

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE INFORMÁTICA



PROJETO FINAL DE CURSO

SiCopaTVD **Sistema para Copa de 2010** **aplicado a plataforma de TV Digital.**

Rodolfo da Costa Ladeira
Graduando em Ciência da Computação

Vladimir Oliveira Di Iorio
(Orientador)

VIÇOSA - MINAS GERAIS
Julho – 2010



RESUMO

SiCopaTVD- Sistema para Copa de 2010 aplicado a plataforma de TV Digital.

Vladimir Oliveira Di Iorio (Orientador)

Rodolfo da Costa Ladeira (Estudante)

RESUMO

Com o advento da interatividade digital em nosso país, surgem grandes oportunidades nas áreas de informação e comunicação. As oportunidades vão desde gerar capital, até mesmo o aprendizado de novas tecnologias e a criação de um novo paradigma de diversão. Especialistas relatam que com a chegada da TV DIGITAL a conhecida internet via PC ou outro periférico parecido cairia em desuso, muitos falam em convergência, outros falam em paridade de tecnologias (ambas andem lado a lado). O importante a ser citado, é que, TV DIGITAL trará com certeza novos campos de trabalho e uma nova forma de assistir TV, tornando-se algo muito mais participativo. Pensar em TV é pensar em futebol, pois futebol gera grandes índices de audiência nas emissoras; e falando em futebol não podemos deixar de lembrar o evento magnífico que terá sua ocorrência em 2010, a “Copa do Mundo da África do Sul” ou “Copa do Mundo da África“, como muitos já estão chamando-a. O Ministro da Comunicação, Helio Costa, prevê um aumento significativo dos usuários de TV DIGITAL na próxima Copa, tornando-a mais popular, observando que tal tecnologia já completa três anos (em nosso país) e em muitos lares onde se tem abrangência de sinal, moradores não sabem nem ao menos o que seria tal tecnologia.

Juntando todo o conjunto de informação passado anteriormente, o gosto por futebol, e em especial a vontade constante em aprender novas tecnologias é que foi pensado em fazer tal projeto. Algo que fosse totalmente novo, que pudesse abranger uma fatia de mercado pouco explorada, que expandisse o conhecimento sobre o assunto e o principal, que criasse um produto final aprimorando o nível de interatividade com o usuário, numa mistura de TV, jogo eletrônico, informação sobre a Copa e álbum de figurinhas.

PALAVRAS-CHAVE

Interatividade digital, comunicação visual, sistemas de informação

ÁREA DE CONHECIMENTO

1.03.03.04-9 - Sistemas de Informação

LINHA DE PESQUISA

(24) Sistemas de Informação



1. Introdução

A TV Digital é um sistema em que as imagens e dados são transmitidos a torre para a televisão no formato de bits (menor unidade de dados digitais). Desta forma, ocorre uma perda muito pequena de dados, e a imagem e o som chegam com alta qualidade (até sete vezes melhor que a analógica). Na cidade de São Paulo, as transmissões da TV Digital (DTV) começaram no dia 2 de dezembro de 2007. O cronograma estabelece que, no primeiro semestre de 2008, chegue ao município do Rio de Janeiro. Até o final de 2008, todas as capitais estarão incluídas no sistema. A previsão é que as transmissões do sinal digital cheguem a todo território nacional até o ano de 2013. Em 2016, o sinal analógico deixará de ser transmitido. Para captar o sinal de TV digital são necessários: uma televisão (de tubo, plasma ou LCD), uma antena de UHF e um conversor (set-top Box). Neste primeiro momento, o sistema digital brasileiro vai apenas melhorar a qualidade da imagem e som das transmissões. Porém, brevemente, serviços interativos serão adicionados, através do padrão Ginga [1]. Votações interativas, dados sobre a programação, games, compras virtuais, movimentação bancária, troca de mensagens entre outros recursos poderão entrar em operação. Quem tem televisor de alta definição (Full HD ou HD Ready) poderá visualizar uma imagem com extrema qualidade. Porém, quem estiver usando um televisor de tubo também vai perceber uma grande melhoria de imagem e som, pois o sistema digital elimina chiados, ruídos e sombras. Uma das vantagens da TV Digital é a mobilidade. Através de televisores específicos portáteis é possível captar a imagem mesmo em movimento. Também é possível assistir os canais de televisão em telefones celulares com este recurso. Como o sistema de transmissão digital é mais compacto, é possível transmitir até oito canais digitais numa mesma frequência do analógico. Portanto, a tendência é que, futuramente, tenhamos mais canais sendo transmitidos pelas emissoras de televisão.

Pensar em TV é pensar em futebol, pois futebol gera grandes índices de audiência nas emissoras; e falando em futebol não podemos deixar de lembrar o evento magnífico que terá sua ocorrência em 2010, a “Copa do Mundo da África do Sul” ou “Copa do Mundo da África“, como muitos já estão chamando-a. O Ministro da Comunicação, Helio Costa, prevê um aumento significativo dos usuários de TV DIGITAL na próxima Copa, tornando-a mais popular, observando que tal tecnologia já completa três anos (em nosso país) e em muitos lares onde se tem abrangência de sinal, moradores não sabem nem ao menos o que seria tal tecnologia. Uma nota importante a se observar é que esta será a primeira copa a ser transmitida com qualidade digital de imagem e som.

Juntando todo o conjunto de informação passado anteriormente, o gosto por futebol, e em especial a vontade constante em aprender novas tecnologias é que foi pensado em fazer tal projeto. Algo que fosse totalmente novo, que pudesse abranger uma fatia de mercado pouco explorada, que expandisse o conhecimento sobre o assunto e o principal, que criasse um produto final aprimorando o nível de interatividade com o usuário, numa mistura de TV, jogo eletrônico, informação sobre a Copa e álbum de figurinhas.



1.2 Objetivos

Como dito anteriormente o software visa misturar áreas e criar um ambiente para grande interatividade em TV DIGITAL, em específico um software que contemple a Copa do Mundo da África.

Objetivos específicos

- Aprender uma linguagem voltada para edição em softwares para TV DIGITAL.
- Utilizar o conhecimento adquirido ao longo do curso nesse projeto.
- Ter um produto final que possa ser reutilizado e ou reprogramado por outros que assim queiram.

2. Base Teórica

Tendo em vista que a tecnologia envolvida neste projeto é algo extremamente novo, a base teórica especificamente será adquirida na web e talvez também através do conhecimento de terceiros, adquiridos em projetos anteriores. Foi feito um estudo prévio (através de informações provenientes de sites) das linguagens para a edição de software para TV DIGITAL, NCL e JAVATV, escolhendo a linguagem que melhor se encaixa nos propósitos do projeto, o padrão Ginga contempla as duas linguagens, porém não teve sua homologação a tempo do projeto para ser escolhido. E, por conseguinte, toda base teórica será adquirida à medida que se haja avanço e necessidade no projeto. É certo que conhecimento sobre banco de dados: o software irá explorar o banco de dados interno (contido no receptor) e o banco de dados da emissora; o conhecimento sobre funcionalidades básicas da linguagem e design gráfico foram necessários no processo de desenvolvimento do software. Foram estudados também padrões que envolvam a TV Digital, aplicando-os, a fim de aumentar a acessibilidade do software. E por fim, para melhorar o processo de desenvolvimento foi adotados documentos para desenvolvimento de softwares, como documentações do tipo (ERSW).

