

UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE CIÊNCIAS - BAURU
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Bruno Fernandes Casella

Desenvolvimento de Aplicações Web
com Base em Modelagem 3D – Módulo Android

UNESP

2012

Bruno Fernandes Casella

Desenvolvimento de Aplicações Web
com Base em Modelagem 3D – Módulo Android

Orientador: Prof. Dr. José Remo Ferreira Brega

Monografia apresentada junto à disciplina
Projeto e Implementação de Sistemas II, do
curso de Bacharelado em Ciência da
Computação, Faculdade de Ciências,
Unesp, campus de Bauru, como parte do
Trabalho de Conclusão de Curso.

UNESP

2012

Bruno Fernandes Casella

Desenvolvimento de Aplicações Web
com Base em Modelagem 3D – Módulo Android

Monografia apresentada junto à disciplina
Projeto e Implementação de Sistemas II,
do curso de Bacharelado em Ciência da
Computação, Faculdade de Ciências,
Unesp, campus de Bauru, como parte do
Trabalho de Conclusão de Curso.

BANCA EXAMINADORA

Prof. Dr. José Remo Ferreira Brega
Orientador
DCo – FC - UNESP – Bauru

Profª. Dra. Simone das Graças Domingues
Prado
DCo – FC - UNESP – Bauru

Prof. Dr. Marco Antônio Sacoman
DCo – FC - UNESP – Bauru

Bauru 29 de outubro de 2012

AGRADECIMENTOS

Primeiramente, aos meus pais, pelo apoio, ensinamentos e conselhos que me deram. Agradeço pelo carinho e amor que sempre tiveram por mim.

À minha família, tão importante na minha vida, que sempre me apoiou e me aconselhou.

Agradeço aos meus amigos, que sempre me acompanharam - alguns antes, outro durante, alguns por um pedaço - nesse percurso, dividindo comigo alegrias e tristezas. Que a distância nunca nos separe.

Aos meus colegas e amigos de trabalho, pelos conselhos, ensinamentos, companheirismo e o bom humor no dia a dia.

A todos os bons professores deste curso, que me guiaram e ensinaram durante minha vida universitária. São vocês que trazem o prestígio que esta faculdade tem.

Ao Prof. Dr. José Remo, pelos conselhos, paciência e apoio dados durante este trabalho.

Ao meu companheiro de projeto, Airton Barbosa, pelo comprometimento e seriedade nos momentos precisos, e pela descontração nos momentos necessários.

Saibam todos, que não estaria onde estou sem vocês.

E a Deus, por me possibilitar ter tantas pessoas a quem agradecer. Obrigado!

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação *Android* que possibilita o acesso a ambientes 3D na web. A aplicação desenvolvida possibilita a maioria dos dispositivos *Android* o acesso a páginas web que contenham conteúdo que utiliza OpenGL para sua renderização. Para tal demonstração de funcionalidade, é utilizada uma aplicação web baseada em modelagem 3D, utilizando tecnologias como X3DOM e HTML5, renderizando os objetos 3D através de WebGL.

Esta aplicação web visa dar ao usuário um ambiente de uma cidade virtual, onde ele pode navegar e interagir com seus objetos. A aplicação *Android* visa trazer essa imersão também ao ambiente mobile.

Para o acesso e armazenamentos de dados da aplicação web foi desenvolvido um Webserver, que traz a aplicação web uma API simples de ser utilizada para acesso ao banco de dados.

ABSTRACT

This project brings the development of an *Android* application which will allow users to access 3D models on web. The application developed allows *Android* devices to access web pages which have code that should use OPENGGL to renderize. To demonstrate this functionality, an example web application was build, using technologies such as X3DOM and HTML5, which uses WebGL to get renderized.

This *web* application gives to the user an environment of a virtual city, where he could surf by and interact with the objects. The *Android* application brings this immersion to the mobile world, also.

The access and storage of data was developed a *Webserver*, which bring to the *web* application a simple API to give access to the database.

LISTA DE FIGURAS

1 Exemplo de código VRML	17
2 Exemplo de construção 3D com sintaxe VRML	18
3 Exemplo de código X3D	19
4 Exemplo de construção com sintaxe X3D	20
5 Exemplo de prédio modelo no FluxStudio	21
6 Exemplo de tela inicial do Blender	22
7 Código-fonte do Fennec no SDK do Android.....	31
8 Página inicial da aplicação web CityFreedom	32
9 Objeto de exemplo criado para a aplicação	33
10 Prédios modelados para a aplicação	33
11 Diagramas Entidade-Relacionamento do BD da aplicação CityFreedom	34
12 Exemplo de código utilizado no webserver.....	36

SÍMBOLOS, ABREVIATURAS, SIGLAS E CONVENÇÕES

ABNT	Associação Brasileira de Normas Técnicas
UNESP	Universidade Estadual Paulista
RV	Realidade Virtual
VRML	Linguagem para Modelagem de Realidade Virtual
JSON	JavaScript Object Notation
API	Interface de Programação de Aplicativos
3D	3 Dimensões
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
SSL	Secure Sockets Layer
XML	eXtensible Markup Language;
SOAP	Simple Object Access Protocol;
IIS	Internet Information Services;
JSON	JavaScript Object Notation;
GUI	Graphical User Interface;
WEBGL	API de desenvolvimento de modelagens 3D;
HTML	HyperText Markup Language;
HTML5	Hypertext Markup Language, versão 5;
GNU	General Public License.

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVOS DO TRABALHO.....	12
1.1.1 OBJETIVO GERAL.....	12
1.1.2 OBJETIVOS ESPECÍFICOS.....	13
1.2 ORGANIZAÇÃO DA MONOGRAFIA.....	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 RELIDADE VIRTUAL E REALIDADE AUMENTADA.....	14
2.2 CONSIDERAÇÕES GERAIS SOBRE A TECNOLOGIA VRML.....	16
2.3 CONSIDERAÇÕES GERAIS SOBRE A TECNOLOGIA X3D.....	18
2.4 FERRAMENTAS DE MODELAGEM – FLUXSTUDIO E BLENDER.....	20
2.5 WEBSERVICES.....	22
2.6 CONSIDERAÇÕES GERAIS SOBRE A TECNOLOGIA WEBGL.....	23
2.7 CONSIDERAÇÕES GERAIS SOBRE A TECNOLOGIA X3DOM.....	24
2.8 CONSIDERAÇÕES GERAIS SOBRE A TECNOLOGIA ANDROID.....	25
2.9 FENNEC.....	26
2.10 TORNADO.....	26
2.11 MYSQL.....	27
2.12 PYTHON.....	28
3 MATERIAIS E MÉTODOS.....	28

3.1 MATERIAL.....	28
3.2 METODOLOGIA.....	29
4 SOFTWARES DESENVOLVIDOS.....	30
5 RESULTADOS OBTIDOS.....	37
5.1 APLICAÇÃO ANDROID COMPATÍVEL COM WEBGL E X3DOM.....	37
5.2 DESENVOLVIMENTO DE UM WEBSERVER.....	37
5.3 INTEGRAÇÃO COM SERVIDOR DE BANCO DE DADOS ATRAVÉS DE WEBSERVER E REQUISIÇÕES EM FORMATO JSON.....	38
6 CONCLUSÕES.....	39
7 TRABALHOS FUTUROS.....	40
7.1 BANCO DE DADOS E CONTROLE DE USUÁRIOS.....	40
REFERÊNCIAS BIBLIOGRÁFICAS.....	41

1. INTRODUÇÃO

Um sistema pode ser chamado de “Sistema Mobile” quando o objetivo dele seja atender ao público que usa *smartphones* e *tablets*, sejam eles voltados para o estudo, trabalho ou entretenimento. Um dos principais usos de tais dispositivos é o acesso a *web*, seja esse acesso feito diretamente por navegadores de *internet*, ou indiretamente – pela própria aplicação, onde o usuário terá uma experiência personalizada voltada para aquela aplicação.

A aplicação proporcionará ao usuário o acesso a ambientes e aplicações *web* que tenham em seu conteúdo ambientes 3D. A motivação para tal veio da falta de estrutura proporcionada pelo ambiente *Android* à ambientes *web* 3D – que não são suportados pelo navegador padrão do mesmo. Por esse motivo, usuários dessa plataforma ficam com uma experiência defasada com relação a imersão proporcionada por tais sistemas *web*.

Primeiramente, uma definição de “Aplicação *web*”, deve ser colocada – já que para um sistema ser chamado de “Aplicação *web*” muito além do limiar que compõe a forma de acessá-lo, ou seja, da necessidade de um navegador para a sua utilização.

Assim, um sistema ou uma solução passa a ser uma Aplicação *web* quando tudo é processado por um servidor, que contém o código, que dali é distribuído para todos os clientes através de uma linha de comunicação, no caso a *internet* ou uma rede particular, sendo portanto algo mais complexo que o funcionamento de um navegador.

Com a motivação de proporcionar a usuários de plataformas *mobile* a mesma experiência que usuários *desktop* teriam ao navegar por uma aplicação *web* 3D, uma aplicação foi desenvolvida, proporcionando o acesso e navegação a aplicações *web* que contenham ambientes 3D.

Na questão de aplicações *web*, a preocupação de compatibilidade foi real por um bom tempo, e muitas alternativas a este problema foram apresentadas. Todavia, algumas boas ideias esbarravam na guerra e disputa das grandes desenvolvedoras e distribuidoras de tecnologia. Cada uma tinha seu próprio caminho de programação, sua própria linguagem e parâmetros. Justamente por isso, o desenvolvimento de aplicações *web* se viu num dilema. Investir em tecnologias próprias para facilitar a compatibilidade entre os sistemas e ver a *internet* e junto a ela, as Aplicações *web* tornarem-se cada vez mais “heterogêneas”, ou simplesmente pensar numa solução que fosse além do sentido inicial de “compatibilidade” imaginado para época? (W3C, 2012). Da necessidade de criar a compatibilidade universal

de comunicação na web, que fosse independente dos sistemas que precisassem se comunicar, surgiu uma ideia revolucionária, o conceito de Webserver.

