

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RAFAEL SILVEIRA MACHADO

**Análise e Implementação de Aplicação Web para
Programação Musical da Rádio da Universidade**

Monografia apresentada como requisito parcial para
a obtenção do grau de Bacharel em Engenharia de
Computação.

Orientador: Profa. Dra. Renata Galante

Porto Alegre
2014

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Engenharia de Computação: Prof. Marcelo Götz

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

Este trabalho consiste na análise e implementação de uma nova aplicação web para programação musical da Rádio da Universidade. O objetivo da ferramenta é fornecer uma solução que substitua os processos realizados de forma manual pelos programadores da emissora. Portanto, o software deve ser capaz de oferecer funcionalidades para cadastro e manutenção das obras do acervo musical além da criação de grades de programações e blocos musicais através de seleção de obras do catálogo.

Palavras-chave: Rádio da Universidade. Aplicação web. Programação musical.

Analysis and Implementation of a Web Application for "Rádio da Universidade" Music Schedule

ABSTRACT

This work is the analysis and implementation of a new web application for music schedule of "Rádio da Universidade". The purpose of the tool is to provide a solution to replace the processes performed manually by the station programmers. Therefore, the software must be able to offer features for registration and maintenance of works of music collection and building the playlist through selection of catalog works.

Keywords: "Rádio da Universidade". Web Application. Music schedule.

LISTA DE FIGURAS

<i>Figura 2.1 – Diagrama do modelo em cascata</i>	12
<i>Figura 2.2 – Diagrama do modelo em espiral</i>	13
<i>Figura 2.3 – Diagrama de uma arquitetura MVC</i>	15
<i>Figura 2.4 – Código que submete uma consulta ao banco com funções nativas do PHP</i>	16
<i>Figura 2.5 – Código que utiliza ORM para realizar uma consulta do banco</i>	16
<i>Figura 2.6 – Código de manipulação de dados utilizando Active Record</i>	17
<i>Figura 4.1 – Exemplo de ficha correspondente a uma obra</i>	26
<i>Figura 4.2 – Espelho da grade de programação de um segunda-feira</i>	28
<i>Figura 4.3 – Trecho de programação disponibilizada na Internet para os ouvintes</i>	30
<i>Figura 4.4 – Modelo ER das tabelas relativas ao acervo musical</i>	34
<i>Figura 4.5 – Modelo ER para armazenar as gravações</i>	34
<i>Figura 4.6 – Modelo ER da parte de programação</i>	35
<i>Figura 4.7 – Protótipo do cadastro de autores</i>	36
<i>Figura 4.8 – Protótipo do cadastro de uma obra</i>	37
<i>Figura 4.9 – Protótipo da tela de programação</i>	38
<i>Figura 4.10 – Protótipo da tela de programação de um bloco musical</i>	39
<i>Figura 4.11 – Tela principal da aplicação de programação</i>	41
<i>Figura 5.1 – Tela da versão anterior da ferramenta</i>	44
<i>Figura 5.2 – Tela de controle do EasyPHP</i>	46
<i>Figura 5.3 – Tela do módulo phpMyAdmin</i>	46
<i>Figura 5.4 – Modelo de dados do acervo musical</i>	48
<i>Figura 5.5 – Modelo de dados da programação</i>	48
<i>Figura 5.6 – Modelo de dados dos usuários da aplicação</i>	49
<i>Figura 5.7 – Trecho do código da classe “Author”</i>	50
<i>Figura 5.8 – Trecho do código do controlador “Author”</i>	51
<i>Figura 5.9 – Trecho de código de uma visão de cadastro de autores</i>	53
<i>Figura 5.10 – Trecho de código responsável por apresentar um menu</i>	56
<i>Figura 5.11 – Tela de login</i>	58
<i>Figura 5.12 – Layout do aplicação</i>	59
<i>Figura 5.13 – Alteração de cadastro Obras</i>	60
<i>Figura 5.14 – Tela de cadastro de autores</i>	61
<i>Figura 5.15 – Tela de cadastro de obras</i>	61
<i>Figura 5.16 – Tela de cadastro de classificação de obras</i>	62
<i>Figura 5.17 – Tela de períodos de programação</i>	63
<i>Figura 5.18 – Grade programação de um período</i>	64
<i>Figura 5.19 – Montagem dos blocos musicais</i>	64
<i>Figura 5.20 – Busca e seleção de obras</i>	65
<i>Figura 5.21 – Cadastro de gravações</i>	65

LISTA DE TABELAS

<i>Tabela 3.1 – Análise comparativa das ferramentas de programação musical.....</i>	<i>22</i>
<i>Tabela 4.1 – Trecho de uma grade de programação</i>	<i>27</i>
<i>Tabela 6.1 – Implementação do requisitos funcionais</i>	<i>66</i>
<i>Tabela 6.2 – Implementação dos requisitos não funcionais.....</i>	<i>66</i>

LISTA DE ABREVIATURAS E SIGLAS

AM	Amplitude Modulada
BPMN	<i>Business Process Model and Notation</i>
CD	<i>Compact Disc</i>
CPD	Centro de Processamento de Dados
KHz	Kilohertz
LP	<i>Long Play</i>
MP3	<i>MPEG-2 Audio Layer III</i>
MVC	Modelo-Visão-Controlé
PHP	<i>PHP: Hypertext Preprocessor</i> (antigamente era <i>Personal Home Page</i>)
SGBD	Sistema de Gerenciamento de Banco de Dados
TCC	Trabalho de Conclusão de Curso
UFRGS	Universidade Federal do Rio Grande do Sul
WAMP	Windows – Apache – MySQL – PHP

SUMÁRIO

RESUMO	3
ABSTRACT	4
LISTA DE FIGURAS	5
LISTA DE TABELAS	6
LISTA DE ABREVIATURAS E SIGLAS	7
SUMÁRIO	8
1 INTRODUÇÃO	9
2 CONCEITOS BÁSICOS	11
2.1 Processo	11
2.1.1 Modelo em cascata.....	11
2.1.2 Modelo em espiral.....	12
2.2 Framework	14
2.2.1 MVC.....	14
2.2.2 Mapeamento Objeto-Relacional.....	16
2.2.3 Caching.....	17
2.2.4 Considerações Finais.....	18
3 TRABALHOS RELACIONADOS	19
3.1 InfoDISC	19
3.2 Cadena Musical	19
3.3 Transmissão MAXXI – Módulo Programação Musical	20
3.4 Playlist Maker 2012	21
3.5 Análise Comparativa	21
4 TRABALHOS PRÉVIOS E LEVANTAMENTO DE REQUISITOS	23
4.1 Rádio da Universidade	23
4.1.1 Organização da Rádio.....	23
4.1.2 Departamento de Programação.....	24
4.2 Descrição do Trabalho do Érico	31
4.2.1 Análise dos Requisitos.....	31
4.2.2 Arquitetura.....	32
4.2.3 Projeto do Banco de Dados.....	33
4.2.4 Protótipos de Interface.....	35
4.2.5 Importação do Dados.....	39
4.2.6 Ferramenta.....	40
5 APLICAÇÃO WEB PARA PROGRAMAÇÃO MUSICAL	42
5.1 Análise de Requisitos	42
5.2 Revisão da Arquitetura	45
5.3 Revisão do Modelo de Dados	48
5.4 Implementação da Aplicação	49
5.5 Aplicação Web	57
5.5.1 Login.....	57
5.5.2 Organização da Aplicação.....	58
5.5.3 Telas de Cadastro do Acervo.....	59
5.5.4 Funcionalidade de Programação.....	62
6 RESULTADOS E TESTE DE USABILIDADE	66
7 CONCLUSÃO	69
REFERÊNCIAS	70
APÊNDICE A – PROCESSO DE PROGRAMAÇÃO MUSICAL	71
APÊNDICE B – SUBPROCESSO DE MONTAGEM DOS BLOCOS MUSICAIS	72

1 INTRODUÇÃO

Este trabalho descreve o desenvolvimento de uma aplicação responsável por gerenciar o acervo musical da Rádio da Universidade e auxiliar na programação da emissora. O sistema foi projetado e desenvolvido inicialmente por Érico Martellet Marcant como parte do seu Trabalho de Conclusão de Curso de Ciências da Computação em 2013.

A Rádio da Universidade¹ é uma emissora vinculada à Universidade Federal do Rio Grande do Sul (UFRGS) e tem o objetivo de irradiar cultura, lazer e informação qualificada. Atualmente, sua grade de horários é composta de programas jornalísticos, de entrevistas e de conteúdo cultural e blocos dedicados à radiodifusão de música erudita.

O Departamento de Programação é responsável por montar a programação musical da emissora. No entanto, esse trabalho é feito manualmente e exige, além de conhecimento técnico e cultural sobre música clássica, domínio sobre o acervo da rádio. A proposta desse trabalho consiste em estender e adicionar novas funcionalidades para uma aplicação capaz de gerenciar e catalogar as obras da Rádio da Universidade e oferecer uma ferramenta adequada para auxiliar a criação de blocos musicais. Além disso, a aplicação será desenvolvida utilizando o mesmo framework Yii adotado pelo CPD da UFRGS, abrindo caminho, portanto, para uma futura integração com os demais sistemas institucionais da universidade.

O desafio de desenvolver uma aplicação web envolve a utilização de muitos dos conceitos discutidos ao longo do curso de Engenharia de Computação. O primeiro obstáculo que surge ao iniciar o desenvolvimento é a identificação do problema e a forma como a ferramenta pode contribuir para auxiliar o processo. Para isso, a Engenharia de Software oferece uma porção de processos que podem ser empregados para otimização dos esforços tais como técnicas de análise de requisitos e análise estruturada. Na área de banco de dados, a adoção de um modelo de dados apropriado contribui para um software com mais qualidade. Por outro lado, a seleção da linguagem de programação e o estilo de programação adotado são aspectos essenciais de uma ferramenta, pois disso dependem a forma como a aplicação é disponibilizada e a sua eficiência. Já na fase de implementação e codificação, identificar e adotar padrões de projeto de software resulta em processo mais rápido e seguro, evitando erros muito comuns nesse tipo de projeto. Enfim, este trabalho emprega diversos conceitos de

¹ <http://www.ufrgs.br/radio>

computação para atingir o seu objetivo que é o de oferecer um software de qualidade à Rádio da Universidade.

O restante texto está organizando da seguinte forma. O Capítulo 2 apresenta os principais conceitos considerados essenciais para o entendimento do trabalho e que envolvem a disponibilização de uma aplicação na Internet, como a necessidade de utilização de um framework eficiente. O Capítulo 3 explora os trabalhos relacionados disponíveis no mercado. O Capítulo 4 apresenta a Rádio da Universidade e descreve o trabalho anterior, mostrando as principais funcionalidades da primeira versão da aplicação. O Capítulo 5 apresenta os novos requisitos e a sua implementação. No Capítulo 6 se discute os resultados do projeto e um teste de usabilidade é definido. O trabalho é finalizado com o Capítulo 7 que descreve as conclusões e trabalhos futuros.

2 CONCEITOS BÁSICOS

Este capítulo é dedicado a apresentar os principais conceitos básicos utilizados nesse projeto.

2.1 Processo

Aplicações web fazem parte de um grupo de softwares que se caracteriza por ser acessível através de um navegador conectado à Internet. No entanto, o desenvolvimento desse tipo de programa enfrenta os mesmos desafios dos demais softwares. Softwares são ferramentas computacionais que reúnem uma série de instruções destinadas a realizar tarefas que auxiliam os usuários. O processo de desenvolvimento de um software envolve muitos aspectos tais como planejamento, identificação do problema e escolha e implementação de uma solução. Com isso, a construção de um software torna-se complexo e com altas chances apresentar um resultado insatisfatório.

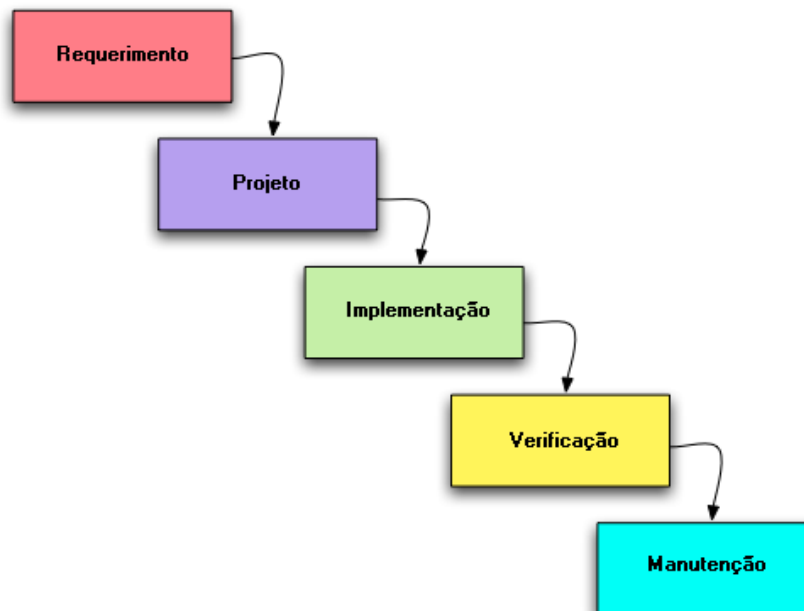
A Engenharia de Software é uma área da computação voltada a estudar soluções e processos de desenvolvimento de software capazes de mitigar os problemas mencionados anteriormente. Entre as metodologias disponíveis no mercado, este projeto utilizou técnicas dos modelos em cascata e em espiral para a realização da ferramenta proposta.

2.1.1 Modelo em cascata

O modelo em cascata é uma metodologia de desenvolvimento de software que organiza o processo em etapas e tem como característica principal estabelecer um fluxo de atividades sem retroceder a etapas anteriores. O modelo foi proposto por Royce (1970) como um conjunto de técnicas vigentes na época de sua publicação.

Existem variações do modelo mas, essencialmente, o processo é dividido em cinco etapas – levantamento de requisitos, projeto, implementação, verificação e manutenção. O fluxo de desenvolvimento é sequencial e avança por cada etapa somente quando as tarefas de cada fase são encerradas. A Figura 2.1 mostra as etapas do modelo em cascata e o fluxo do processo em uma única direção.

Figura 2.1 – Diagrama do modelo em cascata



Fonte: http://pt.wikipedia.org/wiki/Modelo_em_cascata

O modelo é de fácil aplicação e segue um caminho natural para o desenvolvedor. No entanto, o modelo tem muitos questionamentos como a falta de interação maior dos usuários com o processo e a ineficiência na identificação dos requisitos devido a incertezas no início do projeto.

2.1.2 Modelo em espiral

O modelo em espiral foi proposto por Boehm (1988) que integrou diferentes modelos conhecidos à época. O modelo organiza o desenvolvimento em processos em quatro fases iterativas que se sucedem até a obtenção do software final. As fases são: planejamento, análise de riscos, execução e verificação. Cada ciclo engloba uma parte do projeto, podendo ser utilizado até mesmo outro modelo como o em cascata.

A Figura 2.2 mostra um diagrama com as quatro fases e o ciclo de desenvolvimento percorrendo cada uma delas.

Figura 2.2 – Diagrama do modelo em espiral



Fonte: <http://www.cin.ufpe.br/~cadcn/>

O modelo em espiral apresenta algumas vantagens em relação ao em cascata, pois prevê interações com os usuários em vários pontos do projeto. Além disso, versões parciais do software podem surgir durante o desenvolvimento o que aumenta a segurança do usuário em relação ao projeto. Entre as desvantagens, é importante destacar que dependendo do tamanho do projeto, o controle sobre o fluxo e a abordagem é vital e pode ser uma fonte de problemas.

Em relação a este trabalho, o modelo em espiral se mostra como a metodologia mais adequada a ser implantada. Cada ciclo será destinado a executar tarefas inspiradas no modelo em cascata, tais como análise de requisitos, projetos e implementação. Cada ação será planejada e os riscos avaliados. Após a execução, os resultados serão verificados com a participação dos usuários. Assim, será mais fácil detectar possíveis problemas no projeto durante o seu desenvolvimento.

2.2 Framework

A codificação de um programa é um desafio complexo que envolve a resolução de milhares de pequenos problemas que não fazem parte dos requisitos iniciais do produto. Tarefas como acesso a banco, construção de telas e controle de acesso podem consumir inúmeras horas de trabalho e, mesmo assim, podem não funcionar adequadamente. O projeto de software prevê uma porção de técnicas que podem ser utilizadas para minimizar esses problemas. Uma das mais conhecidas é o reaproveitamento de códigos que proporcionam dois tipos de ganhos. Além de acelerar a implementação, a utilização de trechos de códigos já testados diminui a taxa de erros de um sistema.

Um framework se baseia nesses conceitos para oferecer um conjunto de soluções para auxiliar o desenvolvimento de um software. Há inúmeros padrões de desenvolvimento comuns a muitas aplicações. O que o framework faz é oferecer soluções já implementadas para que o programador volte sua atenção exclusivamente para as tarefas do escopo original do projeto.