3. Metodologia

O sistema foi desenvolvido pelo modelo misto, e foi criado módulos, onde a cada fim dos mesmos uma bateria de testes será executada para garantir a sua integridade. O método misto foi escolhido devido a problemas referentes à implementação do banco de dados, fazendo com que se mude a direção da programação. Para testes foi usado protótipos de interfaces. O projeto seguiu a idéia de Desenvolvimento Ágil de Processo, porém não foram usados modelos conhecidos, mas idéias dos mesmos poderão ser aproveitadas e adaptadas (como o Scrum). Sabendo da grandeza da proposta e do curto tempo para execução o processo foi totalmente documentado, a fim de favorecer terceiros que queiram dar continuidade ao projeto, e independentemente do que ocorra o código fonte será livre para possíveis alterações e de maneira alguma deve ser utilizado para fins lucrativos, atentando os objetivos já descritos.



3.1. Requisitos

O sistema deve oferecer um conjunto de funções com navegabilidade pelo controle remoto, criando uma maior interatividade ao se assistir os jogos da Copa do Mundo. Menus consultas, do tipo consulta de jogadores, seleções, estádios, tabelas e classificação e da moeda Lair (que será falada adiante), estarão presentes. Compras de equipamento da seleção preferida também serão possíveis, antemão um cadastro, sendo este, o cadastro, armazenado em um banco de dados interno para facilitar futuros acessos, através de um login. Pode haver também a opção de compra de figurinhas. Este se dará através de uma moeda fictícia denominada Lair. O usuário ganha mais Lares ao assistir mais jogos. A idéia de compra de figurinhas é que o usuário comprará um pacote fechado contendo certo número de figurinhas, estas nunca serão repetidas. O usuário terá uma lista rápida de acesso contendo os próximos jogos com a respectiva emissora transmissora, possibilitando assim o usuário escolher uma das opções de jogos rapidamente. A fim de aumentar a interatividade, expandindo-a extra tela, o usuário pode exportar/importar figurinhas através de alguma mídia de armazenamento. O objetivo principal, com tal exportação/importação, é tornar moderno, a antiga idéia de troca de figurinhas entre adeptos.

Visto a dificuldade em navegar via controle remoto, um teclado virtual deve ser implementado, teclas de atalhos devem existir, e para evitar que o usuário sempre tenha que preencher seus dados; o uso login agiliza o processo e evita que o usuário se canse do programa. Pensando em tornar o projeto agradável para o usuário todo acesso a menus não deve exceder o tempo de 3s e a interface deve ser simples, porém bonita e objetiva.

Ator: Usuário: realiza qualquer operação do sistema.

3.1.1. Requisitos Funcionais

Código	Nome	Descrição
RF1	Consulta de seleções	Usuário pode consultar as seleções participantes e também alguma outra informação sobre a mesma
RF2	Consulta de jogadores	Usuário pode consultar os jogadores de cada seleção participante e também alguma outra informação sobre a mesmo
RF3	Consulta de tabela de jogos e classificação	Usuário pode consultar a uma tabela de jogos, e classificação onde contem também informações do tipo cartões, gols...
RF4	Consulta de figurinhas	Usuário pode consultar as figurinhas de cada jogador adquiridas pelo mesmo
RF5	Comprar novas figurinhas	O usuário deve estar apto a comprar novas figurinhas de jogadores usando os seus Lares
RF6	Compra de equipamento da seleção	O usuário deve estar apto a comprar um equipamento (camisa, meião...) da seleção desejada



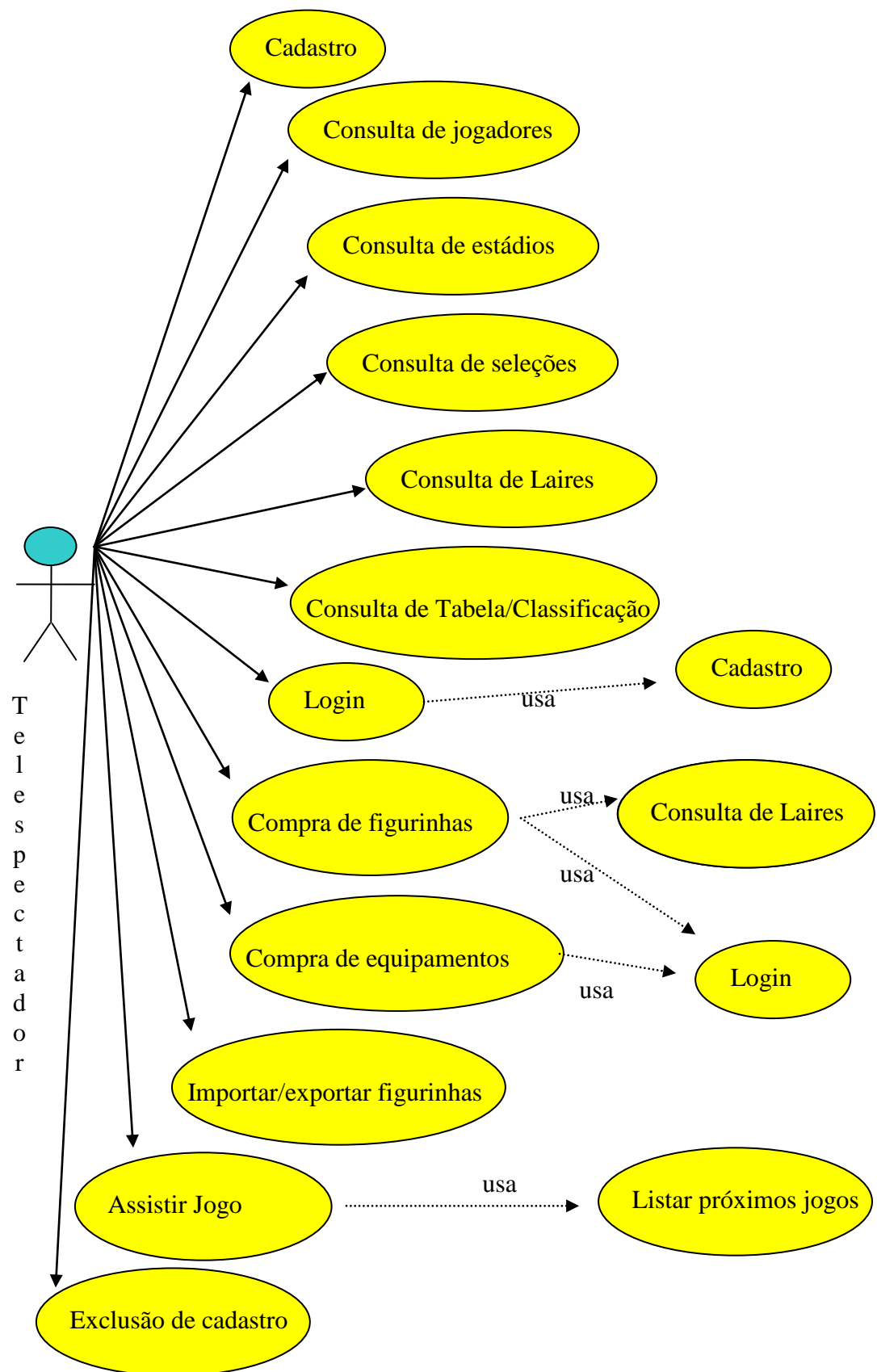
RF7	Consulta de estádios	O usuário pode ver também os estádios e algumas informações sobre o mesmo
RF8	Assistir jogo	Escolhendo-se uma emissora transmissora do jogo o usuário verá o jogo desejado
RF9	Listar próximos jogos	Deve haver uma lista prática de próximos jogos
RF10	Exportar figurinhas	Permitir que o usuário exporte através de algum meio de armazenamento as figurinhas desejadas
RF11	Importar figurinhas	Permitir que o usuário importe através de algum meio de armazenamento as figurinhas desejadas
RF12	Consulta de moeda interna	O usuário pode consultar quanto tem de moeda
RF13	Cadastro de usuário	Cadastra dados importantes pra facilitar (agilizar) serviços.
RF14	Login	Login do usuário, apenas escolhe e carrega as informações sobre um usuário pré-cadastrado
RF15	Exclusão de usuário	Exclui um usuário pré selecionado.

