



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

WESLEY DANIEL SILVA ELIZIARIO

FRAMEWORK GERADOR DE RELATÓRIOS

Assis
2014

Av. Getúlio Vargas, 1200 – Vila Nova Santana – Assis – SP – 19807-634 Fone/Fax:
(0XX18) 3302 1055 homepage: www.fema.edu.br

WESLEY DANIEL SILVA ELIZIARIO

FRAMEWORK GERADOR DE RELATÓRIOS

Trabalho de conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do curso de Análise e Desenvolvimento de Sistemas.

Orientado: Wesley Daniel Silva Elizario

Orientador: Prof. Dr. Luiz Carlos Begosso

Assis
2014

FICHA CATALOGRÁFICA

ELIZIARIO, Wesley D. S.

Framework Gerador de Relatórios / Wesley Daniel Silva Elizario. Fundação Educacional do Município de Assis – FEMA – Assis,2014.

48p.

Orientador: Prof. Dr. Luiz Carlos Begosso.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Framework. 2. Java. 3. Relatório.

CDD: 001.61
Biblioteca da FEMA

FRAMEWORK GERADOR DE RELATÓRIOS

WESLEY DANIEL SILVA ELIZIARIO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, com o requisito do curso de graduação em Análise e Desenvolvimento de Sistemas, analisado pela seguinte comissão examinadora:

Orientador: _____

Analisador (1): _____

Assis
2014

DEDICATÓRIA

Dedico este trabalho primeiramente ao meu Deus, que por sua graça, misericórdia e seu incomparável amor, me deu condições, forças e sabedoria para prosseguir até o fim dessa jornada. Dedico também a minha família, aos meus amigos, ao meu orientador e a todos que me apoiaram direta e indiretamente nestes três anos.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela oportunidade e condições que me concedeu de concretizar a realização, de mais um desejo do meu coração, a minha graduação.

Aos meus pais, por sempre me apoiarem em meus estudos.

Aos amigos de sala de aula, que sempre compartilhamos conhecimentos durante o período de graduação, e ao amigo Eduardo Nicolini Sodré da Silva, pelo incentivo e ajuda desde o início dessa trajetória.

A todos os professores que compartilharam de seus conhecimentos, suas experiências, e pelo aprendizado proporcionado. E em especial ao meu orientador Prof. Dr. Luiz Carlos Begosso, pelo apoio e contribuição para realização deste trabalho.

RESUMO

O presente trabalho refere-se ao desenvolvimento de um framework que agiliza o processo de criação de relatórios, em ambiente desktop, utilizando telas de filtros para realização das consultas ao banco de dados.

O framework é desenvolvido para ambiente desktop, codificado com a linguagem de programação JAVA SE e para banco de dados MySQL.

Nesse sentido, o trabalho inicia-se com uma revisão da bibliografia acerca dos conceitos, padrões e metodologias das linguagens de programação e ferramentas utilizadas para o seu desenvolvimento.

Palavras-chave: Framework; Java; Relatório.

ABSTRACT

This present work refers to the development of a framework that speeds up the reports creation process in a desktop environment, using filter screens to search the database.

The framework itself is built for a desktop environment, coded with JAVA SE as the programming language and MySQL for the database.

This way, the present work begins with a bibliographic revision about the concepts, patterns and methodologies related to the programming languages and tools used in its development.

Keywords: Framework; Java; Report.

LISTA DE ILUSTRAÇÕES

Figura 1 - Esquema de Funcionamento do Framework	24
Figura 2 - Interface de seleção de campos	26
Figura 3 - Interface de propriedades	27
Figura 4 - Interface de seleção de filtros	28
Figura 5 - Interface de seleção de atributos para criação de grupos.....	29
Figura 6 - Interface de seleção de variáveis.....	30
Figura 7 - Exemplo de relatório de produtos	31
Figura 8 - Estrutura Analítica do Projeto	32
Figura 9 - Sequenciamento das Atividades	33
Figura 10 - Diagrama de Caso de Uso.....	35
Figura 11 - Diagrama de Classe.....	36
Figura 12 - Caso de Uso Carregar Relatório	37
Figura 13 - Caso de Uso Manter Campos	38
Figura 14 - Caso de Uso Manter Filtro	39
Figura 15 - Caso de Uso Manter Grupo	41
Figura 16 - Caso de Uso Configurar Relatório	41
Figura 17 - Caso de Uso Executar Relatório	42
Figura 18 - Caso de Uso Abrir Nova Tela	43
Figura 19 - Caso de Uso Salvar Relatório	44

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVOS	12
1.2 JUSTIFICATIVA	12
1.3 MOTIVAÇÃO	12
2 REVISÃO DA LEITURA	13
2.1 LINGUAGEM DE PROGRAMAÇÃO.....	13
2.2 ORIENTAÇÃO A OBJETIVOS.....	13
2.3 JAVA	14
2.4 SWING	15
2.5 XML.....	16
2.6 BANCO DE DADOS	16
2.7 REFLECTION.....	17
3 FRAMEWORK GERADOR DE RELATÓRIOS	18
3.1 FRAMEWORKS	18
3.2 FRAMEWORKS UTILIZADOS PARA O DESENVOLVIMENTO DO GERADOR DE RELATÓRIOS	19
3.2.1 Hibernate.....	19
3.2.2 Jasper Reports	19
3.2.3 JAXB.....	20
3.3 FERRAMENTAS UTILIZADAS COMO SUPORTE	13
3.3.1 Eclipse.....	21
3.3.2 Ireport.....	22
3.3.2 Astah Community.....	22
3.4 FUNCIONAMENTO GERAL DO FRAMEWORK	23
3.5 INTERFACES DO FRAMEWORK	26

4 PLANEJAMENTO DO PROJETO	32
4.1 ESTRUTURAS ANALÍTICAS DO PROJETO (WBS)	32
4.2 SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS	33
4.3 LISTA DE EVENTOS.....	34
4.4 CASO DE USO.....	35
4.5 DIAGRAMA DE CLASSE	36
4.6 ESPECIFICAÇÃO DE CASOS DE USO.....	36
4.6.1 Carregar Relatório	37
4.6.2 Manter Campos	37
4.6.3 Manter Filtro.....	39
4.6.4 Manter Grupo.....	40
4.6.5 Configurar Relatório.....	41
4.6.6 Manter Filtro.....	42
4.6.7 Abrir Nova Tela	43
4.6.8 Salvar Relatório	44
CONSIDERAÇÕES FINAIS	45
REFERÊNCIAS	46

1 INTRODUÇÃO

O conceito de relatório pode ser entendido como a exposição escrita na qual se descrevem fatos verificados mediante pesquisas ou se historia a execução de serviços ou de experiências. É geralmente acompanhado de documentos demonstrativos, tais como tabelas, gráficos, estatísticas e outros (UFPR, 1996). Por outro lado, UNISEPE (2010) define que relatório técnico-científico é o documento original pelo qual se faz a difusão da informação corrente, sendo ainda o registro permanente das informações obtidas. É elaborado principalmente para descrever experiências, investigações, processos, métodos e análises.

IBM (2013) define relatório computacional como um conjunto de dados, com finalidade específica, que pode ser coletado com determinada periodicidade e exibido instantaneamente ou armazenado para posterior exibição. Em um sistema, mesmo que não trabalhe com uma massa muito grande de dados, muitas informações são manipuladas para a geração de relatórios que auxiliarão pessoas em suas tomadas de decisão. Os relatórios são criados por vários objetivos, a fim de organizar esses dados para uma melhor visualização, análise das informações, etc. Porém, criar relatórios não é tão simples e demanda tempo.

No dia-a-dia desenvolvedores, analistas, gerentes de projetos entre outros, se deparam com usuários que constantemente precisam de diferentes relatórios para controle de movimentações em geral, análise, tomada de decisões, etc. Sendo assim, é necessário que possuam uma ferramenta adequada para agilizar, exigindo o mínimo possível de implementação de código, o processo de criação desses relatórios.

Visando a rapidez na criação e integração de relatórios em sistemas o Framework Gerador de Relatórios, proposto neste trabalho, cria automaticamente: as interfaces gráficas de filtros e os comandos necessárias para consultas em banco de dados

(*select*, *inner join*, *order by* e outros), e o relatório de acordo com as configurações salvas em arquivo xml e jrxml.

1.1 OBJETIVOS

O presente trabalho tem por objetivo o desenvolvimento de um framework que agiliza o processo de criação de relatórios gerenciais. Espera-se que o Framework Gerador de Relatórios, como é denominado, possa contribuir com o processo de desenvolvimento de software favorecendo a criação de relatórios e sua respectiva integração com outros sistemas dispensando o desenvolvimento de telas e implementações de códigos, facilitando o trabalho de profissionais que atuam na área de análise, desenvolvimento, suporte entre outros.

1.2 JUSTIFICATIVA

A importância de desenvolvimento deste framework se dá por atender a necessidade de usuários e desenvolvedores de gerarem diferentes relatórios de forma fácil, rápida e exigindo a mínima implementação de linha de código.

1.3 MOTIVAÇÃO

A principal motivação para o desenvolvimento desse aplicativo se dá pelo fato de ser uma tecnologia que ajuda no tempo de desenvolvimento de tarefas básicas, como montar telas de filtro, desenvolvimento de linhas de códigos SQL (Structured Query Language) e facilitar a integração do relatório a outro sistema.

