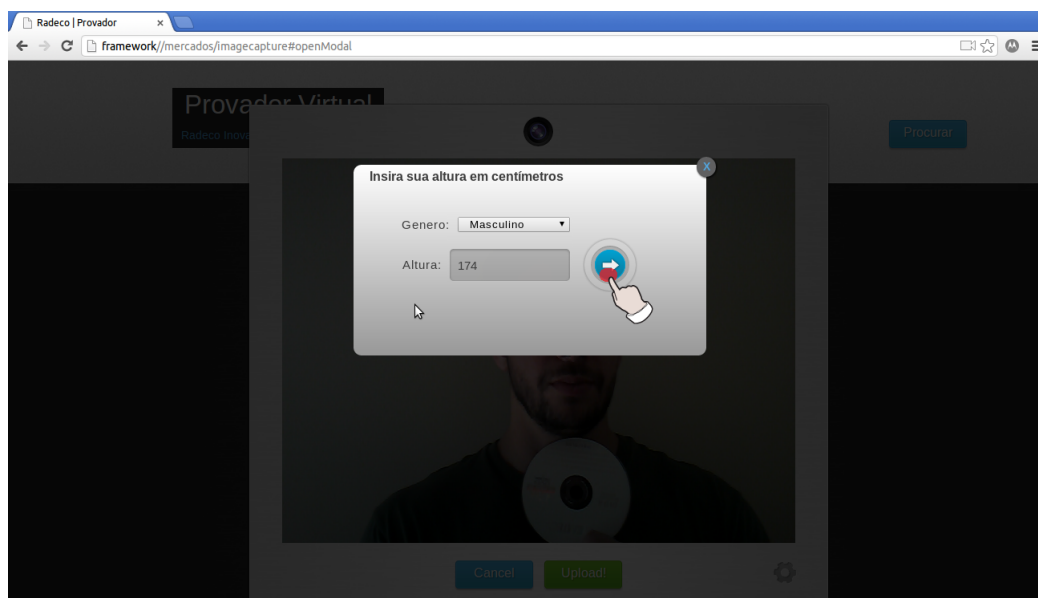


Figura 34: Janela de informações solicitadas ao usuário

Seguindo a indicação do marcador da figura 34, damos entrada dos dados para o *Measurer Module* que fará a aferição de medidas. Antes de mostrar como são exibidos os resultados do *Measurer Module*, vamos voltar para página inicial do Provador Virtual da figura 31 e clicar em “Procurar”. Essa opção é quando o usuário decide carregar uma nova imagem para o sistema. Vide figura 35 abaixo.

