Documentação NerdStore RN

O sistema foi desenvolvido para fazer um cadastro e uma listagem simples de três tipos de objetos Livros, Revistas e Filmes em geral.

OBS: Todo código digitado por mim do sistema tirando a parte do Framework está comentada.

O sistema foi desenvolvido em PHP com Framework Adianti com banco de dados em MYSQL utilizando o MYSQL Workbench

Falando um pouco sobre o Adianti

“Adianti Framework para PHP é uma arquitetura open-source baseada em padrões de projeto, voltada para a criação ágil de aplicações de negócio em PHP. Busca reduzir os custos de desenvolvimento por oferecer componentes de alto nível para criação de formulários, datagrids, relatórios e outros. ”

- Voltado para o desenvolvimento de sistemas;

- Oferece componentes de alto nível;

- Manipulação simplificada do banco de dados;

- Templates para criação de sistemas;

- Controle de login, permissão de acesso, logs.

Site do Framework: <http://www.adianti.com.br/framework>

Falando um pouco sobre o MYSQL:

“O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo. ”

O principal motivo pela escolha dessa linguagem para o sistema foi a fácil integração com o PHP e a configuração com o Adianti que era simples e não necessita de muitos passos.

Um pouco da ideia de usar o Adianti para criação do sistema.

O principal motivo do uso do Adianti para criação do software foi a facilidade de uso de manuais com tutoriais e principalmente o conteúdo que é disponibilizado no próprio site do criador que mostra quase todas as funções de como criar datagrid, cadastros e manipulação de dados dentro de uma tela e mostra também a previsão desses dados mesmo sendo FREE tivemos a oportunidade de ver o livro do criador Pablo Dall'Oglio que por sinal me ajudou a tirar algumas duvidas na parte de integração do banco de dados e também para criação de datagrids específicos dentro do sistema, tem também algumas videoaulas que ele mostra como é o funcionamento do framework bem detalhadamente com detalhes e também fala um pouco sobre novas funcionalidades do mesmo em seu canal no youtube. O Framework também possuí um editor exclusivo com criação dos projetos prontos com templates já inclusos deixando ao usuário a escritura do código mais rápida necessitando apenas utilizar os recursos “quase prontos” do framework e implementar com a sua ideia.

Falando sobre o software criado e suas partes.

O sistema é basicamente composto por uma tela de apresentação 3 telas de cadastro e 3 listagens de cada tipo de cadastro. Vou relatar como o software foi escrito detalhadamente com qual template foi utilizado e como foi feito toda configuração e escritura do código.

1. A criação do projeto para o sistema poder existir foi feito com o Adianti Studio com a criação de um novo projeto sem banco de dados específicos ou seja eu iria deixar para fazer a configuração do banco de dados manual através de um arquivo que fica dentro do projeto que é responsável por essa integração Software X Banco de dados.
2. Após feito a criação do projeto era necessário fazer a configuração dos parâmetros de como o banco de dados iria funcionar ou seja o banco foi criado antes mesmo de o software ser programado para não haver erros na implementação do sistema. Então fizemos a criação do banco de dados da seguinte forma.
3. CREATE TABLE `filme` (
4. `id` int(11) NOT NULL,
5. `titulo` text,
6. `diretor` text,
7. `id\_suporte` int(11) DEFAULT NULL,
8. `id\_genero` int(11) DEFAULT NULL,
9. `dt\_lcto` date DEFAULT NULL,
10. `duracao` int(11) DEFAULT NULL,
11. PRIMARY KEY (`id`)
12. );
13. CREATE TABLE `revistas` (
14. `id` int(11) NOT NULL,
15. `titulo` text,
16. `editora` text,
17. `id\_genero` int(11) DEFAULT NULL,
18. `dt\_lcto` date DEFAULT NULL,
19. `edicao` int(11) DEFAULT NULL,
20. PRIMARY KEY (`id`)
21. );
22. CREATE TABLE `livro` (
23. `id` int(11) NOT NULL,
24. `titulo` text,
25. `editora` text,
26. `genero` text,
27. `isbn` int(11) NOT NULL,
28. `dt\_lcto` date DEFAULT NULL,
29. PRIMARY KEY (`id`)
30. );

O arquivo do banco de dados se encontra na raiz do diretório com nome **cruddatabase.sql**

3- Após isso fizemos a configuração dos parâmetros de ligação do banco de dados com o projeto que no caso deve ser feito através de um arquivo que fica dentro da pasta sample.ini dentro do diretorio app/config que é um arquivo de configuração para conexão com o banco de dados caso queira executar o projeto na rede local ou numa rede externa isso vai depender da sua configuração:

host = localhost

name = crud

user = root

pass =

type = mysql

4-Quando terminado de configurar o banco de dados com o sistema e também fazer a criação do banco com as tabelas dentro do sistema chega a hora de criarmos os atributos que serão jogados como dados para dentro da tabela para isso foi criado um arquivo para cada tipo de objeto no caso Revista.Class, Filme.Class, e Livro.Class que utilizam uma classe TRecord que fica responsável por fazer essa função de colocar os dados dentro da tabela. Esses arquivos são encontrados dentro da pasta “model” do projeto.