2 REVISÃO DA LITERATURA

2.1 LINGUAGEM DE PROGRAMAÇÃO

Os programadores teriam seu trabalho facilitado se os programas fossem escritos em sentenças padronizadas em linguagem humana. Infelizmente, isso não acontece. Os programas têm de ser escritos em uma linguagem de programação, e há uma grande variedade dessas linguagens.

Segundo QUE (1993), as linguagens de programação são divididas de modo geral em duas categorias: aquelas denominadas de baixo nível e as de alto nível. Uma linguagem de programação de baixo nível representa diretamente as instruções em código de máquina e é projetada de modo a ser “compreendida” pelo computador, em lugar de ser entendida pelas pessoas. Um exemplo desse tipo de linguagem é Assembler.

Por outro lado, uma linguagem de alto nível é elaborada para ser mais facilmente compreendida pelas pessoas. As palavras-chaves e funções de uma linguagem de alto nível como C, Delphi, Java entre outros, são em geral muito semelhantes a uma sentença em língua inglesa e são usualmente mais fáceis de aprender, ler e entender. No entanto, as linguagens de alto nível têm de ser, por via de regra, interpretadas e traduzidas em código de máquina de baixo nível. (QUE,1993).

2.2 ORIENTAÇÃO A OBJETOS

O paradigma de programação que utilizada no desenvolvimento deste trabalho foi a programação orientada a objetos. A programação orientada a objetos é um tipo de programação que destaca os objetos do mundo real. Pode-se dizer que é a

abstração da regra de negócio para o mundo real, onde um objeto tem as suas características assim como os objetos do mundo real tem as suas.

SILVA (2011) descreve que na programação orientada a objetos, implementa-se um conjunto de classes que definem os objetos presentes no sistema de software. Cada classe determina o comportamento (definidos nos métodos) e estados possíveis (atributos) de seus objetos, assim, como o relacionamento com outros objetos. Smalltalk, Perl, Python, Ruby, Php, C++, Java e C# são as linguagens de programação mais importantes com suporte à orientação a objetos. Neste trabalho será abordado a linguagem Java.

2.3 JAVA

Java é uma linguagem de programação orientada a objetos e uma plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. É a tecnologia que capacita muitos programas da mais alta qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros, por exemplo. (JAVA, 2014).

Java é uma linguagem de programação de multiplataforma, ou seja, é executado em qualquer sistema operacional aonde se tenha instalado um interpretador denominado máquina virtual JAVA (JVM). JVM é um programa que converte o código Java em comandos que o sistema operacional possa executar.

O Java utiliza um conceito diferente. Ao invés de gerar um código binário diferente para cada plataforma, é gerado um binário que pode ser executado em qualquer plataforma, dentro de uma máquina virtual. Este código binário "universal" é chamado de *bytecode*.

Como Java é uma linguagem que não se prende a uma plataforma, ou a um sistema operacional, ele é utilizado em vários ambientes de desenvolvimento. As plataformas e seus respectivos ambientes de desenvolvimento mais conhecidos são:

- Java SE (*Standard Edition*): plataforma que permite desenvolver e implementar aplicativos Java para desktops;
- Java EE (*Enterprise Edition*): plataforma voltada para desenvolvimento de aplicações de grande porte, onde a execução é realizada em servidores, como aplicações para Web;
- Java ME (*Micro Edition*): plataforma que fornece um ambiente robusto e flexível para aplicativos em execução no incorporado e móvel: microcontroladores, sensores, gateways, telefones celulares, TV set-top boxes, impressoras, etc.

2.4 SWING

Mesmo que Java seja uma linguagem independente de sistemas ainda há a possibilidade de criar programas que dependam de determinado sistema operacional. Assim, no começo da linguagem Java existiam bibliotecas gráficas que executavam somente em alguns sistemas operacionais.

Com o propósito de criar uma biblioteca gráfica completa para o Java, surgiu a API (*Application Programming Interface*) Swing que, conta com um rico conjunto de componentes para tornar as aplicações mais ricas e interativas. O Swing permite criar sistemas multiplataforma, internacionalizáveis, com aparência nativa ou customizada, extensível ao ponto de se poder criar novos componentes e integrá-los à sua aplicação sem dificuldades, totalmente orientado a objetos, e que utiliza padrões de projeto em seu código-fonte. (CAVALÉRO, 2014).

2.5 XML

XML (*Extensible Markup Language*) é uma linguagem de marcação que serve para guardar dados de uma forma estruturada. Essa estrutura é definida pelo próprio usuário ou por um esquema. Um XML é um arquivo de texto puro, portanto independente de plataforma, por isso é muito utilizado para transmitir dados entre diferentes aplicações e sistemas. (CAELUM, 2014).

2.6 BANCO DE DADOS

No início do desenvolvimento de software o armazenamento e acesso a dados eram ainda bastante rudimentares comparados a estrutura dos dias atuais. A maioria dos desenvolvedores armazenavam dados em arquivos de texto. Tais arquivos eram normalmente formados por campos de tamanho fixo, e o acesso a eles não requeria mais do que as operações de leitura e escrita em arquivos, sendo assim, havia grande dificuldade em alterar registros, estruturas ou simplesmente acessar alguns registros.

Com o objetivo de minimizar essa dificuldade surgiram os bancos de dados, estruturas com a capacidade de armazenar dados, controlar registros, controlar estrutura dos dados, sendo estes dados estruturados, que pode ser desde uma simples lista de compras a uma galeria de imagens ou a grande quantidade de informação de uma rede corporativa. Os computadores lidam muito bem com grandes quantidades de dados, e o gerenciamento de banco de dados funciona então como a engrenagem central da computação, seja como utilitários independentes ou como partes de outras aplicações.

2.7 REFLECTION

A API (*Application Programming Interface*) de reflexão é definida pela ORACLE (2014) como sendo a capacidade de um programa de examinar ou modificar o comportamento de aplicações em tempo de execução na *Java Virtual Machine*.

A reflexão é uma ferramenta poderosa, que fornece ao desenvolvedor algumas capacidades extremamente interessantes, como, por exemplo:

- Instanciar objetos cujas classes só são conhecidas em tempo de execução: é possível usar essa facilidade para aplicar o conceito de programar para interface e não para a implementação, permitindo a passagem por parâmetro a classe que efetivamente implementa uma interface;
- Criar *arrays* cujo tamanho e classe só conheceremos em tempo de execução;
- Analisar informações sobre uma classe para implementar restrições de segurança: usando a API de reflexão podemos determinar, por exemplo, o pacote a que uma classe pertence, e permitir que somente classes chamadoras pertencente ao pacote X possam usar certas funcionalidades oferecidas pela classe Y.

3 FRAMEWORK GERADOR DE RELATÓRIOS

Como já citado anteriormente, o objetivo deste trabalho é o desenvolvimento de um framework que pretende agilizar o processo de criação de relatórios sem que haja perda de tempo na implementação, tanto na criação quanto na integração do relatório em um sistema.

O foco do framework está em oferecer um mecanismo automático, onde o desenvolvedor de sistemas não necessite codificar muitas linhas e oferecer todas as funcionalidades para o desenvolvedor construir relatórios e suas telas de filtros.

Para o desenvolvimento do framework foi escolhida a linguagem de programação Java com uma arquitetura em três camadas, obedecendo ao padrão MVC de desenvolvimento de aplicações, utilizando-se de outros frameworks e ferramentas de código aberto que serão abordadas nos tópicos seguintes.

3.1 FRAMEWORKS

Um framework é uma aplicação quase completa, mas com a ausência de algumas partes. Criar uma aplicação sob um framework consiste em prover os pedaços que são específicos para a aplicação. O Framework Gerador de Relatórios é um exemplo de uma aplicação inacabada, desenvolvido com o objetivo de oferecer, a um sistema, recursos necessários para criação de relatórios.

SILVA (2000) define framework como um conjunto de classes inter-relacionadas com o objetivo de facilitar o desenvolvimento de um determinado domínio de aplicação. Os frameworks são compostos de classes concretas e abstratas, sendo que as classes abstratas possuem implementações incompletas que devem ser estendidas para compor as classes completas da aplicação final. A motivação para o

desenvolvimento de frameworks é a reutilização de código e também de projeto, com o intuito de aumentar a produtividade no desenvolvimento de softwares.

Segundo WIRFS-BROCK e JOHNSON (1990), um framework é utilizado através de configuração ou conexão de classes concretas e derivação de novas classes concretas a partir das classes abstratas do framework.

Diferente de uma biblioteca de classes, que por sua vez, cada classe é única e independente das outras, os frameworks provê um modelo de interação ou colaboração (*wired-in interconnections*) entre as instâncias de classes definidas pelo framework.

3.2 FRAMEWORKS UTILIZADOS PARA O DESENVOLVIMENTO DO GERADOR DE RELATÓRIOS

3.2.1 Hibernate

Bauer e King (2005) explicam que o Hibernate é um framework de persistência que tem como finalidade armazenar objetos Java em bases de dados relacionais ou fornecer uma visão orientada a objetos de dados relacionais existentes. Isso se dá porque o framework utiliza arquivos de configuração XML para fazer o mapeamento dos dados contidos nas colunas de uma tabela em uma base de dados relacional para os atributos de uma classe Java. A relação entre o banco de dados e os objetos pode ser feita também através de anotações.