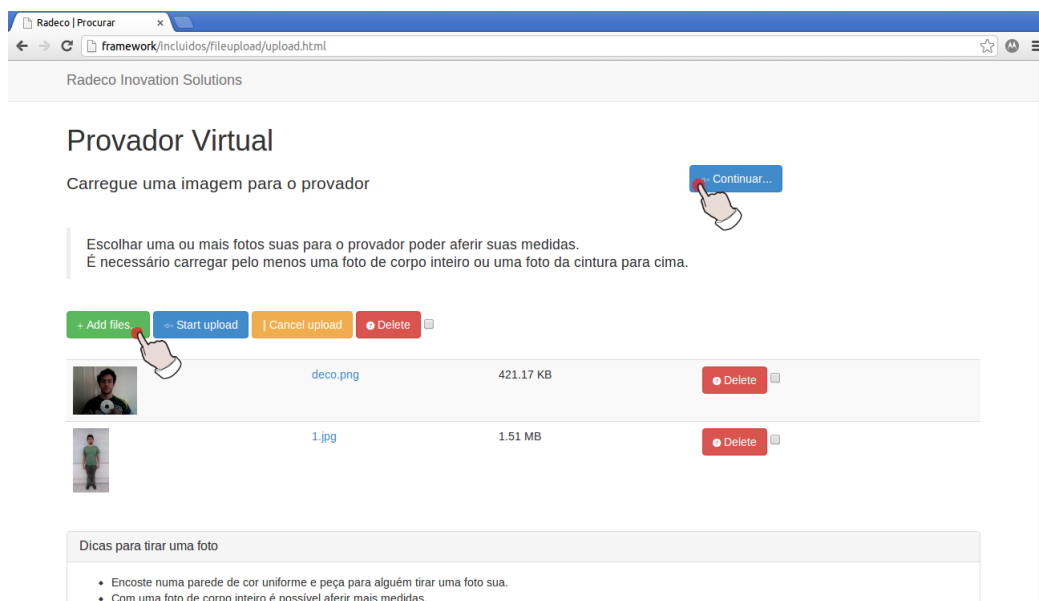


Figura 35: Carregando uma nova imagem

Observando a figura 35, vemos que as imagens que já foram previamente carregadas ficam listadas e que é possível deletá-las. Clicando em “Add Files...” abre-se uma janela, conforme visto na figura 36 abaixo.

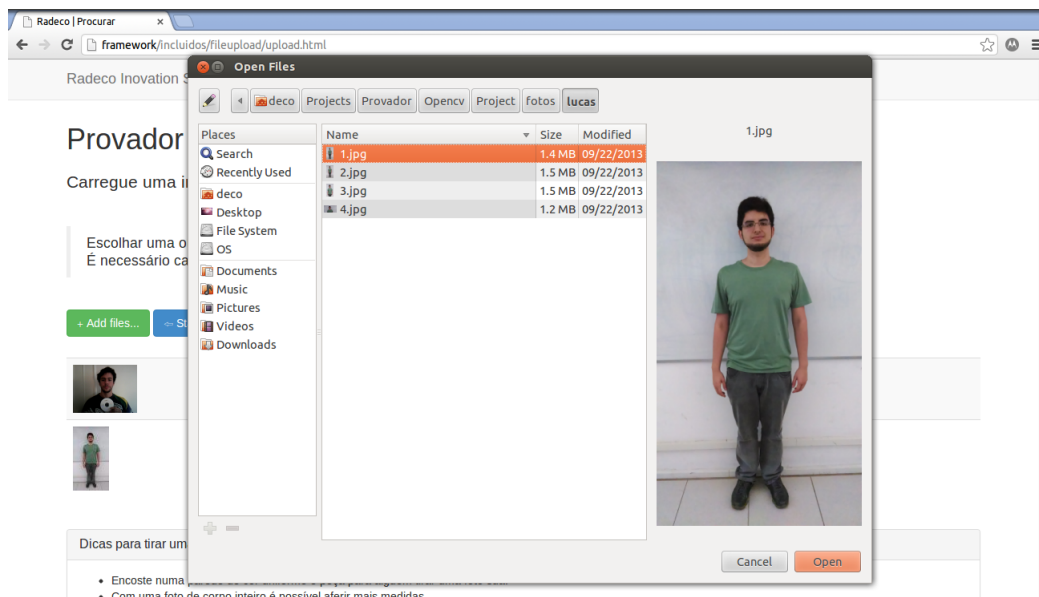


Figura 36: Navegador de arquivos do usuário

A partir daí abrimos o navegador de arquivos do computador do cliente. Assim,

o usuário escolhe uma foto e clica em abrir para dar continuidade ao processo. Assim, que a foto é carregada o usuário clica em “Continuar” como indicado pelo marcador na figura 35 e uma nova janela pedindo dados do usuário é aberta. Como verificamos na figura 37 abaixo.