3.1.2. Requisitos não Funcionais

Código	Nome	Descrição
RNF1	Interface leve e prática	O sistema deve permitir interface leve e prática com acesso a todos os serviços com poucos comandos
RNF2	Desempenho	O sistema deve responder em no máximo 3s a cada entrada de menu
RNF3	Moeda fictícia	O sistema deve oferecer uma moeda, Lair (plural: Laíres), que tenha valor fictício.
RNF4	Teclado virtual	Deve haver um teclado virtual em caso de cadastro pra compra
RNF5	Arquivo temporário	O sistema deve carregar informações do ultimo usuário a fim de agilizar compras.
RNF6	Importe/exporte seguro	No importe e exporte de figurinhas o sistema deve ser seguro para que apenas tal conteúdo seja transferido
RNF7	Atalhos	Deve haver uma tecla de atalho pra cada menu principal



3.2. Diagrama de casos de Uso



3.3. Casos de uso

Consulta de jogadores	
Pré condições:	O usuário deve estar na tela de consulta de jogadores
Requisitos atendidos	RF2
Fluxo principal	1. O usuário escolhe a seleção que queira ver um determinado jogador 2. O usuário escolhe o jogador 3. Vê as características deste

Consulta de seleções	
Pré condições:	O usuário deve estar na tela de consulta de seleção (equipe)
Requisitos atendidos	RF1
Fluxo principal	1. O usuário escolhe a seleção 2. Vê as informações da mesma

Consulta de tabela/classificação	
Pré condições:	O usuário deve estar na tela de consulta de tabela/classificação
Requisitos atendidos	RF3
Fluxo principal	1. Vê as informações desejadas

Consulta de Lares	
Pré condições:	O usuário deve estar na tela de consulta de Lares
Requisitos atendidos	RF12
Fluxo principal	1. Vê as informações dos Lares (quantidade)

Consulta de estádios	
Pré condições:	O usuário deve estar na tela de consulta de estádios
Requisitos atendidos	RF7
Fluxo principal	1. O usuário escolhe o estádio desejado 2. Vê as informações do estádio escolhido

Login de usuário	
Pré condições:	O usuário deve estar na tela de login de usuário
Requisitos atendidos	RF14
Fluxo principal	1. O usuário define dentre opções qual usuário em atividade 2. Confirma cadastro
Fluxo alternativo	1. a) Não há usuários: Sistema retorna uma mensagem de erro. 1.1.a) Sistema retorna a tela principal



Compra de equipamento	
Pré condições:	O usuário deve estar na tela de compra de equipamento
Requisitos atendidos	RF6, RF14
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário "loga" 2. O usuário define qual equipe deseja compra certo equipamento 3. O usuário verifica qual o(s) produtos deseja comprar 4. O usuário define qual a loja sobre cada produto deseja comprar 5. O usuário confirma a compra do equipamento selecionado 6. O sistema retorna a tela principal
Fluxo alternativo	<ol style="list-style-type: none"> 1. a) O usuário não tem login: Sistema entra na tela de cadastro de dados <ol style="list-style-type: none"> 1.1.a) O usuário deve cadastrar os seus dados 1.2.a) Sistema retorna a tela de compra de equipamento 5. a) Caso haja qualquer falha na compra: Sistema aborta a compra <ol style="list-style-type: none"> 5.1.a) Envia mensagem de erro ao usuário 5.2.a) Retorna a tela principal

Compra de figurinhas	
Pré condições:	O usuário deve estar na tela de compra de figurinhas
Requisitos atendidos	RF5, RF12
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário loga 2. O usuário define qual a quantidade de pacotes na compra 3. O usuário confirma a compra de pacotes de figurinhas 4. O sistema mostra as figurinhas adquiridas 5. O sistema retorna a tela principal
Fluxo alternativo	<ol style="list-style-type: none"> 1. a) O usuário não tem login: Sistema entra na tela de cadastro de dados <ol style="list-style-type: none"> 1.1.a) O usuário deve cadastrar os seus dados 1.2.a) Sistema retorna a tela de compra de figurinhas 3. a) Se o usuário não tiver Lares suficiente: O sistema aborta compra <ol style="list-style-type: none"> 3.1.a) Retorna uma mensagem de erro 3.2.a) Volta à tela de compra de figurinhas

Importar/exportar figurinhas	
Pré condições:	O usuário deve estar na tela de exportar/importar
Requisitos atendidos	RF10, RF11
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário escolhe a opção importar ou exportar 2. O usuário escolhe quais figurinhas deseja fazer o processo 3. Sistema confirma o processo com uma mensagem 4. O sistema retorna a tela de exportar/importar
Fluxo alternativo	<ol style="list-style-type: none"> 1. a) Não há dispositivo de armazenamento: Sistema retorna msg de erro 1.1.a) Sistema retorna a tela principal 3. a) Não há figurinhas: o sistema retorna mensagem de erro 3.1.a) Volta à tela de importar/exportar figurinhas