5-Tela de cadastro: Na tela de cadastro é utilizado uma espécie de formulário para preenchimento dos dados para fazer o cadastro do objeto (seja ele livro, revista ou filme) cada um deles possuem uma forma diferente de cadastro.

No filme ele possui variáveis como id, título, diretor, suporte (no caso DVD ou blu-ray), gênero, data de lançamento e duração. Na parte de preenchimento dessas informações foi utilizado um quickform que seria um formulário simples para preenchimento de dados no final desse formulário foi colocado um botão de salvar e listar cada um com um uma função diferente:

Salvar: Faz a conexão com o banco de dados e a inserção do dado dentro da tabela filmes.

Listar: Ele leva a página de listar os filmes já cadastrados.

No livro ele possui basicamente a mesma ideia do cadastro dos filmes, porém com variáveis diferentes como editora, ISBN o restante do formulário é o mesmo dos outros

Na revista segue a mesma ideia dos outros dois cadastros.

Cada cadastro como relatado possuí um arquivo separado dentro da pasta os arquivos são os seguintes:

FilmeForm.Class = responsável pela parte de cadastro dos filmes.

LivroForm.Class = responsável pela parte de cadastro dos livros.

RevistaForm.Class = responsável pela parte de cadastro das revistas.

No final de cada cadastro eles fazem uma conexão diferente com cada tabela respectiva, ou seja, nenhuma tabela tem uma ligação direta uma com a outra devido a serem cadastro diferentes.

6- Tela de listagem de dados:

Nessa tela foi feito um datagrid para cada tipo de objeto (seja ele livro, revista ou filme) vou citar como foi projetado as três listagens:

Foi utilizado um QuickDatagrid com a função QuickColumn dentro da página esse método de datagrid não permite com que seja ordenado as informações ou seja se quiser que seja ordenado por nome ou por data de lançamento ou por gênero e etc. nos registros das listagens é possível que o dado seja editado ou pagado dependendo do desejo do usuário quando clicado em editar ele abre a tela de cadastro do objeto (seja ele livro, revista ou filme) com os dados preenchidos para serem alterado ele utiliza um método “on reload” para executar esse processo de preenchimento e dentro da listagem ele faz uma busca dentro das tabelas para serem mostrado os dados dentro das columns do datagrid.

Os arquivos dessas telas ficam dentro da pasta app/control:

RevistaList.Class Listagem de revistas.

LivroList.Class Listagem de livros.

FilmeList.Class Listagem de filmes.

OBS: é um pouco complicado descrever essa tela dentro de apenas um pequeno resumo é importante que esteja com o código aberto que também possui comentários para melhor entendimento do usuário.

7- Tela de apresentação e arquivos auxiliares.

Na tela de apresentação eu fiz um pequeno resumo de com o projeto funciona todo ele HTML pois possui apenas texto e hipertexto, nessa tela consta informações básicas como os integrantes do grupo um pouco sobre o modelo de 4 camadas os integrantes do grupo como foi criado as tabelas do sistema e também o parâmetro de configuração do sistema com o banco de dados.

Arquivos auxiliares

Arquivos auxiliares seriam basicamente arquivos que foram utilizados para melhor customização da parte de interface do projeto como arquivos de ícones arquivos dos menus e a parte de templates:

O template utilizado foi o template 1 do Adianti, porém modificado em alguns aspectos como os logos do adiante e os links que faziam parte desses logos que no caso são botões

Na parte de menus do sistema ele possui um arquivo chamado menu.xml na pasta raiz do diretório que fica responsável por editar a parte de menus e dos link’s do mesmo.

A parte de ícones do Adianti foram utilizados os ícones padrão do sistema que no caso todos vem incluso dentro do projeto não sendo necessário colocar algum externo mas caso necessário por opção do usuário pode ser adicionado outros ícones baixados e colocados na pasta app/images.

Programas utilizados para fazer o programa funcionar:

Xampp = responsável por fazer a conexão PHP com banco de dados e rodar o sistema no navegador.