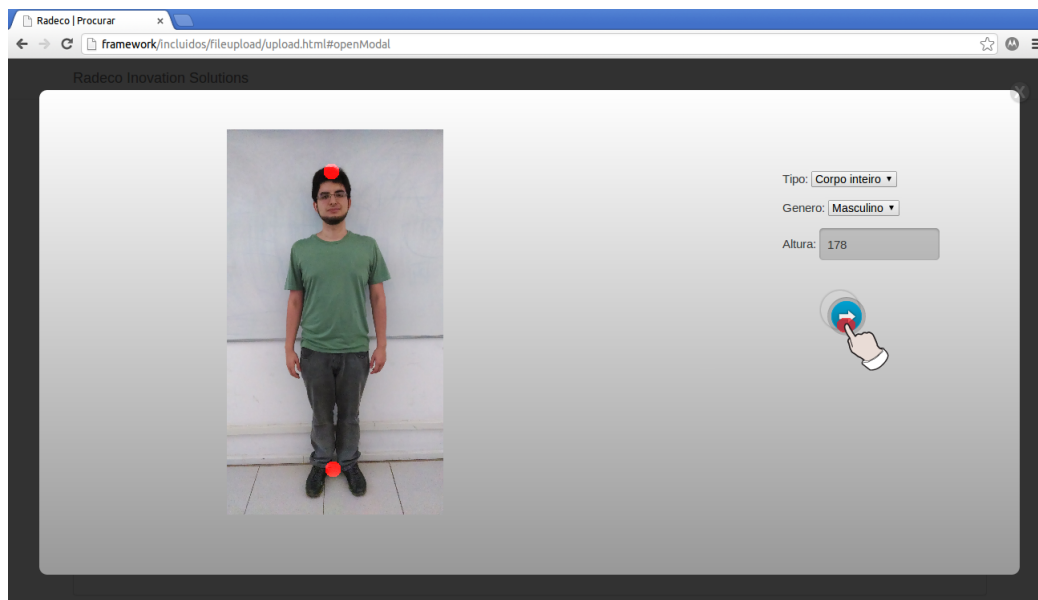


Figura 37: Navegador de arquivos do usuário

Conforme verificamos na figura 37, na parte esquerda é solicitado ao usuário o tipo de imagem, o gênero e a altura. Para tipo de imagem, o sistema solicita se o usuário está dando como entrada uma foto de corpo inteiro ou uma foto apenas da parte superior. Para uma foto apenas da parte inferior é solicitado que o usuário segure um CD próximo ao corpo. Para uma foto de corpo inteiro é solicitado que o usuário marque na foto a ponta de sua cabeça e o começo dos pés (pontos em vermelho na foto do lado esquerdo da figura 37).

Clicando no botão indicado pelo marcador na figura 37, enviamos então os dados para processamento no *Measurer Module*. Os resultados das medições são exibidos conforme na figura 38 abaixo.

Provador Virtual
Radeco Innovations Solutions

Resultados das medidas Antropométricas

Seu tamanho de roupa é: M

Distância entre ombros	49 cm		Tórax	22 cm	
Largura do pescoço	14 cm		Braço	57 cm	
Altura	170 cm		Perna	90 cm	
Tamanho do tronco	49 cm		Coxa	42 cm	

Próximo

Fonte: radeco.com.br

Figura 38: Resultado das medidas antropométricas

Na figura 38, vemos uma tabela com uma sugestão de tamanho de roupa na primeira linha e as medidas antropométricas obtidas pelo sistema nas demais linhas da tabela. As medidas obtidas são: (i) distância entre ombros; (ii) largura do pescoço; (iii) altura; (iv) tamanho do tronco; (v) comprimento do tórax; (vi) comprimento dos braços; (vii) comprimento das pernas; (viii) comprimento das coxas.

A primeira fase do provador está completa. O usuário já aferiu suas medidas e também já sabe seu tamanho de roupa para a roupa escolhida na página do *e-commerce* (figuras 29 e 30). Agora entrará em ação o *Chooser Module* que selecionará o *avatar* mais adequado e o *Display Module* que exibirá o *avatar*. Para continuar o usuário clica em “Próximo” e, assim, próxima página é vista na figura 39 abaixo.