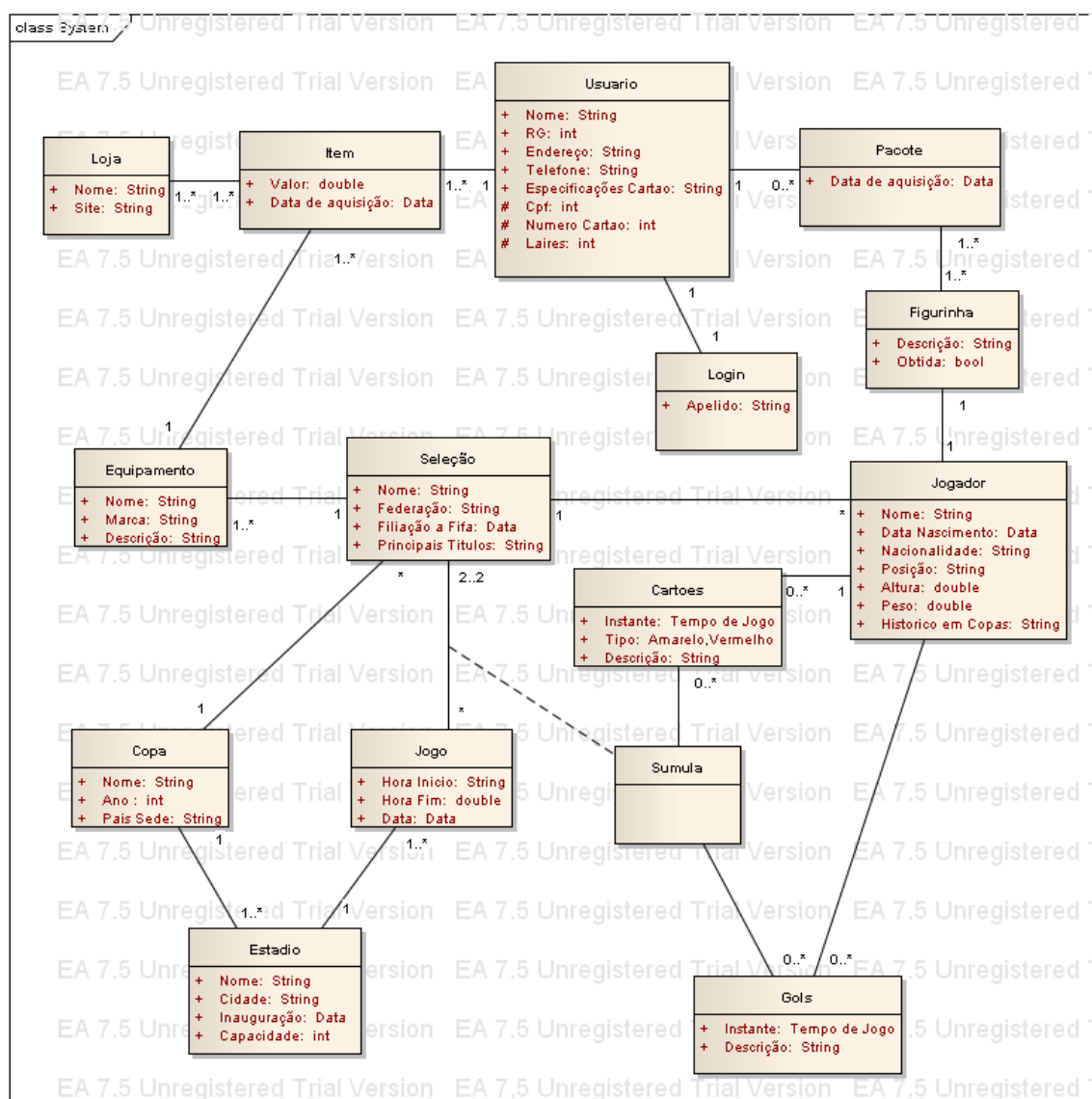
Assistir jogos	
Pré condições:	O usuário deve estar na tela principal do programa
Requisitos atendidos	RF8, RF9
Fluxo principal	<ol style="list-style-type: none"> 1. Através da lista de rápida de jogos o usuário escolhe um jogo a assistir 2. O usuário escolhe a emissora a qual deseja assistir 3. Sistema entra no modo jogo
Fluxo alternativo	<ol style="list-style-type: none"> 1. a) O jogo ainda não começou: Sistema retorna uma mensagem de erro. 1.1.a) Sistema retorna a tela principal

Cadastro de usuário	
Pré condições:	O usuário deve estar na tela de cadastro de usuário
Requisitos atendidos	RF13
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário preenche seus dados pessoais (identificador, nome..) 2. O usuário preenche dados de localidade (endereço) 3. O usuário preenche dados de cartão 4. Confirma cadastro 5. Sistema armazena em arquivo temporário as informações



Exclusão de usuário	
Pré condições:	O usuário deve estar na tela de exclusão de usuário
Requisitos atendidos	RF15
Fluxo principal	1. O usuário define através do login qual usuário a ser excluído 2. Confirma exclusão 3. Sistema retorna a tela principal
Fluxo alternativo	1.a) O usuário escolhido está logado: Sistema retorna mensagem de erro 1.1.a) O Sistema retorna a tela principal

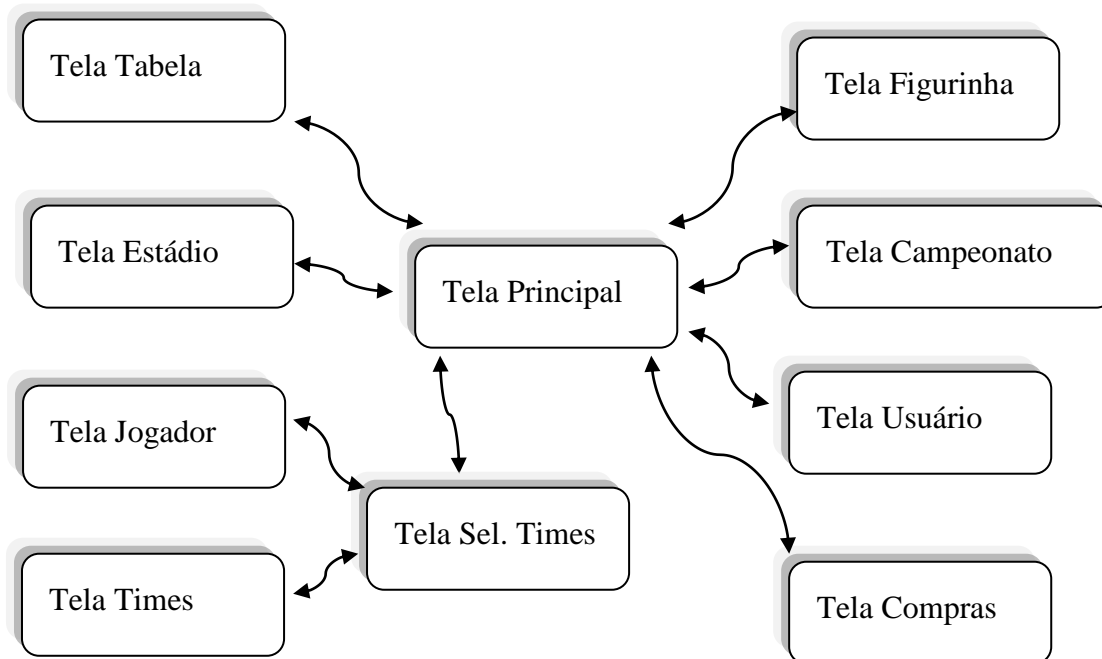
3.4. Diagrama de classe de domínio



3.5. Diagrama de fluxo de telas

Podemos observar na Figura 3.5.1, abaixo como é o fluxo das telas do programa.

Figura 3.1.1-Fluxo de telas do programa



3.6. A implementação

Antes mesmo de iniciar o nosso relato sobre a implementação utilizada é importantíssimo frisar que a tecnologia utilizada está constantemente em evolução, sendo assim muitas diferenças da tecnologia atual podem ser verificadas. Relataremos brevemente sobre a tecnologia utilizada a fim de guiar o leitor. Para edição de códigos o eclipse Europa versão 3.3.2 foi utilizado. A api javaTV[5] versão 1.0 deu o suporte ao uso da tecnologia de TV digital para JAVA. Para visualização das xlets usou-se o emulador xletview versão 0.36. Para edição de imagens foi usado o Photoshop CS3. A princípio foi estipulado que o desenvolvimento do programa seria do tipo bottom-up, porém como a tecnologia utilizada é muito nova e a maior parte do programa está concentrada na parte gráfica visto que é um programa televisual, passou a ser desenvolvida de forma top-down. Em exceção a parte de persistência, que foi desenvolvido primeiro, pois a parte mais importante do programa é seu banco, contendo as informações necessárias a serem repassadas para o usuário. Com isto o modelo adotado pode ser considerado o misto. A tecnologia do emulador não suporta ferramentas como derby e hibernate que facilitariam a criação de bancos, era necessário então criar uma forma de criar um banco antes mesmo de começar a trabalhar com o projeto, esta forma foi lidar com arquivos de entrada e saída. As sessões posteriores são

para explicar como cada parte do projeto que foi desenvolvida, a separação em itens facilita a compreensão isolada.