Este projeto selecionou o Yii Framework para implementar a aplicação de programação musical. O Yii é um framework livre, de alta performance, voltado a aplicações web em PHP. O Yii possui muitas das características comuns a outros frameworks web. As subseções a seguir apresentarão as principais características e componentes utilizados nesse projeto.

2.2.1 MVC

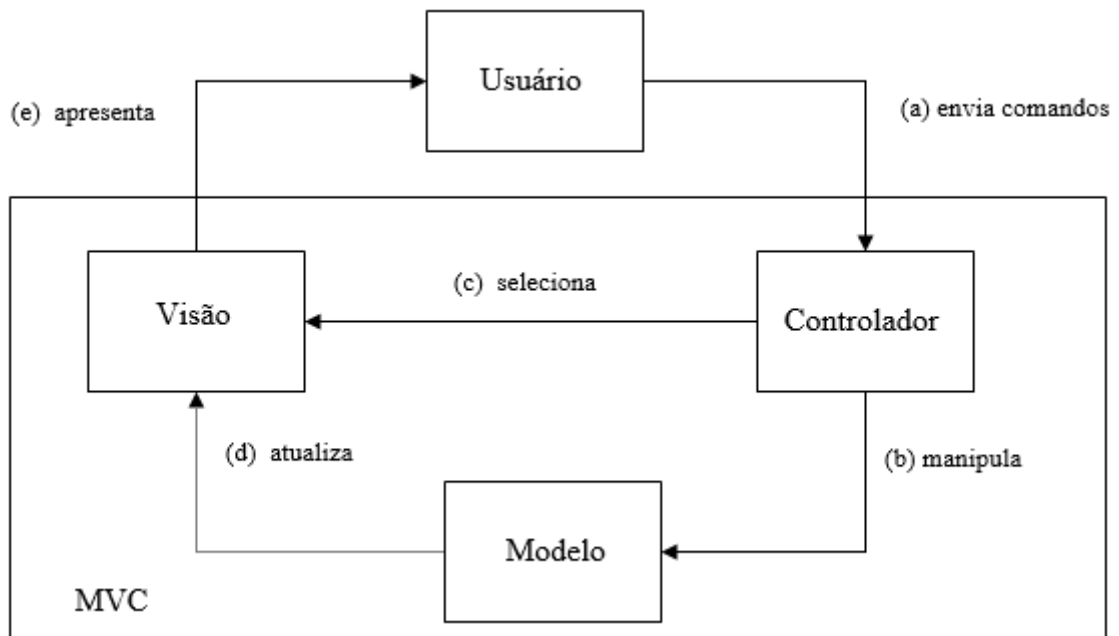
A organização modelo-visão-controlador (MVC) é um dos modelos de arquitetura de software mais populares em aplicações para Web. A maioria dos frameworks estão organizados de modo a representar esse modelo.

MVC (BURBECK, 1987) consiste em separar em camadas a interação dos usuários com o software. O modelo (*model*) é a representação dos dados, regras de negócios, lógica e funções da aplicação. A visão (*view*) é qualquer saída de dados para o usuário, como páginas HTML, imagens e arquivos entre outros. Já o controlador (*controller*) faz a mediação das entradas ao convertê-las em comandos para os modelos e visões.

A Figura 2.3 representa o fluxo de informações em uma arquitetura típica em MVC. O usuário acessa o controlador enviando comandos do que deseja fazer. O controlador, por sua vez, interpreta esses comandos e manipula o conjunto de dados, regras de negócios e funções

preestabelecidos. Após, é selecionado um formato de saída dentro da coleção de visões. Essas saídas são alimentadas indiretamente pelo modelo. Logo, quando um modelo é manipulado a saída que é a representação desse item sofre alteração também. Por fim, os dados de saída são apresentados ao usuário.

Figura 2.3 – Diagrama de uma arquitetura MVC



Fonte: Elaborada pelo autor

Embora criado e desenvolvido para qualquer tipo de software e linguagem de programação, são nas aplicações web que o padrão MVC encontrou força e é largamente utilizado. A maioria dos frameworks utiliza os conceitos fundamentais da arquitetura em camada com pequenas variações. Por ser uma plataforma essencialmente cliente-servidor, o usuário interage com o programa através de uma página da Internet em um navegador. Nessa página, é possível enviar comandos ao servidor através de requisições HTTP do tipo GET e POST. Essas requisições são recebidas pelo servidor e processadas pelo controlador. Após manipular os modelos, uma visão é selecionada e enviada ao cliente para ser observada pelo usuário em seu navegador novamente.

O principal benefício do MVC é a separação de conceitos que ocorre na arquitetura. Cada camada é responsável por uma tarefa bem delimitada e isso permite que o desenvolvimento possa ser feito por equipes diferentes. Com isso, é possível substituir elementos dentro de um projeto sem precisar refatorar todo o código. Um exemplo ocorre quando uma determinada aplicação trabalha com um banco de dados específico. Ao abstrair a

interação com essa base utilizando um conjunto de modelos responsáveis pela comunicação com o controlador, é possível migrar o banco para outra base alterando apenas os modelos correspondentes. Há ainda frameworks que fazem com que essa tarefa torne-se ainda mais simples ao exigir apenas algumas alterações no arquivo de configuração do projeto.

Outra vantagem que surge com a utilização do padrão é a reutilização de códigos. Uma vez que cada camada trabalha de forma independente e atendem a determinadas tarefas, é mais fácil encontrar trechos de códigos de outros projetos que podem ser reaproveitados e compartilhados entre si. Isso faz com o tempo de desenvolvimento dos projetos caia significativamente.

2.2.2 Mapeamento Objeto-Relacional

O mapeamento objeto-relacional (ORM, do inglês *object-relational mapping*) é uma técnica de desenvolvimento de software que utiliza a orientação a objetos para abstração de tabelas do banco de dados em classes e os registros em objetos (FOWLER, 2003).

Ao utilizar uma ferramenta de ORM, o programador não necessita elaborar comandos SQL para acessar registros armazenados em tabelas de banco de dados relacionais. Nesses casos, basta instanciar a classe correspondente e acessar as propriedades necessárias. As Figura 2.4 e Figura 2.5 ilustram trechos de códigos em PHP que mostram a diferença das duas formas de se obter o mesmo registro.

Figura 2.4 – Código que submete uma consulta ao banco com funções nativas do PHP

```
1 <?php
2
3 string $comando='select nome from Colaboradores where id=5';
4 resource $rs=mssql_query($comando, $conexao);
5 string $nome=mssql_result($rs,0,'nome');
6
```

Fonte: Elaborada pelo autor

Figura 2.5 – Código que utiliza ORM para realizar uma consulta do banco

```
1 <?php
2
3 object $colaborador= new Colaborador(5);
4 string $nome= $colaborador->nome;
5
```

Fonte: Elaborada pelo autor

A utilização da técnica ORM está em acordo com a programação em camadas MVC. Já que os códigos das classes podem ser desenvolvidos de forma independente do restante do projeto e formando o conjunto de modelos disponíveis para serem manipulados.

Há diferentes ferramentas que implementam ORM com algumas variações. A mais comum e presente nos frameworks PHP é o *Active Record* (AR). Esse padrão engenharia de foi criado em 2003 por Martin Fowler e consiste em encapsular as tabelas do banco de dados em classes que implementam interfaces que possuem métodos de inclusão, alteração e deleção de registros. As interfaces também fornecessem métodos para acesso aos atributos da tabela como propriedades do objeto instanciado. A Figura 2.6 mostra um exemplo de implementação utilizando *Active Record*.

Figura 2.6 – Código de manipulação de dados utilizando *Active Record*

```
1 <?php
2
3 object $colaborador= new Colaborador();
4 $colaborador->nome= 'Rafael Machado';
5 $colaborador->data_nascimento='25/03/1982';
6 $colaborador->save();
7
```

Fonte: Elaborada pelo autor

2.2.3 Caching

Caching é uma técnica de armazenamento temporário de dados que visa diminuir a latência na busca de informações (HENNESSY; PATTERSON, 2012). Dentro da computação, é largamente utilizada em várias soluções tanto em hardwares como em softwares.

A linguagem PHP, quando utilizada em aplicações web, não dispõem de mecanismos de armazenamento de dados entre uma requisição de usuário e outra. Um sistema de loja virtual, por exemplo, precisa apresentar o catálogo de produtos constantemente quando o cliente navega pelas páginas. Isso faz com que a aplicação necessite consultar o banco de dados inúmeras vezes. É interessante, nesses casos, utilizar *caching* para armazenar o resultado dessas consultas a fim de otimizar o acesso à base de dados.

Os principais frameworks oferecem soluções de *caching* para armazenar estruturas dos mais diversos tipos. A utilização mais comum é para dados de consulta a banco. No entanto, é possível armazenar objetos, trechos de códigos e até mesmo páginas HTML.

As implementações de *caching* também variam quanto a forma de armazenamento. O framework pode salvar os dados em variáveis de sessão, arquivos, memória e em estruturas específicas para isso em um banco de dados.

2.2.4 Considerações Finais

Ao implementar uma aplicação web com o Yii, o desenvolvedor dificilmente deixará de utilizar a arquitetura MVC. O framework convencionou a separação de trechos de códigos em três camadas. É até possível burlar essa organização, e escrever um conjunto de códigos estruturados de uma forma diferente, no entanto, há uma série de restrições que o desenvolvedor teria que contornar para que esse sistema funcionasse.

Portanto, ao selecionar um framework como o Yii, o projetista está também optando por utilizar uma estrutura MVC e todos os benefícios que surgem com esse padrão.

Já durante a implementação da ferramenta, é preciso tomar outras decisões em relação à escolha dos componentes a serem utilizados. Neste trabalho, optou-se pelo uso do *Active Record* para manipular o banco de dados. As classes produzidas com o padrão permitem abstrair os registros do banco de dados em objetos manipuláveis. Com essa técnica, economiza-se tempo de desenvolvimento ao não precisar criar longas declarações de consultas SQL. Além disso, o código passa a ser mais simples e de fácil manutenção.

A utilização de *caching* foi outra decisão tomada durante a implementação. Uma aplicação complexa como a deste trabalho necessita fazer inúmeras consultas ao banco de dados, o que pode tornar o sistema lento na visão do usuário. No projeto, foi utilizado *caching* para manter modelos de dados, informações sobre a identidade do usuário e consultas a obras. Além disso, optou-se por armazenar os dados de *caching* em arquivos gerenciados pelo Yii.

3 TRABALHOS RELACIONADOS

O projeto anterior apresentou uma análise comparativa de sistemas de apoio emissoras de rádio disponíveis no mercado. Este capítulo é dedicado a retomar e expandir essa análise. Os três softwares foram revisitados além de uma quarta ferramenta incluída na avaliação.

3.1 InfoDISC

O InfoDISC² é um sistema destinado a emissoras de rádio que possibilita a criação de estruturas para a geração automática de programação. O software é desenvolvido pela Informa. A ferramenta oferece a possibilidade de montar uma programação manual e a automação desse processo através da criação de regras de reprodução. É possível elaborar regras respeitando estilos, intérpretes e ocorrências por faixa de horários.

O sistema oferece as seguintes funcionalidades:

- Criação de perfis musicais para programação;
- Listas de músicas por estilo com destaque para as preferidas pelo público;
- Cadastro de vinhetas;
- Ajustes na programação automáticas;
- Organização do acervo musical;
- Acesso ao áudio de qualquer ponto do sistema;
- Geração de roteiro musical;
- Geração de diversos relatórios como estrutura dos programas, ECAD, vinhetas, rankings, mídias e informações patrimoniais.

3.2 Cadena Musical

Cadena Musical³ é um módulo de gerenciamento musical de emissoras. O sistema é desenvolvido pela Cadena Sistemas. A ferramenta permite controlar o acervo e criar listas de programação de forma manual ou automatizada.

² <http://informabr.com/infodisc>

³ <http://www.cadena.com.br/brazilian/solucoes/cadena-musical>

As seguintes funcionalidades são oferecidas:

- Cadastro de CDs, MDs e outras mídias;
- Cadastro detalhado de cada música;
- Listas de trabalhos onde é possível informar quantas vezes cada música deve ser executada por dia e o sistema é capaz de distribuí-las homoganeamente;
- Relatório mensal para envio ao ECAD;
- Criação de modelos de programação com blocos musicais, comerciais, vinhetas, participações de ouvintes entre outros;
- Cadastro de vinhetas, abertura e encerramentos de programas;
- Programação automática de acordo com modelos definidos;
- Possibilidade de ajustes na programação;
- Integração com os demais sistemas da empresa voltados ao mercado.

A empresa oferece outras ferramentas voltadas a emissoras de rádio e TV tais como Cadena Automação, Cadena Comercial, Cadena Ouvintes, Cadena Financeiro, Cadena Jornalismo, Cadena Censura e Cadena Locutor.

3.3 Transmissão MAXXI – Módulo Programação Musical

Transmissão MAXXI⁴ é um sistema desenvolvido pela ACCESS voltado a emissoras de rádio. Possui um modelo chamado Programação Musical especializado em gerenciamento e elaboração de grades de programação musical. O módulo oferece as seguintes funcionalidades:

- Cadastro de emissoras permitindo gerar diferentes grades de programação;
- Cadastro de mídias físicas e digitais;
- Cadastro de música por categoria, gênero, tipo de música, ritmo intérpretes e demais características de uma música;
- Controle de repetição;
- Controle de similaridade de músicas;

⁴ <http://www.accessweb.com.br/Sistema-de-Gestao-Esportiva/modulo-discotecario.html>

- Programação manual da programação onde são oferecidos diversos controles para auxiliar o programador na tarefa;
- Simulação musical onde é possível gerar várias grades de programação conforme regras estabelecidas;
- Diversos relatórios para ECAD, controle de músicas e de programação.

3.4 Playlist Maker 2012

Playlist Maker 2012⁵ é um software da Playlist Software Solutions voltado a produção de programação e organização do acervo de emissoras de rádio. A ferramenta permite a geração de grades de programação de forma rápida e eficiente.

As principais funcionalidades do software são as seguintes:

- Interface gráfica voltada a facilitar a execução de todas as tarefas;
- Geração simplificada de *playlists*;
- Gerenciamento de acervo;
- Pesquisa ao acervo permitindo localizar qualquer música através dos seus atributos cadastrados;
- Bloqueio de conteúdo para não repetir a programação;
- Sistemas de mensagens que auxiliam o programador;
- Acesso ao áudio de forma simplificada e em qualquer parte do sistema;
- Sugestão musical;
- Gráficos estatísticos da programação;
- Sistema multiusuário podendo ser acessado de forma simultânea;
- Possibilidade de acessar o sistema de forma remota;

3.5 Análise Comparativa

Através da análise das ferramentas da seção anterior, é possível perceber que muitas das funcionalidades são parecidas entre si. No entanto, nenhuma delas apresentou funcionalidades dirigidas a execução de música erudita como é o caso da Rádio da

⁵ http://www.playlist.com.br/en/prod_playlistmaker2012.asp

Universidade. Para melhor avaliação a Tabela 3.1 foi montada evidenciando as principais funcionalidades e a forma como elas estão disponíveis em cada software.

Tabela 3.1 – Análise comparativa das ferramentas de programação musical

	<i>InfoDisc</i>	<i>Cadena Musical</i>	<i>Transmissão MAXXI</i>	<i>Playlist Maker 2012</i>
Controle do acervo	Sim e controle de mídias	Sim e controle de mídias	Sim e controle de mídias	Sim
Classificação das músicas	Por estilos, intérpretes e ocorrências	Sim mas não informado quais tipos de classificações	Categoria, gênero, tipo, ritmo, intérpretes entre outras	Sim mas não informado quais tipos de classificações
Controle da grade de programação	Sim	Sim	Sim	Sim
Programação musical manual	Sim	Sim	Sim	Sim
Programação musical automática	Sim, baseado em estruturas definidas por estilos e demais classificações	Sim, de acordo com modelos definidos	Simulação musical baseadas e regras	Não
Relatórios	ECAD, vinhetas, rankings, mídias e informações patrimoniais	ECAD	ECAD, controle de músicas	Não
Diferencial		Integração com demais ferramentas da empresa		Interface gráfica diferenciada

Fonte: Elaborada pelo autor

4 TRABALHOS PRÉVIOS E LEVANTAMENTO DE REQUISITOS

Este capítulo tem o objetivo de apresentar o trabalho prévio realizado pelo Érico em 2013. A primeira seção descreve a Rádio de Universidade, sua estrutura e, em especial, o Departamento de Programação, responsável pelos processos de manutenção do acervo musical da emissora e a programação da grade de horários.

A seguir o trabalho anterior é apresentado. O entendimento dos requisitos e da primeira versão ferramenta são essenciais para o desenvolvimento deste projeto.

4.1 Rádio da Universidade

Esta seção é dedicada a Rádio da Universidade e é dividida em duas partes, mostrando como a emissora se organiza e a forma como os processos de programação musical é feito pelo departamento responsável.