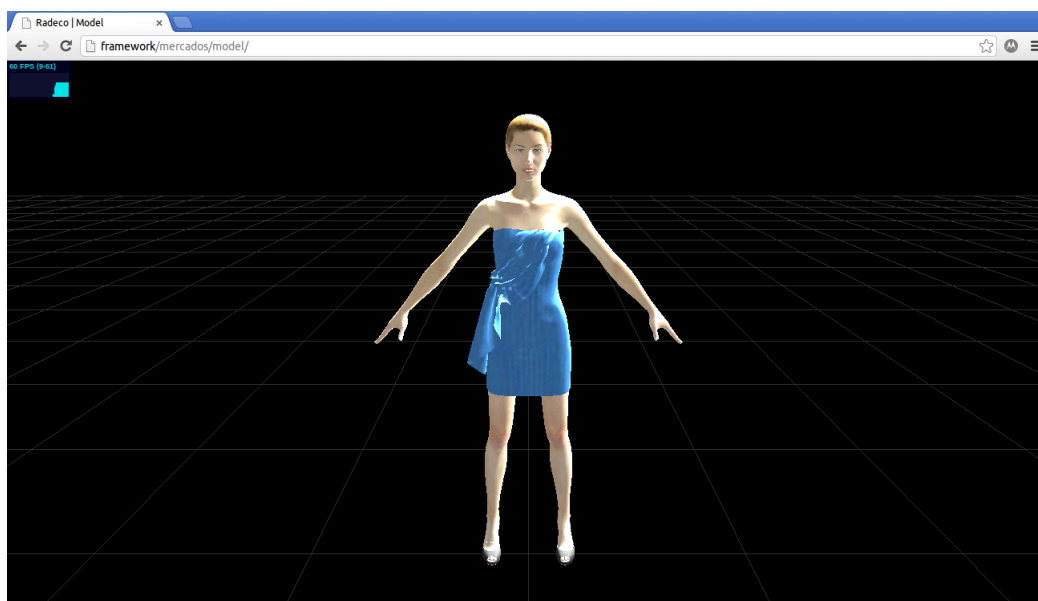


Figura 39: Avatar magro e alto vestido com a roupa selecionada

O avatar da figura 39 é um avatar feminino alto e magro. Como explicamos no capítulo das soluções escolhidas, formamos um banco de dados com diversos avatares. Nas figuras 40 e 41 podemos ver outros avatares femininos.

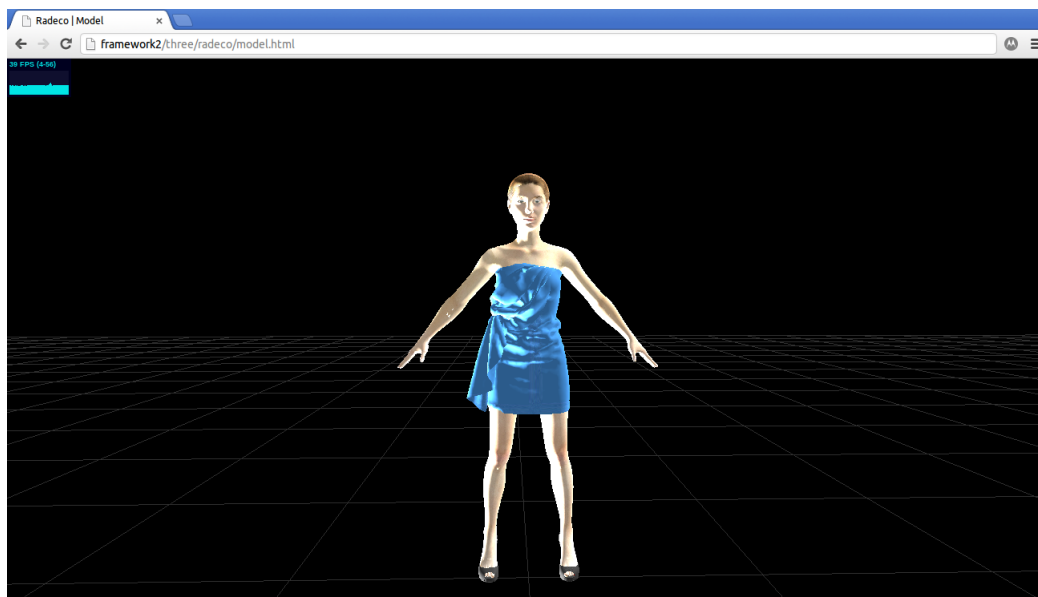


Figura 40: Avatar magro e baixo vestido com a roupa selecionada

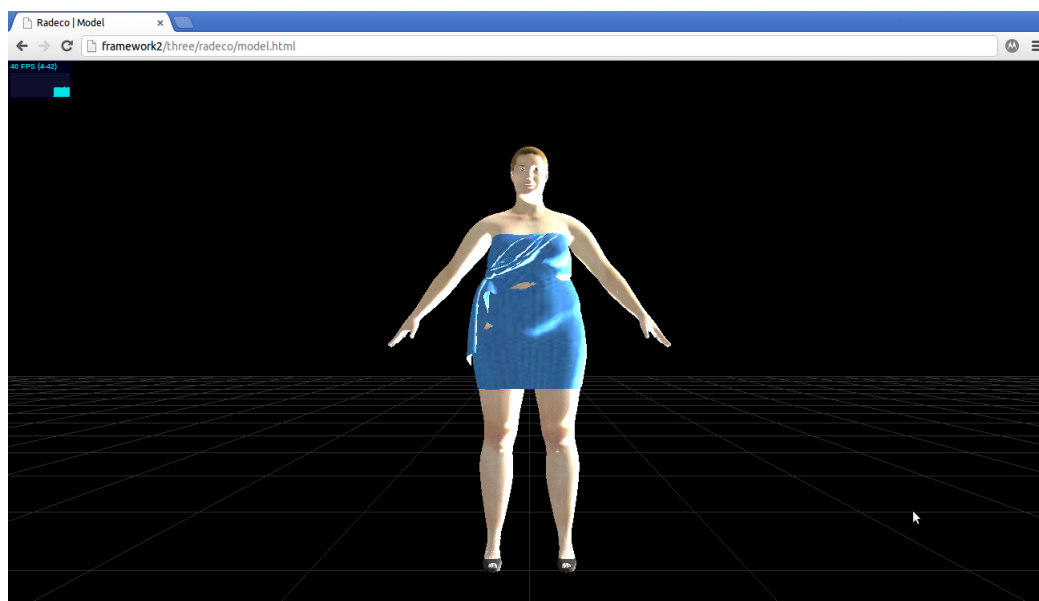


Figura 41: *Avatar* gordo e alto vestido com a roupa selecionada

Comparando as figuras 39, 40 e 41 percebemos que para o *avatar* da figura 40 (magro e baixo) o vestido está comprido e para o *avatar* da figura 41 (alto e gordo) o vestido está tão apertado que chega a rasgar. Portanto, percebemos que esse vestido é ideal para o *avatar* da figura 39 (magro e alto).

12 CONCLUSÕES

Afirmamos ao longo deste projeto que, o sistema desenvolvido neste projeto, denominado Proveedor Virtual, tem por escopo facilitar e propiciar as aquisições de vestuário pela internet, diminuindo a alta taxa de devolução das roupas adquiridas por tal meio.

Iniciamos o projeto procedendo ao levantamento dos dados, para entender como é feita a produção de roupas em larga escala e do estado da arte. Ao analisarmos os dados sobre o *e-commerce* no Brasil e as tecnologias utilizadas para tal fim, constatamos que existe um nicho de mercado não explorado adequadamente e de extremo potencial.

Buscando atender às necessidades do usuário de aquisições de vestuário pela internet, as funcionalidades do projeto foram definidas. Todas as alternativas de soluções foram esmiuçadas, com o intuito de mostrar dentre as tecnologias atuais aquelas que poderiam auxiliar na confecção de um proveedor virtual.

Para a escolha das soluções que melhor se enquadravam ao escopo do projeto, usamos quatro critérios: tempo de execução do projeto, qualidade, integração, custo (possibilidade de utilização de softwares livres) e desempenho. A partir destes critérios, algumas soluções foram experimentadas para, finalmente, adotarmos as soluções pertinentes.

Para finalizar, o plano gerencial do projeto foi construído juntamente com o início do seu desenvolvimento. Estabelecido o cronograma, garantimos o prazo

de entrega do projeto para dezembro e com o EAP demonstramos que o valor de seu custo total será de aproximadamente R\$ 20.640,00 (vinte mil, seiscentos e quarenta reais). Restou comprovada, portanto, a viabilidade deste projeto.

O protótipo do projeto cumpre com todas as funcionalidade descritas nesse trabalho, quais sejam a aferição de medidas, a sugestão de tamanho de roupa e a visualização do avatar vestido. Nesse sentido o projeto atende a todos requisitos técnicos e aos objetivos de melhora e incrementação da venda de roupas de vestuário pela internet a que se prôpos.