A novidade dos Webservers foi a introdução da linguagem XML, que é um conjunto de regras para definir uma linguagem de marcação que divide um documento em partes, identificando cada uma.

Independentemente da maneira como o sistema fosse construído, da linguagem da qual ele surgiu ou do sistema operacional a rodar por trás do navegador, as informações serão transmitidas em formato XML (HAROLD, 1999).

Pensando na palavra base da nova tecnologia que deu origem aos Webservers e tomando-a como referência, cabe a pergunta: assim como “compatibilidade” associada à inovação é a essência dos novos alicerces dos grandes Aplicações *web*, o que ainda restringe o uso e a criação de soluções *web* com base nas tecnologias 3D? Pode-se muito bem introduzir algo novo, trabalhar também com a compatibilidade para permitir que os browsers, através da *internet*, forneçam aos usuários a oportunidade única de interagir com novos sistemas baseados em modelagem 3D. Pode-se também tornar os elementos 3D, a modelagem e a navegação por ambientes criados virtualmente uma realidade não só para aplicações complexas como jogos, mas também para a navegação diária e simples. Muito além, apresentar e incorporar às aplicações *web* a imersão no mundo 3D. Talvez seja o próximo passo a seguir, diante das tendências a preencher as lacunas de possibilidades dos sistemas de computador, sejam eles, aplicações *web* ou não.

1.1. Objetivos do Trabalho

1.1.2 Objetivo geral

O objetivo geral foi desenvolver um sistema que tenha como base de criação a interação entre modelagem 3D de ambientes e as técnicas clássicas das aplicações *web*.

Com base nessa modelagem criar uma solução que corresponda a um serviço para qualquer pessoa que tenha acesso a internet no intuito de obter informações a respeito de lugares, estabelecimentos ou mesmo pontos de referência de uma cidade.

Esta aplicação poderá ser acessada por qualquer usuário *desktop* ou *mobile* a fazer uso de um dos browsers compatíveis com as tecnologias abordadas, sem que seja necessário instalar qualquer outro *software* externo ou *plugin*. As modelagens 3D, bem

como todos os recursos do aplicações *web* dependerão única e exclusivamente de um browser compatível.

Ao sistema foi dado o nome de “CityFreedom”, que representa a liberdade de poder navegar por uma cidade de maneira diferente e revolucionária. É sentir-se livre para, visitar bairros, conhecer estabelecimentos ou mesmo explorar aquela parte da cidade que a pessoa sempre quis conhecer. É a liberdade concedida pela modelagem 3D em aplicações *web*.

1.1.2 Objetivos específicos

Primeiramente, desenvolver uma aplicação que possibilitasse o acesso a um sistema *web* que tenha uma interação entre modelagem 3D de ambientes e as técnicas clássicas das aplicações *web*, possibilitando assim que qualquer pessoa que tenha acesso a *internet* através da plataforma *Android* possa navegar nesse ambiente sem que seja necessário instalar qualquer outro *software* externo ou *plugin* além do aplicativo. As modelagens 3D, bem como todos os recursos do aplicações *web* dependerão única e exclusivamente da aplicação.

O segundo objetivo do trabalho foi desenvolver um *webserver*, que possibilitasse o acesso ao banco de dados através de uma API simples e segura, além de proporcionar facilidade em futuras modificações e incrementos necessários. O *webserver* também deve ser rápido e escalável, além de funcionar como um modulo totalmente independente da aplicação que deseja fazer a requisição ao banco, assim como funcionar com diferentes programas de banco de dados. Através de *webserver*, qualquer aplicação poderia ter acesso a banco desejado pela mesma API, de forma transparente e sem a necessidade de módulos de terceiros.

1.2 Organização da monografia

- No Capítulo 1 deste relatório são apresentados alguns conceitos introdutórios, sobre os assuntos abordados dentro deste trabalho.
- No Capítulo 2 deste relatório é apresentada a fundamentação teórica realizada, visando abordar as tecnologias utilizadas no desenvolvimento do trabalho.

- No Capítulo 3 deste relatório são apresentados e a metodologia utilizados para desenvolvimento do trabalho.
- No Capítulo 4 deste relatório são apresentados os programadas desenvolvidos neste trabalho.
- No Capítulo 5 são mostrados quais foram os resultados obtidos.
- No Capítulo 6, serão mostrados as conclusões obtidas com este trabalho.
- No Capítulo 7, serão mostrados trabalhos futuros que poderão ser realizados.

2. FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é elucidar os conceitos técnicos e teóricos para facilitar a compreensão das tecnologias X3DOM, X3D, Webservers, *Android* e do desenvolvimento de aplicações web, bem como das ferramentas utilizadas neste trabalho. Serão também mostrados nesse capítulo aspectos relativos à realidade virtual e aumentada.

2.1 Realidade Virtual e Realidade Aumentada

A Realidade virtual (RV) é uma técnica avançada de interface onde o usuário pode interagir e navegar em um ambiente tridimensional fazendo uso de dispositivos não convencionais de entrada e saída para trabalhar de maneira única os sentidos humanos (visão, audição e tato) (MACHADO, 2002).

Os dispositivos podem ser luvas eletrônicas, capacete de visualização, visão eletroscópica, dentro outros. De forma extremamente natural e equivalente ao que fariam se estivessem interagindo com o mundo real. A ideia é que uma pessoa munida de tais equipamentos possa interagir com o ambiente virtual de maneira tão natural e espontânea à ponto de literalmente sentir-se parte do cenário (ADAMS, 1994).

A realidade virtual ainda é considerada como a união de três pilares principais, os

quais compõem a base da estruturação da tecnologia: imersão (sensação de estar dentro do ambiente.), Interação (possibilidade do usuário interferir com o que acontece no ambiente, e vice-versa) e envolvimento (capacidade do ambiente motivar o usuário a participar) (SGARBI, 2003).

Tendo em vista que o objetivo deste trabalho é a criação de uma aplicação *web* torna-se importante discorrer primeiramente a respeito da modelagem 3D em si, a qual será o foco do estudo de Realidade Virtual possibilitando ao usuário a identificação dos três pilares da realidade virtual (imersão, interação e envolvimento), mesmo que o único sentido envolvido seja a visão mediante restrições da tecnologia implantada, no caso um simples browser compatível com renderização WebGL.

Para tal atenta-se às ferramentas, técnicas e experiência. No que diz respeito à modelagem, aquilo que ficou muito claro no âmbito tecnológico nos últimos anos foi o surgimento de diversos sistemas de modelagem gratuitos. Neste sentido, a modelagem 3D passa de algo fantasioso e tido como complexo e inviável para algo cada vez mais palpável. Diante desta premissa alguns bons exemplos de modeladores e programas próprios para a criação de cenários 3D fornecem simplicidade e desenvolvimento intuitivo para quem se prontificar a gastar o mínimo de tempo no estudo das técnicas envolvidas nestes programas. (KIRNER, 2011)

Neste quadro de inovações aliadas às práticas cada vez mais singelas surge uma das ferramentas amplamente utilizadas neste projeto, o *FluxStudio*, o qual nada mais é que uma ferramenta de modelagem 3D capaz de gerar arquivos nos formatos VRML e X3D. Apesar de o *FluxStudio* ser gratuito, ele evoluiu para outro sistema muito mais complexo e enriquecido com diversas outras funcionalidades, chamado Vizard. O Vizard, no entanto, tem um custo na ordem de vários milhares de dólares, o que confirma a ideia de que a renderização mais profissional e detalhada envolve grande quantidade de recursos e investimentos em *software* e *hardware*. (KIRNER, 2011)

2.2 Considerações gerais sobre a tecnologia VRML

Não há como analisar ou estudar a linguagem X3D, a qual se consolidou como a

base para o desenvolvimento de modelagens tridimensionais para aplicações *web*, sem antes discorrer a respeito de sua predecessora, a linguagem VRML (Virtual Reality Modeling Language).

O VRML é uma linguagem para criação de ambientes tridimensionais utilizado para projetar cenários tanto para *Browser* (com a utilização de *plugins*) e aplicações *web* quanto para sistemas Desktop (os quais possuem naturalmente maior poder de processamento e renderização). A linguagem VRML possibilita a criação de objetos, manipulação de cores, brilho e transparência além da aplicação de texturas. Um diferencial da linguagem VRML foi a introdução de sensores possibilitando ao usuário a sensação de interação com o ambiente modelado. (FESTA, BORLAND, 2005)

A linguagem VRML não foi necessariamente abandonada. A linguagem X3D, sua sucessora possui todos os pontos positivos e os recursos anteriormente empregados, a diferença a mudança de sintaxe, focada e vinculada na tecnologia XML, a qual por questões de compatibilidade e desempenho é uma alternativa muito mais interessante para a modelagem 3D para aplicações *WEB*.