4.1.1 Organização da Rádio

A Rádio da Universidade é uma emissora universitária pioneira começando sua operação em janeiro de 1951. Inicialmente, a Rádio estava autorizada a transmitir apenas conteúdos educacionais, palestras e informações do observatório astronômico da Universidade. No entanto, em dezembro de 1953, a emissora foi retirada do ar pois estava transmitindo músicas em sua grade, o que não lhe era permitido. Após nova solicitação ao Governo Federal, a Rádio obtém sua autorização definitiva de funcionamento e volta a operar em novembro de 1957 na frequência de 1080 kHz, ocupando essa faixa até os dias atuais.

Em 1960, a Rádio foi transferida para o local da atual sede da emissora no Campus Central da UFRGS onde antigamente funcionava a Seção de Meteorologia do Observatório Astronômico da Escola de Engenharia.

Já em 1990, a Rádio se moderniza e passa a ser a primeira emissora de Porto Alegre a transmitir conteúdo através armazenado digitalmente em *Compact Discs* (CD). Essa modernização, prosseguiu nos anos seguintes com a aquisição de novos equipamentos tais como mesas de áudio, reprodutores de CD e gravadores.

Em 1997 foi lançada a página na Internet da Rádio, e, um ano depois, passou a transmitir pela rede a sua programação através de *streaming* de áudio.

Atualmente a emissora está organizada em departamentos (Administrativo, Jornalismo, Programação, Operações Técnicas/Estúdios e Operações Técnicas/Transmissores) sob a direção do Professor André Prytoluk.

4.1.2 Departamento de Programação

O Departamento de Programação é responsável por manter e gerenciar o acervo musical e elaborar a grade de programação da emissora. Além disso, também seleciona as obras eruditas a serem executadas nos blocos dedicados à música. Esse procedimento é feito manualmente e sofreu pouca variação no passar dos anos.

O departamento é chefiado pelo Jornalista Roberto Busato, no local ainda trabalham os funcionários Manoela Faria Nogueira e Juliano Dupont. Foram realizadas diversas entrevistas com o jornalista Busato durante a realização do trabalho do Érico. Já no projeto atual, as reuniões tiveram a participação da Manoela e do Juliano que começaram a trabalhar em 2014. Nessas reuniões foram detalhadas as atividades desenvolvidas no departamento. Essa compreensão é essencial para a realização do projeto, pois o aplicativo tem o objetivo informatizar esses processos.

Como todas as outras atividades do departamento, a manutenção do acervo é feita de forma manual desde que o processo foi criado. Antigamente, o catálogo era composto por um conjunto de discos de vinis (LP). Com a atualização técnica nos anos 90, a emissora passou a adotar exclusivamente conteúdo musical no formato digital e armazenado em CDs. Atualmente, o acervo é composto por mais de 4000 CDs organizando em uma estante na sala de programação.

Cada obra catalogada possui duas e m papel que são utilizadas no processo de criação de um bloco musical. Essas fichas possuem as informações necessárias para localizar a obra na estante, como o número dos CD e das faixas necessárias para a reprodução. É importante salientar que uma obra pode se estender por mais de uma faixa ou CD. Há, por exemplo, a “Sonata N° 3, Opus 5 em Fá Menor” de Johannes Brahms, que se divide em cinco movimentos, cada um armazenado em uma faixa diferente no CD. Outro bom exemplo é a ópera “A Força do Destino” de Verdi que se estende por três diferentes CDs. Por outro lado, um determinado disco pode conter mais uma obra. Portanto, cada ficha contém uma lista de CDs e faixas necessários para a localização e reprodução da obra.

Além dos dados de localização física do CD, as fichas contêm o título, autor, intérprete, movimentos e a duração da obra. Essas informações são utilizadas tanto na hora da

montagem do bloco musical como pelo locutor da emissora. O locutor recebe uma cópia da ficha para poder anunciar e a obra que será transmitida. Nessa fala, são apresentados o título, autor, intérprete e divisões quando for o caso. Após a execução, o locutor volta a comentar sobre a obra por um breve período, esta fala é denominada desanúncio. A outra ficha é entregue ao operador técnico da emissora. Ele é responsável por colocar no ar a obra e, para isso, precisa conhecer o número do CD e as faixas que necessitam ser reproduzidas, além da duração do áudio para maior controle da transmissão.

Após a transmissão de um bloco musical, as fichas retornam ao departamento de programação e são atualizadas em um espaço no verso da folha com a data da execução. Essa informação é necessária para que o programador não selecione a obra novamente em curto espaço de tempo e dê prioridade a outras.

Desde 2008, as fichas são impressas a partir de um arquivo digital no formato *Microsoft Word*. O arquivo fica armazenado localmente em um computador da sala de programação e, se necessário, é reimpresso para substituir as folhas que possam estar extraviadas ou em mau estado. A Figura 4.1 ilustra o exemplo de ficha correspondente a uma obra.

Figura 4.1 – Exemplo de ficha correspondente a uma obra

OPERADOR		LOCUTOR
NÚMERO: CD - 21		TÍTULO: CONCERTO PARA BANDONÉON E ORQUESTRA
CD	FXS: 1, 2, 3	
CD	FXS:	
CD	FXS:	
DURAÇÃO: 20' 50"		AUTOR: ASTOR PIAZZOLLA
OBS.: x 751		
INTÉRPRETES: ASTOR PIAZZOLLA, BANDONÉON E ORQUESTRA DE ST. LUKAS, SOB A DIREÇÃO DE LALO SCHIFRIN.		
MOVIMENTOS: ALLEGRO MARCATO MODERATO PRESTO		

Fonte: MARCANT, 2013

Outra tarefa importante do departamento de programação é a elaboração da grade de programação da emissora. Essa atividade é dividida em duas partes. Na primeira é criado um espelho da programação contendo horários de início e fim dos programas e blocos musicais que irão ao ar. Na segunda etapa, é feita a seleção das obras do acervo que farão parte de cada bloco. A Tabela 4.1 mostra a programação de dois dias da emissora.

Tabela 4.1 – Trecho de uma grade de programação

<i>Horários</i>	<i>Segunda</i>	<i>Terça</i>
07:00 às 09:00	Música	Música
09:00 às 09:10	Jornalismo 1080	Jornalismo 1080
09:10 às 10:00	Música	Música
10:00 às 10:10	Jornal da UFRGS	Jornal da UFRGS
10:10 às 10:15	Toque de Arte	Toque de Arte
10:15 às 13:00	Música	Música
13:00 às 13:30	Fronteiras da Ciência	ADUFRGS no Ar
13:30 às 14:00	Música	Música
14:00 às 14:15	UFRGS Entrevista	UFRGS Entrevista
14:15 às 16:00	Música	Música
16:00 às 16:05	Literatura	Literatura
16:05 às 17:00	Música	Música
17:00 às 17:10	Jornal da UFRGS 2ª Edição	Jornal da UFRGS 2ª Edição
17:10 às 18:00	Música	Música
18:00 às 18:05	Jornalismo 1080	Jornalismo 1080
18:05 às 19:00	Universidade Revista	Universidade Revista
19:00 às 20:00	A Voz do Brasil	A Voz do Brasil
20:00 às 20:30	Tangos em la Noche	Tangos em la Noche
20:30 às 00:00	Música	Música
00:00 às 00:03	Boletim Astronômico	Boletim Astronômico
00:03 às 07:00	Música	Música

Fonte: <http://www.ufrgs.br/radio>

A grade de programação é feita com antecedência de no mínimo uma semana. Para montar a grade, o programador baseia-se em um conjunto de programas acordados previamente com a direção da emissora, formando o que é denominado espelho da grade de programação. Eventualmente, algum programa é alterado para atender a eventos sazonais (como a Feira do Livro), boletins extraordinários ou cancelamento de algum programa. Quando necessário, o programador pode, então, alterar as durações dos blocos musicais de forma a cobrir alguma faixa de horário.

Nos espaços dedicados à execução de obras, é possível adicionar pequenas chamadas gravadas anteriormente. Essas chamadas servem para anunciar outros programas de dentro da grade. A Figura 4.2 mostra um exemplo de espelho da grade. É possível notar a indicação das chamadas através da utilização do rótulo CH associado a um número que denota qual gravação deverá ser executada.

Figura 4.2 – Espelho da grade de programação de um segunda-feira

RÁDIO DA UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Programação para segunda-feira,

07.00 -

CH - 001/02
CH - 020/01

09.00 - JORNALISMO 1080

09.10 -

CH - 05
CH - 07

10.00 - JORNAL DA UFRGS (1ª edição)

CH - 020/06

10.10 - TOQUE DE ARTE

10.15 -

CH - 22

13.00 - FRONTEIRAS DA CIÊNCIA

13.30 - CH - 08

14.00 - UFRGS ENTREVISTA

14.15 -

16.00 - LITERATURA

16.05 - CH - 020/02

CH - 13

17.00 - JORNAL DA UFRGS (2ª edição)

17.10 - CH - 001/01

CH - 26

18.00 - JORNALISMO 1080

18.05 - UNIVERSIDADE REVISTA

19.00 - A VOZ DO BRASIL

20.00 - TANGOS EN LA NOCHE

20.30 - CH - 020/03

24.00 - BOLETIM ASTRONÔMICO

PROGRAMAÇÃO ESPECIAL NOTURNA

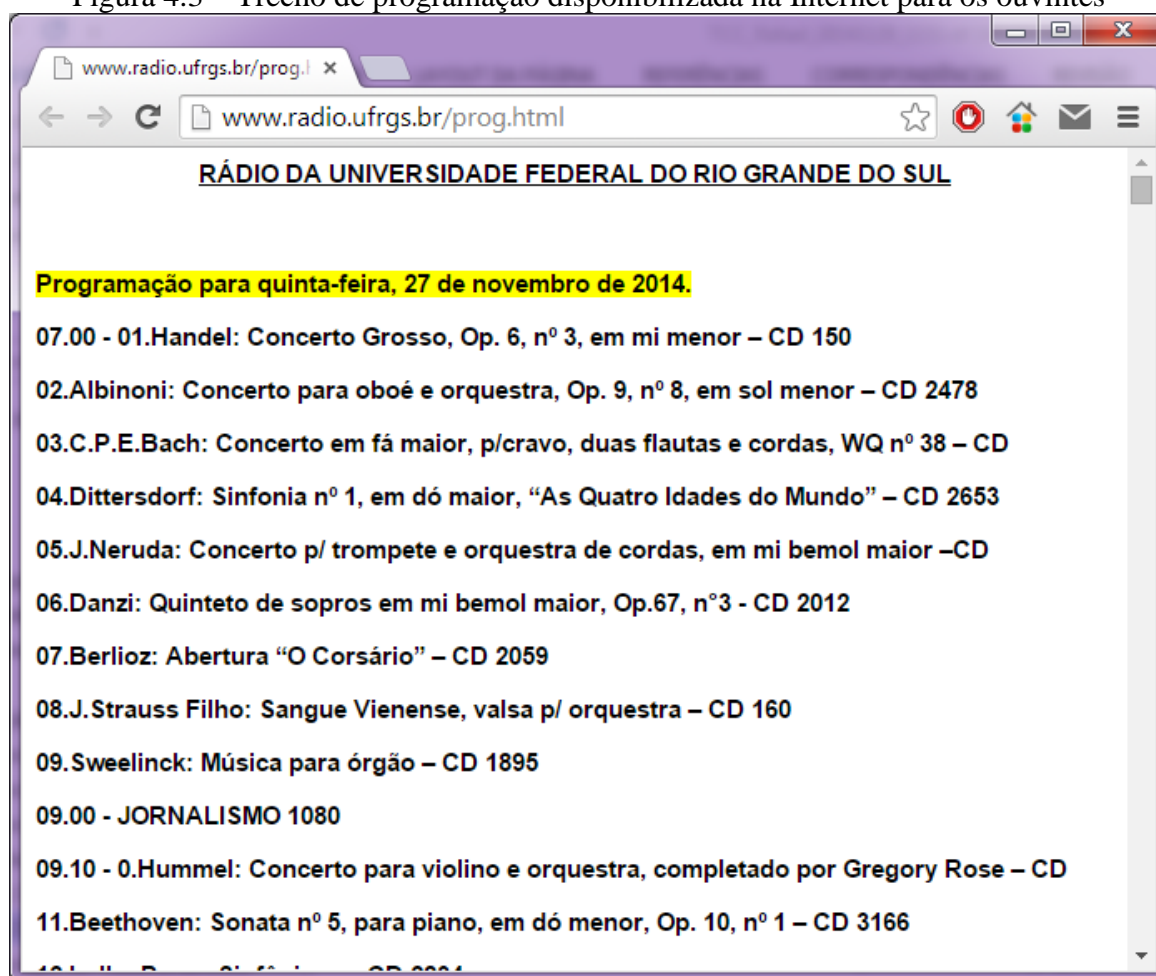
Fonte: MARCANT, 2013

Com a definição do espelho, sabe-se o tempo disponível para cada bloco musical. Nesse ponto é iniciada a segunda parte do processo de programação, que é a definição das obras que serão veiculadas nas janelas disponíveis. Os critérios de seleção das obras são subjetivos, e cabe, ao programador, conhecer o acervo musical da emissora e utilizar do seu conhecimento técnico para realizar a tarefa. Há algumas recomendações a serem seguidas e a principal delas é manutenção de equilíbrio na seleção das obras. Isso significa evitar repeti-las em curto um espaço de tempo. Para atender à orientação, o programador toma como base a programação realizada de 80 a 90 dias antes, reorganizando em uma nova sequência e fazendo alterações pontuais a seu critério. Ao fazer a substituição, as datas das últimas execuções presente no verso das fichas é levada em consideração nesse momento.

O controle de tempo dos blocos é feito de forma manual com o auxílio de uma calculadora. O programador precisa considerar a duração de cada obra ao selecioná-la. Também leva em consideração os tempos de anúncio e desanúncio de cada obra que não são fixos e podem sofrer variações na ordem de alguns segundos a mais de um minuto. A presença de chamadas, spots e vinhetas também devem ser avaliados. No fim, o somatório dos tempos deve se aproximar do tamanho do bloco, mas não pode ultrapassá-lo de forma alguma, sob risco de a execução não seja completada ou ainda que atrase a programação da rádio. Nos casos de sobra de tempo, algumas obras marcadas como “Encontro de Horários” são utilizadas para completar o bloco. Essas peças são curtas e permitem flexibilizar a grade de programação.

No fim, a grade completa, incluindo a lista de obras que serão transmitidas, é compilada em um documento único e disponibilizada na página da Internet da emissora para que os ouvintes possam identificar as obras e os autores antes de irem ao ar (ver Figura 4.3). Como a programação é feita de forma antecipada, normalmente o documento publicado contém a programação de sete dias consecutivos.

Figura 4.3 – Trecho de programação disponibilizada na Internet para os ouvintes



Fonte: <http://www.ufrgs.br/radio>

Para o dia da programação, todos os CDs e uma versão das fichas de cada obra são entregues ao operador da rádio. Ele é responsável por reproduzir a programação na ordem correta, controlar o tempo de execução e respeitar as pausas de anúncio e desanúncio das obras. O locutor recebe a outra versão das fichas para que possa ler o título, autor, intérpretes e divisões das obras. Essas operações são feitas ao vivo no horário entre 7 e 19 horas. Fora desse período, as apresentações são gravadas no formato MP3 e transmitidas nos horários corretos.

Durante a transmissão podem ocorrer por problemas técnicos e o operador se vê obrigado a alterar a lista das obras. Quando isso ocorre, o programador é informado e atualiza as informações de reprodução. Após o dia da programação, as fichas e o CDs voltam para o Departamento de Programação e o registro de execução da obra é adicionado no verso das fichas. Só após isso, a programação é dada como encerrada.

4.2 Descrição do Trabalho do Érico

A proposta deste trabalho é prosseguir com o projeto original desenvolvido pelo Érico Martellet Marcant em seu Trabalho de Conclusão de Curso de Ciências da Computação. Portanto, é necessário dedicar um espaço para relatar as atividades por ele. Para desenvolver a ferramenta, Érico adotou uma versão simplificada da metodologia tradicional em cascata (ROYCE, 1970). O processo foi dividido nas seguintes etapas: levantamento de requisitos, definições de arquitetura, modelagem de dados, elaboração de protótipos não-funcionais, importação de dados e desenvolvimento.

O levantamento de requisitos foi feito através de uma série de entrevistas com os servidores da emissora a seguir: Prof. André Prytoluk, diretor da rádio, Paulo Busato e Ivana Caselli do Departamento de Programação e Prof. Luiz Sperotto, diretor do Departamento Técnico. Nessas reuniões foram detalhados a estrutura da rádio, os processos referentes à manutenção do acervo e programação musical e os requisitos que serviriam como escopo inicial da ferramenta.