13 REFERÊNCIAS

- [1] <http://opencv.org/>
- [2] <http://www.blender.org/>
- [3] <http://www.marvelousdesigner.com/>
- [4] <http://www.xbox.com/en-US/kinect/>
- [5] <http://www.opengl.org/>

14 BIBLIOGRAFIA

- R. C. Gonzalez, and R. E. Woods, Digital Image Processing - second edition, Prentice Hall, 2002.
- G. Bradski and A. Kaehler, Learning OpenCV, O'Reilly, 2008.
- H. Pedrini, and W. R. Schwartz, Análise de Imagens Digitais - Princípios, Algoritmos e Aplicações, Thomson, 2008.
- H. Pedrini, and W. R. Schwartz, Análise de Imagens Digitais - Princípios, Algoritmos e Aplicações, Thomson, 2008.
- H. Pedrini, and W. R. Schwartz, Análise de Imagens Digitais - Princípios, Algoritmos e Aplicações, Thomson, 2008.
- Al Bovik (ed.), Handbook of Image Video Processing - second edition, Elsevier, 2005.
- L. F. Costa, and R. M. Cesar Jr., Shape Analysis and Classification, CRC Press, 2001.
- D. Salomon, Data Compression - The Complete Reference - 3rd ed., Springer, 2004.
- T. M. Mitchell, Machine Learning, WCB McGraw-Hill, 1997.
- J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hugues, Computer Graphics: Principles and Practice, Addison-Wesley, 1990.

- A. S. Glassner, Principles of Digital Image Synthesis, Morgan Kaufmann, 2 volumes, 1995.
- R. D. Lopes, Capítulo 3 ? Síntese de Imagens, em: O Multicomputador TRGR e a Paralelização da Síntese de Imagens, dissertação de mestrado, Escola Politécnica da USP, 1993.

APÊNDICE A – MODELO EM CASCATA

A.1 O Modelo

O modelo em cascata é um modelo de desenvolvimento de software seqüencial no qual o desenvolvimento é visto como um fluir constante para frente (como uma cascata) através das fases de análise de requisitos, projeto, implementação, testes (validação), integração, e manutenção de software. A origem do termo cascata é frequentemente citado como sendo um artigo publicado em 1970 por W. W. Royce; ironicamente, Royce defendia uma abordagem iterativa para o desenvolvimento de software e nem mesmo usou o termo cascata. Royce originalmente descreve o que é hoje conhecido como o modelo em cascata como um exemplo de um método que ele argumentava ser um risco e um convite para falhas.

A.2 Uso do Modelo

No modelo em cascata original de Royce, as seguintes fases são seguidas em perfeita ordem:

- Elicitação de requisitos
- Projeto
- Construção (implementação ou codificação)
- Integração

- Teste e depuração
- Instalação
- Manutenção de software

Conforme visto na figura 42 abaixo.