Exemplo da sintaxe VRML na figura abaixo:

```
#VRML V2.0 utf8
#Color example: a pyramid
Shape {
  appearance Appearance{
    material Material { }
  }
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        # bottom
        -1.0 -1.0 1.0, #vertex 0
        1.0 -1.0 1.0, #vertex 1
        1.0 -1.0 -1.0, #vertex 2
        -1.0 -1.0 -1.0, #vertex 3
        # top
        0.0 1.0 0.0 #vertex 4
      ]
    }
    colorPerVertex FALSE #so each face will have one of the colors
    color Color {
      color [
        1.0 0.0 0.0,#color 0
        0.0 1.0 0.0,#color 1
        0.0 0.0 1.0,#color 2
        1.0 1.0 0.0,#color 3
        0.0 1.0 1.0,#color 4
      ]
    }
    colorIndex [ 2 3 4 0 0 ] #the first face will have color 2, the 2nd color3...
    coordIndex [
      #bottom square
      0, 3, 2, 1, 0, -1,
      #side1
      0, 1, 4, -1,
      #side2
      1, 2, 4, -1,
      #side3
      2, 3, 4, -1,
      #side1
      3, 0, 4, -1,
    ]
  }
}
```

Figura 1. Exemplo de código VRML

Exemplo de representação gráfica do código VMRL acima, na figura abaixo:

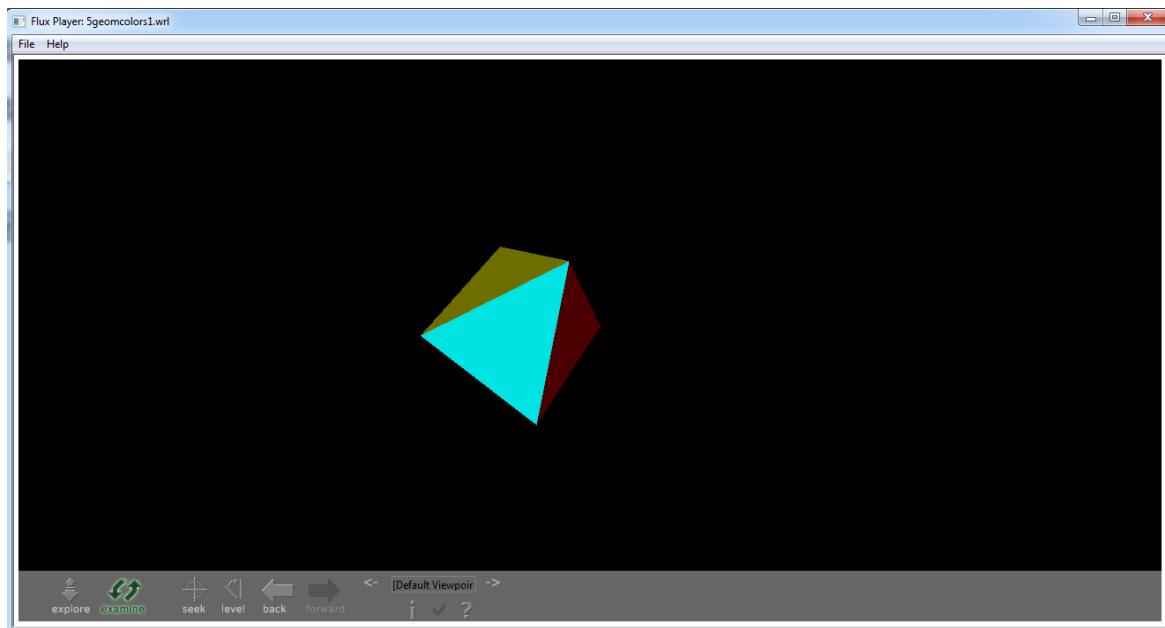


Figura 2: Exemplo de construção 3D com sintaxe VRML

2.3 Considerações gerais sobre a tecnologia X3D

Surgindo da revisão da especificação do padrão VRML, é uma tecnologia de padrão aberto para distribuir conteúdo 3D, incorporando os avanços dos recursos disponíveis nos últimos dispositivos gráficos comerciais e também incorpora melhorias na sua arquitetura. É uma linguagem desenvolvida com a sintaxe XML capaz de representar e comunicar cenas 3D e objetos. A tecnologia X3D dá suporte para gráficos 3D, transformações de geometria, iluminação, materiais, texturas, mapeamento, pixels, vértices e aceleração de *hardware*. Permite animação com temporizadores e interpoladores de condução contínua. Permite também interação com o mouse e entradas de teclado. A navegação no ambiente acontece por meio do uso de câmeras, com características de colisão, proximidade e visibilidade, detecção e vários tipos de iluminação (BRUTZMAN, DALY, 2007)

Para o foco deste projeto, convém elucidar o termo “*Browser X3D*”. *Browser X3D* é aquele que possibilita não só a leitura e interpretação de arquivos ou código X3D como também possibilita a navegação por *viewpoints*, características de iluminação e animação anteriormente citadas. Os *browsers X3D* ainda interpretam os elementos X3D com base na linguagem HTML e nos novos elementos *HTML5*, ou seja, um arquivo X3D “puro” não

necessariamente terá funcionamento amplo na interpretação dos browsers (como Mozilla Firefox ou Google Chrome). Para tanto é necessário converter o código X3D nativo em tags HTML possibilitando assim a renderização das cenas 3D de maneira a tornarem-se apresentáveis ao usuário final. (BRUTZMAN, DALY, 2007)