Nas fases seguintes, foram definidos a arquitetura do projeto, o modelo de dados inicial. Alguns protótipos de interface também foram projetados para validações com os usuários.

Foi dedicada atenção especial à importação dos dados do acervo da emissora. A utilização de dados reais da emissora foi importante para detectar problemas durante o desenvolvimento da ferramenta.

Por fim, foi feita a codificação do sistema. Nesta etapa, ocorreram reuniões com os usuários e algumas modificações foram feitas no projeto.

4.2.1 Análise dos Requisitos

Através de reuniões com os usuários foi possível chegar a uma série de requisitos para a ferramenta. O principal problema apontado é a não informatização dos processos de manutenção do acervo e programação musical da emissora. As fichas de papel contêm todos os dados a respeito das obras e não há forma de localizar determinada informação sem fazer busca extensiva a essas fichas ou utilizar da memória dos recursos humanos da emissora. Por outro lado, os procedimentos manuais realizados na programação possuem forte dependência dos conhecimentos e experiência adquiridos pelo programador.

Com foco nesses problemas, foi montada uma proposta de sistema que seria responsável por auxiliar nos dois processos apontados. As informações do acervo musical deveriam ser normalizadas em uma base de dados para serem acessadas de forma mais dinâmica. Além disso, o processo de programação também seria informatizado de forma a se integrar com o acervo e facilitar a seleção de obras e o controle de tempo das execuções.

Nessas reuniões foram definidos os seguintes requisitos funcionais da ferramenta a ser implementada:

- Possibilidade de manutenção do acervo através de novos cadastros, alterações e exclusões de obras;
- Ferramenta para auxiliar a programação, onde possa ser acessado o acervo para incluir as obras no bloco musical;
- Possibilidade de alterar a grade devido a problemas de última hora na execução das obras;
- Armazenar as grades de programação para possibilitar consultas;
- Armazenar a última data de execução de cada obra;
- Gerar relatórios de programação para divulgação na página da Internet da emissora.

Como a Rádio da Universidade não possui um departamento de informática, foi requisitado que a aplicação fosse de fácil manutenção e empregasse apenas software livre. Não foram definidos outros requisitos não-funcionais para a ferramenta.

4.2.2 Arquitetura

Para a construção e funcionamento da ferramenta, foi elaborada uma arquitetura que permitisse disponibilizar o aplicativo na *web*.

Para armazenar os dados foi escolhido o SGBD relacional MySQL (MYSQL, 2013). O sistema apresenta-se como uma solução robusta sendo considerada a ferramenta livre mais utilizada no mundo entre bancos de dados (DBENGINES, 2013). MySQL utiliza a linguagem SQL para realizar consultas e não requer infraestrutura sofisticada para ser instalado.

A linguagem de programação utilizada na ferramenta é o PHP. A linguagem é amplamente utilizada para criar esse tipo de aplicação para *web* pois é livre, de código aberto e possui grande quantidade de plug-ins, bibliotecas e frameworks disponíveis na web que podem ser utilizados para enriquecer o projeto. Também optou-se pela utilização de um framework por ser uma opção que fornece um conjunto de funcionalidades que mantém a

organização da aplicação e a qualidade do código. O framework selecionado foi o CakePHP (CAKEPHP, 2013).

Para instalar todas essas ferramentas, o autor utilizou uma pilha WAMP que é um pacote que reúne todos os softwares necessários para criar um *webserver* em um único computador com o sistema operacional Windows. A instalação utilizada possui Apache 2.4, MySQL 5.6 e PHP 5.5.3. O sistema operacional foi o Windows 7 64 bits. Já a versão do framework selecionado é o CakePHP 2.4.

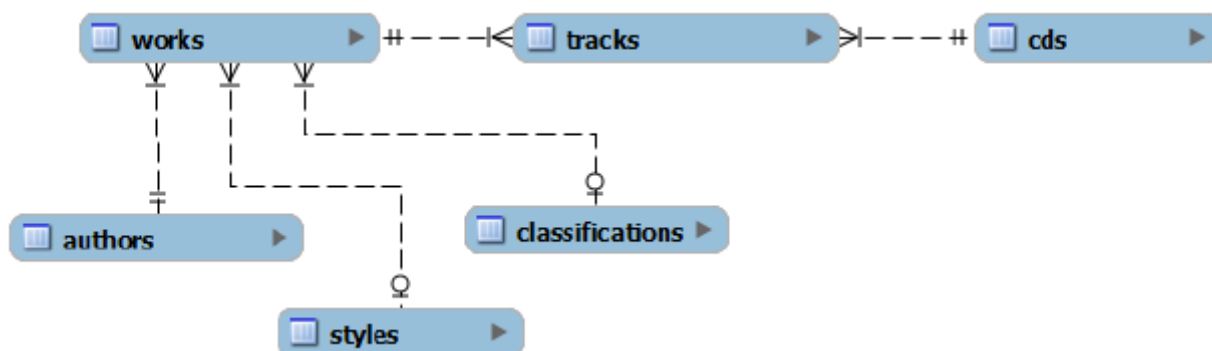
4.2.3 Projeto do Banco de Dados

O framework CakePHP possui uma série de recomendações sobre definições do modelo de dados que visam fornecer facilidades na criação de aplicações. As principais regras tratam da nomenclatura de tabelas, atributos e chaves primárias. Os nomes das tabelas devem estar em inglês e no plural. A chave primária deve ser simples e nomeada como “Id”. As chaves estrangeiras devem possuir o nome da tabela a qual se referencia seguido do sufixo “_id”. Campos de armazenamento de datas de criação e alteração de registros devem ser nomeados como “created” e “modified”. O autor procurou atender a todas essas recomendações a fim de maximizar as vantagens oferecidas pelo framework.

O modelo de dados foi elaborado com foco no acervo da rádio. Baseada na utilização das fichas de papel, criou-se uma entidade denominada “works” para armazenar as informações a respeito de cada obra. Essa tabela possui como atributos o título, autor, duração e código da obra herdado das fichas. O autor é representado através de relacionamento com entidade denominada “authors”. Essa tabela possui atributo para o nome do autor e um campo para armazenamento de observações. A tabela de obras possui ainda outros dois atributos para classificação a nível de estilo musical (barroco, romântico, moderno, etc.) e qualidade (para possibilitar com que obras de qualidade mais elevada passam ser pesquisadas e executadas com mais frequência). Esses atributos estão relacionados com as entidades “styles” e “classifications” respectivamente.

Para a localização das obras foi definida uma entidade chamada “cds” para registro das mídias físicas. Considerando que uma obra pode estar em mais de um CD e um CD pode conter mais de uma obra, conclui-se que entre as entidades há relacionamento de cardinalidade N para N. Para atender essa representação, criou-se uma nova entidade denominada “tracks” para representar as faixas musicais que estão presentes nos CDs e fazem parte de uma obra. Esse conjunto de tabelas poder ser visualizada na Figura 4.4.

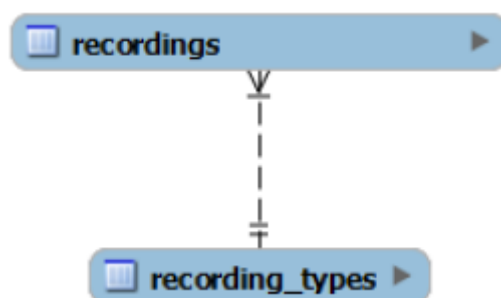
Figura 4.4 – Modelo ER das tabelas relativas ao acervo musical



Fonte: MARCANT, 2013

A etapa seguinte consistiu em montar o modelo para armazenar os dados do processo de programação da rádio. Foi criada uma entidade denominada “recordings” para responsável por manter o cadastrado de todos programas, recorrentes ou não, chamadas, spots, vinhetas e blocos musicais da emissora. Essa tabela possui os atributos para nome, duração, regras de recorrência e tipo da gravação (programa, spot, etc.). Foi criada ainda outra tabela para armazenar esses tipos e se relacionais com a tabela de gravações denominada “recording_types”. A Figura 4.5 ilustra essa parte do modelo.

Figura 4.5 – Modelo ER para armazenar as gravações

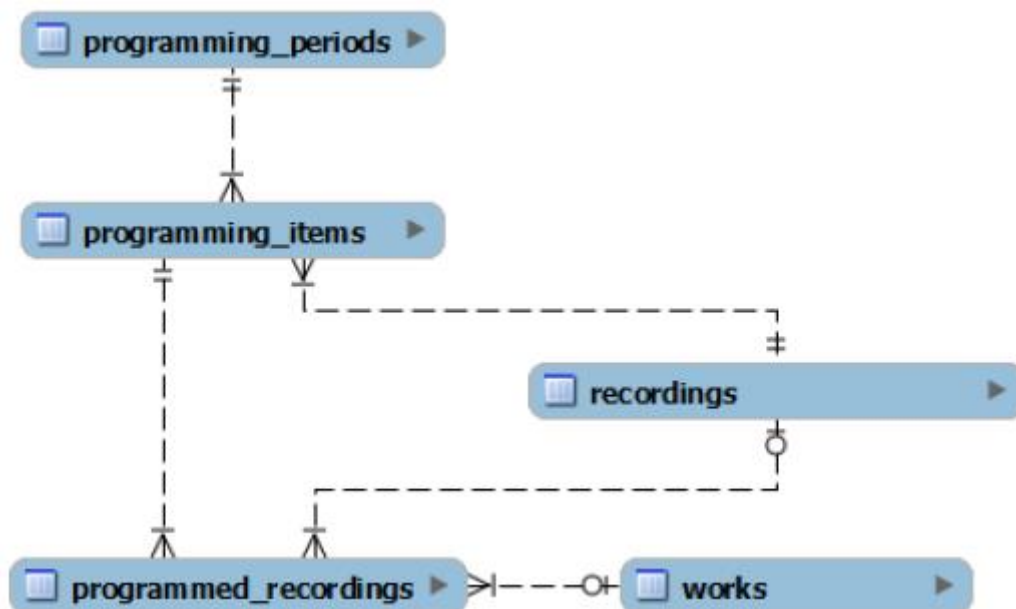


Fonte: MARCANT, 2013

A programação é registrada em outra entidade denominada “programming_periods” que possui os atributos de data e hora de início e fim do período de programação e outro campo de status para indicar se a programação está sendo elaborada ou concluída. Cada período de programação é composto por um conjunto de gravações que compõem uma agenda. A tabela “programming_items” serve para armazenar os itens de cada programação, a entidade possui atributos para identificar o período, a gravação, e data e hora de início e fim da execução do programa. Para registrar a sequência de obras de um bloco musical, há ainda a

entidade “programmed_recordings” para armazenar a identificação da obra, o número sequencial da peça dentro do bloco e o bloco de gravação ao qual ele está referenciado. A Figura 4.6 ilustra esse conjunto de tabelas e os seus relacionamentos.

Figura 4.6 – Modelo ER da parte de programação



Fonte: MARCANT, 2013

4.2.4 Protótipos de Interface

Durante o desenvolvimento do projeto, as reuniões com os usuários foram constantes e a utilização de protótipos de interface foram utilizados para facilitar a compreensão da ferramenta e, conseqüentemente, validar o projeto. Os protótipos foram elaborados com base nos requisitos levantados e envolveram apenas algumas funcionalidades. Muitas das funções são semelhantes como as telas de cadastrados de autores, obras, estilos e classificações.

A primeira tela elaborada foi a de cadastro de autores. A imagem possuía um campo para fazer buscas e uma tabela de autores já cadastrados com paginação. Na linha de cada autor havia botões para edição ou exclusão do registro. A tela ainda tinha um outro botão para permitir o cadastro de novos autores com mostra a Figura 4.7.

Figura 4.7 – Protótipo do cadastro de autores

Cadastro de Autores

Autor

COD	AUTOR	
1	SERGEY RACHMANINOFF	<input type="checkbox"/> <input type="checkbox"/>
2	JOHANNES BRAHMS	<input type="checkbox"/> <input type="checkbox"/>
3	JOHAN HALVORSEN	<input type="checkbox"/> <input type="checkbox"/>

Fonte: MARCANT, 2013

Ao acessar a opção de incluir novo registro, o usuário é levado a outra tela que permita fazer essa operação. Nessa tela, a aplicação apresenta em forma de campos os dados que podem ser preenchidos. Para a operação de edição de registros, a mesma tela é apresentada mas com os dados já preenchidos, bastando ao usuário modificar apenas o que deseja e salvar as alterações. Como protótipo foi projetada a tela de cadastro de uma obra apresentada na Figura 4.8.

Figura 4.8 – Protótipo do cadastro de uma obra

Cadastro de Obras – Nova Obra

Número CD: Título:

Duração: Autor:

Nota: Período:

Faixas	Movimentos
4	ALLEGRO MODERATO
5	ADAGIO
6	MINUETO

Intérpretes	Tipo
ORQUESTRA DE CAMARA "QUODBLIET MUSICUM"	ORQUESTRA
AURELIAN POPA	DIREÇÃO

Observação:

Primeira Execução Última Execução

Fonte: MARCANT, 2013

A tela inicial da função de programação é semelhante aos de outros cadastrados. Ao acessar uma grade, o usuário é levado ao esqueleto da programação quem contém a lista das gravações já em ordem de exibição como mostrado na Figura 4.9. Ainda não apresentados o tempo de duração de cada programa e o tempo disponível a ser programado. Há também botões para exclusão edição e exclusão da gravação.

Figura 4.9 – Protótipo da tela de programação

Programação - Esqueleto

Programação de a

Data:

Hora	Programa	Duração	Tempo Disponível	
07:00	Bloco Musical	60'	60'	<input checked="" type="checkbox"/> X
09:00	Jornalismo 1080	10'	0'	X
09:10	Bloco Musical	50'	50'	<input checked="" type="checkbox"/> X
10:00	JORNAL DA UFRGS 1 EDIÇÃO	10'	0'	X

Fonte: MARCANT, 2013

As gravações do tipo Bloco Musical possuem ainda a necessidade de localizar e selecionar as obras que serão executadas. Para isso, a tela de programação dos blocos foi elaborada e contém um espaço dedicado à lista de obras e autores selecionados. Há um cabeçalho de identificação da programação e do bloco selecionado e um botão para a função de adição de novas obras ao bloco. A Figura 4.10 ilustra o protótipo utilizado na validação com os usuários.

Figura 4.10 – Protótipo da tela de programação de um bloco musical

Programação – Bloco Musical

Programação de a Data: Bloco:

Duração Tempo Disponível

Ordem	Autor	Obra	Duração	
1	Kraus	Abertura "Olimpia"	07'35"	X
2	J.Haydn	Quarteto de cordas, nº32, em dó maior, Op.33 nº3, "O Pássaro"	13'25"	X
3	Dussek	Concerto para harpa e orquestra, em mi bemol maior, Op.15	10'00"	X

Fonte: MARCANT, 2013

É importante salientar que esse conjunto de protótipos foi organizado para facilitar a compreensão da proposta da ferramenta junto os usuários. A partir dessas reuniões algumas modificações foram incorporadas ao projeto de forma a tornar a aplicação mais semelhante aos processos da rádio.

4.2.5 Importação do Dados

A informatização do acervo da rádio exigiu um esforço de migração dos dados das fichas de papel para o domínio da aplicação. O acervo é composto por mais de 4000 obras, e a Direção e o Departamento de Programação estavam cientes que esse passo poderia demandar um longo esforço para cadastramento das obras. No entanto o autor dedicou uma parte do projeto na busca de soluções para facilitar a migração desses dados.

A partir das fichas armazenadas em arquivos do Microsoft Word e com a utilização de macros de programação, foi possível extrair as informações referentes às obras em uma planilha eletrônica do Microsoft Excel. Após a utilização de fórmulas para dispor os dados em forma mais organizada, foi possível gerar scripts SQL para que permitisse popular as tabelas do banco de dados do acervo. Esses scripts conseguiram extrair as informações sobre título de obras, autores, duração, observações e localização física. Os dados referentes ao estilo e classificação foram deixados em branco para posterior preenchimento pelos usuários.

A partir de observações da locução ao vivo da rádio foi possível extrair uma relação aproximada entre os tempos de anúncio e desanúncio com o tamanho da informação textual da obra. Utilizando essa relação, foi possível gerar mais scripts para popular os atributos de anúncio e desanúncio relacionados às obras.

Essa importação foi importante na migração dos dados. Com isso foi possível alimentar o banco de dados suficientemente para, em um primeiro momento, gerar informações reais durante a codificação da ferramenta e, posteriormente, oferecer um conjunto de dados mínimos para que o Departamento de Programação conseguisse atualizar e manter o acervo digital com maiores chances de sucesso.

4.2.6 Ferramenta

A etapa de codificação da ferramenta foi dividida em três grupos de funcionalidades. A primeira etapa foi desenvolver as funções relativas ao acervo musical que envolve cadastro de obras, autores, estilos, classificações das obras. Além disso, o cadastro de gravações foi incluído nessa etapa pela similaridade das funções. A segunda etapa consistiu na elaboração das funções de programação da emissora e montagem dos blocos musicais. Por fim, a construção de relatórios foi feita na terceira etapa.