3.6.1. Do Banco de Dados

O banco de dados do programa é dividido em duas partes: banco local e banco dinâmico. Esta divisão ocorre, pois para o nosso problema sempre teremos um conjunto de informações que é fixa e outra que é atualizada à medida que se passa o evento (copa). Podemos pensar que sempre teremos os mesmos jogadores em seus times ao iniciar o programa, mas não podemos garantir que a tabela será a mesma ao iniciar o programa, pois a mesma pode sofrer alterações (resultados de jogos). Daí para o banco fixo, também chamado banco local, usamos arquivos de entrada simples com extensão “sdb”, este é um formato fictício desenvolvido apenas para o programa, contendo as informações requeridas. Podemos verificar um arquivo de entrada do banco na Figura 3.6.1.1

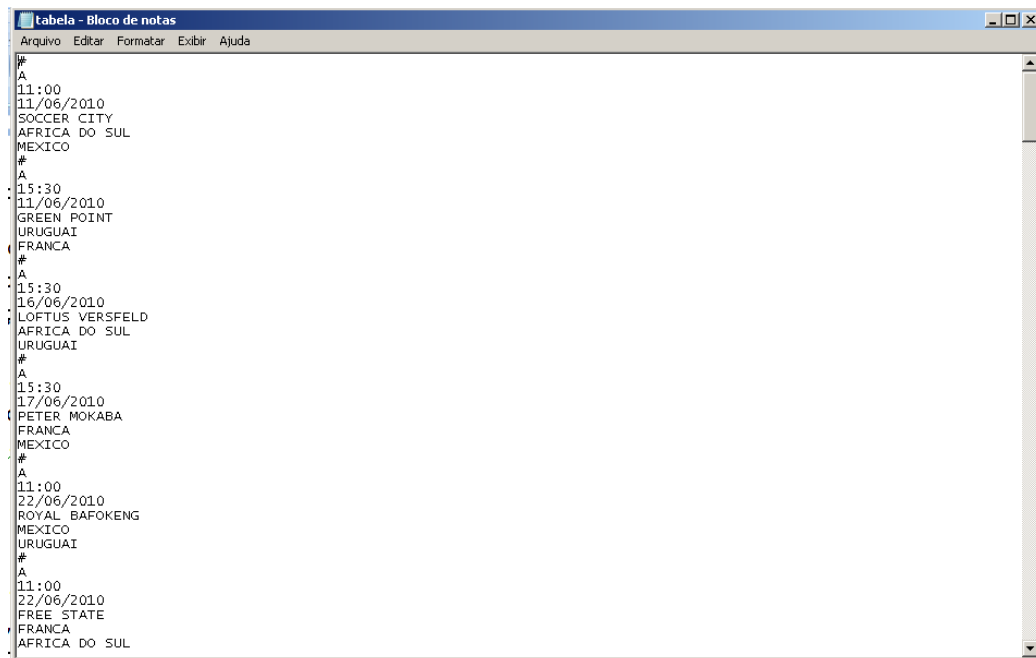


Figura 3.6.1.1 – Figura com parte do arquivo tabela.sdb aberta, contém informação de toda a tabela da fase de grupos, na figura está sendo mostrada apenas o grupo A.

Assim tais informações são lidas do arquivo e preenchidas em objetos específicos. Ou seja, o programa tem um objeto Copa, que segue o padrão singleton, que contem objetos times, estádios, jogadores e tabela. Então, ao iniciar o programa o banco local é preenchido, e as informações básicas já podem ser visualizadas. Se pensarmos em termos de código, o programa para acessar um objeto externo usa a idéia de carrossel [6]. Para explicar melhor, podemos ver a Figura 3.6.1.2.

```

private static void scanJogadores(String dbJogadores)
    throws IOException {

    CarouselFile carousel = new CarouselFile("banco de dados local/jogadores.sdb");
    RandomAccessFile randomFile = new RandomAccessFile(carousel.getAbsolutePath(), "r");

    String jogadorIdentificador = "#";
    String outroJogador = randomFile.readLine();

    // lendo as linhas do arquivo

```

Figura 3.6.1.2 - Trecho do método scanJogadores, pertencente a classe ScanDb com uso de objeto carrossel para escanear um arquivo de entrada.

Ao se criar um objeto do tipo CarouselFile, o mesmo tem a raiz localizada na pasta do emulador xletview e o restante do caminho é repassado como parâmetro do construtor. Ao contrário de imagens, apenas com o uso de objetos carrossel podemos acessar arquivos do tipo texto. Logo após isto, pode-se usar a classe RandomAccessFile, pertence ao pacote utilitário de Java, passando o caminho absoluto da localização do carrossel e no nosso caso “r”(apenas leitura) como parâmetros do construtor para acesso a arquivos de io.

3.6.2. Da Interface Gráfica

Toda parte gráfica do programa SiCopaTVD foi desenvolvido pelo programa Photoshop CS3, desenvolvido pela Adobe. A idéia foi criar um conjunto de imagens onde o usuário navegasse entre figuras, que são mostradas em forma de menu e pudesse acessar as informações que desejasse. Uma dada cena pode conter várias imagens diferentes, tornando-a mais dinâmica. O programa tem a noção de qual conjunto de imagens utilizar devido a separação em objetos, onde cada tela tem um objeto característico.

3.6.2.1. Tela Menu Principal

Vamos falar um pouco sobre esta tela, ou classe, pois assim pode ser verificada mais claramente a forma utilizada para edição das telas. A Figura 3.6.2.1.1 mostra a tela do menu principal, a imagem padrão é a exibida na figura.



Figura 3.6.2.1.1 – Emulador XletView com o xlet do programa SiCopaTVD carregado na tela de menu principal.

O usuário pode navegar no controle remoto virtual e explorar esta tela. Abaixo, na Figura 3.6.2.1.2, temos o conjunto de imagens possíveis para a cena (ou tela corrente).



Figura 3.6.2.1.2 – Conjunto de imagens possíveis para a tela de menu principal, visualizadas no emulador XletView

Como já mencionado o menu tela principal é um objeto que contém um conjunto de imagens que podem variar dentro de uma cena, dependendo da navegação do usuário. Ao invocar uma instância da classe, o conjunto de imagens é carregado à medida que é necessário, então a cena é repintada. Pensando em termos de código a cena nada mais é que um container, contendo um conjunto de elementos que constrói a imagem a ser passada. Podemos observar uma estruturação de uma tela na Figura 3.6.2.1.3.

```
public TelaMenuPrincipal(HScene cena, int menuCorrente) {
    this.cena = cena;
    this.controladorDeEvento = EventManager.getInstance();
    this.menuCorrente = menuCorrente;
    this.animacao = new HAnimation();
}

public void inicia() {
    // criando a imagem padrao
    Image imagenPadrao = imagens
        .carregarImagem(urls.menuPrincipalJogadores);

    imagemMenu = new HIcon(imagenPadrao, 0, 0, cena.getWidth(), cena
        .getHeight());

    logo = new LogoPrograma();
    logo.setVisible(true);

    // criando atalhos de botoes
    criaAtalhos();

    // adicionando componentes a cena
    cena.add(logo);
    cena.add(atalhos);
    cena.add(animacao);
    cena.add(imagemMenu);
    cena.repaint();
}
```

Figura 3.6.2.1.3 - Figura contendo trecho do código da classe TelaMenuPrincipal, em que é mostrado o construtor e um método inicia

O funcionamento de toda tela, ou melhor, de todo menu, é sempre criar o objeto e logo após iniciar os componentes, atributos, desse menu. Na parte de inicialização é que se faz a inserção dos componentes que participaram da cena naquele dado estado.