Veja, na figura baixo, um exemplo de codificação X3D, e na figura 4, um objeto X3D no modo de edição.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
"http://www.web3d.org/specifications/x3d-3.0.dtd"/>
<X3D profile='Immersive' >
<head>
  <meta name='Vizthumbnail'
    content='Thumb_exemploMonografia_x3d393001351052894.jpg' />
  <meta name='ExportTime' content='2:28:14' />
  <meta name='ExportDate' content='10/24/2012' />
  <meta name='FluxStudioVersion' content='659' />
</head>
<Scene>
<WorldInfo
  title='Untitled'
  info='This Web3D Content was created with Flux Studio, a Web3D authoring tool'
  "www.mediamachines.com" />
<Transform DEF='dad_Box1'
  translation='4.53006 0 2.26503'>
  <Shape DEF='Box1'
    containerField='children'>
    <Appearance
      containerField='appearance'>
      <Material DEF='Red'
        containerField='material'
        ambientIntensity='0.200'
        shininess='0.200'
        diffuseColor='1 0 0' />
      </Appearance>
      <Box DEF='GeoBox1'
        containerField='geometry'
        size='1 1 1' />
      </Box>
    </Shape>
  </Transform>
<Transform DEF='dad_Sphere1'
  translation='4.57582 1.32699 2.31079'>
  <Shape DEF='Sphere1'
    containerField='children'>
    <Appearance
      containerField='appearance'>
      <Material
        containerField='material'
        USE='Red' />
      </Appearance>
      <Sphere DEF='GeoSphere1'
        containerField='geometry'
        radius='1.000' />
      </Sphere>
    </Shape>
  </Transform>
</Scene>
</X3D>
```

Figura 3. Exemplo de código X3D

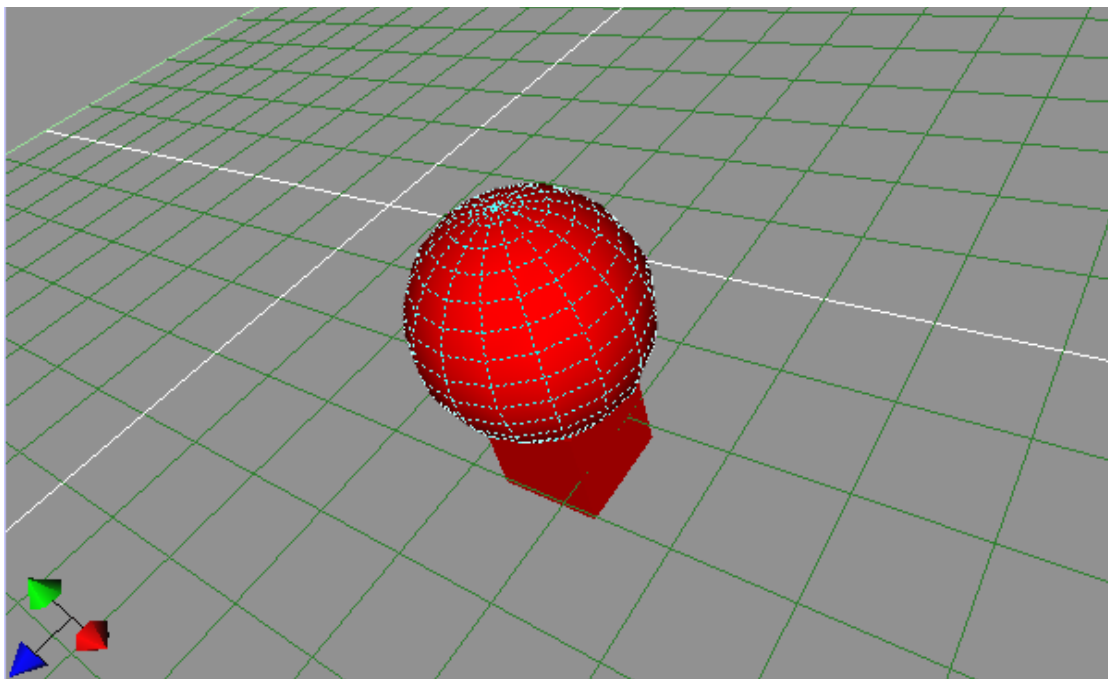


Figura 4. Exemplo de construção com sintaxe X3D

2.4 Ferramentas de modelagem – FluxStudio e Blender

O *FluxStudio* é uma ferramenta de modelagem simples e intuitiva, que gera arquivos no formato X3D, os quais, mediante testes preliminares se apresentaram estáveis na leitura dos *browsers* dependendo do nível de detalhamento e da quantidade de objetos modelados. Por outro lado, esta ferramenta não possibilita o tratamento de modelagem necessário para criar objetos complexos o suficiente para uma exploração talvez mais abrangente da modelagem em si, caso as técnicas de aplicação de tais modelos sejam viáveis para um sistema *web*. Sendo assim, a tendência a trabalhar em conjunto utilizando a aplicação do FluxStudio e a aplicação Blender (também voltada para modelagem 3D).

O *Blender* é um *software* gratuito para modelagem e trabalhos com gráficos 3D. Com ele é possível criar animações, jogos complexos e manipulação de vídeos. Ele foi concebido mediante a licença GNU/LINUX e está disponível também para *Windows* e *Mac*. Ao contrário do *FluxStudio*, o Blender é extremamente complexo e basicamente não é possível criar cenários 3D utilizando-o sem prévio conhecimento a respeito das funcionalidades da ferramenta. Por outro lado, ao utilizá-lo a possibilidade de criação de objetos cada vez mais complexos e ao mesmo tempo “leves” o bastante para não comprometer o desempenho da aplicação torna-se extremamente viável. [JAMES, DANIEL. 2010].

Com o *Blender* os modelos e objetos ficarão naturalmente mais condizentes com o nível de tecnologia atual, resta saber se os modelos são complexos o bastante a ponto de não permitir o desenvolvimento plausível de uma aplicação voltada à *web*. Se for este o caso, a modelagem terá de ser restrita ao FluxStudio, o qual, como dito, possui uma modelagem mais simples e intuitiva cujos objetos podem ser facilmente criados usando um sólido qualquer preenchido com uma textura.

Abaixo, um exemplo do modelo utilizado nos testes sendo modelado no *FluxStudio* (figura 5) e em seguida um cubo sendo modelado no Blender (figura 6).

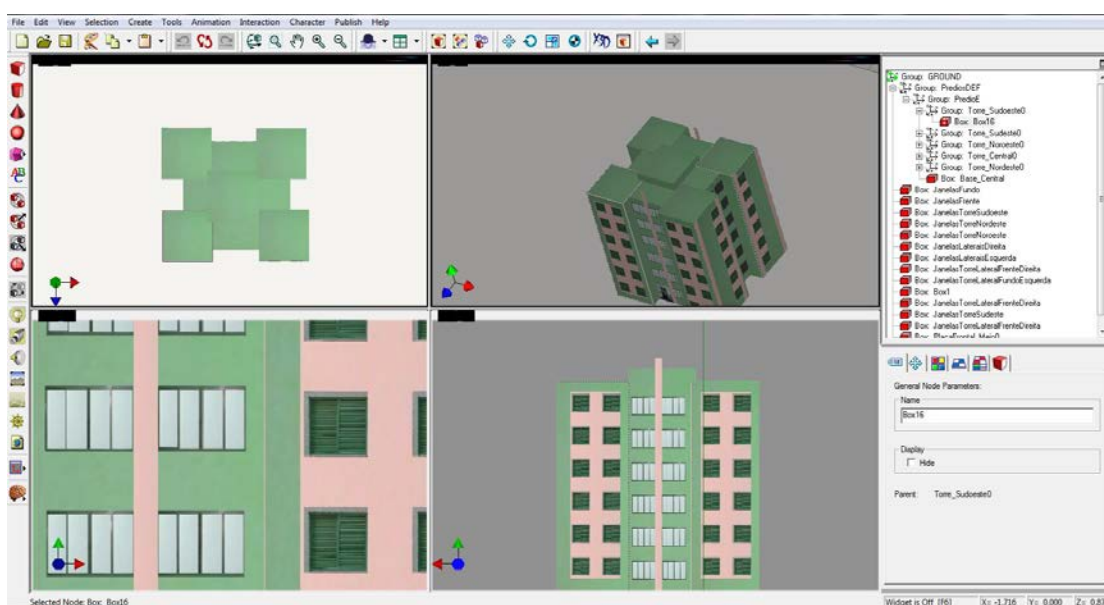


Figura 5. Exemplo de prédio modelado no FluxStudio

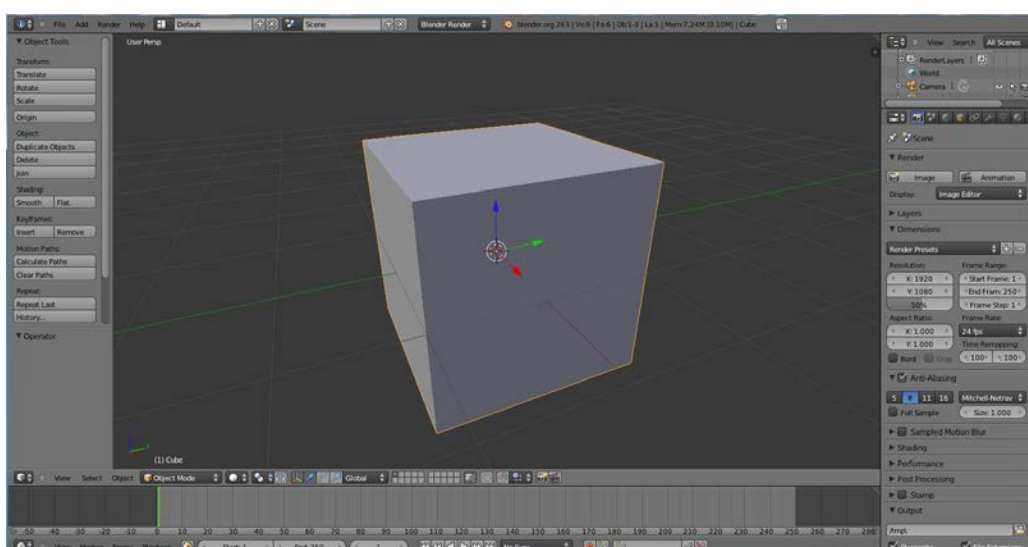


Figura 6. Exemplo da tela inicial do Blender

2.5 Webservers

Outro ponto importante é o desenvolvimento da vertente do sistema voltada aos Webservers. Foi citada nesta proposta a importância da nova tecnologia no advento da programação *web*, e foi discutido a respeito do desenvolvimento de uma nova técnica que talvez se tornasse tão importante e cotidiana como tal. Nada melhor, portanto, do que fazer uso desta tecnologia para encorpar a criação e desenvolvimento de outra.

Dessa maneira, será dada uma atenção especial ao desenvolvimento do *Webserver* para a aplicação no caso de resolver uma possível falta de compatibilidade entre as modelagens e a aplicação em si. Nesse caso, se for constatado que objetos no formato X3D não possuem viabilidade de desenvolvimento para browsers ou aplicações voltadas ao ambiente *web*, a solução será “chamar” o conteúdo online para tradução e interpretação segundo os conceitos e aplicações locais, o que sabidamente é uma alternativa que, a princípio, não apresentaria problemas em interpretar as modelagens.

No entanto, apesar de ser uma proposta absolutamente viável, no que diz respeito à experiência do usuário, deve-se atentar a fato de que, ainda sim, esta é uma opção secundária ou de complementação, tendo em vista a não padronização e a necessidade de um “*software*” especialmente para gerar a modelagem 3D ao usuário final. Isso poderia inviabilizar o plano de sucesso do projeto, o qual tem como embasamento, principalmente a criação de tendência no desenvolvimento de aplicações para *web*.

Em outras palavras, seria absolutamente relevante ver o ambiente 3D da cidade virtual interpretado de forma máxima, com o maior número de detalhes possível, mas a partir do momento que uma aplicação local é necessária para tal fim, a ideia de aplicação *web*, representatividade e facilidade perde força. A respeito de Webservers, o texto abaixo apresenta uma breve introdução a respeito da tecnologia, mediante informações contidas no livro citado. É uma ótima fonte de conhecimento a respeito dos Webservers. Vejamos:

Like many other incipient technologies, *web* services are still surrounded by a substantial level of noise. This noise results from the always dangerous combination of wishful thinking on the part of research and industry and of a lack of clear understanding of how *web* services came to be.

On the one hand, multiple contradictory interpretations are created by the many attempts to realign existing technology and strategies with *web* services. On the other hand, the emphasis on what could be done

with *web* services in the future often makes us lose track of what can be really done with *web* services today and in the short term. (ALONSO, 2010).

2.6 Considerações gerais sobre a tecnologia WebGL

De nada adianta criar cenários completos em linguagem X3D organizados perfeitamente em nós XML se o *Browser* não for capaz de interpretá-los. Diante deste problema surgiu em 2011 uma API concebida com base em JavaScript a partir do elemento Canvas do *HTML5*. Esta API era capaz de aplicar os recursos da modelagem 3D aos browsers e fazer com que os cenários e objetivos modelados segundo a linguagem X3D se tornassem parte integrante do desenvolvimento de aplicações *web*. Sem a necessidade de *plugins* ou *softwares* de terceiros para que isso fosse possível. (KHRONOS, 2010)

A API WebGL tornou-se a base para o desenvolvimento de inúmeras outras ferramentas de criação de cenários 3D nesse período de pouco mais de um ano de seu lançamento. Dela surgiu, por exemplo, o *X3DOM* que é uma biblioteca que maximiza a utilização de WebGL em aplicações *web* tornando as renderizações mais eficientes e dinâmicas, no contínuo e incessante esforço de melhorar cada vez mais o desempenho das modelagens numa plataforma teoricamente tão limitada, que são os browsers. (X3D e WebGL, 2012).

Este projeto fará uso da biblioteca *X3DOM* para a criação de seus respectivos cenários e modelagens 3D visando a possibilidade de aproximar-se o máximo possível da realidade dos objetos criados, para que os conceitos de navegação e reconhecimento da cidade virtual tenham para o usuário o que há de melhor no que diz respeito a imersão no universo 3D.

2.7 Considerações gerais sobre a tecnologia X3DOM

X3DOM é uma tecnologia, ainda em fase experimental, *opensource* que tem como objetivo dar suporte a discussões de como integrar as tecnologias como *HTML5* e um ambiente *web* 3D declarativo, e permite incluir elementos X3D como parte do código *HTML5* de uma página *web*.

Surgindo da revisão da especificação do padrão VRML (*Virtual Reality Modeling Language*), é uma tecnologia de padrão aberto para distribuir conteúdo 3D, incorporando

os avanços dos recursos disponíveis nos últimos dispositivos gráficos comerciais e também incorpora melhorias na sua arquitetura. É uma linguagem desenvolvida com a sintaxe XML capaz de representar e comunicar cenas 3D e objetos.

A tecnologia X3D dá suporte para gráficos 3D, transformações de geometria, iluminação, materiais, texturas, mapeamento, *pixels*, vértices e aceleração de *hardware*. Permite animação com temporizadores e interpoladores de condução contínua (W3C, 2012). Permitem também interação com o mouse e entradas de teclado. A navegação no ambiente acontece por meio do uso

de “*viewpoints*” (pontos de visão manipuláveis dentro da cena 3D), com características de colisão, proximidade e visibilidade, detecção e vários tipos de iluminação. Todos estes recursos da tecnologia *X3DOM* estão presentes no projeto proposto.

As implementações atuais do *X3DOM* funcionam em browsers que têm implementado WebGL, a exemplo do Firefox 4.0 para desktop ou móbile, Google Chrome, e as últimas versões do Opera para desktop e *Android*.

2.8 Considerações gerais sobre Android

Android é o sistema mobile mais popular do mundo (ANDROID, 2012). Ele é desenvolvido com base no núcleo do *Linux*, e conta com mais de trezentos parceiros em hardware, *software* e operadoras (ANDROID DEVELOPER, 2012). O *Android* permite aos desenvolvedores criar *softwares* na linguagem *Java*, utilizando bibliotecas desenvolvidas pela Google.

Elas podem ser utilizadas através de sua SDK (Software Developer Kit), disponíveis gratuitamente na *web* (ANDROID DEVELOPER, 2012). A SDK do *Android* é compatível com várias das versões mais recentes da JDK (Ambiente de Desenvolvimento *Java*), assim como com a IDE Eclipse, mas ambos não são distribuídos juntamente com a SDK do *Android* (Mednieks, Dornin. 2011).

O sistema teve seu nascimento comercial em 2008, desde então lançou mais de seis novas atualizações do SO, todas com melhorias significativas tanto para o usuário – em termos de interface, funcionalidades, segurança e economia de bateria – quanto para o desenvolvedor – mais funções nas bibliotecas de sua plataforma de desenvolvimento, melhorias para a padronização da interface, entre inúmeras outras (ANDROID DEVELOPER, 2012).

Devido ao seu grande público-alvo e bom ambiente de desenvolvimento, assim como grande variedade de *hardware*, que possibilita avaliar o desempenho de nossa aplicação em vários casos, ele é o sistema *mobile* que terá foco nesse projeto.

O sistema tem muitas funcionalidades desenvolvidas, que são feitas para funcionar com qualquer peça que tenha em seu *hardware* a capacidade de rodar tal funcionalidade. O *Android* tem em suas especificações as funções abaixo.

- Adaptabilidade de tela: o sistema se adapta a vários tamanhos e resoluções de tela
- Drivers gráficos: têm biblioteca tanto para gráficos 2D quanto para gráficos 3D baseado na tecnologia OpenGL 2.0
- Armazenamento: utiliza do banco de dados SQLite, um DB relaciona leve e funcional
- Conectividade: suporta as tecnologias GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC e WiMax
- Suporte a vários idiomas
- Suporte a inúmeros formatos de arquivos de áudio e vídeo, assim como streaming de ambos
- Multitouch: suporte nativo a telas multitoque
- Implementação multitarefa

2.9 Fennec

Para o funcionamento da aplicação *web* em dispositivos móveis, é necessário que o browser suporte HTML5 e renderização via WebGL. Infelizmente este não é suportado pelo browser padrão da maioria dos dispositivos móveis que rodam o sistema *Android*. Para utilizar tal funcionalidade era necessária instalação de browser de terceiros, como o Firefox Web Browser e o Opera. Infelizmente o suporte a WebGL não é prioridade, e se mostra funcional em algumas versões, e não funcional em outras. Para que se tenha dependência desse versionamento, foi utilizado como base o browser Fennec.

Fennec é o codinome para a versão móvel do Firefox. Ele dá aos desenvolvedores uma plataforma para criar conteúdo e aplicações usando as tecnologias *web* mais recentes, como HTML5, CSS e JavaScript (FENNEC, 2012).

Por ser um *web* browser de código aberto, é possível compilar um browser viewpoints, o que nos dá várias vantagens. Primeiramente, teremos um aplicativo, o que possibilita tanto customização quanto colocá-lo no *market* do *Android* para posterior *download*. Há também a certeza de que este será um browser funcional para a aplicação, pois uma vez certo que de a versão feita suporta HTML5, X3DOM e renderiza WebGL, teremos sempre um versão funcional do aplicativo, não dependendo de desenvolvimento de terceiros.

Temos a aplicação *web* rodando através desse browser. Após compilar seu código fonte, assim como fazer o build da aplicação *Android*, ele se mostrou estável e funcional. Obtivemos resultados satisfatórios quanto ao desempenho de navegação em ambientes X3DOM, o que possibilita uma boa experiência para o usuário em ambientes mobile.

2.10 Tornado

Uma ferramenta poderosa para escrever aplicações *web*, que são simples de criar, expandir e integrar. É rápido e escalável. (DORY, PARRISH. 2012). Escrito em *Python*, uma linguagem de alto nível, dinâmica, que se destaca entre umas das mais populares e poderosas [L. E. Borges. *Python* para Desenvolvedores], é robusto o suficiente para suportar grande quantidade de tráfego. Foi desenvolvido por Bret Taylor, e mais tarde seu código foi aberto pelo *Facebook* quando este comprou a FriendFeed (DORY, PARRISH. 2012).

Foi feito um WebServer com base no *Tornado*, que irá tratar de todas as requisições no banco de dados de nossa aplicação. Para tal, modelou-se todas as tabelas, chaves e visões feitas no BD, e todas as requisições ao banco serão tratadas por ele. No WebServer modelado, as requisições serão feitas através do protocolo HTTPS – que combina SSL, um protocolo de segurança que pode ser usado por qualquer aplicação que use TCP como meio de transporte (THOMAS, 2001) – trazendo assim tráfego encriptado de dados ao *Tornado*. O WebServer está em um servidor separado da aplicação, rodando no sistema operacional Ubuntu.

As informações retornam dados do banco em formato JSON – um formato para troca de dados fácil para humanos lerem e escreverem, e fácil para máquinas gerarem e captarem (JSON, 2012) – o que facilita para a aplicação *web* fazer a captação dos resultados de suas requisições *web* ao banco de dados.

Todas essa troca de informações entre aplicação *web*, WebServer e banco de dados ocorre de forma segura, simples, e extremamente rápida, devido às tecnologias utilizadas no desenvolvimento do projeto.

2.11 MySQL

O MySQL é um sistema de gerenciamento de banco de dados que tem como base a utilização da linguagem SQL para seu funcionamento. Existente há quase vinte anos, e mantido pela Oracle desde 2009, é utilizado por inúmeras empresas ao redor do mundo.

Suas principais características são a portabilidade, suportando praticamente qualquer plataforma atual, compatibilidade com inúmeras linguagens de programação, suporte a várias tecnologias de banco de dados, como *triggers*, cursos e funções. Também possui um bom desempenho e estabilidade, exigindo pouco *hardware* para seu funcionamento.

Por ser considerado um banco de dados ótimo para aplicações de médio porte, assim como bom para aplicações de grande porte, foi o banco escolhido para ser utilizado no *Webserver* desenvolvido, além de ser um *software* livre com base na GPL, o que implica na não necessidade de uma licença comercial.

2.12 Python

Python é uma linguagem de programação de alto nível orientada a objetos e interpretada . Combinando uma sintaxe concisa e clara, priorizando a facilidade de leitura do código, e conta com recursos poderosos de sua biblioteca padrão e por módulos de terceiros (L. E. BORGES, 2010).

É uma linguagem ao mesmo tempo poderosa e fácil de fazer implementações e futuras modificações. Por ter inúmeros módulos tem compatibilidade com vários bancos de dados. É também multiplataforma, o que levou a ser a linguagem escolhida para o desenvolvimento do *webserver* utilizado no projeto.

3. Materiais e Métodos

3.1 Material

- Notebook com processador Intel i3 com placa de vídeo integrada, com sistema operacionais Ubuntu 12.04 64bits, Ubuntu 10.04 64 bits e *Windows* 7 64bits, com 6GB de memória
- *Desktop* com hardware preparado para testes de desempenho das modelagens (Placa de vídeo GeForce 460GTX, AMD Phenom II, 8Gb memória, sistema operacional Ubuntu 12.04 64bits)
- Tablet Samsung Galaxy Tab com sistema *Android* 3.0
- Smartphone Motorola Milestone2 com sistema *Android* 2.3
- Máquina Virtual com sistema operacional Ubuntu 12.04 para *Webserver* e Banco de Dados
- *Android* SDK e *Android* NDK
- Software de modelagem 3D *FluxStudio* 2.1
- Biblioteca JQUERY 1.7+
- API *X3DOM*
- Navegadores Google Chrome, Mozilla Firefox e Opera (Não há suporte à tecnologia tratada neste trabalho para o *Browser* Internet Explorer e para o Safari os recursos funcionam apenas mediante uso de *plugins*)
- Software conversor X3D, *HTML5* (x3dom.org)

3.2 Metodologia

O desenvolvimento deste trabalho contou com as seguintes etapas:

- Estudo a respeito das técnicas de modelagem
- Estudo sobre a tecnologia *Android*
- Estudo sobre *webserver* e as tecnologias envolvidas no desenvolvimento do mesmo
- Desenvolvimento de uma aplicação *Android* que suportasse X3DOM, com

base no Fennec, em paralelo com a criação de algumas modelagens de exemplo

- Desenvolvimento de um *Webserver*
- Integração do portal CityFreedom, desenvolvido por Airtton Barbosa, com o *Webserver*, o qual será acessado pelo aplicação *Android*

A primeira parte foi dividida em:

- A partir de exemplos simples modelados trabalhar a renderização das modelagens nos *browsers*;
- Encontrar maneiras para melhorar o desempenho da aplicação mediante modelagens mais complexas;
- Estudar as ferramentas de modelagem para criar cenários cada vez melhores e mais eficientes.

A segunda parte foi dividida em:

- Estudo das IDE de programação *Android*
- Estudo do projeto do Fennec
- Compilar o código fonte do Fennec

A terceira parte foi dividida em:

- Buscar a melhor solução para o objetivo descrito na construção de um *webserver* para o projeto
- Estudo do projeto *Tornado*
- Escolha do banco de dados

A quarta parte foi dividida em:

- Criação de uma aplicação base a partir do Fennec
- Criação de modelos X3D
- Conversão dos modelos
- Teste da aplicação em dispositivos *Android* utilizando os modelos gerados

A quinta parte foi dividida em:

- Modelagem de como seria desenvolvida a API de acesso ao *webserver*
- Desenvolvimento do código do *webserver*
- Integração do *webserver* com o bando de dados MySQL

A sexta parte foi dividida em:

- Trabalhar com conjunto com Airton Barbosa (coautor do projeto) para unir o *Webserver* criado à aplicação “CityFreedom”.
- Testes funcionais dessa integração

4. Softwares Desenvolvidos

A aplicação desenvolvida voltada para o ambiente mobile tem como foco dispositivos *Android*. Nessa aplicação foi aplicado o conhecimento adquirido para criar uma aplicação que dá ao usuário novas possibilidades de uso ao dispositivo.

O objetivo é fornecer ao usuário uma aplicação onde ele possa acessar aplicações web que contenham ambientes 3D, podendo também interagir e navegar neste ambiente, sem a necessidade de plugins específicos, e sim apenas a aplicação desenvolvida.

Para se criar tal aplicação foi utilizado como base o código fonte do Fennec, codinome para a versão mobile do navegador Firefox. Tem o código em mãos, foi compilado utilizando o compilador GCC da plataforma *Linux*. Com a utilização da plataforma NDK, que possibilita utilizar a linguagem C++ para a criação de aplicações para a plataforma *Android*, é possível criar uma aplicação que pode ser instalada em qualquer dispositivo, assim como personalizar o visual da mesma.

Na figura 7 é mostrado o código fonte do Fennec aberto no *Android* SDK.

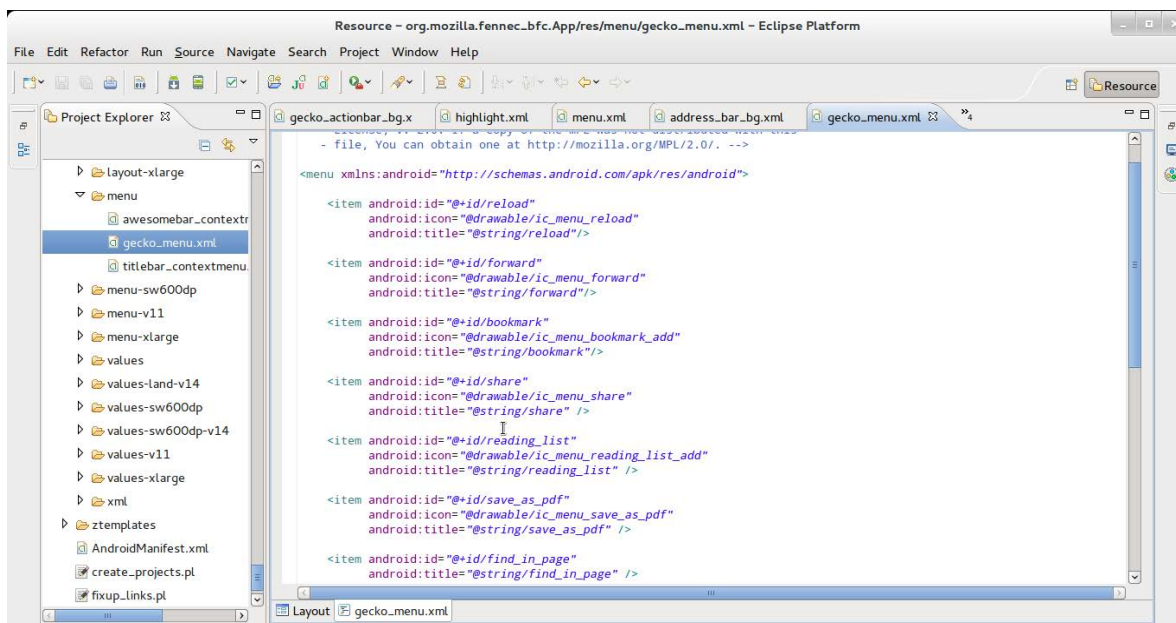


Figura 7. Código-fonte do Fennec no SDK do *Android*

Para testar a navegabilidade em ambientes web 3D, foram modeladas alguns exemplos de modelos 3D. Para tal, foi utilizada a ferramenta FluxStudio, que permite modelar de maneira simples, e cria um objeto X3D como saída. Esse objeto poderia ser incluído em uma página HTML, mas para a renderização de tal objeto seria necessário a utilização de plugins em um browser desktop, e impediria a sua utilização em uma ambiente mobile.

Optou-se então por analisar uma tecnologia recente, a API WebGL. A partir dessa API e de uma de suas vertentes, foi possível traduzir o código X3D em linguagem HTML5 através da criação de nós, como em XML, já suportados por browser desktop modernos, e com suporte feito à aplicação *Android* aqui feita. Entretanto, essa conversão gerava um número exagerado de linhas, o que resultava em um baixo desempenho.

Diante deste problema uma ferramenta oferecida pela própria desenvolvedora da biblioteca X3DOM oferece a solução para a conversão de um arquivo X3D para a codificação HTML5, embora vértices e curvas complexas não sejam corretamente traduzidas. Assim, para um protótipo inicial, a criação das modelagens se deu com base em manipulação de texturas aplicadas a objetos simples, no intuito de resolver esse problema.

Assim, tornou-se possível utilizar o código gerado em uma página HTML5, o que comprovou a funcionalidade da aplicação, que renderizou de forma satisfatória um objeto 3D.

Para um teste mais delicado, a aplicação web “CityFreedom” - cuja página inicial pode ser vista, no ambiente *Android*, na figura 7 - desenvolvida pelo Airton Barbosa, foi utilizada para comprovar que uma aplicação web 3D complexa seria renderizada para o usuário da aplicação *Android*.

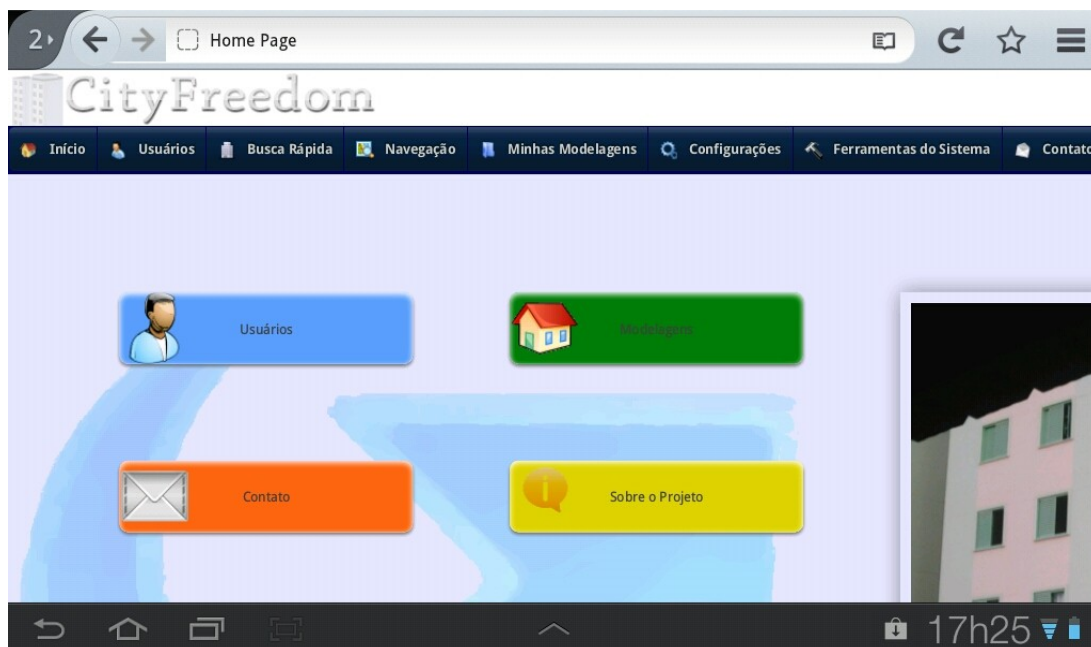


Figura 8. Página inicial da aplicação web CityFreedom

A aplicação mostrou-se navegável e consistente no ambiente *Android* nos ambientes 3D, como é possível ver na figura 9, que consiste em um objeto de teste, e também na figura 10, que consiste em vários objetos modelados.

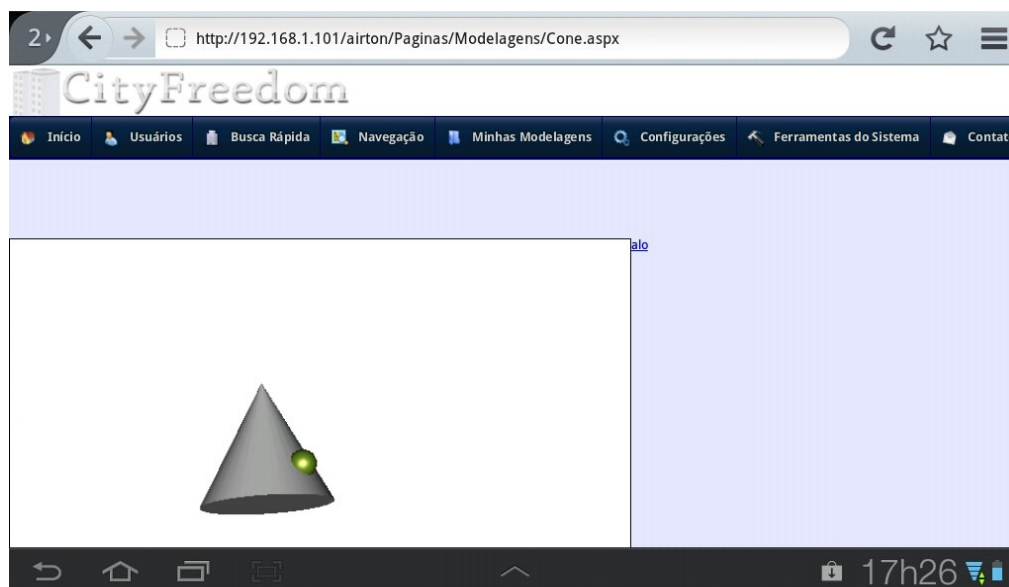


Figura 9. Objeto exemplo criado para a aplicação

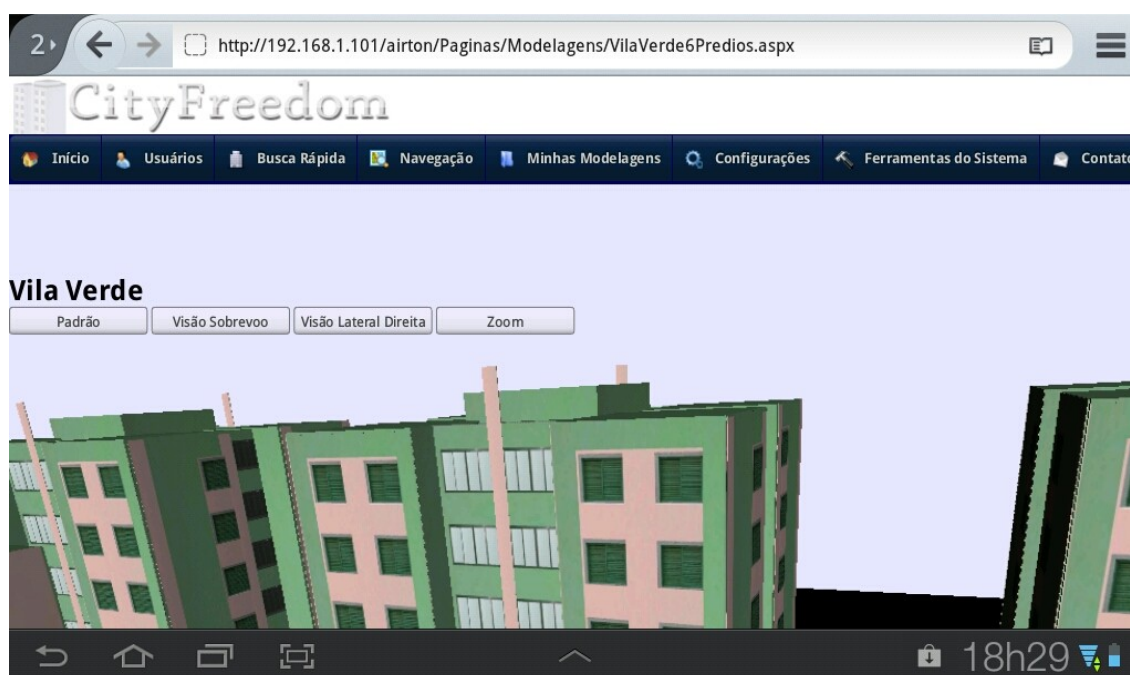


Figura 10. Prédios modelados para a aplicação

Essa aplicação web, que consiste em uma cidade modelada, onde cada bairro é um modulo 3D navegável e possível de interagir, utilizou-se de um Webserver, que possibilita não só ela, mas a qualquer aplicação web utilizar da API feita para um acesso fácil e seguro ao banco de dados escolhido. A estrutura das tabelas que serão usadas pela aplicação web CityFreedom pode ser vista na figura 11.

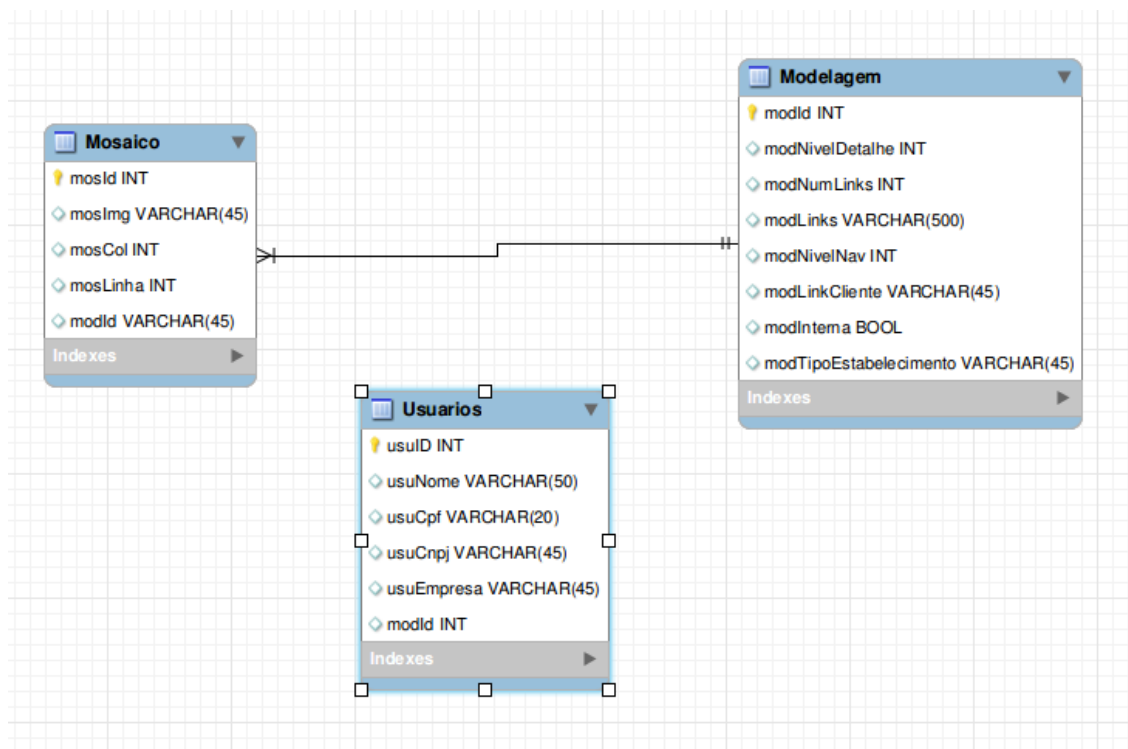


Figura 11. Diagramas Entidade-Relacionamento do BD da aplicação CityFreedom

O desenvolvimento desse webserver se deu na plataforma *Linux*, e seu funcionamento independe do servidor da aplicação web, que no caso irá rodar em um server IIS em ambiente *Windows*.

O webserver utiliza o *Tornado*, que possibilita criar este com facilidade, bom desempenho, e possibilidade de posterior expansão e integração. A escolha dessa tecnologia levou em conta a interdependência que será feita entre os módulos necessários para o webserver, e como a solução criar uma API simples e segura para acesso ao banco de dados.

Com a solução criada é possível acessar o banco de dados da aplicação web desejada pelo protocolo HTTP, que terá sua segurança feita pela camada SSL. Consultas retornarão dados em JSON, um formato leve para intercâmbio de dados computacionais. Inserções no banco poderão ser feitas enviados os dados via HTTP através de métodos GET ou POST. É mostrado na figura 12 o exemplo do código que cuida do recebimento de pedidos no webserver.

O modelo de API criado permite que outras aplicações utilizem métodos padrões para consultas, inserções, atualizações e deleções de forma padronizada e sem a necessidade de módulo de terceiros.

Assim, foi obtida a integração entre a aplicação web “CityFreedom”, que utiliza o webservice desenvolvido, e a aplicação *Android*, que navega na aplicação web renderizando seu conteúdo 3D, levando a interação e funcionalidades desse ambiente aos usuários mobile.

```
#!/usr/bin/python
# coding: utf-8
# Bruno Fernandes Casella <brunocasella@gmail.com>

import database as db
import logging
from request.controllers import BaseController

class MosaicoController(BaseController):

    logger = logging.getLogger("droidServerRequest")
    handlers = [r'/(?P<controller>mosaico)/?',
                r'/(?P<controller>mosaico)/(?P<id>[\d]+)/?',
                r'/(?P<controller>mosaico)/(?P<action>mosImg)/?',
                r'/(?P<controller>mosaico)/(?P<action>mosImg)/(?P<id>.+)/?',
                r'/(?P<controller>mosaico)/(?P<action>mosCol)/?',
                r'/(?P<controller>mosaico)/(?P<action>mosCol)/(?P<id>[0-9]+)/?',
                r'/(?P<controller>mosaico)/(?P<action>mosLinha)/?',
                r'/(?P<controller>mosaico)/(?P<action>mosLinha)/(?P<id>[0-9]+)/?',
                r'/(?P<controller>mosaico)/(?P<action>modId)/?',
                r'/(?P<controller>mosaico)/(?P<action>modId)/(?P<id>[0-9]+)/?',
                r'/(?P<controller>mosaico)/(?P<action>new)/?',
                r'/(?P<controller>mosaico)/(?P<action>update)/?',
                r'/(?P<controller>mosaico)/(?P<action>delete)/?',
                ]
    model_class = db.models.Mosaico
    form_remove_list = ['type']

    def list(self, id=None):

        if id is None:
            BaseController.list(self, id)
        else:
            s = db.utils.get_sql_session()
            try:
                q = s.query(db.models.Mosaico). \
                    filter(db.models.Mosaico.mosId == id)
                self.write(self.query2json(q))
            except Exception as e:
                self.sendHttpError(500)
                self.logger.error("{0}".format(e))
            finally:
                s.close()