3.2.2 Jasper Reports

JasperReports é um poderoso framework open-source para geração de relatórios e totalmente escrito em Java. Seu principal objetivo é ajudar a criar documentos

prontos para impressão de uma forma simples e flexível. Entre as funcionalidades do JasperReports pode-se destacar:

- Exporta relatórios para diversos formatos diferentes, tais como PDF, HTML, XML, XLS, etc;
- Aceita diversas formas de entrada de dados, tais como um arquivo XML ou CSV, conexão com o banco de dados, uma sessão do Hibernate, uma coleção de objetos em memória, etc;
- Permite o uso de diagramas, gráficos e até códigos de barras;

A API (*Application Programming Interface*) JasperReports organiza dados recuperados de uma fonte de dados de acordo com um design de relatório definido em um arquivo JRXML. (JASPERSOFT, 2014).

3.2.3 JAXB

Segundo IBM (2014), JAXB (*Java Architecture for XML Binding*) é uma tecnologia de ligação XML-para-Java que possibilita a transformação entre esquema e objetos Java e entre documentos da instância XML e instâncias do objeto Java. A tecnologia JAXB consiste em uma API de tempo de execução e ferramentas de acompanhamento que simplificam o acesso aos documentos XML. É possível usar as APIs JAXB e as ferramentas para estabelecer os mapeamentos entre as classes Java e o esquema XML. Um esquema XML define os elementos de dados e a estrutura de um documento XML. A tecnologia JAXB fornece a ferramenta para possibilitar que você converta seus documentos XML para e a partir dos objetos Java. Os dados armazenados em um documento XML podem ser acessados sem a necessidade de entender a estrutura de dados XML.

3.3 FERRAMENTAS UTILIZADAS COMO SUPORTE

3.3.1 Eclipse

Na implementação de software é adotado alguma ferramenta de desenvolvimento para aumentar a produtividade. Essas ferramentas são chamadas IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado). Uma IDE é um ambiente integrado de desenvolvimento que provê um conjunto de ferramentas que trabalham de forma integrada, oferecendo ao desenvolvedor facilidades para realizar as principais tarefas relacionadas ao desenvolvimento de um software. (CASAVELLA, 2014).

No caso específico da plataforma Java, a IDE utilizada no desenvolvimento do presente trabalho é o Eclipse. O Eclipse é uma ferramenta de filosofia de código aberto (*Open Source*). Destacam-se algumas funcionalidades no Eclipse, para o desenvolvimento em Java, que são:

- Auto-complete;
- Organização de *imports*;
- Auto-formatação de código;
- Mapeamento de referências;
- Compilação imediata;
- Geração de trechos de códigos;
- Incorporação de plugins: TO-DO, TASKS, etc.

3.3.2 Ireport

O iReport é uma ferramenta desenvolvida pela mesma empresa do JasperReports, a JasperForge, e por isso é muito comum ver os dois sendo usados em conjunto. Uma das dificuldades ao trabalhar com os relatórios, está na definição do *layout*. É complicado escrever o layout totalmente em XML, sem ter que se aprofundar em todas as *tags* e atributos possíveis, e além disso posicionar todos os elementos corretamente. Na prática, é muito raro alguém editar o JRXML manualmente, e sim apenas para fazer alguns pequenos ajustes quando necessários. O processo normal é utilizar alguma ferramenta para gerar o JRXML automaticamente, e o iReport é utilizado com esse propósito. (K19, 2014).

3.3.3 Astah Community

Astah Community é uma ferramenta gratuita voltada para a modelagem de diagramas UML (*Unified Modeling Language*). Além do Astah Community, existem outras três versões: Astah UML, Astah Professional e Astah Share que disponibilizam outras funcionalidades além da modelagem UML, porém, sua licença é comercial.

A ferramenta Astah Community é conhecida por sua praticidade e simplicidade em elaborar diagramas, como por exemplo: diagramas de classe, caso de uso, sequência, atividade, comunicação, máquina de estado, componentes, implantação, estrutura de composição, objetos e pacotes. (TUTORIAL, 2014).

3.4 FUNCIONAMENTO GERAL DO FRAMEWORK

O framework é distribuído em um arquivo JAR, onde nesse arquivo existem as classes necessárias para: a persistência no banco de dados, manipulação dos componentes swing e manipulação de arquivos XML e JRXML.

Ele é integrado a outro sistema que obtém as informações das classes persistentes do sistema onde estiver integrado. A cada relatório criado pelo framework, são gerados dois arquivos: um com extensão XML e outro com a extensão JRXML. No arquivo XML são gravadas todas as informações necessárias para criação do relatório e montagem de tela de filtro. O arquivo JRXML é o modelo de layout do relatório e onde serão exibidos os dados das consultas no banco de dados. Para editar um relatório criado, o framework carrega as informações do relatório presentes no arquivo XML.

A Figura 1 ilustra o esquema de funcionamento do framework Gerador de Relatórios:

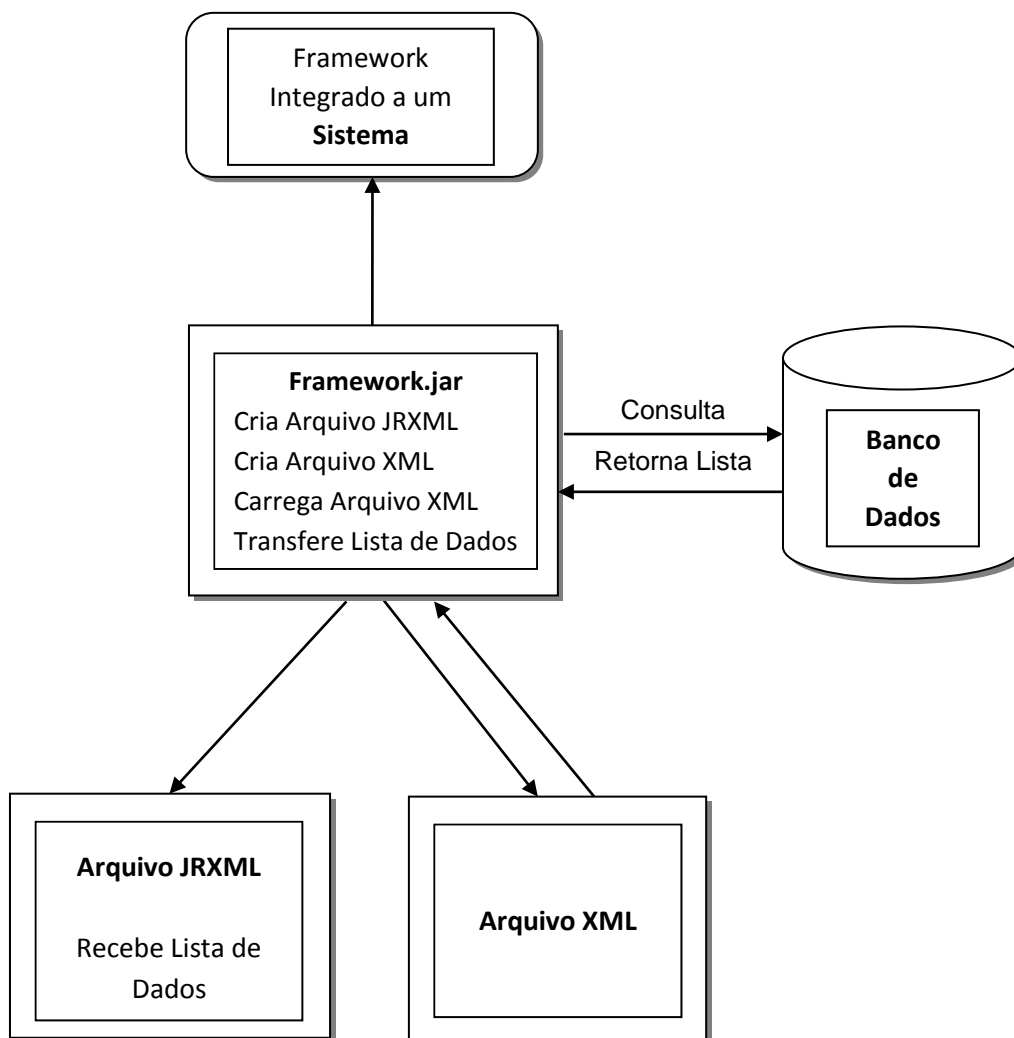


Figura 1 - Esquema de Funcionamento do Framework

Por ser um framework, é necessário a integração do mesmo com outros sistemas que, para seu funcionamento, devem possuir os seguintes requisitos:

- Ser um sistema de ambiente desktop;
- Obedecer o padrão de projetos MVC;
- Utilizar banco de dados MySQL, Hsqldb ou Oracle;

- Utilizar framework de persistência Hibernate;

Além dos requisitos citados acima, para efetuar a integração do Framework Gerador de Relatórios, é necessário adicionar o arquivo de extensão JAR ao projeto do sistema. Em seguida basta executar em qualquer parte da codificação sistema as seguintes linhas de código:

```
TelaConfigRel tela = new TelaConfigRel();  
tela.setVisible(true);
```

3.6 INTERFACES DO FRAMEWORK

A Figura 2 ilustra a interface de seleção dos atributos, de uma determinada classe, que compõe o relatório.

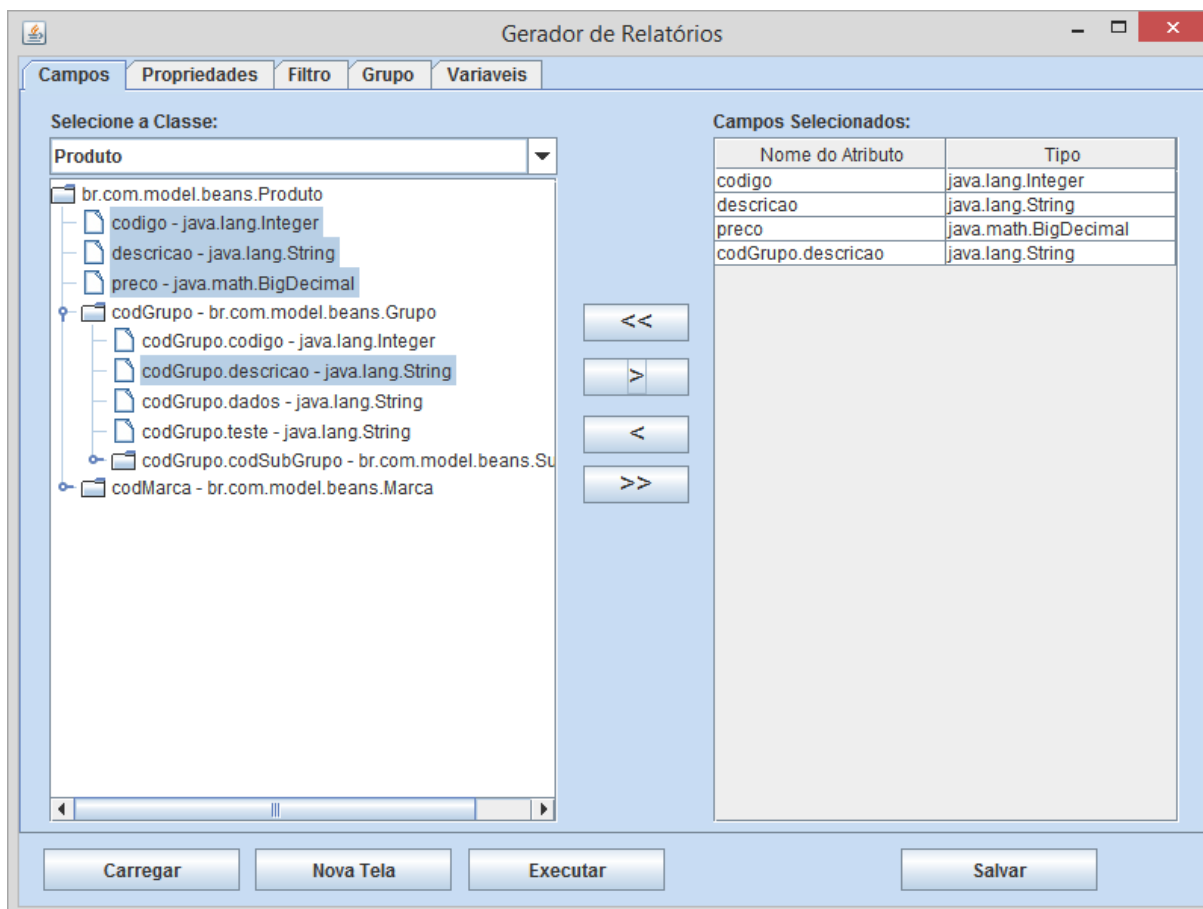


Figura 2 - Interface de seleção de campos

Esta é a interface para selecionar os campos (atributos) responsáveis pela exibição dos dados consultados no banco de dados. No painel, ao lado direito, exibe as informações dos atributos referente a classe selecionada no componente combobox. A tabela, que aparece no lado direito da tela, diz respeito aos atributos selecionados através dos botões localizados no centro da tela.

A Figura 3 ilustra a interface de propriedades dos atributos selecionados e do layout do relatório.

Nome do Atributo	Tamanho	Título	Linha	Coluna	Tam. Font	Alinh.(E-C-D)
codigo	133	Cod	1	0	10	E
descricao	133	Desc	1	133	10	E
preco	133	Preco	1	266	10	E
codGrupo.descri...	133	Desc Grupo	1	399	10	E

Formulário

Caminho do Menu (Ex: Relatorios,Cadastros): Nome do Arquivo:

Título do Relatório: Descrição do Menu:

Exibir dados em: Margens
 Superior Inferior Esquerda Direita

Largura: Altura: Alt. Linha Det: Qde de Linhas: Tem Linha Det:

Carregar Nova Tela Executar Salvar

Figura 3 - Interface de propriedades

Esta tela é referente a configuração do relatório. É utilizado o botão "Atualizar" para que os campos selecionados na tela anterior apareçam na tabela, que aparece no centro da tela. Nessa tabela é possível inserir algumas propriedades dos campos que serão exibidos no relatório como: o tamanho, o título, a posição da linha e da coluna, o tamanho da fonte e o alinhamento. Abaixo, um formulário com as propriedades do layout do relatório como: o caminho do menu, o título do relatório, o

nome do arquivo, a descrição do menu, a largura da página, a altura da página e, as margens também podem ser configuradas.

A Figura 4 ilustra a interface de seleção dos atributos, de uma determinada classe, responsáveis pela construção da tela de filtro.

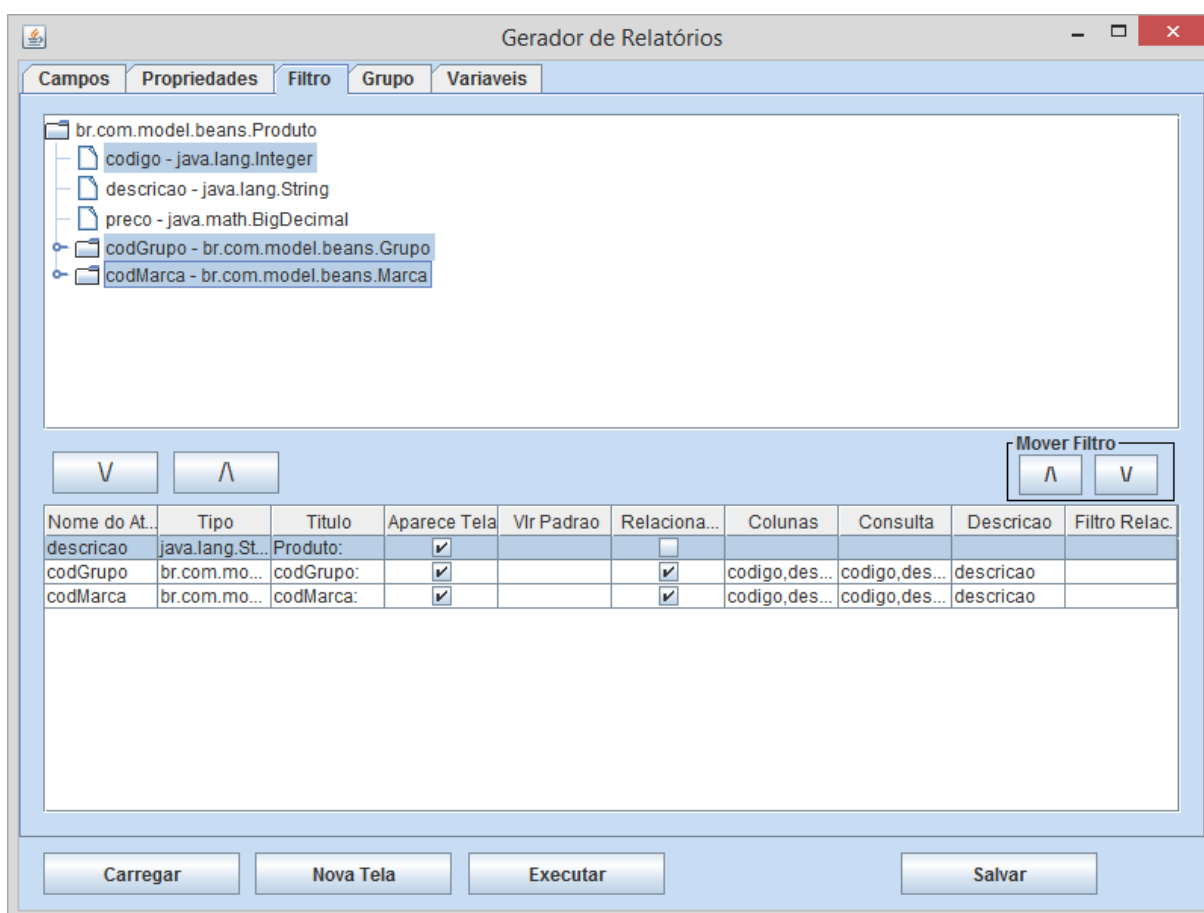


Figura 4 - Interface de seleção de filtros

Esta interface é para selecionar os filtros que serão utilizados ao fazer uma consulta no banco de dados. O painel acima, diz respeito aos atributos que representam os campos das tabelas do banco. Abaixo, na tabela, aparecem as propriedades dos atributos como: o título, se o campo vai aparecer na tela, o valor padrão para consulta, se campo tem relacionamento com outra tabela, as colunas que serão

exibidas na tela de filtro, as colunas para consulta em outra tabela e a descrição exibida para os dados selecionados de outra tabela.

A Figura 5 ilustra a interface de seleção de atributos, de uma determinada classe, responsáveis pela construção de agrupamentos de dados no relatório.