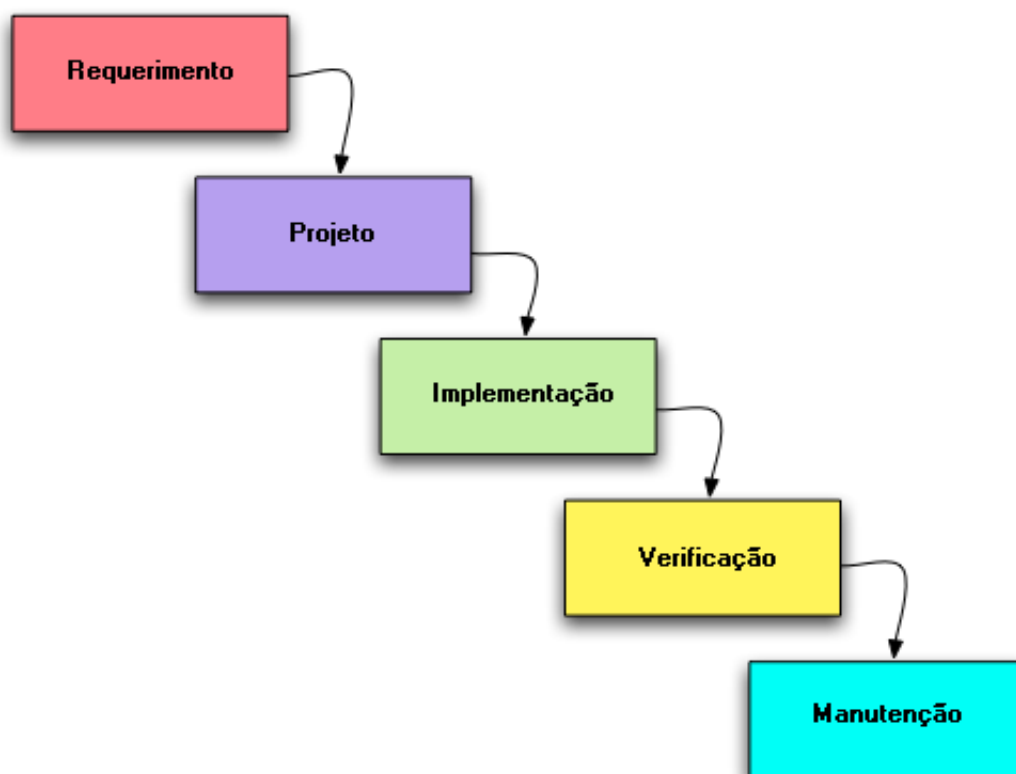


Figura 42: O Modelo em cascata estático. O andamento do processo flui de cima para baixo, como uma cascata

Para seguir um modelo em cascata, o progresso de uma fase para a próxima se dá de uma forma puramente sequencial. Por exemplo, inicialmente completa-se a especificação de requisitos elaborando um conjunto rígido de requisitos do software (Por exemplo, os requisitos para Wikipédia devem permitir edições anônimas de artigos; Wikipédia deve permitir às pessoas procurar pelas informações),

embora as especificações dos requisitos reais sejam mais detalhados, em um procedimento para projeto. O software em questão é projetado e um *blueprint* é desenhado para implementadores seguirem, este projeto deve ser um plano para implementação dos requisitos dados. Quando e somente quando o projeto está terminado, uma implementação para este projeto é feita pelos codificadores. Encaminhando-se para o próximo estágio da fase de implementação, inicia-se a integração dos componentes de software construídos por diferentes times de projeto. (Por exemplo, um grupo pode estar trabalhando no componente de página web da Wikipedia e outros grupos pode estar trabalhando no componente servidor da Wikipedia. Estes componentes devem ser integrados para juntos produzirem um sistema como um todo). Após as fases de implementação e integração estarem completas, o produto de software é testado e qualquer problema introduzido nas fases anteriores é removida aqui. Com isto o produto de software é instalado, e mais tarde mantido pela introdução de novas funcionalidades e remoção de defeitos.

Portanto o modelo em cascata move-se para a próxima fase somente quando a fase anterior está completa e perfeita. Desenvolvimento de fases no modelo em cascata são discretas, e não há pulo para frente, para trás ou sobreposição entre elas.

Contudo, há vários modelos em cascata modificados (incluindo o modelo final de Royce) que podem ser incluídos como variações maiores ou menores deste processo.

⁰fonte: wikipédia

APÊNDICE B – O MODELO *SCRUM*

B.1 O *Scrum*

O *Scrum* é um processo de desenvolvimento iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de software.

Scrum possui seu foco no gerenciamento de projeto da organização onde é difícil planejar à frente. Mecanismos do Controle de Processo Empírico, onde ciclos de *feedback* constituem o núcleo da técnica de gerenciamento que são usadas em oposição ao tradicional gerenciamento de comando e controle. É uma forma de planejar e gerenciar projetos trazendo a autoridade da tomada de decisão a níveis de propriedade de operação e certeza.

Apesar de a palavra não ser um acrônimo, algumas empresas que implementam o processo a soletram com letras maiúsculas como *Scrum*. Isto pode ser devido aos primeiros artigos de Ken Schwaber, que capitalizava *Scrum* no título.

Scrum não é um processo prescribente, ou seja, ele não descreve o que fazer em cada situação. Ele é usado para trabalhos complexos nos quais é impossível prever tudo o que irá ocorrer. Apesar de *Scrum* ter sido destinado para gerenciamento de projetos de software, ele pode ser utilizado em equipes de manutenção de software ou como uma abordagem geral de gerenciamento de projetos/programas.

B.2 *Sprint* (corrida, tiro)

Um *sprint* é a unidade básica de desenvolvimento em Scrum. *Sprints* tendem a durar entre uma semana e um mês, e são um esforço dentro de uma caixa de tempo (ou seja, restrito a uma duração específica) de comprimento constante. Vemos um *sprint* representado na figura 43 abaixo.

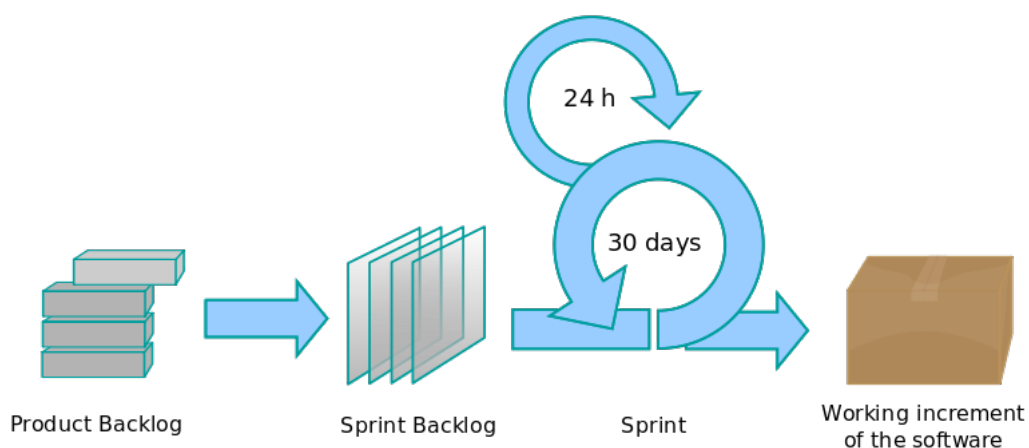


Figura 43: Processo de *Scrum*, um *sprint*

Cada *sprint* é precedido por uma reunião de planejamento (*Sprint Planning*), onde as tarefas para o *sprint* são identificadas e um compromisso estimado para o objetivo do *sprint* é definido e seguido por uma reunião de revisão ou de retrospectiva, onde o progresso é revisto e lições para os próximos *sprints* são identificadas.

Durante cada *sprint*, a equipe cria um incremento de produto potencialmente entregável (por exemplo, software funcional e testado). O conjunto de funcionalidades que entram em um *sprint* vêm do “*frontlog*” do produto, que é um conjunto de prioridades de requisitos de alto nível do trabalho a ser feito.

Quais itens do *backlog* que entram para o *sprint* são determinados durante a reunião de planejamento do *sprint* (*Sprint Planning*). Durante esta reunião, o

Product Owner informa a equipe dos itens no backlog do produto que ele ou ela quer concluídos.

A equipe então determina quantos eles podem se comprometer a concluir durante o próximo sprint, e registram isso no *backlog* do *sprint*. Durante um *sprint*, ninguém está autorizado a alterar o *backlog* do *sprint*, o que significa que os requisitos são congelados para esse *sprint*.

O desenvolvimento de cada *sprint* deve terminar na "caixa de tempo" prevista. Se os requisitos não são completados por qualquer motivo, eles são deixados de fora e voltam para o *backlog* do produto. Depois que um sprint é completado, a equipe demonstra como usar o software.

O *Scrum* permite a criação de equipes auto-organizadas, encorajando a co-localização de todos os membros da equipe e a comunicação verbal entre todos os membros e disciplinas da equipe no projeto. Um princípio chave do *Scrum* é o reconhecimento de que, durante um projeto, os clientes podem mudar de idéia sobre o que eles querem e precisam (muitas vezes chamados requisitos churn), e que os desafios imprevisíveis não podem ser facilmente tratados de uma maneira preditiva ou planejada tradicional. Como tal, o *Scrum* adota uma abordagem empírica, aceitando que o problema não pode ser totalmente entendido ou definido, focando na maximização da habilidade da equipe para entregar rapidamente e responder às necessidades emergentes.

⁰fonte: wikipédia