Dos componentes de uma cena

Como a cena é um container ela pode conter outros componentes (containers). Para facilitar a compreensão e a edição do código algumas cenas contém containers com funcionalidades específicas. Podemos dar um exemplo significativo ao falar do menu estádio, para tal daremos uma breve olhada em seu código. O mesmo pode ser visto na Figura 3.6.2.1.4

```
public class TelaMenuEstadio implements UserEventListener {

    // gerenciadores
    private EventManager controladorDeEvento;
    private Imagens imagens = Imagens.getInstance();

    private HScene cena;

    // padrao singleton
    private static TelaMenuEstadio Instance = null;
    // componentes presentes na cena
    public mapaDeSedes mapaDeSedesMenu;
    private HIcon iconeInfo, iconeInfoImagem;
    private HIcon imagemPadrao;
    private mapaAfricaDoSul mapaAfricasDoSulMenu;
    public logoPrograma logo;
    private HStaticText infoSede;
    private HIcon imagemSede;

    // sede corrente identificando a informacao de qual sede mostrar
    private Estadio estadioCorrente;

    // ajuda/atalhos
    private listaDeAtalhos atalhos;
    private boolean mostrandoListaDeAtalhos = false;
```

Figura 3.6.2.1.4 – Parte do código do menu estádio mostrando os containers que estão presentes em sua cena.

Na figura acima pode ser verificado que um dado objeto TelaMenuEstadio terá atributos do tipo mapaDeSedes, mapaAfricaDoSul entre outros, que são objetos container que compõe a cena do objeto TelaMenuEstadio. As partes do objeto (containers) terão um conjunto de funcionalidades que no fim deixam o código mais limpo e tomam pra si o dever sobre uma dada função. Para ser mais claro, podemos citar o caso do objeto mapaDeSedes que a medida que o usuário caminha sobre o menu ele tem o dever de mudar o cursor de lugar, ou no caso de mapaAfricaDoSul mudar o local geográfico, entre outras funções. Podemos verificar na Figura 3.6.2.1.5 a separação do menu Estádio.

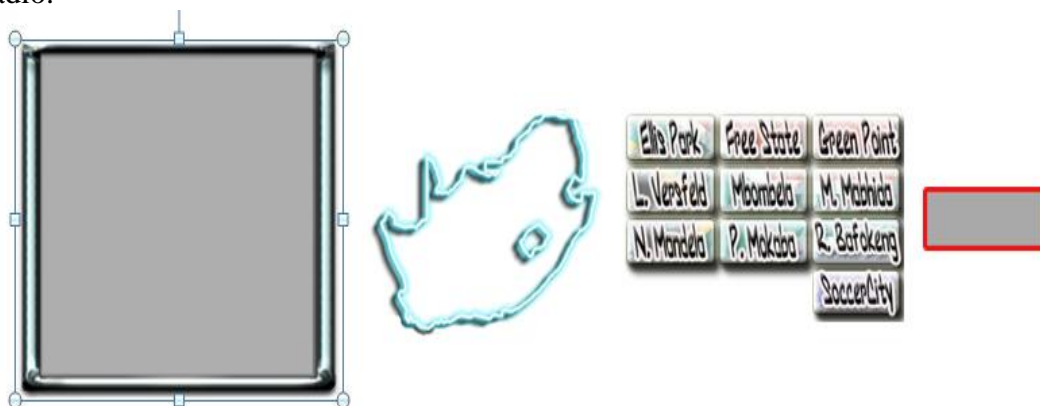


Figura 3.6.2.1.5 – Figura contendo partes(imagens) do menu tela Estádio. Tela de info, mapa da África do Sul, mapa das Sedes, cursor.

A montagem da cena do menu Estádio pode ser verificada na Figura 3.6.2.1.6.

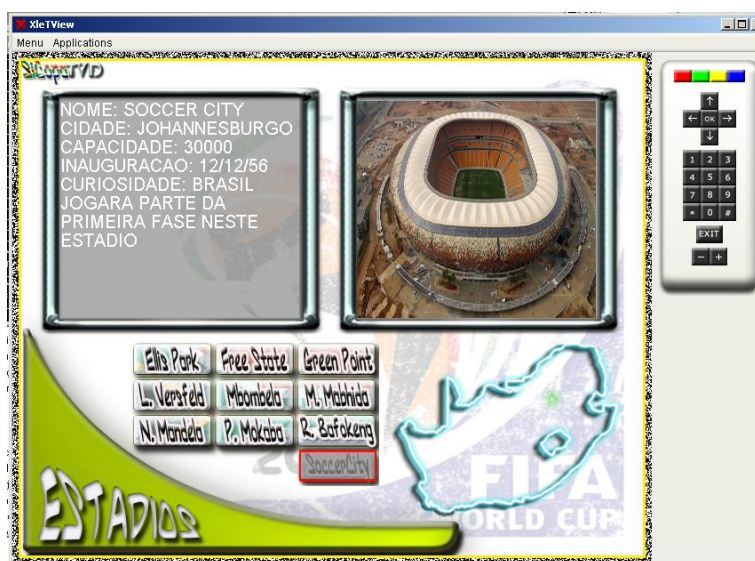


Figura 3.6.2.1.6 – Emulador xletview o programa SiCopaTVD rodando sobre a tela de menu de estádios.

Adotando esta técnica a utilização de componentes torna-se mais clara e fácil até mesmo para reutilização.

3.6.3. Dos testes e verificação de erros

É necessária tal seção, pois, como a plataforma utilizada para edição do código, o Eclipse, não é a mesma usada para a emulação da xlet; ou melhor, o programa eclipse não consegue rodar internamente xlets, para isto temos o emulador xletview. Para o uso de debug deve ser configurada uma porta de comunicação com o eclipse para acesso externo ao mesmo. A porta que mais foi utilizada foi a 8000. A técnica de usar o debug externo ao eclipse [2] foi muito utilizada vista a pouco conhecimento a alguns aspectos da tecnologia utilizada o que facilita muito a correção de erros. Podemos verificar a Figura 3.6.2.1.7, abaixo, com o trecho de chamada do emulador pelo prompt usando o debug.

```
Administrador: C:\Windows\system32\cmd.exe - java -Xdebug -Xnoagent -Xrunjdwp:transport=dt_socket,
Extra...false
Adminstrador: C:\Windows\system32\cmd.exe - java -Xdebug -Xnoagent -Xrunjdwp:transport=dt_socket,
Extra...false
tem-se selecoes nesse grupo0
Extra...false
tem-se selecoes nesse grupo0
Extra...false
tem-se selecoes nesse grupo0
Extra...false
tem-se selecoes nesse grupo0
Extra...false
Extra...false

D:\Documents\rodolfo\faculdade\Projeto final\programas\emulador\Xletview\java -X
debug -Xnoagent -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n -J
ar Xletview.jar
Listening for transport dt_socket at address: 8000
XletView, Copyright (C) 2003 - 2004 Martin Sveden
XletView comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute
it under certain conditions;
see license document for details.
*****
setting properties...
```