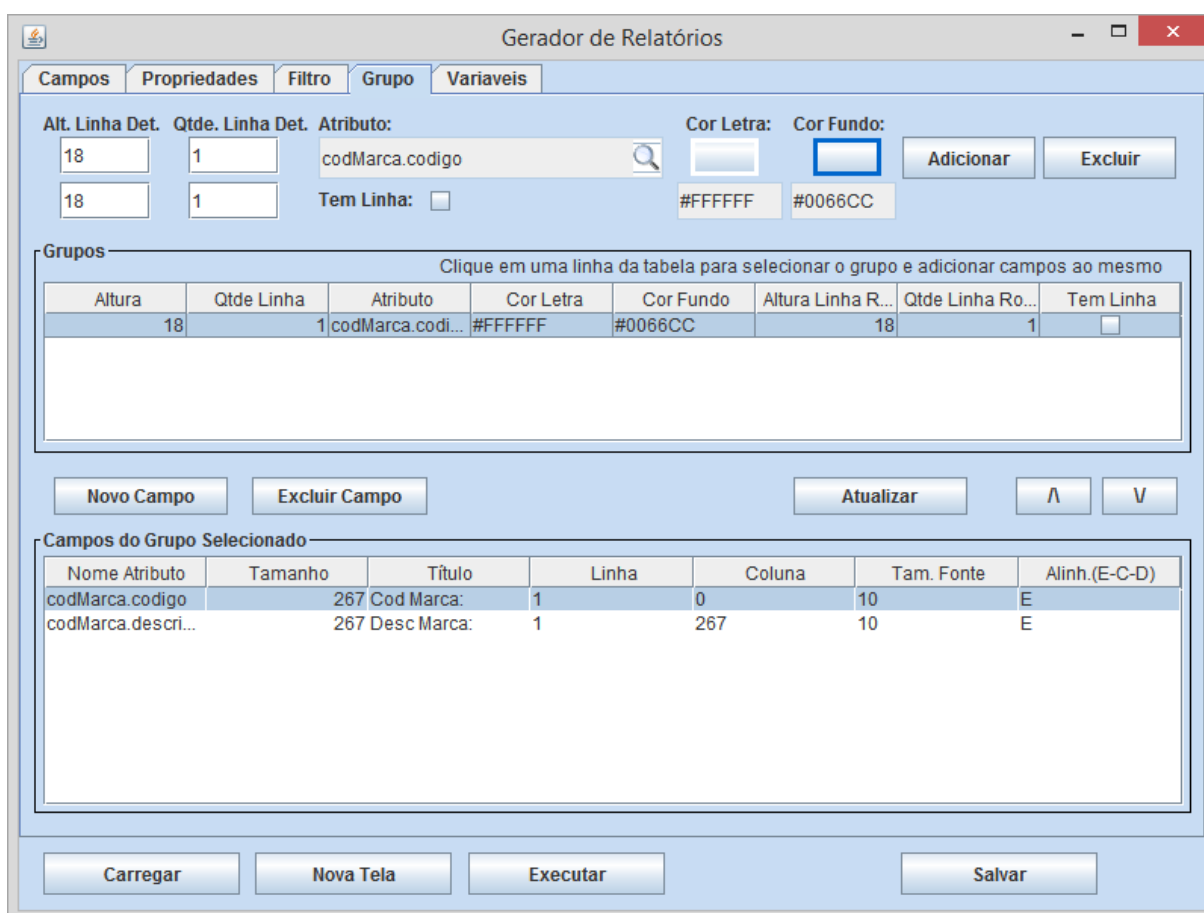


Figura 5 - Interface de seleção de atributos para criação de grupos

Na interface para criação grupo, é possível criar agrupamentos nos dados que serão exibidos no relatório. A tabela, no centro da tela com o título "Grupos", diz respeito aos grupos criados, e a tabela abaixo, exibe os campos adicionados a cada grupo criado.

A Figura 6 ilustra a interface de seleção de atributos, de uma determinada classe, responsáveis pela construção de somatórios e contagem.

The screenshot shows the 'Gerador de Relatórios' window with the 'Variáveis' tab selected. The interface is divided into several sections:

- Radio Buttons:** Variável, Parâmetro, Lb Fixo
- Fields:**
 - Nome: total
 - Tipo: java.lang.Double
 - Calculo: (Sum -> Somatório) Sum
 - Resetar Quando: Report
 - Incrementar Quando: None
 - None para agrupamento: (empty)
 - Resetar Quando (Expression): \$F(valorTotal)
 - Expressão: new Double(0)
 - Aparece em: Column
- Buttons:** Adicionar, Excluir, Atualizar
- Table:**

Nome	Tamanho	Texto	Linha	Coluna	Tam. Fonte	Alinh.(E-C-D)	Evaluation	EvaluationGro...
total	100	Total:	1	435	10	D		
- Bottom Buttons:** Carregar, Nova Tela, Executar, Salvar

Figura 6 - Interface de seleção de variáveis

Esta interface objetiva a criação de variáveis. É possível criar várias variáveis para fazer cálculos, como: soma, subtração, divisão, multiplicação. No centro da tela existe um formulário para configuração das variáveis criadas, e abaixo, a tabela que exhibe as variáveis criadas e mais outras configurações.

A Figura 7 ilustra um exemplo de relatório gerado pelo Framework Gerador de Relatórios.

Quinta-feira 25 Setembro 2014

Relatorio de Produtos

Cod Marca: 6		Desc Marca: MOBENZANI		Site Marca: www.MOBENZANI.com		
Contagem:	Cod	Desc	Desc Grupo	Estoque	Valor	Estoq X Valor
1,00	2551	COXIM MOTOR FORD CORCEL II/PAMPA. FO-2522	PECAS	0	6,73	0,00
2,00	2553	SUSP FORD - FUTOBLOC STD (BAIXO).	SUSPENSAO	12	0,00	0,00
3,00	4245	SUSP FORD - BUCHA BAND CORCEL 1978>1984	SUSPENSAO	38	3,27	124,26
4,00	7056	SUSP FORD - FUTOBLOC 37MM (ALTO).	SUSPENSAO	2	5,04	10,08
5,00	9076	SUSP GM - COXIM AMORT DIANT CORSA	SUSPENSAO	10	8,23	82,28
Estoque: 62,00				Valor: 23,27	Estoq X Vlr: 216,62	
Cod Marca: 7		Desc Marca: RICAQUIMICA		Site Marca: www.RICAQUIMICA.com		
Contagem:	Cod	Desc	Desc Grupo	Estoque	Valor	Estoq X Valor
1,00	3118	GRAFITE EM P	UTILIDADES DO LAR	367	1,36	499,11
2,00	5717	COLA JUNTAS MOTOR SEMI- SECATIVA BRANCA	LUBRIFICANTES	0	6,40	0,00
3,00	5787	COLA ADESIVO UNIVERSAL	PECAS	12	1,44	17,28
4,00	5938	TRAVA QUIMICA RICACOL ALTO TORQUE 10G.	PECAS	27	4,42	119,45
5,00	6888	COLA ADESIVO EPOXI BI- COMPONENTE	PECAS	89	6,32	562,84
Estoque: 495,00				Valor: 19,95	Estoq X Vlr: 1.198,67	
Cod Marca: 8		Desc Marca: INDISA		Site Marca: www.INDISA.com		
Contagem:	Cod	Desc	Desc Grupo	Estoque	Valor	Estoq X Valor
1,00	9500	BOMBA DAGUA FORD 0944 ESCORT/GOL. INDISA	ARREFECIMENTO	4	0,00	0,00
2,00	9838	BOMBA DAGUA GM 0147 CORSA 8V INDISA 252002	ARREFECIMENTO	3	0,00	0,00
3,00	9839	BOMBA DAGUA FIAT 0766 PALIO FIRE	ARREFECIMENTO	2	0,00	0,00
Estoque: 9,00				Valor: 0,00	Estoq X Vlr: 0,00	
Cod Marca: 9		Desc Marca: SCHADEK		Site Marca: www.SCHADEK.com		
Contagem:	Cod	Desc	Desc Grupo	Estoque	Valor	Estoq X Valor
1,00	2001	BOMBA OLEO VW 10100 KOMBI DIESEL SCHADEK	MOTOR	2	54,40	108,80
Estoque: 2,00				Valor: 54,40	Estoq X Vlr: 108,80	
Total Estoq: 568,00				Total Valor: 1.524,09		

Relatorio Desenvolvido pelo Framework Gerador de Relatorios Pagina 1 de 1

Figura 7 - Exemplo de relatório de produtos

4 PLANEJAMENTO DO PROJETO

4.1 ESTRUTURA ANALÍTICA DO PROJETO (WBS)

Para o desenvolvimento do Framework Gerador de Relatórios elaborou-se a seguinte Estrutura Analítica do Projeto, ilustrada pela figura 8:

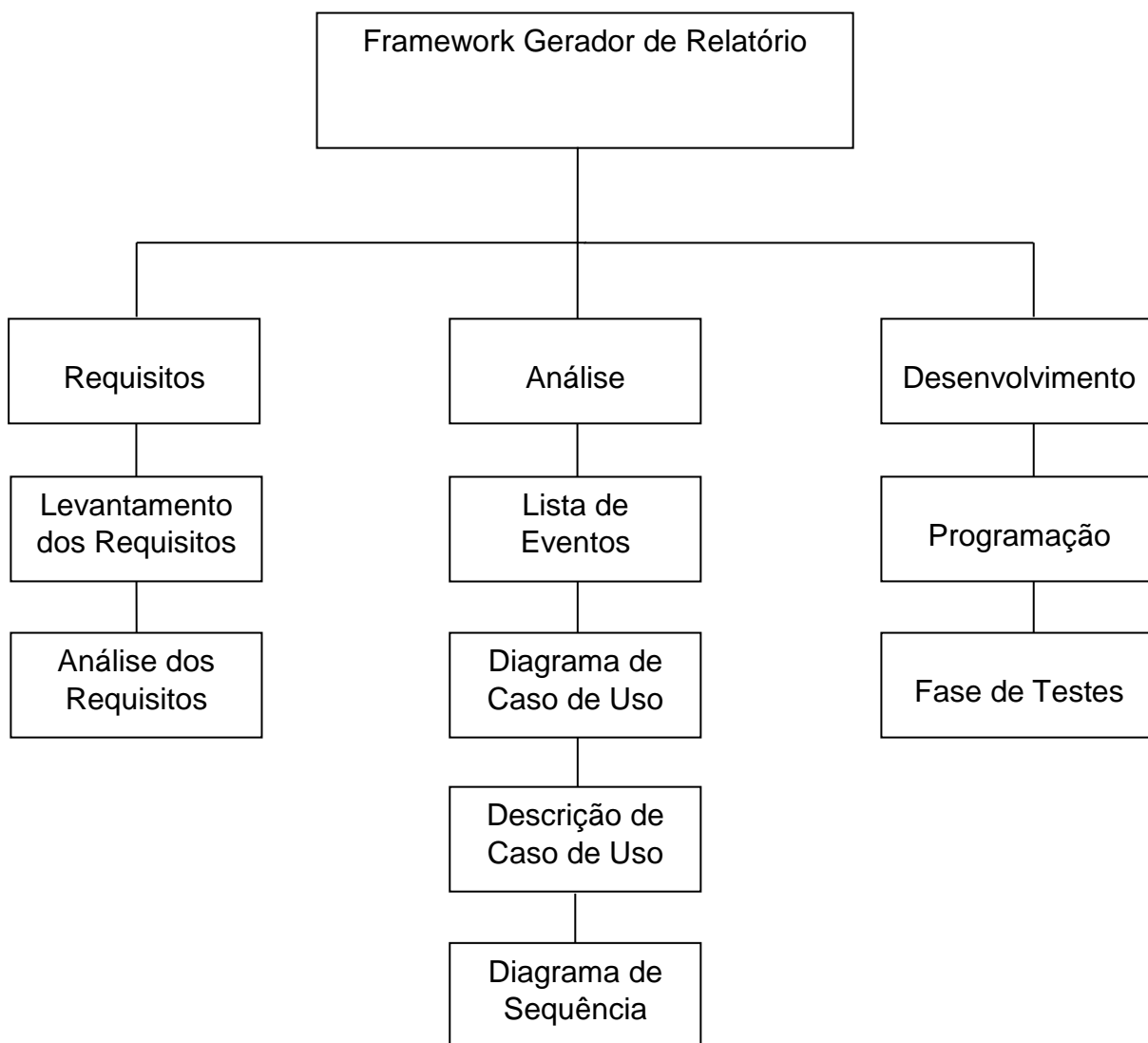


Figura 8 - Estrutura Analítica do Projeto

4.2 SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS

O Sequenciamento de Atividades envolve estabelecer uma relação lógica das atividades a serem desenvolvidas ao longo do projeto. A figura 9 ilustra o sequenciamento das atividades estabelecidas e desenvolvidas durante o processo de análise e implementação do sistema:

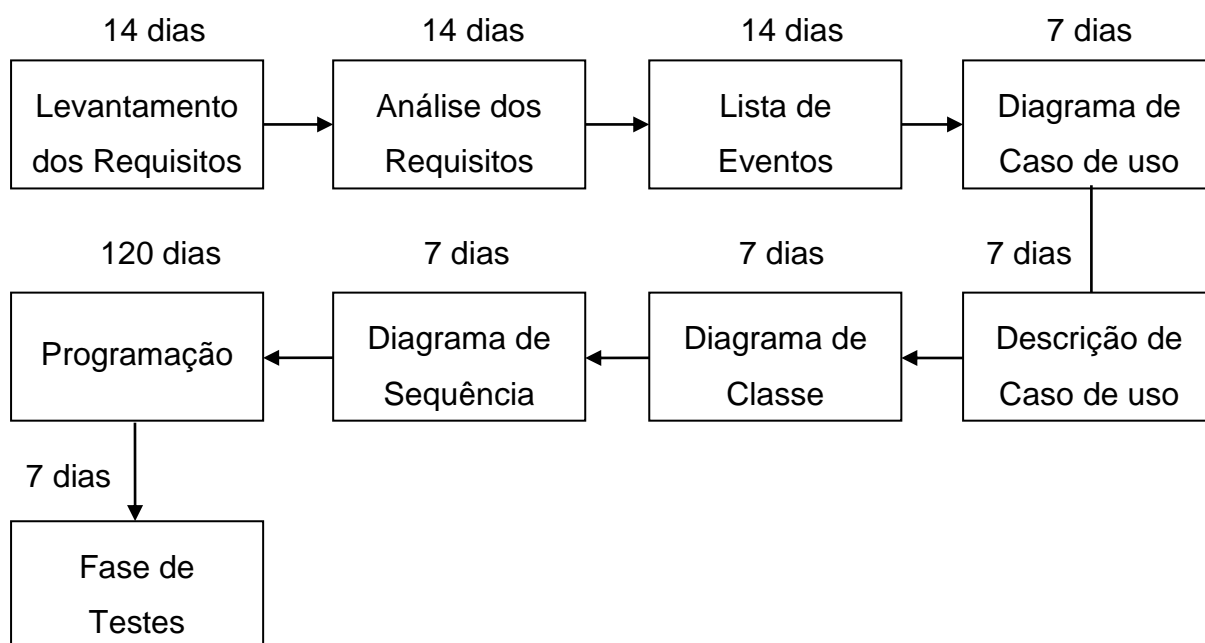


Figura 9 - Sequenciamento das Atividades

4.3 LISTA DE EVENTOS

N ^a	Nome	Objetivo	Caso de Uso
1	Usuário Carrega um Relatório	Seleção de um Relatório já Existente no Computador	Carregar Relatório
2	Usuário Mantém Campos	Tela de Manter Campos	Manter Campos
3	Usuário Mantém Grupo	Tela de Manter Grupo	Manter Grupo
4	Usuário Mantém Variáveis	Tela de Manter Variáveis	Manter Variáveis
5	Usuário Mantém Filtro	Tela de Manter Filtro	Manter Filtro
6	Usuário Configura Relatório	Tela de Configuração de Relatório	Configurar Relatório
7	Usuário Executa Relatório	O Usuário tem a Opção de Executar um Relatório	Executar Relatório
8	Usuário Abre uma Nova Tela	O Usuário tem a Opção de Manipular mais de um Relatório	Abrir Nova Tela
9	Usuário Salva o Relatório	Salvar Relatório no Computador	Salvar Relatório

4.4 CASO DE USO

A figura 10 apresenta o Diagrama de Casos de Uso do sistema desenvolvido.

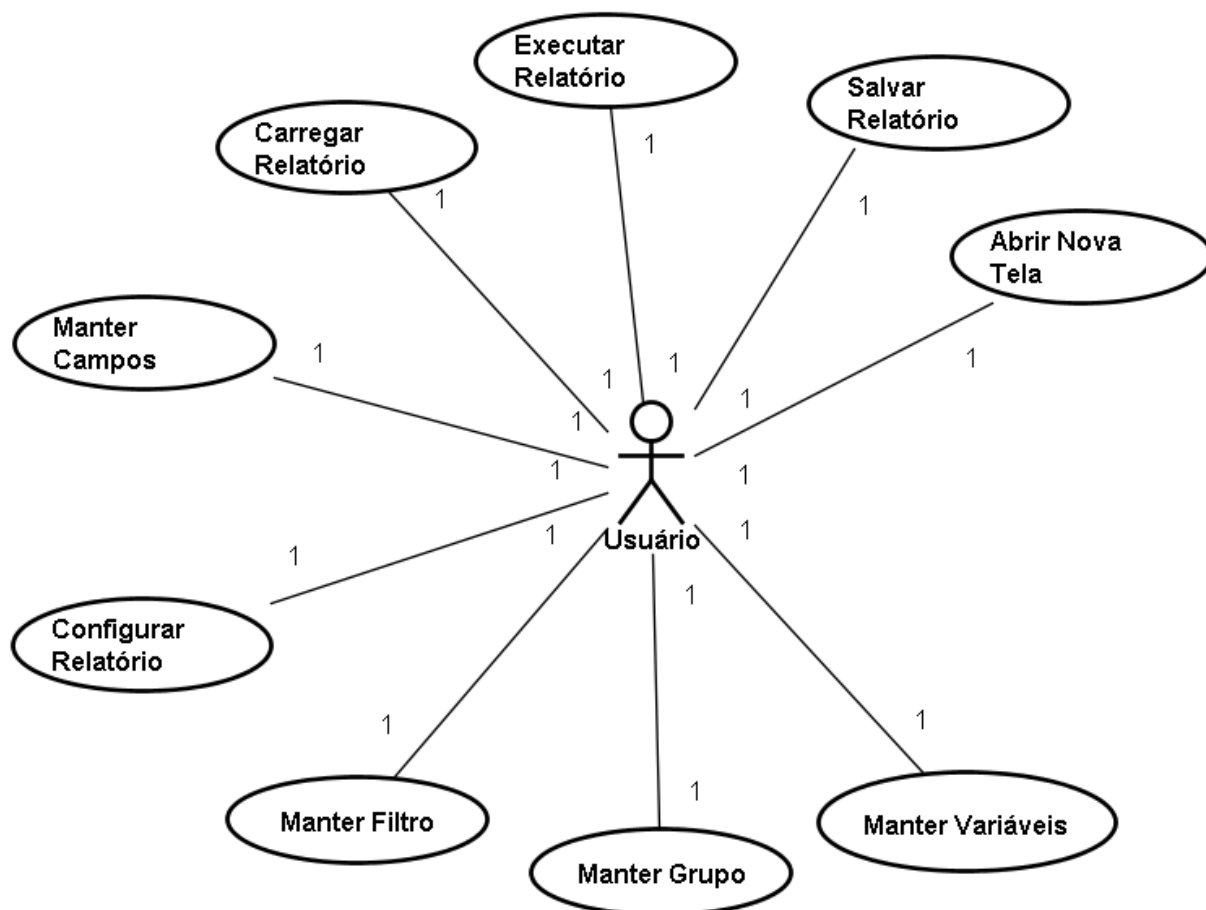


Figura 10 - Diagrama de Caso de Uso

4.5 DIAGRAMA DE CLASSE

A figura 11 ilustra o diagrama de classes construído para o sistema Framework Gerador de Relatórios.