Os cadastros são fundamentais para o processo de manutenção do acervo. Esse tipo de serviço são essencialmente inclusão e manipulação de registro na base de dados. Por isso, o cadastro de estilos, classificações, autores, e obras disponibilizam atividades de criação, visualização, alteração e deleção de registros.

A programação envolve tarefas como criação e manipulação de períodos, montagem de blocos musicais através da seleção de obras do acervo e o cadastro de gravações. Cada período de programação dá origem a uma grade de programação que é composta por um conjunto de gravações e blocos musicais.

Por último, os relatórios são visualizações da grade programação em formatos específicos e apropriados para atender diferentes demandas dos processos da rádio.

A Figura 4.11 apresenta a tela principal da primeira versão da ferramenta. É possível observar todas as funcionalidades dispostas no menu lateral.

Figura 4.11 – Tela principal da aplicação de programação

Fonte: MARCANT, 2013

As principais funcionalidades dessa versão da ferramenta são:

- Criação, visualização, alteração e exclusão de registros de obras;
- Criação, visualização, alteração e exclusão de registros de autores;
- Criação, visualização, alteração e exclusão de registros de estilos;
- Criação, visualização, alteração e exclusão de registros de classificações;
- Criação, visualização, alteração e exclusão de registros de tipos de gravações;
- Criação, visualização, alteração e exclusão de registros de gravações;
- Criação, visualização, alteração e exclusão de registros de períodos de programação;
- Montagem de blocos musicais;
- Emissão de relatórios de obras por bloco de programação;
- Emissão de relatórios de grade para o operador técnico;
- Emissão de relatórios de obras para locução;
- Emissão de relatórios de grade para disponibilização do site da rádio.

5 APLICAÇÃO WEB PARA PROGRAMAÇÃO MUSICAL

Este capítulo descreve as atividades realizadas no desenvolvimento deste projeto. São apresentados os novos requisitos e forma como foram identificados. Em seguida, tanto a arquitetura quando o modelo de dados são reavaliados a fim de se verificar se ainda atendem à demanda. O processo de implementação da ferramenta é detalhado, apontando e justificando as decisões tomadas no desenvolvimento em relação a utilização de padrões de projetos e software. Ao fim do capítulo a nova versão da Aplicação Web de Programação Musical é apresentada.

5.1 Análise de Requisitos

Com a retomada do projeto, foi preciso realizar uma nova rodada de reuniões e entrevistas para levantar os novos requisitos. Inicialmente houve uma reunião com todos os envolvidos usuários da ferramenta. Participaram desse encontro o diretor da rádio Prof. André Prytoluk, Paulo Busato e Juliano Dupont do Departamento de Programação e Manoela Nogueira do Departamento de Jornalismo. Inicialmente foi apresentada a ferramenta desenvolvida no trabalho anterior. Foram mostradas todas as funcionalidades implementadas e discutidas novas melhorias. Outras duas reuniões aconteceram na sequência envolvendo apenas os profissionais do Departamento de Programação, nas quais foram apontadas algumas inconformidades da aplicação com o processo executado no departamento e discutidas novas funcionalidades a serem desenvolvidas.

Desde o início, foi possível detectar duas preocupações dos usuários. O Prof. André Prytoluk apontou a importância da ferramenta na informatização dos processos de manutenção do acervo e programação da grade. Já os funcionários do Departamento de Programação estavam preocupados com a complexidade da ferramenta e a necessidade de o software conseguir se adaptar aos processos existentes sem alterar as atividades já desenvolvidas no departamento. Essa última observação, portanto, se mostrou como um desafio ao projeto e serviu como inspiração para a realização do trabalho.

Durante as reuniões as funcionalidades relacionadas ao cadastro e manutenção do acervo foram consideradas suficientes para atender à demanda da rádio. Apenas algumas sugestões pontuais foram apontadas como, por exemplo, a tela de exibição das informações de determinado autor passe a exibir uma lista de suas obras. Essas sugestões serão abordadas e listadas mais adiante. Já as funcionalidades que envolvem o processo de programação foram

alvo de maior atenção. Foi possível perceber a dificuldade no entendimento das atividades realizadas pela ferramenta e quais as opções oferecidas aos usuários para facilitar a construção de uma grade programação.

Diante dessas dificuldades optou-se iniciar pela realização de um mapeamento do processo de programação com o objetivo de facilitar o entendimento dos usuários. Para isso, foi utilizada a notação BPMN (*Business Process Model and Notation*). A construção do diagrama foi baseada na descrição do processo (MARCANT, 2013) e na versão preliminar da ferramenta. Os Apêndices A e B contêm a representação do processo de programação da emissora e do subprocesso de montagem dos blocos musicais.

Com a criação dos artefatos e validação junto ao usuário, foi possível reunir mais requisitos a serem implementados na nova fase da ferramenta. A primeira solicitação foi a ampliação da busca por obras, inserindo filtros quanto a estilo, classificação, duração e principalmente por autor. A busca por torna-se particularmente importante ao permitir a procura por parte do nome do autor como utilizar a entrada “Mozart” para encontrar o cadastro de Wolfgang Amadeus Mozart. Junto a isso, foi solicitada a opção de ordenação das listas de autores pelo sobrenome.

A organização da ferramenta também precisou ser modificada. O menu principal da aplicação não apresenta de forma clara as três grandes funcionalidades de manutenção do acervo, programação e relatórios. A Figura 5.1 é um exemplo de tela da ferramenta. É possível verificar os links para todas as telas são apresentadas constantemente ao usuário podendo induzi-lo ao erro. Foi proposto, portanto, uma nova organização dos menus em dois níveis. Onde o primeiro nível apresentaria as funcionalidades relacionadas ao acervo e à programação, e o segundo nível exibiria apenas os links relacionadas à funcionalidade acessada. A funcionalidade de relatórios seria integrada à de programação pela relação estreita entre elas.

Figura 5.1 – Tela da versão anterior da ferramenta

Programação Musical - Rádio da Universidade 1080 AM

Ações

Novo Período de Programação

Programação

Períodos

Cadastros

Obras

Autores

Estilos

Classificações

Tipos de Gravação

Gravações

Relatórios

Programação

Períodos de Programação

Nome	Início	Fim	Status	Ações
Programação de 07/11/2013 a 08/11/2013	2013-11-07 07:00:00	2013-11-08 06:59:00	Rascunho	Visualizar Editar Excluir
Programação de 31/10/2013 a 07/11/2013	2013-10-31 07:00:00	2013-11-07 06:59:00	Rascunho	Visualizar Editar Excluir
Programação de 24/10/2013 a 31/10/2013	2013-10-24 07:00:00	2013-10-31 06:59:00	Completo	Visualizar Excluir
Programação de 10/10/2013 a 17/10/2013	2013-10-10 07:00:00	2013-10-17 06:59:00	Rascunho	Visualizar Editar Excluir
Programação de 03/10/2013 a 10/10/2013	2013-10-03 07:00:00	2013-10-10 06:59:00	Completo	Visualizar Excluir

Página 1 de 1, exibindo 5 registros do total de 5, iniciando no registro 1, terminando no 5

< Anterior Próximo >

Fonte: MARCANT, 2013

No entanto, o requisito mais significativo foi a inclusão de suporte a autenticação e autorização de usuários. Por se tratar de uma ferramenta disponibilizada na web era especialmente importante a incorporação de mecanismos de segurança. Apenas as pessoas autorizadas deveriam ter acesso à ferramenta. Para isso, foi sugerido limitar o acesso mediante processo de login onde cada usuário passaria a ser identificado através de utilização de credenciais pessoais. Com a utilização de mecanismo de autenticação surge a necessidade de nova funcionalidade relativas a gerência de credenciais. Portanto foi preciso incluir uma nova área para cadastrado de usuários e senhas.

Com o encerramento dessa etapa de levantamento dos requisitos, foi possível separar aqueles que são funcionais dos não funcionais. A seguinte lista compila os principais requisitos não funcionais:

- Reorganização da ferramenta de forma a torná-la mais intuitiva e simples para os usuários;
- Inclusão de mecanismos de segurança para proteção das informações e impedir que pessoas não autorizadas consigam acessar a ferramenta;
- Nova funcionalidade de administração da ferramenta para permitir o cadastro das credenciais dos usuários.

Já os requisitos funcionais abaixo foram identificados:

- Permitir busca de obras por estilo, classificação, duração e autor;
- Exibir lista de obras no cadastro de autor;
- Permitir ordenação de lista de autores por sobrenome;
- Permitir cadastrar novos usuários e definir senhas;

- Função de troca ou recuperação de senha.

Adicionalmente a esses requisitos, outros foram identificados mas deixados de fora do escopo do projeto e sugeridos como sugestão de futuros trabalhos relacionados à ferramenta. Por outro lado, algumas mudanças adicionais discutidas durante o andamento das etapas seguintes do projeto foram incorporadas à aplicação. Na maioria das vezes, essas mudanças envolviam apenas pequenas alterações de interface.

5.2 Revisão da Arquitetura

Com a definição dos novos requisitos a arquitetura definida do trabalho anterior precisou ser reavaliada. Com as mudanças planejadas foi preciso verificar se as definições anteriores ainda atendiam às necessidades da nova ferramenta.

Não houve mudança quanto à disponibilização da ferramenta na web, portanto a linguagem PHP é indicada para a construção de aplicativos como esse. PHP é simples, livre, não exige infraestrutura elaborada e possui uma grande comunidade de programadores. A linguagem, portanto, possui todas as características necessárias para atender a demanda do projeto.

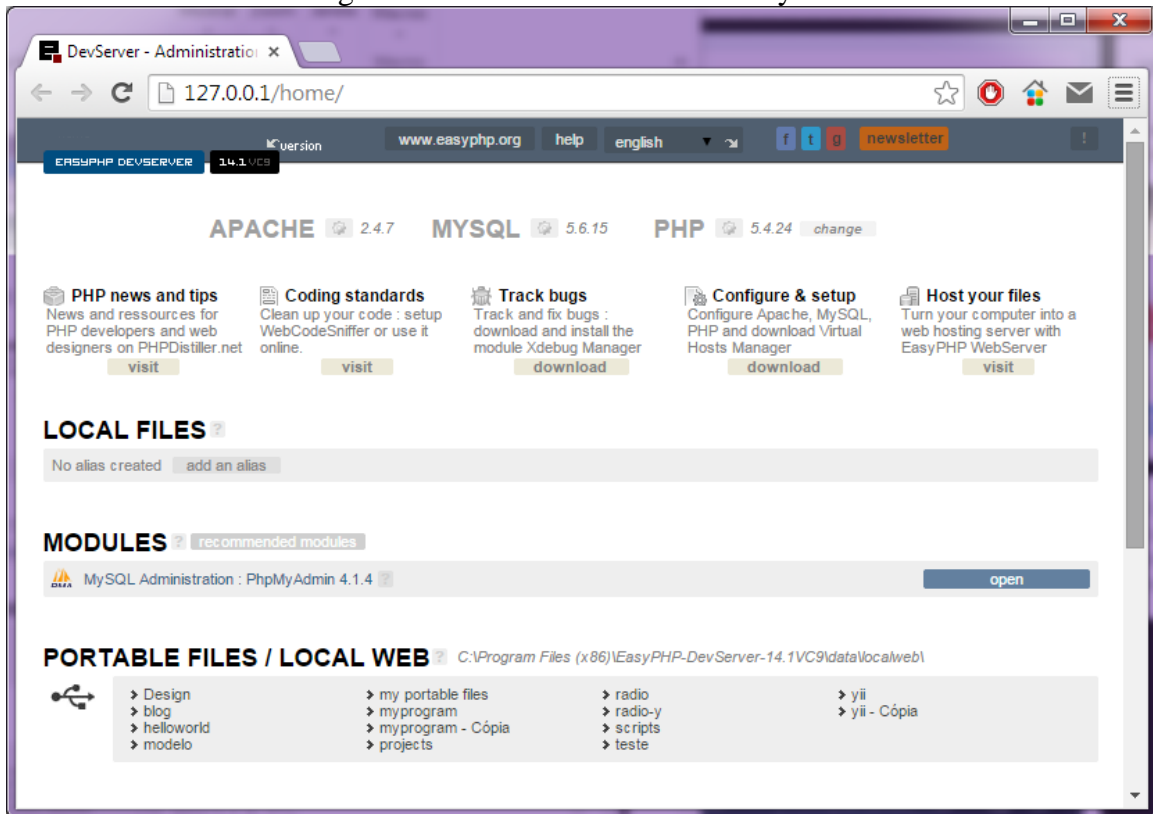
O banco de dados relacional MySQL continua sendo o mais indicado para o projeto. O SGBD é livre e largamente utilizado por aplicações ao redor do mundo⁶. A linguagem SQL é utilizada para consulta e manipulação de dados. PHP possui alta integração com o MySQL pois oferece suporte nativo para conexão. Além disso, há a disposição uma grande variedade de ferramentas e componentes que dão suporte a essa integração.

A instalação desses softwares foi feita através do pacote WAMP denominado EasyPHP⁷ que oferece versões estáveis do PHP e MySQL. O pacote disponibiliza também um servidor HTTP Apache (APACHE, 2014) responsável por prover a acessibilidade entre o usuário a aplicação. A Figura 5.2 mostra a tela de configuração do pacote. O EasyPHP oferece ainda o módulo phpMyAdmin que é uma interface gráfica para acesso banco de dados como pode ser visto na Figura 5.3. A versão do EasyPHP utilizada neste projeto foi a 14.1 VC9 que possui o Apache 2.4.7, MySQL 5.6.15 e o PHP 5.4.24.

⁶ <http://db-engines.com/en/ranking>

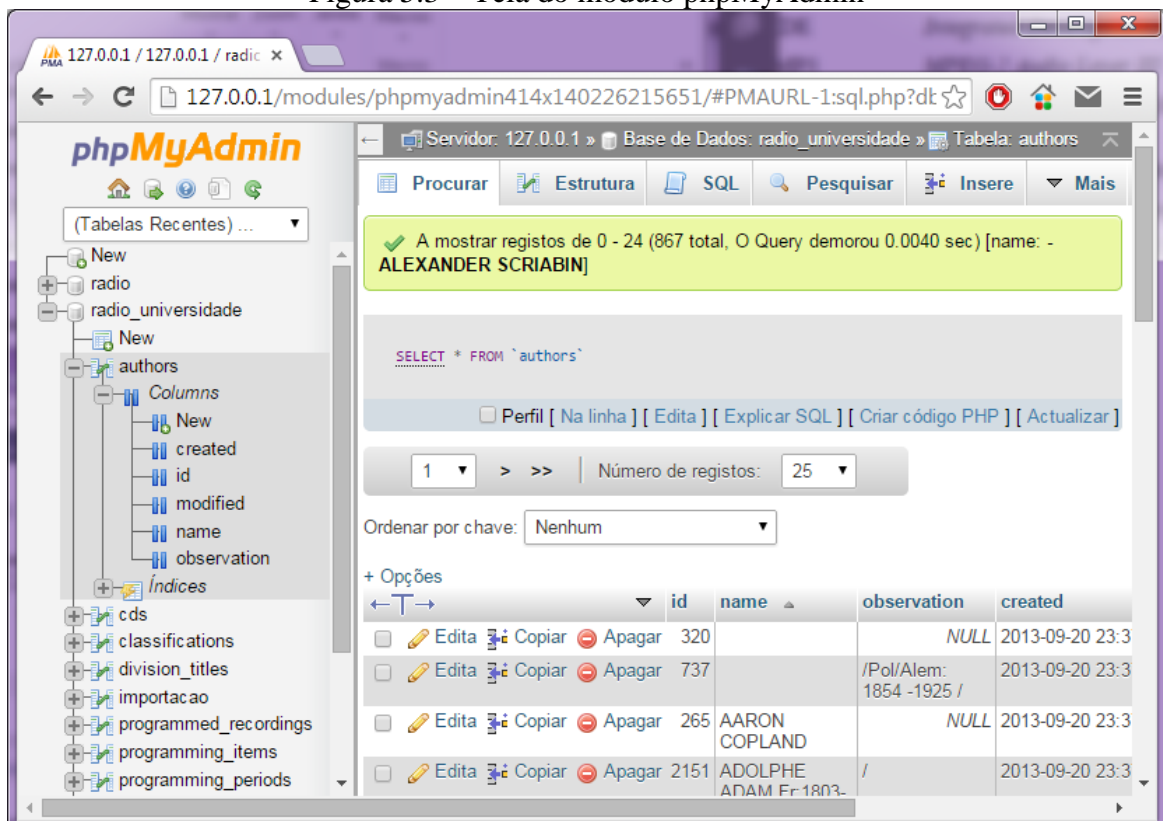
⁷ <http://www.easyphp.org/>

Figura 5.2 – Tela de controle do EasyPHP



Fonte: Elaborada pelo autor

Figura 5.3 – Tela do módulo phpMyAdmin



Fonte: Elaborada pelo autor

A utilização do framework CakePHP para desenvolvimento da ferramenta também foi avaliada. Frameworks para aplicações web são soluções que proveem uma série de componentes de visam aliviar a sobrecarga de atividades comuns a muitas aplicações na web, como acesso a banco de dados e tratamento de exceções, entre outros. Há inúmeras vantagens na adoção de um framework. A reutilização de códigos acelera o desenvolvimento e aumenta a qualidade do software ao empregar componentes já testados por outros programadores. Ao mesmo tempo, a abstração de determinadas atividades leva o desenvolvedor a investir mais tempo na tarefa a ser cumprida do que na tecnologia a ser adotada para alcançar tais objetivos. O CakePHP é um framework bastante popular e apresenta as principais características comuns às soluções similares, tais como *Active Record*, convenção sobre configuração e desenvolvimento em camadas MVC.