Figura 3.6.2.1.7 – Prompt de comando fazendo a chamada do emulador xletview com o uso da porta 8000 para debug

Quanto aos testes, tirando o desenvolvimento do banco, todo o resto foi testado dentro do emulador parte a parte, à medida que era construído era testado, a fim de evitar uma possível dispersão numa de erros. Em especial para o banco de dados foi criado um modelo de testes que testava internamente ao eclipse o uso do banco (se tal estava sendo



acessado e construído de forma correta). Esta técnica por hora foi enganosa, ao usá-la a leitura dos arquivos de entrada estava correta, porém ao transpor para o emulador não obtínhamos o resultado esperado, como dito anteriormente, o emulador não consegue acessar um arquivo io, daí a necessidade do uso de carrossel para o acesso de arquivos io. À medida que tal foi usado, os problemas de acesso foram selecionados.

4. Resultados

Esta seção trará os resultados sobre o programa SiCopaTVD. Os resultados não serão mostrados em forma de gráficos, ou comparativos, visto que o mesmo é apenas um programa de consulta. Por isso, para tornar o processo de compreensão dos resultados mais claro, separamos em itens. A seguir vamos falar do resultado sobre as telas, sobre o banco de dados, e as regras de negócio figurinha e moeda Lair.

4.1. Telas

Como o programa tem um conjunto de telas o resultado sobre as mesmas está separado perante esta divisão. Nos itens será falado um pouco sobre o menu em questão e as opções do controle remoto do usuário. O importante frisar que toda tela tem uma subtela de atalhos no qual mostra quais as teclas e suas funções ativas para a dada tela, definida pela tecla *, e a tecla vermelha também presente em muitas retorna a tela anterior.

4.1.1. Tela Principal

Esta é a tela principal do programa, nesta tela o usuário escolhe qual o sub-menu deseja entrar. Pode haver navegação na mesma trocando apenas as imagens que designam qual o sub-menu o usuário acessará(caso aperte ok). Figura 4.1.1.1 contém a imagem da tela referente.



Figura 4.1.1.1 - Emulador xletview com SiCopaTVD carregado na tela principal

Atalhos:

←, → Troca de menu

↑,↓ Troca o sub-menu

OK acessa o sub-menu



4.1.2. Tela Jogadores

Tela onde contém informações dos jogadores de uma seleção escolhida na tela de seleções (tela anterior que dá acesso a esta). Se a seleção tiver mais do que onze jogadores haverá páginas de lista de jogadores. Figura 4.1.2.1 contém a imagem da tela referente.



Figura 4.1.2.1 – Emulador xletview com SiCopaTVD carregado e na tela de jogadores da seleção brasileira

Atalhos:

←, → Troca a pagina da lista de jogadores

↑, ↓ Troca o jogador

4.1.3. Tela de times

Tela onde contém informações sobre um dado time escolhido na tela de seleções (tela anterior). Esta tela é simples e não tem funcionalidades do controle. Figura 4.1.3.1 contém a imagem da tela referente.



Figura 4.1.3.1 – Emulador xletview com SiCopaTVD carregado na tela de seleções com a Espanha selecionada.

4.1.4. Tela de estádios

Tela que contém informações sobre os estádios presentes na copa do mundo. Nesta tela tem-se um mapa de estádios onde o usuário navega e define qual estádio visualizar as informações. Tem-se também um mapa da África do Sul com um cursor localizando geograficamente a posição da sede selecionada. Figura 4.1.4.1 contém a imagem da tela referente.

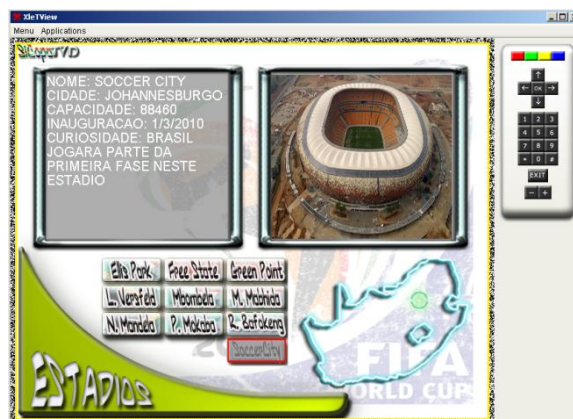


Figura 4.1.4.1 – Emulador xletview com SiCopaTVD carregado na tela de estádios

Atalhos:

←, →, ↑, ↓ Navega entre as sedes respeitando os limites do mapa de sedes.

4.1.5. Tela de figurinhas

Tela que exibe as figurinhas adquiridas. Não contém funções especiais do controle. Figura 4.1.5.1 contém a imagem da tela referente.



Figura 4.1.5.1 – Emulador xletview com SiCopaTVD carregado na tela de figurinhas.

4.1.6. Tela de sobre

Tela que exibe informações sobre o país sede e sobre a Copa. Não há navegação no controle. Figura 4.1.6.1 contém a imagem da tela referente.

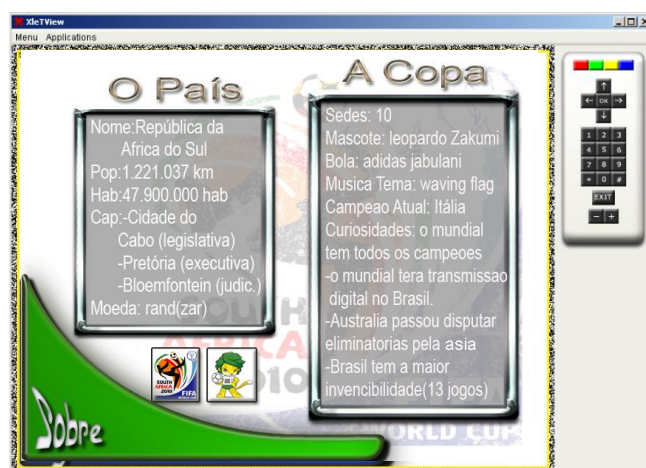


Figura 4.1.6.1 – Emulador xletview com SiCopaTVD carregado na tela sobre

4.1.7. Tela de tabela

Tela que contém informação sobre a tabela do campeonato. Nesta tela pode-se acessar a fase de grupos ou a fase eliminatória. Como o objeto carrossel não foi implementado para acesso dinâmico, a atualização dos placares também não é feita de forma dinâmica através de certo canal. Figura 4.1.7.1 contém a imagem da tela referente.



Figura 4.1.7.1 – Emulador xletview com SiCopaTVD carregado na tela tabela, com visualização da fase de grupos e eliminatórias(oitavas).

Atalhos:

- ←, → Troca o grupo quando ativo.
- ↑, ↓ Troca a fase eliminatória quando ativa.
- 0 alterna entre fases grupos e eliminatórias.

4.1.8. Tela de classificação

Tela e funcionalidades não implementadas.

4.1.9. Tela de compra

Tela e funcionalidades não implementadas.

4.1.10. Tela de usuário

Tela e funcionalidades não implementadas.

4.2. Banco de dados

O banco de dados não foi preenchido completamente. Estádios, seleções e copa estão com todas as informações completas. Agora a parte de jogadores, que tem mais de 600 jogadores, não foi preenchida totalmente. Desses apenas quatorze do Brasil, um da Alemanha, um da África do Sul e um da Alemanha foram mapeados. Algumas informações deveriam atualizar-se dinamicamente como é o caso dos placares e da parte das figurinhas. Como não foi tratado o objeto carrossel para tal finalidade está funciona a atualização está desabilitada para essa versão, logo para o arquivo de tabela a atualização é feita manualmente e direto no arquivo.

4.3. Figurinhas

A idéia do compartilhamento das figurinhas não está contida nesta versão, portando não tem como trocar com outros usuários qualquer tipo de figurinha. As figurinhas que são contidas nesta versão foram inseridas diretamente (manualmente) no local base do arquivo carrossel, não foram conquistadas perante a certo número de pontos, como foi proposto.

4.4. Moeda Lair e Usuário

O programa teria diversos usuários com uma quantidade de moedas cada um, porém devido à escassez do tempo tal funcionalidade não se encontra presente nesta versão. A parte de moedas não foi implementada e, por conseguinte nem a parte de usuários.

4.5. Requisitos funcionais

Foi implementado os requisitos funcionais RF1, RF2, RF3, RF4, RF5 sendo que o requisito RF3 foi desmembrado em dois sub-menus.



5. Conclusão

A tecnologia de aplicativos para TV Digital está apenas “engatinhando”, muitas linguagens ainda estão sendo testadas. O padrão Ginga a pouco foi estabelecido. Porém com tal padrão a vida dos desenvolvedores de software para o ambiente será facilitada. As dificuldades em dar progresso a uma tecnologia são enormes, isto com padrões bem definidos, agora imagine sem eles. Este foi o nosso caso, sem um padrão coerentemente definido e com pouco suporte. A linguagem adotada, JAVATV, prometia dominar o setor. Se pensássemos em outras vertentes, veríamos que tal promessa realmente poderia ocorrer, temos Java dominando por tudo que é situação. Porém com TV digital esse fator não ocorreu. Talvez seja o descaso da Sun no setor, não dando continuidade na evolução em sua api JavaTv, ou talvez o padrão Ginga que traz a opção de uma outra linguagem, além de Java, com um contexto um pouco diferente. Apesar de tudo, a escolha pela api para o desenvolvimento do projeto nos garantiu pelo menos os objetivos propostos. Apesar do projeto não conter todos os requisitos propostos, a linguagem se mostrou capaz de supri-los. A falta de documentação e suporte interferiu um pouco para que todos os requisitos fossem completados. O emulador usado para a visualização das xlets tem muitas barreiras de compatibilidade com a linguagem Java. Um fator decisivo que atrapalhou o projeto foi a necessidade de objetos carrossel para acessar arquivos io (esta é a única maneira). Até que se notasse tal situação, muito tempo em buscas e pesquisas pela solução foi desperdiçado. No entanto começar algo totalmente do zero e conseguir um resultado considerável é algo gratificante. Quando se fala em resultado considerável, queremos dizer, um programa agradável de usar, com imagens bem trabalhadas e que retorna as informações coerentemente a que foi proposto. Tal resultado com certeza teve base no conhecimento adquirido ao longo do curso, não no conteúdo das disciplinas cursadas em si, mas sim na evolução que cada uma proporcionou. O objetivo de aprender uma nova linguagem foi contemplado, e as barreiras desse novo ambiente foram inicialmente desmistificadas. A partir disto é criada uma grande oportunidade para expansão do conhecimento na área e até mesmo uma aplicação do conhecimento adquirido no ramo comercial. Portar o programa para a plataforma física (receptor) da TV digital não era o real objetivo, mas é interessante pensar neste tipo de teste, que ficará para oportunidades futuras, à medida que se tenha contato maior com o receptor. Por fim, fica também a esperança que o projeto possa contribuir para futuros projetos baseado nesta tecnologia, e que os erros aqui cometidos sejam soluções para os demais, pois a evolução de uma tecnologia depende de trocas de experiência.



6. Testes de aceitação

1 Teste de aceitação para caso de uso__01

Procedimento de teste

Identificação	CU01
Objetivo	Testar consulta de jogadores
Requisitos especiais	Estar na tela de jogadores
Requisitos verificados	RF01
Fluxo	No menu principal navegue até o sub-menu jogadores Escolha um time na tela de escolha de times Verifique as informações dos jogadores desejados

Caso de teste

Identificação	CT01	
Itens a testar	Consulta de jogadores	
Entrada	Campo	Valor
	Menu principal	Sub-menu jogadores
	Tela Escolha de Times	Brasil
	Jogador	Luis Fabiano
Saídas esperadas	Campo	Valor
	Informações	Robson Silva, 30 anos,...
	Informações rápidas	Robinho, sem figurinha
Ambiente	Emulador XletView	
Procedimento		
Dependências	Nenhuma	

7. Referências bibliográficas

- [1] www.ginga.org.br
- [2] <http://b4dtv.blogspot.com/2009/07/tutorial-debug-no-xletview-usando-o.html>
- [3] flagpedia.net
- [4] <http://www.fifa.com/worldcup/index.html>
- [5] <http://java.sun.com/javame/technology/javatv/index.jsp>
- [6] http://www.interactivetvweb.org/tutorials/mhp/file_access
- [7] http://www.teleco.com.br/tutoriais/tutorialtvd/pagina_3.asp
- [8] <http://www.dtv.org.br/>
- [9] <http://g1.globo.com/Noticias/Tecnologia/0,,MUL1403871-6174,00.html>
- [10] www.grupogingagoias.com.br/
- [11] <http://www.havi.org/>
- [12] <http://www.davic.org/>
- [13] <http://gingarn.wikidot.com/>
- [14] <http://lince.dc.ufscar.br/home>
- [15] <http://javafree.uol.com.br/topic-861397-orgdvbinternet-conexao-com-a-internet.html>
- [16] JavaTV™ API Technical Overview: The JavaTV API Whitepaper - Bart Calder, Jon Courtney, Bill Foote, Linda Kyrnitszke, David Rivas, Chihiro Saito, James Van Loo, Tao Ye
- [17] Ferramentas de Acessibilidade para TV Digital Interativa com Java™ - Marcos Vinícius Henke Arnoldo, Karen Selbach Borges
- [18] Utilização da api JavaTV para o desenvolvimento de aplicações para a TV digital interativa – Eli Candido Junior
- [19] Specification of the Home Audio/Video Interoperability (HAVi) Architecture, Chapter 8 – Level 2 User Interface – HAVi, Inc.



Cronograma

Atividades	ago/09	set/09	out/09	nov/09	dez/09	jan/10	fev/10	mar/10	abr/10	mai/10	jun/10	jul/10
Definição do tema do projeto final de	x	x										
Levantamento de dados			x	x								
Estudo da literatura aux.					x	x	x	x	x	x	x	
Levantamento dos requisitos e casos de uso				x	x	x						
Modelagem conceitual					x	x						
Projeto e implementação						x	x	x	x	x	x	
Testes e relatório										x	x	x