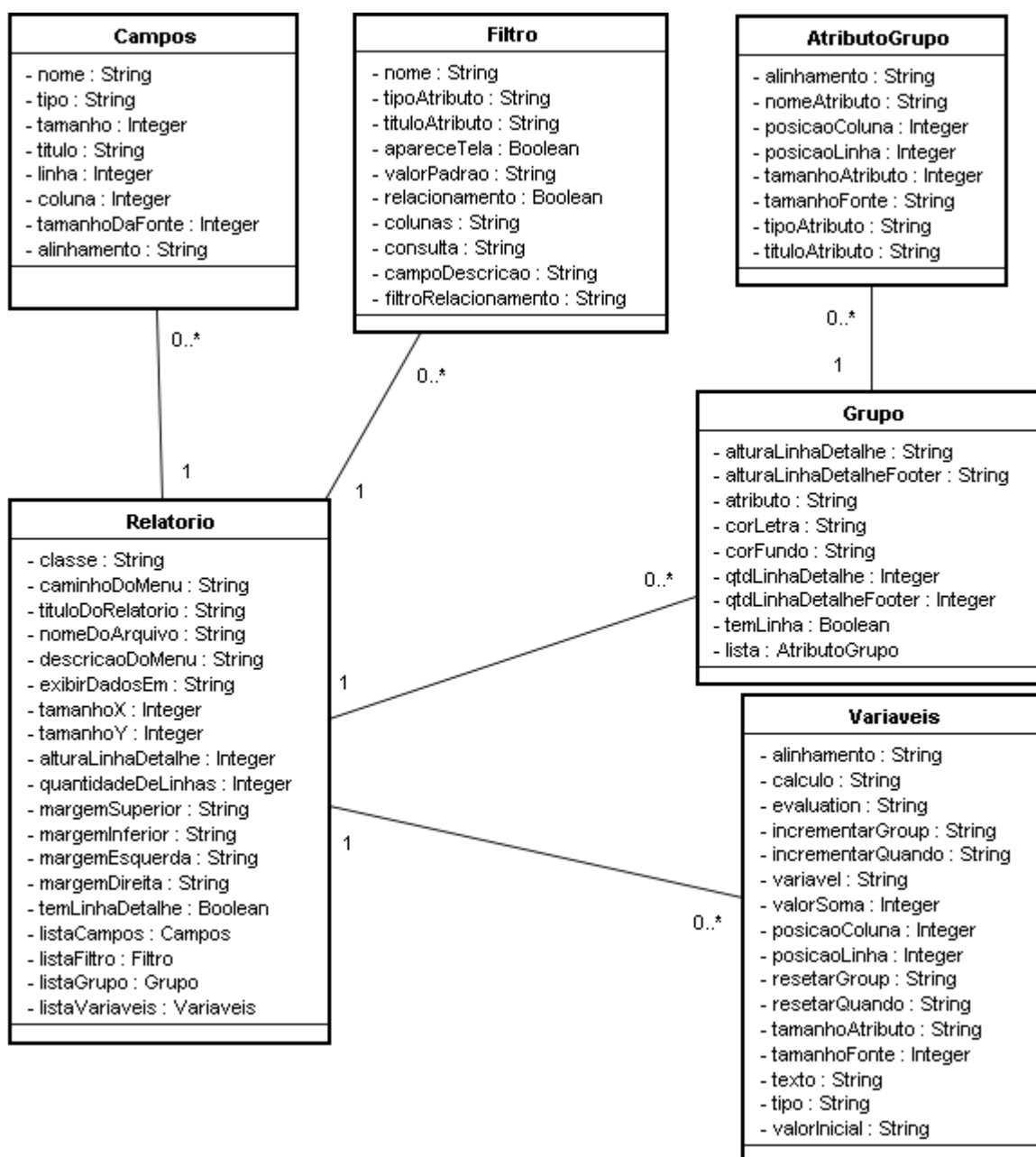


Figura 11 - Diagrama de Classe

4.6 ESPECIFICAÇÃO DE CASOS DE USO

4.6.1 Carregar Relatório

Nome de Use Case	Carregar Relatório
Finalidade/Objetivo	Carregar um as informações de um relatório através de um arquivo de extensão xml.
Ator	Usuário
Pré-Condições	Existir um relatório salvo.
Eventos Iniciais	O Usuário da início clicando no botão "Carregar".
Fluxo Principal	1 – Clicar no botão "Carregar". 2 – O sistema exibe a janela para encontrar o arquivo. 3 – Selecionar o arquivo.

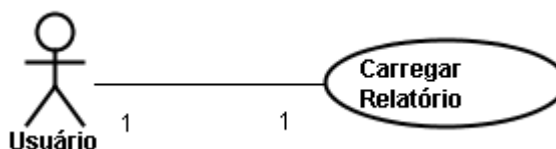


Figura 12 - Caso de Uso Carregar Relatório

4.6.2 Manter Campos

Nome de Use Case	Manter Campos
Descrição	A função deste caso de uso é selecionar os atributos de uma classe, que serão exibidos no relatório.
Ator	Usuário
Pré-Condições	Ter um relatório carregado.

Evento Inicial	O usuário seleciona a aba "Campos".
Fluxo Principal	<p>1 – Selecionar a classe desejada clicando na caixa de combinação. [A1]</p> <p>2 – Selecionar campo clicando em cima do nome do atributo da classe. [A2] [A3]</p> <p>3 – Excluir campo selecionado da tabela de campos selecionados. [A4] [A5]</p> <p>4 – Selecionar a aba "Propriedades".</p> <p>5 – Atualizar campo clicando no botão "Atualizar". [A6]</p>
Fluxo Alternativo	<p>[A1] Alterar Classe.</p> <p>1 – O usuário Escolhe outra classe para alterar.</p> <p>[A2] Selecionar Um Atributo.</p> <p>1 – O usuário clica no botão ">".</p> <p>[A3] Selecionar Todos Atributos da Classe.</p> <p>1 – O usuário clica no botão ">>"</p> <p>[A4] Excluir Um Atributo.</p> <p>1 – O usuário clica no botão "<".</p> <p>[A5] Excluir Todos Atributos Selecionados.</p> <p>1 – O usuário clica no botão "<<".</p> <p>[A6] Calcular Campos</p> <p>1 – O usuário clica no botão "Sim" para calcular ou clica no botão "Não" para não calcular.</p>

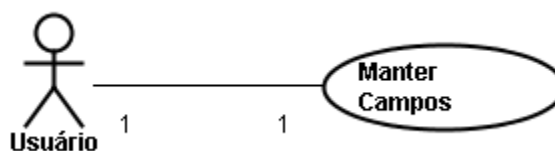


Figura 13 - Caso de Uso Manter Campos

4.6.3 Manter Filtro

Nome de Use Case	Manter Filtro
Descrição	A função deste caso de uso é selecionar os atributos de uma classe, que serão responsáveis pela consulta no banco de dados.
Ator	Usuário
Pré-Condições	Ter um relatório carregado.
Evento Inicial	O usuário seleciona a aba "Filtro".
Fluxo Principal	1 – Selecionar campo clicando em cima do nome do atributo da classe. [A1] 3 – Excluir campo selecionado da tabela de filtros selecionados. [A2]
Fluxo Alternativo	[A1] Selecionar Um Atributo. 1 – O usuário clica no botão "V". [A2] Excluir Um Atributo. 1 – O usuário clica no botão "Λ"

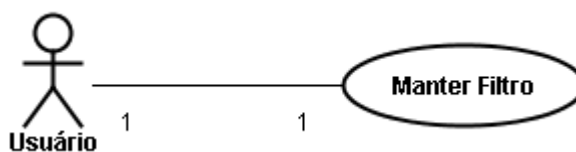


Figura 14 - Caso de Uso Manter Filtro

4.6.4 Manter Grupo

Nome de Use Case	Manter Grupo
Descrição	A função deste caso de uso é selecionar os atributos de uma classe, que serão responsáveis pelo agrupamento de dados.
Ator	Usuário
Pré-Condições	Ter um relatório carregado.
Evento Inicial	O usuário seleciona a aba "Grupo".
Fluxo Principal	<p>1 – Selecionar atributo em que será agrupado clicando no botão dentro do campo de texto. A[1]</p> <p>2 – Selecionar Grupo adicionado. A[2] A[3] A[4] A[5]</p> <p>3 – Excluir campo selecionado da tabela de filtros selecionados. [A2]</p>
Fluxo Alternativo	<p>[A1] Adicionar Grupo.</p> <p>1 – O usuário seleciona um campo.</p> <p>2 – Clicar no botão "Adicionar".</p> <p>[A2] Excluir Grupo.</p> <p>1 – Clicar no botão "Excluir".</p> <p>[A3] Adicionar Campos</p> <p>1 – Clicar no botão "Novo Campo".</p> <p>2 – Selecionar campo.</p> <p>A[4] Excluir Campos.</p> <p>1 – Selecionar Campo.</p> <p>2 – Clicar no botão "Excluir Campo".</p> <p>A[5] Alterar Campo.</p> <p>1 – Alterar dados do campo.</p> <p>2 – Clicar no botão "Atualizar".</p>

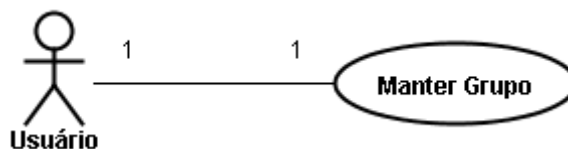


Figura 15 - Caso de Uso Manter Grupo

4.6.5 Configurar Relatório

Nome de Use Case	Configurar Relatório
Descrição	A função deste caso de uso é adicionar as informações básicas para criação do modelo de relatório, como: nome do arquivo xml e jxml, título do relatório, largura e altura da área de exibição dos dados, etc.
Ator	Usuário
Pré-Condições	Ter um relatório carregado.
Evento Inicial	O usuário seleciona a aba "Propriedades".
Fluxo Principal	1 – Preencher formulário de configuração.