No entanto, neste projeto optou-se deixar de usar o CakePHP e utilizar o Yii⁸ Framework para o desenvolvimento da ferramenta. O Yii é um framework igualmente popular, eficiente e apresenta as mesmas características do CakePHP. Mas a decisão de alteração foi tomada com o objetivo de alinhar a ferramenta com as demais aplicações institucionais da UFRGS. Desde 2010 o CPD utiliza o Yii como framework de desenvolvimento PHP. Segundo Hubert Ahlert, diretor do Departamento de Sistemas de Informação do CPD, em novembro de 2014, 50% dos sistemas de ensino, pesquisa, extensão e administrativos utilizam o Yii e a tendência é que os demais sistemas também sejam migrados a médio e longo prazo.

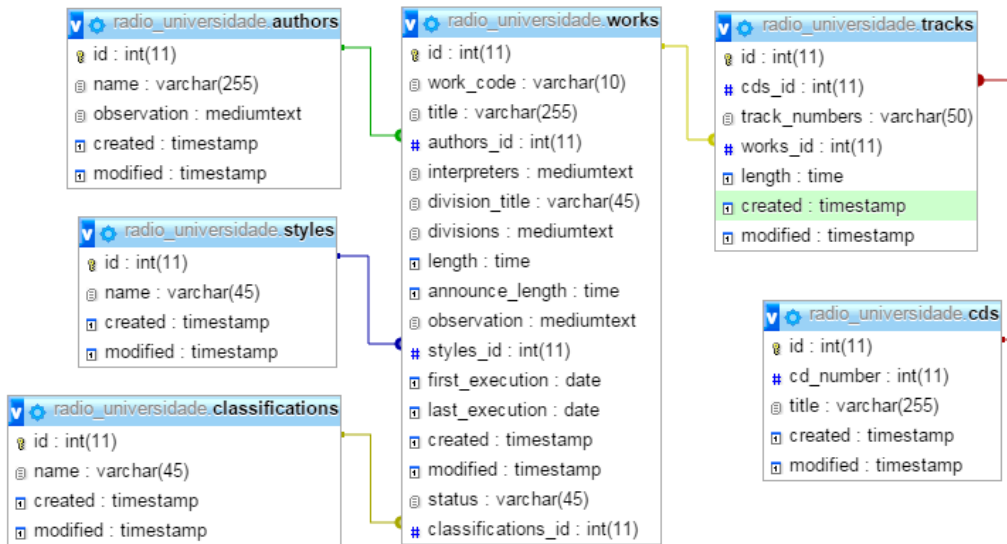
Portanto, ainda que o trabalho de conversão do código fonte pese contra a decisão alterar o framework, a possibilidade de uma futura integração da ferramenta com os sistemas institucionais da universidade cria um conjunto de novas possibilidades para qualificar a ferramenta, tais como infraestrutura mais eficiente para disponibilização da ferramenta e suporte para manutenção da aplicação.

⁸ <http://www.yiiframework.com/>

5.3 Revisão do Modelo de Dados

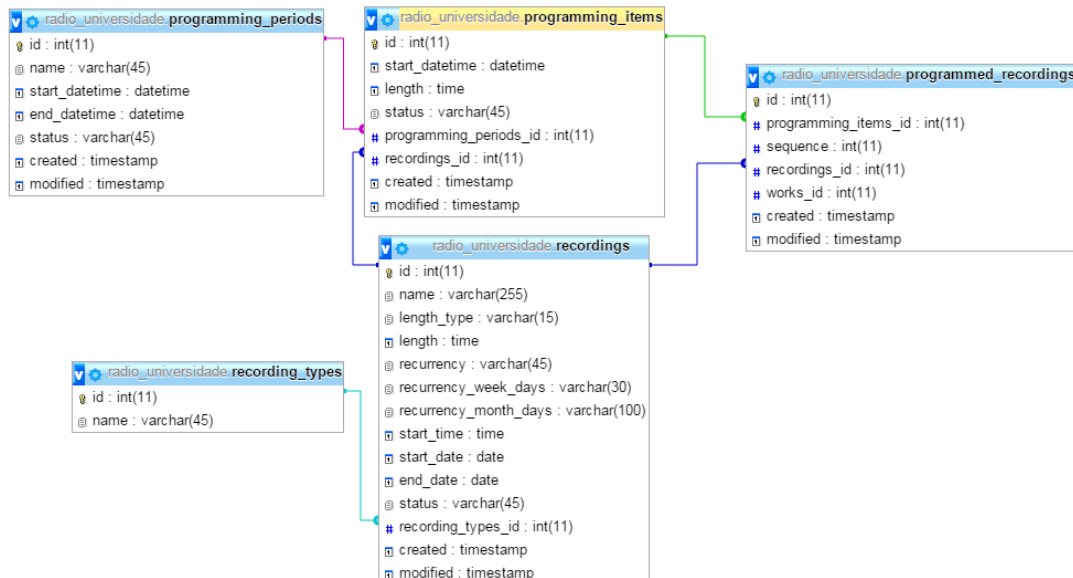
Com as novas requisições modelo de dados relacional utilizado no projeto anterior precisou ser revisto. No entanto não foi preciso fazer grandes alterações no que já estava previsto. Os modelos referentes ao acervo musical (Figura 5.4) e programação (Figura 5.5) são apresentados de forma mais completa.

Figura 5.4 – Modelo de dados do acervo musical



Fonte: Elaborada pelo autor

Figura 5.5 – Modelo de dados da programação



Fonte: Elaborada pelo autor

A alteração mais significativa foi a criação de uma nova entidade denominada “users” para registrar as credenciais de acesso dos usuários. Além de possuir atributos como usuário e senha, o e-mail pode ser armazenado também para o que o procedimento de recuperação de senha possa ser implementado. A Figura 5.6 mostra o diagrama da nova tabela.

Figura 5.6 – Modelo de dados dos usuários da aplicação



Fonte: Elaborada pelo autor

5.4 Implementação da Aplicação

O desenvolvimento da ferramenta foi feito em etapas que eram encerradas com a validação junto aos usuários. O objetivo era dividir a aplicação em módulos funcionais capazes de executaram atividades previstas nos requisitos funcionais do trabalho anterior somados aos dos novos requisitos. Ao validar esses módulos alguns ajustes eram solicitados e a implementação era retomada para atender às novas demandas. Cada etapa da implementação só era encerrada quando o módulo era homologado com os usuários.

A primeira etapa consistiu na migração dos códigos fontes para o novo framework. O objetivo era deixar a aplicação com as mesmas funcionalidades apresentadas no trabalho anterior. A implementação iniciou com a criação dos modelos de dados. Na arquitetura MVC, os modelos são abstrações que representam os dados da aplicação.

O Yii disponibiliza a técnica denominada *Active Record* para acessar e manipular o banco de dados. Para isso, é preciso fazer o mapeamento de cada tabela em uma classe diferente. Portanto, as classes a seguir foram implementadas e adicionadas ao projeto: “Author”, “Cd”, “Classification”, “ProgrammedRecording”, “ProgrammingItem”, “ProgrammingPeriod”, “ProgrammingRecording”, “Recording”, “RecordingType”, “Style”, “Track” e “Work”. Inicialmente, cada classe de modelo contém apenas o nome da tabela à qual se refere e as relações com as outras tabelas do banco. A Figura 5.7 representa um trecho

da classe “Author” que define o relacionamento da tabela “authors” com a tabela “works”. Dentro de um projeto Yii, essas classes são chamadas de modelos e disponibilizadas dentro do diretório denominado “models”.

Figura 5.7 – Trecho do código da classe “Author”

```
24▼ /**
25   * @return array relational rules.
26   */
27   public function relations()
28▼ {
29       return array(
30           'works' => array(self::HAS_MANY, 'Work', 'authors_id'),
31       );
32▼ }
--
```

Fonte: Elaborada pelo autor

Após a criação dos modelos, a implementação prosseguiu com a criação dos controladores. Começando com a funcionalidade de gerência do acervo, foram criados os controladores para ações relativas às obras, autores, estilos e classificações. Cada controlador é definido como uma classe que disponibiliza um conjunto de métodos denominados “actions” capazes de interpretar uma requisição de usuário, manipular os modelos disponíveis e instanciar uma visão capaz apresentar o resultado da requisição ao usuário. Os controladores implementados para a processo do acervo são bastante simples e semelhantes, pois disponibilizam “actions” para visualizar, criar, editar e excluir registros do acervo. A Figura 5.8 ilustra uma parte do código do controlador “Author”, responsável por manipular o cadastro de autores da rádio. É possível observar que o trecho em questão contém a definição da “action” responsável por criar e salvar um novo autor na base de dados. A linha 27 instancia um objeto da classe “Author” e o trecho compreendido entre as linhas 32 e 36 é responsável por receber os atributos relacionados ao autor, atribuir ao objeto criado anteriormente e executar o salvamento do objeto na base de dados. Caso essa tarefa tenha sido executada com sucesso, a ação “view” é invocada para apresentar as informações salvas do novo autor inserido. Caso contrário, a tela de criação é acionada para voltar a apresentar o formulário de cadastrado juntamente com as informações de erro geradas ao tentar salvar os dados.

Figura 5.8 – Trecho do código do controlador “Author”

```

21▼ /**
22  * Creates a new model.
23  * If creation is successful, the browser will be redirected to the 'view' page.
24  */
25  public function actionCreate()
26▼ {
27      $model=new Author;
28
29      if(isset($_POST['Author']))
30▼ {
31
32          $model->attributes=$_POST['Author'];
33
34          if($model->save())
35              $this->redirect(array('view','id'=>$model->id));
36      }
37
38      $this->render('create',array(
39          'model'=>$model,
40      ));
41▼ }

```

Fonte: Elaborada pelo autor

A Figura 5.8 apresenta ainda um exemplo de como o *Active Record* consegue abstrair o acesso ao banco de dados. Os dados referentes a um autor podem ser acessados, alterados e salvos utilizando orientação a objetos. O exemplo ilustrado mostra a criação de um objeto vazio do tipo “Author” a fim de que um novo autor seja cadastrado. No entanto, é possível também instanciar outros objetos representando autores já cadastrados na base de dados. Para isso, seria necessário passar como parâmetro o valor referente à chave primária do autor que se deseja acessar. Os objetos do tipo *Active Record* oferecem uma porção de métodos que auxiliam as atividades básicas de manipulação de dados. O método “attributes” realiza um casamento de dados atribuindo os valores de um vetor às propriedades do objeto, ou em outras palavras, aos atributos da tabela representada pelo objeto. Já o método “save” é responsável salvar os dados atribuídos, gerando os comandos SQL necessários para tal tarefa e submetendo-os ao banco de dados. É possível, portanto, observar nesse pequeno exemplo, como tarefas complicadas como acesso ao banco, manipulação de dados e salvamentos podem ser simplificadas ao utilizar um arcabouço de soluções presentes em um framework.

O restante das ações representadas em um controlador funcionam de forma semelhante ao descrito anteriormente. Os controladores vinculados às entidades correspondentes a obras, estilos e classificações são análogos ao de autores, pois basicamente possuem “actions” para visualização, criação, alteração e exclusão de registros. Dentro da estrutura do Yii, os controladores são disponibilizados dentro do diretório “controllers”.

Concomitantemente à implementação dos controladores, os códigos referentes à camada de visão são desenvolvidos também. As “views” são trechos de códigos implementados com HTML responsáveis por apresentar ao usuário uma resposta à sua requisição. O exemplo da Figura 5.8 mostra como o acionamento de uma visão é feito. O comando da linha 38 executa uma chamada à visão “create” passando como parâmetro um objeto do tipo “Author”. Nesse caso o objeto pode possuir dois valores diferentes. Em um primeiro caso, o objeto pode ser vazio, e isso indica que um formulário para cadastro de novo autor deva ser apresentado ao usuário como todos os campos em branco. Em um segundo caso, o objeto passado como parâmetro pode ter uma porção de dados associados frutos da tentativa de salvamento frustrado da linha 34. Nesse caso, a tela de cadastro de autor é também apresentada, mas os campos aparecem já preenchidos com os dados inseridos anteriormente. A tela ainda pode mostrar informações sobre o erro acontecido para que o usuário execute a sua tarefa de forma mais apropriada.

A Figura 5.9 apresenta o código de uma visão que apresenta a tela de cadastro de autores. É possível observar foi utilizando um tipo de componente chamado *widget* para a geração de códigos HTML. O *widget* “ActiveForm”, declarado na linha 2, é responsável por gerar marcadores para apresentação de um formulário ao usuário. As linhas 9, 10, 17 e 18 executam métodos que exibem os campos do formulário. Por fim, a linha 26 apresenta um botão de salvamento do formulário. O código apresentado é suficientemente genérico para ser acionado em situações distintas. Tais como a tela de cadastro de um novo autor ou a tela de alteração cadastral de um autor já presente na base dados. O funcionamento do formulário depende do valor do objeto representado variável “\$model”. Sendo vazio, a tela é apresentada como um novo cadastro, caso contrário, é uma tela de alteração cadastral.

Figura 5.9 – Trecho de código de uma visão de cadastro de autores

```

1 <!-- author/_form -->
2 <?php $form=$this->beginWidget('CActiveForm'); ?>
3 <fieldset>
4     <table>
5         <tbody>
6             <tr>
7                 <td>
8                     <div class="input text">
9                         <?php echo $form->labelEx($model,'name'); ?>
10                        <?php echo $form->textField($model,'name'); ?>
11                    </div>
12                </td>
13            </tr>
14            <tr>
15                <td>
16                    <div class="input text">
17                        <?php echo $form->labelEx($model,'observation'); ?>
18                        <?php echo $form->textField($model,'observation'); ?>
19                    </div>
20                </td>
21            </tr>
22        </tbody>
23    </table>
24 </fieldset>
25 <div class="row buttons">
26     <?php echo CHtml::submitButton($model->isNewRecord ? 'Create' : 'Save'); ?>
27 </div>
28 <?php $this->endWidget(); ?>
29 <!-- end author/_form -->

```

Fonte: Elaborada pelo autor

Diferentemente dos modelos e dos controladores, os códigos da camada de visão não são encapsulados em classes. As visões são arquivos PHP cujo o nome é a forma com a qual são acionados. Tipicamente, um conjunto de visões é vinculado a um único controlador. Por isso, esses arquivos disponibilizados em diretórios como o nome do controlador ao qual pertence, e esses diretórios localizados dentro de outro denominado “views”. Esse tipo de estrutura faz com que a visão “create” do controlador “author” seja diferente da visão de mesmo nome mas do controlador “work”. Embora a estrutura e comportamento dessas visões sejam semelhantes, há diferenças nos formulários apresentados, e cada código é implementado de forma diferente.

Ainda referente a implementação das visões. O Yii convencionou a existência de uma visão com escopo especial denominado “main”. Essa visão é automaticamente acionada e executada antes das outras visões e serve para gerar trechos de códigos HTML comuns a todas as telas da aplicação. Essa funcionalidade é utilizada para agrupar a criação de temas e layouts em um único lugar. Por ter o comportamento especial, essa visão não é vinculada a

nenhum controlador e, por isso, ela deve ser salva em um diretório denominado “layouts” e este, dentro do “views”.

A primeira etapa da implementação foi dedicada a fazer a migração dos fontes do projeto anterior para o novo framework. A funcionalidade de gerência do acervo da rádio consistia basicamente de um conjunto de telas de cadastrados para manipular os dados referentes a obras, autores, estilos e classificação. Pela similaridade de comportamento, o cadastro de gravações também foi incluído nessa primeira leva desenvolvimento. A etapa continuou com a implementação dos processos de programação e geração de relatórios. No entanto muitas das atividades presentes nesses processos não foram concluídas e, portanto, os processos não estavam completamente funcionais. Como o objetivo de cada ciclo da implementação era avaliar os módulos junto aos usuários, foi necessário complementar as tarefas descritas no projeto anterior além de migrar as já implementadas para que resultasse em um módulo completamente funcional.

Na funcionalidade correspondente à programação foram criados os controladores “ProgrammingPeriod”, “ProgrammingRecording” e “ProgrammingItem”, responsáveis por agregar as ações correspondentes aos cadastros dos períodos de programação, dos programas de cada período e das obras correspondente a cada bloco musical respectivamente. Esses controladores atuam de forma semelhante aos descritos anteriormente, no entanto, as telas correspondentes aos cadastros possuem comportamentos diferenciados com o objetivo de auxiliar o programador a executar suas atividades. Tomando como exemplo o cadastro de um novo período de programação, a tela de criação de períodos deve informar o usuário a não cadastrar um período que tenha colisão de horários com outro já existente. Essas regras são implementadas nos arquivos de visão e inseridas também no modelo correspondente para que a validação dos dados seja feita de forma eficiente e a ferramenta não propague a inconsistência do ponto de vista do negócio para o banco de dados.

Outra tela com funcionamento singular no sistema é a de montagem dos blocos musicais. Ao acessar a tela de edição de um bloco pela primeira vez, o usuário é apresentado a uma lista vazia de obras. A seleção de novas obras é feita através do acionamento de um botão que apresenta uma nova tela de pesquisa de obras. Após a localização da obra desejada, o programador seleciona o item e a tela da lista de blocos musicais volta a ser apresentada com a adição da obra desejada. A implementação desse tipo de funcionalidade é feita nos arquivos de visões com o auxílio de requisições assíncronas AJAX aos controladores, como por exemplo, submeter uma pesquisa e apresentar os resultados dela.

Por isso, a implementação das visões ligadas à funcionalidade de programação exigiu o maior tempo de dedicação dessa primeira etapa. Das 35 horas registradas para fazer a conversão, 20 horas foram dedicadas exclusivamente ao desenvolvimento das telas de montagem de blocos, seleção de obras e demais tarefas de auxílio à programação.

Ao fim da migração, a ferramenta foi validada junto aos usuários e a implementação passou à etapa seguinte que foi dedicada a incorporação de mecanismos de segurança previsto na nova lista de requisitos.

A primeira parte dessa etapa consistiu em criar uma nova funcionalidade dedicada ao cadastro e gerência de usuários. Para isso, foi criado o modelo “User”, o controlador de mesmo nome e as visões correspondente às telas de visualização, criação, edição e deleção de usuários do sistema.

A seguir, foi implementada a autenticação de usuários. O Yii oferece um conjunto de funcionalidades para gerência de identidades nas aplicações. Primeiramente um novo modelo foi adicionado ao projeto. Esse modelo, diferente dos citados até o momento, não representa uma entidade do banco de dados. Ao invés disso, a classe “LoginForm” é uma abstração do processo de login de usuários e possui propriedades e métodos específicos como os atributos de usuário e senha além de métodos para métodos de autenticação de senha e a criação de um novo objeto define que um usuário é considerado logado no sistema. A classe “UserIdentity” também foi criada, e implementa a relação entre o processo de login e um usuário. Por isso, um método denominado “authenticate” foi definido com as regras de validação de credenciais. Essas regras comparam os valores passados como usuário e senha com os valores armazenados no cadastro de usuários.

O processo de login é acessado pelo controlador denominado “Site” que entre outras ações de auxílio à administração da aplicação, possui uma denominada “Login”. A ação é responsável por acionar a visão que apresenta um formulário de autenticação e, posteriormente, receber os dados informados e disparar o processo de login. Caso a autenticação não tenha sido bem sucedida, a tela com o formulário volta a ser apresentada com a mensagem de erro gerada.

Ao estabelecer a figura de usuário logado, é necessário inserir em cada tela do sistema comandos de filtragem de usuários com permissão de acesso. No Yii, essa tarefa é bastante simples de ser feita, uma vez que é possível inserir métodos nos controladores para informar que tipo de controle de acesso deve ser feito e a quais regras eles devem obedecer. Foi criado, portanto, um controlador genérico denominado “Controller” que possui todas as propriedades e métodos que devem ser herdados pelos demais controladores. Nesse caso, foi implementado

o método “filters” que indica que o controle de acesso “accessControl” deve ser utilizado e um outro método chamado “accessRules” que fornece um vetor de autorizações. Na implementação, foi definido que todos os usuários do sistema têm acesso ao controlador responsável pelo login (“Site”) e os demais controladores só poderiam ser acessados por usuários logados.

Como função complementar ao processo de login, foi criada também a ação denominada “Logout” que destrói a identidade do usuário e altera o estado do indivíduo de usuário logado para não logado.

Concluída essa segunda etapa da implementação, a aplicação foi validada novamente junto aos usuários a funcionalidade de autenticação de usuários considerada concluída.

A terceira etapa de implementação foi dedicada a nova organização da ferramenta. A visão responsável por montar o layout de forma a exibir um novo menu em dois níveis. O menu principal foi inserido em uma barra horizontal na parte superior da tela e os links estariam sempre visíveis aos usuários logados apresentando as funcionalidades de “Acervo”, “Programação”, “Configurações” e a ação de saída do sistema chamada “Sair”. O menu de segundo nível foi colocando no lugar do anterior e somente os links vinculados à funcionalidade acessada estariam disponíveis.

Para a montagem dos menus, foi utilizando um widget específico fornecido pelo Yii para esse tipo de estrutura. A Figura 5.10 apresenta um exemplo desse tipo de implementação. O menu é montado somente quando o usuário acessa a funcionalidade “Acervo”.

Figura 5.10 – Trecho de código responsável por apresentar um menu

```

51 if(in_array($this->controller, array('work','author','style','classification')))
52     $this->widget('zii.widgets.CMenu', array(
53         'items'=>array(
54             array('label'=>'Obras', 'url'=>array('work/index')),
55             array('label'=>'Autores', 'url'=>array('author/index')),
56             array('label'=>'Estilos', 'url'=>array('style/index')),
57             array('label'=>'Classificações', 'url'=>array('classification/index')),
58         ),
59     ));

```

Fonte: Elaborada pelo autor

Ainda em relação a nova organização, as tarefas de emissão de relatórios foram incorporadas ao processo de programação e a nova área chamada de “Configurações” agregaria as funções de controle e gerência de usuários.

A implementação feita nesta etapa, embora simples, foi a que gerou maior mudança visual da ferramenta. Durante a validação das tarefas efetuadas, foi possível notar os

primeiros sinais de aprovação da ferramenta pelos futuros usuários do Departamento de Programação.

A quarta etapa do desenvolvimento consistiu na implementação dos demais requisitos funcionais considerados essenciais para que a ferramenta possa ser utilizada. Alterações como acréscimo de atributos nos cadastros de autores e obras exigiram pequenas modificações nos respectivos arquivos da camada de visão.

Já a tela a tarefa de localização de obras foi alterada para parametrizar as buscas através de autores, estilos, classificações, intérpretes, movimentos, intervalos de duração e de data da última execução.

Foram feitas muitas alterações nesta etapa. Os encontros realizados para validação da ferramenta serviram também para identificar e, conseqüentemente, implementar novas mudanças tais como troca de nomes de atributos e disposição das informações na tela.

Ao término dessa etapa, a ferramenta estava pronta para ser disponibilizada para os usuários a fim de começarem a testar as funcionalidades disponíveis.

5.5 Aplicação Web

Esta seção é dedicada a apresentar a aplicação disponibilizada para os usuários da Rádio da Universidade. A seção é dividida em subseções que visam cobrir as diferentes funcionalidades da ferramenta de forma a sinalizar as principais mudanças implementadas.

5.5.1 Login

A utilização da aplicação é somente autorizada a usuários autenticados. Portanto, ao acessar a ferramenta a tela de login é apresentada independente da requisição que é feita. Ou seja, mesmo que o usuário acesse através da URL de alguma funcionalidade válida dentro da aplicação, se ele não estiver autenticado, é automaticamente redirecionado para o login no sistema. A Figura 5.11 apresenta a tela de login com os campos de usuário e senha disponíveis para preenchimento. Só após a autenticação, as demais funcionalidades são apresentadas.

Figura 5.11 – Tela de login



Fonte: Elaborada pelo autor

5.5.2 Organização da Aplicação

A aplicação foi organizada de forma a agrupar as funcionalidades relacionadas aos processos da Rádio da Universidade. A Figura 5.12 mostra o menu principal na parte de cima tela com os links para as áreas que tratam do acervo e da programação. O exemplo ilustra ainda a tela que lista as obras cadastradas no sistema. Com isso, o menu secundário na parte lateral contém apenas os links dos outros cadastrados relacionados ao processo de manutenção do acervo.

Figura 5.12 – Layout do aplicação

Número da Obra	Título	Duração	Autor
0001-0	TESTE FICHA N 1, OP. 1, EM SOL MAIOR, PARA TECLADO DE MICRO E IMPRESSORA, DENOMINADA "ALELUIA"	00:01:50	LEOPOLD MOZART
0001-1	CONCERTO Nº 4 PARA PIANO E ORQUESTRA, EM SOL MAIOR, OP. 38	00:37:40	LUDWIG VAN BEETHOVEN
0003-1	UM RÉQUIEM ALEMÃO, PARA SOLISTAS, CORO E ORQUESTRA, SOBRE VERSÍCULOS DO VELHO E NOVO TESTAMENTO	01:16:23	JOHANNES BRAHMS
0004-2	CANÇÃO DAS PARCAS, OP. 89, PARA CORO E ORQUESTRA, SOBRE TEXTO DE GÖETHE	00:08:40	JOHANNES BRAHMS
0004-3	RAPSÓDIA PARA CONTRALTO, COROMASCULINO E ORQUESTRA, BASEADA EMFRAGMENTO DA "VIAGEM DE INVERNO PELOHARZ", DE GÖETHE.	00:14:24	JOHANNES BRAHMS
0004-4	NÂNIE, OP. 82, PARA CORO E ORQUESTRA, SOBRE TEXTO DE SCHILLER	00:14:15	JOHANNES BRAHMS
0004-5	A CANÇÃO DO DESTINO, OP. 54, PARA CORO E ORQUESTRA, SOBRE TEXTO DEFRIEDRICH	00:16:37	JOHANNES BRAHMS

Fonte: Elaborada pelo autor

5.5.3 Telas de Cadastro do Acervo

As telas de cadastros são simples e são, basicamente, formulários onde cada campo representa uma informação a respeito da entidade alvo do cadastro. Tomando como exemplo a atualização dos dados de uma obra específica (ver Figura 5.13), os atributos como título, nome do autor, duração da obra e identificação da mídia, entre outras informações, são disponibilizados ao usuário para modificação. Nota-se que os campos já aparecem preenchidos pois trata-se apenas de uma alteração. Quando o formulário de novo cadastro é requisitado, outra tela contendo o mesmo formulário é apresentada. Nesse caso, no entanto, os campos aparecem vazios e o usuário deve preenchê-los com as informações necessária.

Figura 5.13 – Alteração de cadastro Obras

Programação Musical - Rádio da Universidade 1080 AM

Acervo Programação Configurações Sair (admin)

Cadastros **Editar Obra**

Obras
Autores
Estilos
Classificações

Número da Obra *
0006-5

Título *
O DANÚBIO AZUL, VALSA PARA ORQUESTRA

CDs e Faixas

CD	Faixas
6	5

Duração
00:08:27

Autor *
JOHANN STRAUSS FILHO

Duração do Anúncio
00:00:15

Estilo
ROMÂNTICO

Intérpretes *
ORQUESTRA POP DE CINCINATTI, SOB A DIREÇÃO DE ERICH KUNZEL

Fonte: Elaborada pelo autor

As telas dos cadastros de autores (Figura 5.14), estilos (Figura 5.15) e classificações (Figura 5.16) são bastante semelhantes ao de obras. Todos estes serviços são acessados a partir de uma tela inicial onde é possível visualizar uma lista de registros armazenados e os botões para visualização, alteração e exclusão de registros são disponibilizados. Há ainda um botão da inclusão de novos registros próximo do menu secundário.

Figura 5.14 – Tela de cadastro de autores

Programação Musical - Rádio da Universidade 1080 AM

Acervo Programação Configurações Sair (admin)

Ações

Novo Autor

Cadastros

Obras

Autores

Estilos

Classificações

Autores

Nome do Autor	Observação	
FRANCESCO MANCINI	/ 1672-1737 /	
ARCANGELO CORELLI		
FRANCESCO MANCINI		
NICOLÒ PAGANINI		
MAURO GIULIANI		
GIOVANI PIERLUIGI DA PALESTRINA		
GIOVANNI BATTISTA PERGOLESI	/1710-1736/	
GIOACCHINO ROSSINI		
GIUSEPPE VERDI		
CARL PHILIPP EMANUEL BACH		

< Anterior 1 2 3 4 5 6 7 8 9 10 Próximo >

Powered by **Yii Framework**.

Fonte: Elaborada pelo autor

Figura 5.15 – Tela de cadastro de obras

Programação Musical - Rádio da Universidade 1080 AM

Acervo Programação Configurações Sair (admin)

Ações

Nova Estilo

Cadastros

Obras

Autores

Estilos

Classificações

Estilos

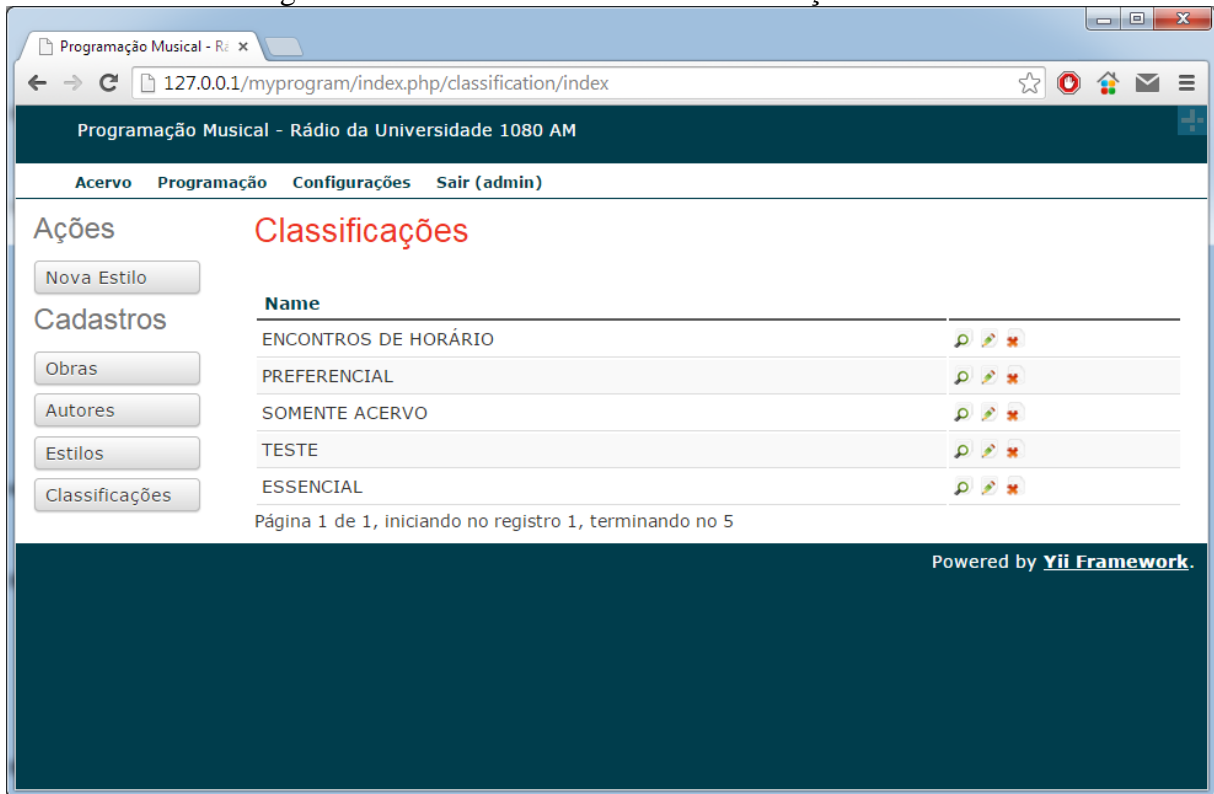
Name	
INDETERMINADO	
EM BRANCO	
ANTIGO	
CLÁSSICO	
BARROCO	
ROMÂNTICO	
MODERNO	
CONTEMPORÂNEO	

Página 1 de 1, iniciando no registro 1, terminando no 8

Powered by **Yii Framework**.

Fonte: Elaborada pelo autor

Figura 5.16 – Tela de cadastro de classificação de obras



Fonte: Elaborada pelo autor

5.5.4 Funcionalidade de Programação

A funcionalidade de programação contém os serviços de cadastro de gravações, tipos de gravações e períodos de programação. A Figura 5.17 apresenta a tela inicial do cadastro dos períodos de programação onde é possível observar os links para os serviços disponibilizados no menu secundário.

Figura 5.17 – Tela de períodos de programação

Programação Musical - Rádio da Universidade 1080 AM

Acervo Programação Configurações Sair (admin)

Ações **Períodos de Programação**

Novo Período

Programação

Gravações

Tipos

Períodos

Nome	Início	Fim	Status
Programação de 20/11/2014 a 27/11/2014	20/11/2014 07:00:00	27/11/2014 06:59:00	Rascunho
Programação de 24/10/2013 a 31/10/2013	24/10/2013 07:00:00	31/10/2013 06:59:00	Completo
Programação de 03/10/2013 a 10/10/2013	03/10/2013 07:00:00	10/10/2013 06:59:00	Completo
Programação de 31/10/2013 a 07/11/2013	31/10/2013 07:00:00	07/11/2013 06:59:00	Rascunho
Programação de 07/11/2013 a 08/11/2013	07/11/2013 07:00:00	08/11/2013 06:59:00	Rascunho
Programação de 19/12/2013 a 20/12/2013	19/12/2013 07:00:00	20/12/2013 06:59:00	Rascunho

Página 1 de 1, iniciando no registro 1, terminando no 6

Powered by **Yii Framework**.

Fonte: Elaborada pelo autor

Ao acessar um período de programação específico, é possível listar a grade de programação gerada automaticamente a partir das gravações configuradas previamente. A Figura 5.18 mostra a grade de programação após a seleção de um período cadastrado. Programas como o “Jornalismo 1080 10min” são inseridos na grade automaticamente pois este tipo de programa está configurado como de periodicidade diária, iniciando sempre às dez horas da manhã e com duração de dez minutos. Durante o processo de criação da grade, os intervalos entre um programa e outro são preenchidos como blocos musicais. O exemplo mostra a ocorrência desses blocos ao longo da grade.

Figura 5.18 – Grade programação de um período

Programação

Data	Horário	Duração	Nome	Status	
10/10/2013 Qui	07:00	02:00:00	Música	Programado	
10/10/2013 Qui	09:00	00:10:00	Jornalismo 1080 10min	Programado	
10/10/2013 Qui	09:10	00:50:00	Música	Agendado	
10/10/2013 Qui	10:00	00:10:00	Jornal da UFRGS 10 min	Programado	
10/10/2013 Qui	10:10	00:05:00	Toque de Arte	Programado	
10/10/2013 Qui	10:15	00:45:00	Música	Agendado	
10/10/2013 Qui	11:00	00:55:00	Entrevista Coletiva	Programado	
10/10/2013 Qui	11:55	01:05:00	Música	Agendado	
10/10/2013 Qui	13:00	00:20:00	Visão Social	Programado	
10/10/2013 Qui	13:20	00:40:00	Música	Agendado	
10/10/2013 Qui	14:00	00:15:00	UFRGS Entrevista	Programado	
10/10/2013 Qui	14:15	01:45:00	Música	Agendado	
10/10/2013 Qui	16:00	00:05:00	Literatura	Programado	
10/10/2013 Qui	16:05	00:55:00	Música	Agendado	
10/10/2013 Qui	17:00	00:10:00	Jornal da UFRGS 2ª Edição	Programado	

Fonte: Elaborada pelo autor

O processo de programação segue com a edição dos blocos musicais. Para isso cada bloco deve ser montado individualmente pelo programador da rádio. A Figura 5.19 mostra a lista de obras de um bloco em edição. É possível observar as obras já inseridas na lista e botões para alterar a ordem e excluir a obra. Há informações sobre o tempo de execução e um botão para localizar e inserir novas obras.

Figura 5.19 – Montagem dos blocos musicais

Adicionar Obras

#	Código	Título	Autor	Duração	Anúncio	Ordenar	
1	0001-0	N 1, OP. 1, EM SOL MAIOR, PARA TECLADO DE MICRO E IMPRESSORA, DENOMINADA "ALELUIA"	LEOPOLD MOZART	00:01:50	2 x 00:01:01		
2	1872-6	DOZE VARIAÇÕES SOBRE UMA ÁRIA DE "PAPAGENO", DA ÓPERA A FLAUTA MÁGICA DE MOZART, PARA VIOLONCELO E PIANO, OP. 66	LUDWIG VAN BEETHOVEN	00:09:55	2 x 00:00:22		
Tempo Total						00:14:31	

Página 1 de 1

Salvar

Fonte: Elaborada pelo autor

Já a seleção das obras para inclusão em um bloco é feita através de um formulário de busca como ilustrado na Figura 5.20. É possível combinar buscas complexas a fim de localizar a obra desejada.

Figura 5.20 – Busca e seleção de obras

Título Número da Obra

Autor

Estilo Classificação

Duração Maior Que Duração Menor Que

Número da Obra	Título	Nome do Autor	Duração	Duração do Anúncio	
0001-0	TESTE FICHA N 1, OP. 1, EM SOL MAIOR, PARA TECLADO DE MICRO E IMPRESSORA, DENOMINADA "ALELUIA"	LEOPOLD MOZART	00:01:50	00:01:01	Selecionar

Página 1 de 1

Fonte: Elaborada pelo autor

Já o cadastro de gravações e tipos de gravações se comportam de forma similar ao já mencionados anteriormente. A Figura 5.21 mostra uma parte da lista de gravações registradas.

Figura 5.21 – Cadastro de gravações

Gravações

Nome	Tipo de Duração	Duração	Recorrência	Dias da Semana	Hora de Início	Status	Tipo de Gravação	
Boletim Astronômico	Fixo	00:03:00	Semanal	Dom,Seg,Ter,Qua,Qui,Sex,Sáb	00:00:00	Ativo	Programa	 
Jornalismo 1080 10min	Fixo	00:10:00	Semanal	Seg,Ter,Qua,Qui,Sex	09:00:00	Ativo	Programa	 
Jornal da UFRGS 10 min	Fixo	00:10:00	Semanal	Seg,Ter,Qua,Qui,Sex	10:00:00	Ativo	Programa	 
Toque de Arte	Fixo	00:05:00	Semanal	Seg,Ter,Qua,Qui,Sex	10:10:00	Ativo	Programa	 
Entrevista Coletiva	Fixo	00:55:00	Semanal	Qui	11:00:00	Ativo	Programa	 
Fronteiras da Ciência	Fixo	00:30:00	Semanal	Seg	13:00:00	Ativo	Programa	 
ADUFRGS no Ar	Fixo	00:15:00	Semanal	Ter	13:00:00	Ativo	Programa	 
Sintonia da Terra	Fixo	00:20:00	Semanal	Qua	13:00:00	Ativo	Programa	 

Fonte: Elaborada pelo autor

6 RESULTADOS E TESTE DE USABILIDADE

Este capítulo é dedicado a apresentar os resultados do desenvolvimento do projeto e propor um teste de usabilidade a ser aplicado aos futuros usuários da aplicação.

Os requisitos detectados tanto nesse projeto quanto no anterior foram compilados em duas tabelas. A Tabela 6.1 a lista de requisitos funcionais validados, a classificação se o requisito foi detectado no projeto anterior ou somente no trabalho atual e se foi implementado na versão mais recente da aplicação.

Tabela 6.1 – Implementação do requisitos funcionais

<i>Requisitos Funcionais</i>	<i>Requisito detectado somente neste projeto</i>	<i>Implementado</i>
Cadastro de obras do acervo	Não	Sim
Cadastro de programação da rádio	Não	Sim
Montagem de blocos musicais	Não	Sim
Tela de consulta às grades de programação	Não	Sim
Registro de data de última execução de cada obra	Não	Sim
Relatório de divulgação da grade para os ouvintes	Não	Sim
Localização de obras por estilo, classificação, duração e autor	Sim	Sim
Lista de obras na consulta do cadastro do autor	Sim	Sim
Ordenação de lista de autores por sobrenome	Sim	Sim
Cadastro de usuário e definição de senhas	Sim	Sim
Troca ou recuperação de senha	Sim	Sim

Fonte: Elaborada pelo autor

A Tabela 6.2 é semelhante à anterior mas apresenta os requisitos não funcionais.

Tabela 6.2 – Implementação dos requisitos não funcionais

<i>Requisito Não Funcionais</i>	<i>Requisito detectado somente neste projeto</i>	<i>Implementado</i>
Utilização de software livre	Não	Sim
Sistema de fácil manutenção	Não	Sim
Reorganização da aplicação por processos	Sim	Sim
Reorganização do <i>layout</i> da aplicação	Sim	Sim
Mecanismos de segurança (autenticação)	Sim	Sim
Administração da aplicação (gerência de credenciais de acesso)	Sim	Sim

Fonte: Elaborada pelo autor

No entanto, o sucesso de uma aplicação como a deste projeto só pode ser medido através da visão dos usuários. Para isso é proposto a realização de um teste usabilidade.

O público alvo do teste são os três funcionários do Departamento de Programação da rádio. Durante o desenvolvimento do projeto foi possível identificar claramente dois perfis. O primeiro é composto pelo Jornalista Paulo Busato que executa as atividades de criação da programação da emissora há mais de 20 anos. É um usuário experiente e ajudou a consolidar as tarefas no formato de processo atual. O segundo grupo é formado pelos servidores Manoela Nogueira e Juliano Dupont. Eles começaram a trabalhar na rádio em 2014 e não possuem muita experiência em relação às atividades do departamento.

O teste é dividido em duas partes. Na primeira, há uma lista de tarefas que o usuário deve desempenhar utilizando a ferramenta e ao término delas, informar se conseguiu realizá-las e se foi fácil ou difícil. A lista a seguir contém sete tarefas sugeridas para avaliação:

- Identificar e acessar as funcionalidades de manutenção do acervo e programação musical;
- Cadastrar uma nova obra no acervo;
- Cadastrar um novo autor;
- Localizar e visualizar as informações de uma obra com duração entre 10 e 12 minutos;
- Criar uma grande de programação;
- Criar um bloco musical com obras de um único autor;
- Gerar um relatório de locução de obras para um dia de programação.

Junto às tarefas o usuário deve responder uma pergunta simples com o objetivo de avaliar o seu comportamento. Para isso, cada tarefa deve ser seguida da pergunta “Você conseguiu realizar essa tarefa?”, cuja resposta no formato de múltipla escolha deve ser selecionada da lista a seguir:

- Sim e foi fácil;
- Sim e foi difícil;
- Não consegui realizar.

A segunda parte do teste consiste em um grupo questões discursivas que avaliam a percepção do usuário com a ferramenta. As seguintes perguntas devem ser feitas:

- A aplicação está de acordo com o processo de manutenção do acervo da rádio?
- A aplicação está de acordo com o processo de programação da rádio?
- As informações dispostas nas telas foram suficientes para realizar as tarefas?
- As expectativas em relação ao comportamento da aplicação foram atendidas?
- O que você mudaria na aplicação?

As duas partes do teste devem ser feitas em conjunto por cada usuário, pois as tarefas propostas na primeira etapa servem como base para as questões da fase seguinte.

Após a aplicação dos testes é necessário realizar uma análise para detectar se a aplicação cumpre com a sua proposta inicial. Caso o resultado seja negativo, é possível também detectar os pontos onde a realização de projeto falhou.

7 CONCLUSÃO

Este trabalho teve como proposta continuar o desenvolvimento de uma ferramenta web capaz de auxiliar a programação musical da Rádio da Universidade. O projeto foi iniciado pelo Érico Martellet Marcant em 2013 como parte do seu trabalho de conclusão de curso de Ciências da Computação.

O trabalho iniciou com a revisão do projeto do anterior. Foi importante detectar a metodologia adotada, os requisitos encontrados, a arquitetura utilizada e a implementação feita. Como base nisso e nas sugestões de futuros trabalhos, uma nova metodologia foi empregada para o projeto da ferramenta. Foi adotado um modelo de projeto de software inspirado no modelo em cascata e no modelo em espiral para separar as etapas do processo com validações sucessivas com os usuários do Departamento de Programação.

Durante a análise de requisitos foram aplicadas diferentes técnicas como reuniões, prototipagem e criação de diagramas para a identificação dos requisitos.

Em relação à arquitetura do projeto, decidiu-se pela utilização de novo framework motivada pela decisão de alinhar a ferramenta com os demais sistemas institucionais da UFRGS disponibilizados pelo CPD. Espera-se com isso, uma futura integração entre esses ambientes.

A implementação foi feita seguindo todas as recomendações do guia de programação do framework. Foram utilizados os padrões de projetos mais adequados para resolução dos problemas, o que tornou o desenvolvimento mais eficiente. Todos os requisitos planejados foram implementados na versão final da ferramenta. Para encerramento deste projeto, foi elaborado um teste de usabilidade a ser aplicado aos usuários da aplicação.

Como trabalhos futuros, entende-se que a primeira tarefa a ser realizada é aplicação do teste de usabilidade proposto e a análise dos resultados. A integração da aplicação com os sistemas institucionais e a utilização da infraestrutura do CPD é recomendada para a disponibilização da ferramenta para os usuários. Além disso, sugere-se a implementação da funcionalidade de criação automatizada de grades de programação. Este recurso está presente nos sistemas similares disponíveis no mercado e poderia representar um grande avanço para a qualidade da programação musical da Rádio da Universidade.

REFERÊNCIAS

ACCESS. **Transmissão MAXXI**. Disponível em <<http://www.accessweb.com.br/Sistema-de-Gestao-Esportiva/modulo-discotecario.html>>. Acesso em dezembro de 2014.

BOEHM, B. W. **A Spiral Model of Software Development and Enhancement**. [S.I.]: 1988. Disponível em <<http://csse.usc.edu/csse/TECHRPTS/1988/usccse88-500/usccse88-500.pdf>>. Acesso em dezembro de 2014.

BURBECK, S. **Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)**. [S.I.]: 1987. Disponível em <<http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>>. Acesso em dezembro de 2014.

CADENA Sistemas. **Cadena Musical**. Disponível em <<http://www.cadena.com.br/brazilian/solucoes/cadena-musical>>. Acesso em dezembro de 2014.

FOWLER, M. **Pattern of Enterprise Application Architecture**. [S.I.]: Addison-Wesley, 2003.

HENNESSY, J. L.; PATTERSON, D. A. **Architecture: A Quantitative Approach**. 5 ed. [S.I.]: Elsevier, 2012. 493 p.

INFORMA Consultoria e Sistemas. **InfoDISC – Programação Musical**. Disponível em <<http://informabr.com/infodisc>>. Acesso em dezembro de 2014.

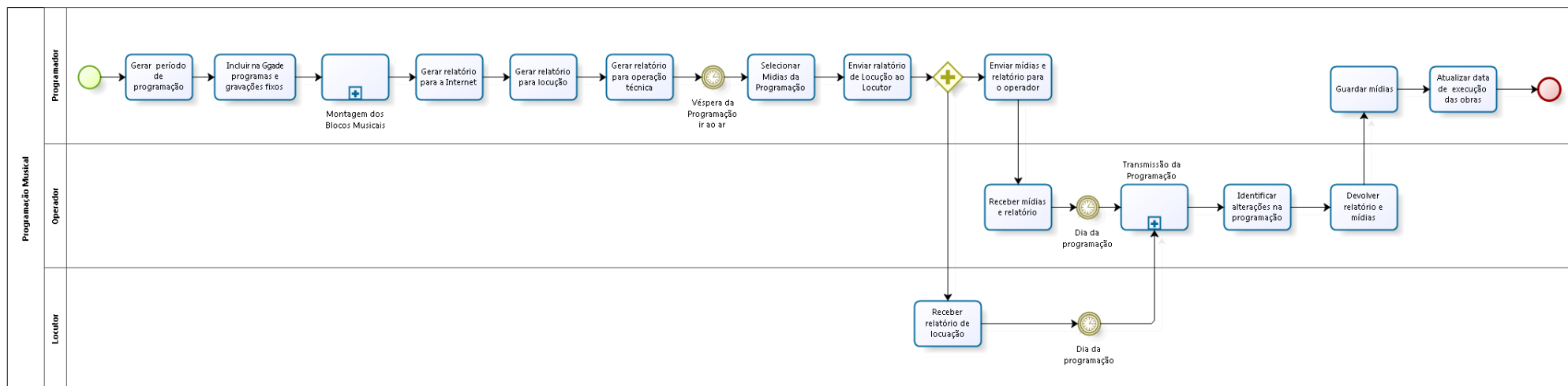
MARCANT, E. M. **Ferramenta Web para Programação da Rádio da Universidade**. 2013. 57 f. Trabalho de Graduação (Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

PLAYLIST Software Solutions. **Playlist Maker 2012**. Disponível em <http://www.playlist.com.br/en/prod_playlistmaker2012.asp>. Acesso em dezembro de 2014.

ROYCE, Winston. **Managing the Development of Large Software Systems**. Proceedings of IEEE WESCON 26, August 1970.

SOMMERVILLE, I. **Software Engineering**. [S.I.]: Person Education, 2011.

APÊNDICE A – PROCESSO DE PROGRAMAÇÃO MUSICAL



APÊNDICE B – SUBPROCESSO DE MONTAGEM DOS BLOCOS MUSICAIS