    def mosImg(self, id=None):
        if id is None:
            BaseController.list(self, id)
        else:
            s = db.utils.get_sql_session()
```

Figura 12. Exemplo de código utiliza no *webserver*

5. Resultados Obtidos

5.1 Aplicação Android compatível com WebGL e X3DOM

A ideia inicial do projeto consistia em possibilitar integração entre a aplicação e modelos 3D na plataforma *mobile* de forma que o usuário pudesse navegar livremente pelo ambiente.

Nenhuma solução para ambiente *mobile* existia de forma definitiva, sendo assim que o usuário dependia de navegadores de terceiros, assim como tinha dependência de uma versão funcional para que pudesse navegar em um ambiente 3D.

Para tanto foi criado um *Browser* próprio, com base no código aberto do Fennec (distribuição do Mozilla Firefox para dispositivos móveis) no intuito de possibilitar o melhor funcionamento do sistema na plataforma *Android*.

Assim, a melhor solução encontrada foi aplicada no projeto, que consistiu em construir um aplicativo com base em um *Browser* com suporte a WebGL, e garantir em todas as versões desse aplicativo uma navegação funcional em ambiente *X3DOM*.

Temos então um aplicativo que pode ser utilizado em qualquer dispositivo *Android*, tendo em seu desempenho variando apenas conforme a diferença entre o hardware que está rodando.

5.2 Desenvolvimento de um Webserver

Foi criado um Webserver, que proporciona a uma aplicação *web* o acesso ao banco de dados através de uma API. Esse modelo de tráfego de dados é simples, seguro e rápido, atendendo assim as necessidades geradas pela aplicação “CityFreedom”, desenvolvida por Airton Barbosa.

Com esse modelo também temos segurança e facilidade de novos módulos serem adicionados ao *Webserver*.

Criar um WebServer que comporta o banco de dados nos dá a possibilidade de, acessar os dados do banco da mesma maneira (Figura 13) independentemente de qual linguagem foi utilizada para o desenvolvimento do portal.

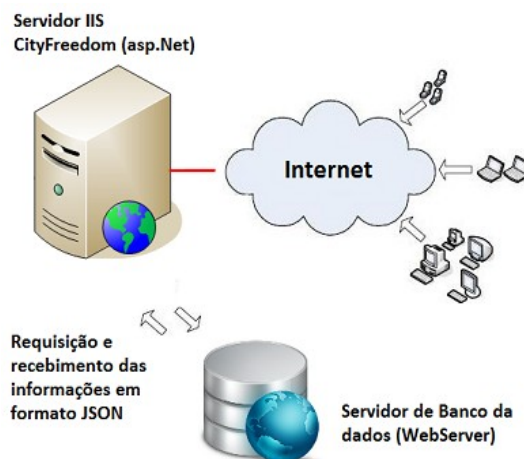


Figura 13. Estruturação do sistema

As requisições ao servidor de banco de dados são isoladas por meio de codificação JSON criptografada pelo protocolo SSL, trazendo segurança para o acesso aos dados - algo relevante para o sistema, que manipula informações de usuários. Através destas requisições, informações não só dos usuários do sistema como também das modelagens relacionadas a estes usuários são a resposta do servidor de banco de dados para o servidor da aplicação. Seguindo também a vertente de otimização, é através das requisições ao banco que o CityFreedom organiza e renderiza as modelagens de acordo com a opção de navegação, trazendo referências às páginas com mais ou menos detalhes - no banco são guardados os identificadores das modelagens, que variam dependendo da quantidade ou qualidade dos detalhes.

5.3 Integração com servidor de banco de dados através de WebServer e requisições em formato JSON

Foi desenvolvido um servidor de banco de dados que se comunicará com a aplicação “CityFreedom” por meio de requisições HTTP em formato JSON (semelhante ao XML, porém possibilita maior compactação de informações).

A integração das duas partes do projeto proporciona a criação de um sistema *web* completo, no que diz respeito à aplicação de todas as técnicas revolucionárias e pioneiras de programação oriundas dos estudos dos últimos anos, tudo isso unido à possibilidade oferecida de navegação e imersão em um universo 3D bem ao alcance de um simples browser.

6. Conclusões

Este trabalho apresenta uma visão das novas tecnologias com relação a aplicações *Android*, modelagem 3D e webserver, e a partir do estudo dessas tecnologias apresenta uma proposta de desenvolvimento sobre tais tecnologias.

Durante esse desenvolvimento foram discutidos vários aspectos sobre tais tecnologias, assim como estratégias para melhor desempenho e funcionalidade. Dada essas análises, foram sugeridas novas possibilidades, e escolhidas os melhores caminhos a seguir no desenvolvimento.

No que diz respeito a modelagem 3D, o resultado final foi obtido através do uso de *X3DOM* e *HTML5*, pois essas tecnologias satisfazem a qualidade descritas no objetivo, pois são suportadas pelo aplicativo *Android* desenvolvido, não necessitam de outros *plugins* e tem um bom desempenho.

Através do desenvolvimento de um aplicativo baseado no Fennec, é aberto a possibilidade, para usuários *Android*, da navegação em ambientes 3D, algo não possível com o browser padrão de tais dispositivos.

Com o desenvolvimento de um *Webserver* e sua API, foi possível fazer uma fácil integração entre a aplicação “CityFreedom”, possibilitando assim seu completo funcionamento, e seu posterior funcionamento através do acesso *web* pelo dispositivo móvel.

Foi possível, portanto, trazer a integração entre um aplicação *web* e o banco de dados através do Webserver, assim como trazer o mundo de um ambiente 3D ao usuários mobile.

Para trabalhos futuros, a aplicação *Android* deve ser personalizada, possibilitando uma melhor experiência aos usuários. Com relação ao Webserver, ele deverá ser acrescentado em funcionalidades conforme a demanda necessário pelas aplicações web que farão através dele seu acesso a um banco de dados.

7. Trabalhos Futuros

7.1. Banco de Dados e Controle de Usuários

Há melhorias a serem feitas no tratamento das informações de usuários e elas são relacionadas às modelagens de cada um destes usuários. Num primeiro momento, foi preparada a base de dados para que fosse possível demonstrar o funcionamento e a associação de um usuário ao sistema. No entanto resta criar as telas e vincular os usuários às modelagens específicas, o que possibilitará recursos como configurações e maneira de visualização dos objetos e cenários.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMS, L. **Virtualização e Realidade Virtual**, São Paulo, Ed Makron Books, p. 255-259.

ALONSO, GUSTAVO; CASATI FABIO; KUNO HARUMI; MACHIRAJU VIJAY. **Web Services: Concepts, Architectures and Applications**. 2010.

ARMSTRONG, E.; BALL, J.; BODOFF, S.; CARSON, D. B.; EVANS, I.; Dale, G.; HAASE, K.; JENDROCK, E. **ORACLE, Getting Started with Web Applications**. 2005. Disponível em: <http://docs.oracle.com/javaee/1.4/tutorial/doc>. Acesso em Jul. 2012.

ANDROID DEVELOPER, **Android, the world's most popular mobile platform**. Disponível em <http://developer.Android.com/about/index.html> acesso em Jul. de 2012

ANDROID, **Discover Android**. 2012. Disponível em <http://www.Android.com/about> acesso em Jul. de 2012.

BRUTZMAN, D.; DALY, L. **X3D, Extensive 3D Graphics for Web Authors**. 1ª Ed. [S.I.] Editora Morgan Kaufmann Publishes. 2007, p. 20-30.

DANIEL, J. **Crafting Digital Media**. 1ª Ed. [S.I.] Editora Apress. Parte 2, p. 145-147.

DELILLO, B. **Khronnon Group. WebGL – Open GL ES 2.0 for the Web**. 2009. Disponível em: <http://bjartr.blogspot.com.br/2009/10/more-webgl-progress-now-with-video.html>. Acesso em Jul. 2012.

FENNEC **Mozilla Firefox Mobile**. Disponível em [http://www.mozilla.com/en-US /mobile](http://www.mozilla.com/en-US/mobile) acesso em Ago. de 2012.

FESTA, P; BORLAND, J. **Is a 3D web more than just empty promises?** 2005. Disponível em www.zdnet.co.uk. Acesso em abril de 2012.

HAROLD, R. **XML Bible**. 1st Ed. [S.I.]: Editora Wiley. 1999.

JSON - **Introducing JSON** – Disponível em <http://www.json.org> acesso em Jul. de 2012

KIRNER, C; PINHO, M. **Uma Introdução à Realidade Virtual. Jornada de Autualização em Informática. Congresso Nacional da Sociedade Brasileira de Computação. ANAIS**. Recife, Agosto 1996.

KIRNER, C. 2011. **Realidade Virtual e Aumentada: Aplicações e Tendências**. 2011. Disponível em: <http://www.ckirner.com/realidadevirtual/?PUBLICA%C7%D5ES>. Acesso em Jun. 2012.

L. E. BORGES. **Python para Desenvolvedores – 2ª edição**. 2012

M. DORY, A. PARRISH, B. BERG. **Introduction to Tornado**. 2012

SGARBI, R. , ROCHA FERNANDES A. M. **Um Tutorial para Ensino de Modelagem de Modas Baseado em Ambiente 3D**. 2003

VALÉRIO, A; MACHADO, L. **Realidade Virtual – Definições, Dispositivos e Aplicações**. 2002. Disponível em: http://www.de.ufpb.br/~labteve/publi/2002_reic.pdf. Acesso em Jun, 2012.

X3DOM Group. **API and Examples**. 2012. Disponível em: <http://x3dom.org/docs/dev/>. Acesso em Jun. 2012.

X3D e WebGL – Disponível em http://www.x3dom.org/?page_id=2 acesso em Ago. de 2012.

Z. MEDNIEKS, L. DORNIN, G.B. MEIKE, M. NAKAMURA. **Programming Android**. 2011.

W3C Web Services Architecture. 2004. Disponível em <http://www.w3.org/TR/ws-arch>. Acesso em Julho de 2012