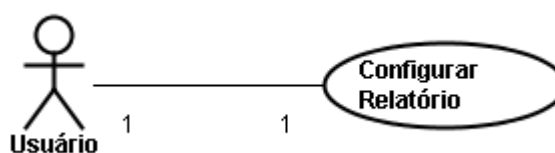


Figura 16 - Caso de Uso Configurar Relatório

4.6.6 Executar Relatório

Nome de Use Case	Executar Relatório
Descrição	A função deste caso de uso é montar a tela de filtro e o relatório de acordo com as informações dos arquivos xml e jrxml.
Ator	Usuário
Pré-Condições	Existir um relatório salvo.
Fluxo Principal	1 – Clicar no botão "Executar". [A1] [A2]
Fluxo Alternativo	[A1] Salvar Relatório. 1 – Para salvar clicar no botão "Sim" ou para não salvar clicar no botão "Não". A[2] Fazer Filtro 1 – Para fazer filtro clicar no botão "Sim" ou para não fazer filtro clicar no botão "Não".

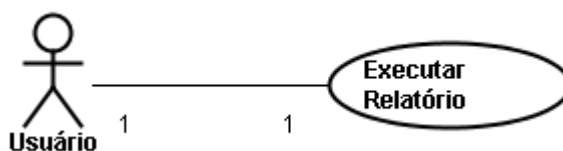


Figura 17 - Caso de Uso Executar Relatório

4.6.7 Abrir Nova Tela

Nome de Use Case	Abrir Nova Tela
Descrição	A função deste caso de uso é abrir a tela de gerar relatório, caso o usuário deseje manipular mais de um relatório.
Ator	Usuário
Pré-Condições	Ter uma aplicação do gerador de relatórios em execução.
Fluxo Principal	1 – Clicar no botão "Nova Tela".

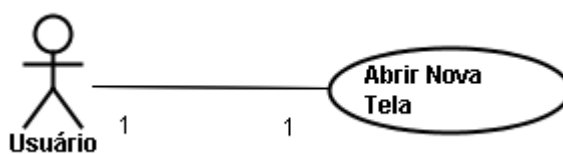


Figura 18 - Caso de Uso Abrir Nova Tela

4.6.8 Salvar Relatório

Nome de Use Case	Salvar Relatório
Descrição	A função deste caso de uso é salvar as informações do relatório no arquivo xml e jrxml.
Ator	Usuário
Pré-Condições	Ter um relatório carregado.
Evento Inicial	Clicar no botão "Executar".
Fluxo Principal	1 – Salvar relatório. [A1] 2 – Escolher fazer filtro. [A2]
Fluxo Alternativo	[A1] Salvar. 1 – Para salvar clicar no botão "Sim" ou para não salvar clicar no botão "Não". A[2] Fazer Filtro 1 – Para fazer filtro clicar no botão "Sim" ou para não fazer filtro clicar no botão "Não".

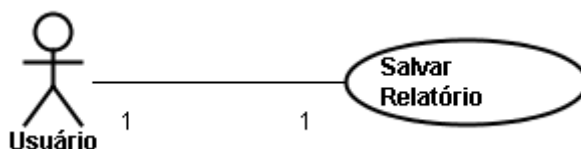


Figura 19 - Caso de Uso Salvar Relatório

CONSIDERAÇÕES FINAIS

O presente trabalho objetivou o desenvolvimento de um framework para agilizar a criação de relatórios dispensando o desenvolvimento de telas para consulta no banco de dados e implementações de códigos.

Este framework foi codificado pela linguagem de programação JAVA SE para o ambiente desktop, que utilizou-se de diferentes de tipos de frameworks para a construção do mesmo.

Nesse sentido o trabalho inicia-se com uma revisão da bibliografia acerca dos conceitos, padrões e metodologias das linguagens de programação e ferramentas utilizadas neste.

O desenvolvimento deste trabalho proporcionou conhecimentos específicos de tecnologias e outros frameworks existentes no mercado os quais o autor ainda não possui conhecimento.

Como trabalhos futuros pretende-se desenvolver a criação de sub-relatórios e gráficos.

REFERÊNCIAS

BAUER, Christian; KING, Gavin. Hibernate in action. Greenwich: Manning Publications, 2005.

CAELUM, Ensino e Inovação. JAXB – XML e Java de mãos dadas. Disponível em <blog.caelum.com.br/jaxb-xml-e-java-de-maos-dadas/>. Acesso em: 22 fev. 2014.

CASAVELLA, Eduardo. Ambientes Integrado para o Desenvolvimento em Linguagem C. Disponível em <<http://linguagemc.com.br/ambientes-integrados-de-desenvolvimento-em-linguagem-c/>>. Acesso em: 16 ago. 2014.

CAVALÉRO, Pedro. Swing: O Desktop Java - Revista Easy Java Magazine 12. Disponível em <<http://www.devmedia.com.br/swing-o-desktop-java-revista-easy-java-magazine-12/22860>>. Acesso em: 04 mar. 2014.

IBM. Utilizando o JAXB Para Ligação de Dados XML. Disponível em <http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.express.doc%2Finfo%2Fexp%2Fae%2Ftwbs_jaxbind.html>. Acesso em: 22 fev. 2014.

IBM. Report Definitions. Disponível em <http://pic.dhe.ibm.com/infocenter/director/v5r2/index.jsp?topic=/capman_5.20/fqn0_c_report_definitions.html>. Acesso em: 05 nov. 2013.

JAVA. Disponível em <http://www.java.com/pt_BR/download/faq/whatis_java.xml>. Acesso em: 04 mar. 2014.

JAVA FREE. Tutorial Java: O que é Java?. Disponível em <<http://javafree.uol.com.br/topic-871498-Tutorial-Java-O-que-e-Java.html>>. Acesso em: 04 mar. 2014.

JAVA. Saiba mais sobre a tecnologia Java. Disponível em <http://www.java.com/pt_BR/about/>. Acesso em: 04 mar. 2014.

JASPERSOFT, COMMUNITY. JasperReports Library - Tutorial. Disponível em <<http://community.jaspersoft.com/project/jasperreports-library>>. Acesso em: 04 mar. 2014.

K19, Treinamentos. Relatórios em Java - JasperReports e iReport. Disponível em <<http://www.k19.com.br/artigos/relatorios-em-java-jasperreports-e-irepor/>>. Acesso em: 23 fev. 2014.

ORACLE. Why Java?. Disponível em <<http://www.oracle.com/us/technologies/java/overview/index.html>>. Acesso em: 04 mar. 2014.

ORACLE. Trail: The Reflection API. Disponível em <<http://docs.oracle.com/javase/tutorial/reflect/>>. Acesso em: 08 mar. 2014.

QUE, Development Group. Introdução à Programação. Tradução de Vandenberg Dantas de Souza. Rio de Janeiro: Campus, 1993.

SILVA, Eduardo N. S. DESENVOLVIMENTO DO FRAMEWORK JAVA-FÁCIL. 2011. 41p. Trabalho de Conclusão de Curso Fundação. Educacional do Município de Assis - FEMA/Instituto Municipal de Ensino Superior de Assis – IMESA, SP, Assis, 2011.

SILVA, Ricardo P. Suporte ao desenvolvimento e uso de frameworks e componentes. Tese para a obtenção do grau de Doutor em Ciências da Computação. Universidade Federal do Rio Grande do Sul, UFRGS. Porto Alegre: 2000.

TUTORIAL. Guia Prático de utilização da ferramenta Astah Community. Disponível em <<http://www-pet-si.inf.ufsm.br/images/consultoriodesoftware/Astah.pdf>>. Acesso em 12 mar. 2014.

UNIVERSIDADE FEDERAL DO PARANÁ. Biblioteca Central. Normas para apresentação de trabalhos: teses, dissertações e trabalhos acadêmicos. 5.ed. Curitiba: Ed. UFPR, 1996.

UNIÃO DAS INSTITUIÇÕES DE SERVIÇOS, ENSINO E PESQUISA LTDA. Manual de Relatório Científico (MRC) (GRADUAÇÃO). 2010. 8p. São Paulo: UNISEPE, 2010.

WIRFS-BROCK, Rebecca J.; JOHNSON, Ralph. E. Surveying current research in object-oriented design. Communications of the ACM. v.33, n.9, sep, 1990.