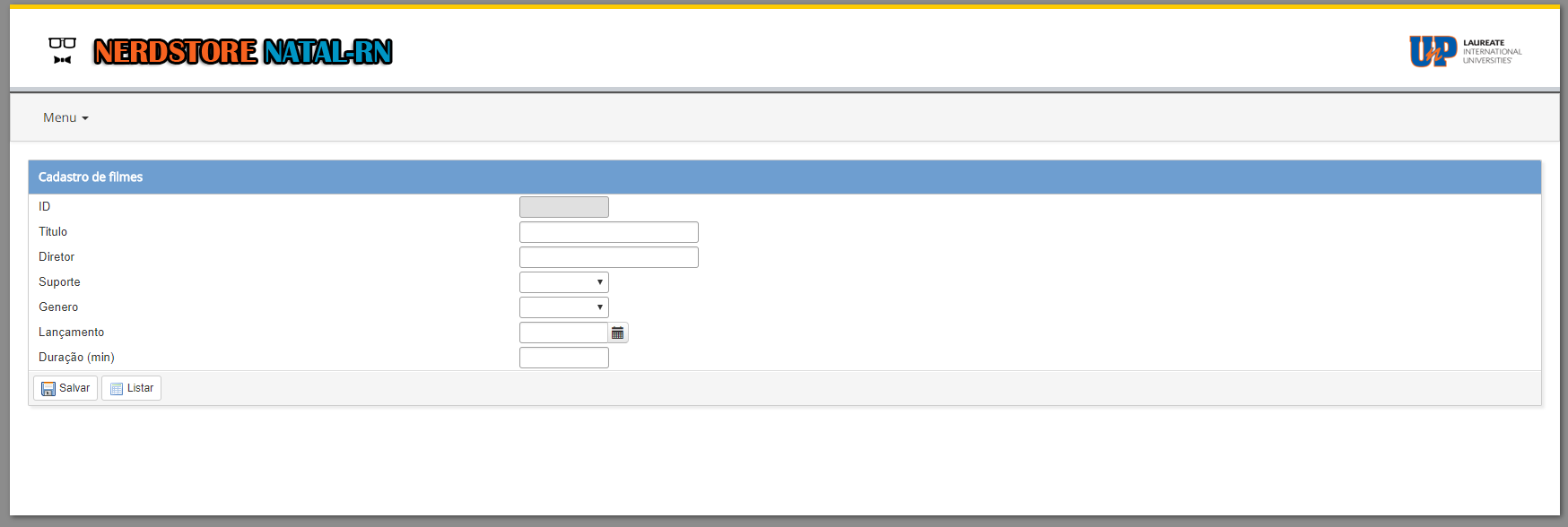
MYSQL Workbench = SGBD para criação e manipulação das tabelas do sistema.

Adianti Studio = para edição e criação do sistema NerdStore.

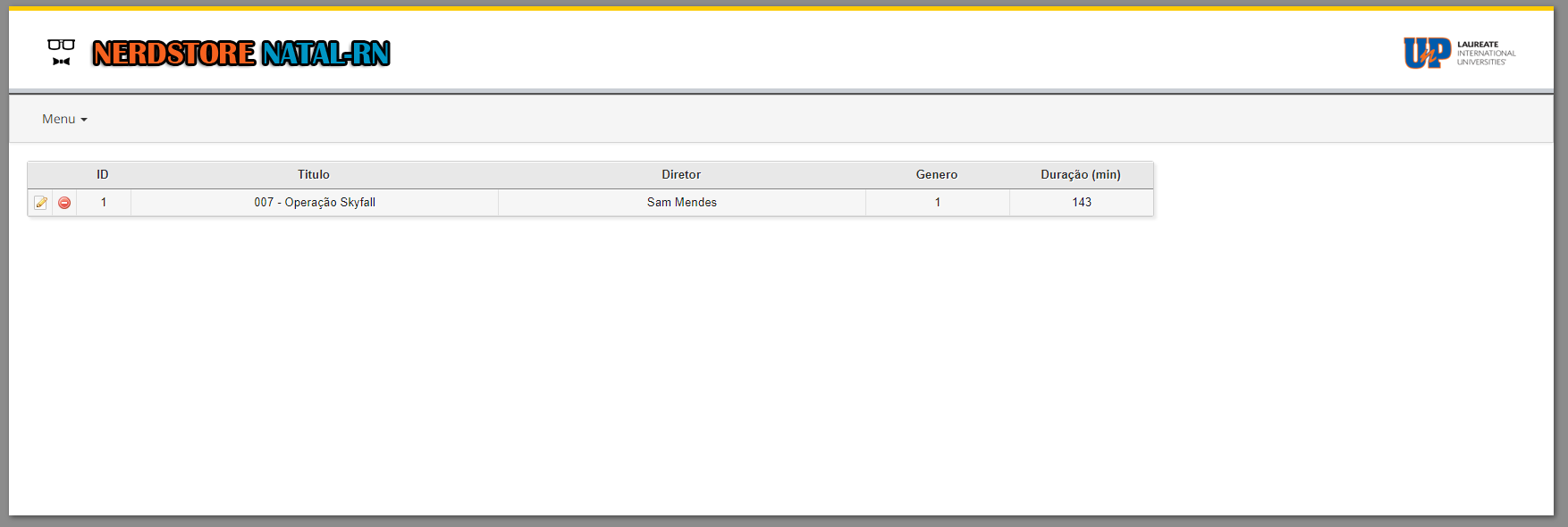
Navegador de internet = para visualização do projeto e verificar o funcionamento do mesmo. (Mozilla Firefox, Google Chorme, Internet Explorer, Opera, NetScape, IceWeasel, etc).

Telas do sistema

Tela de cadastro:



Tela de listagem:



Tela de apresentação:

