

UNIVERSIDADE DE CAXIAS DO SUL
CENTRO DE COMPUTAÇÃO E TECNOLOGIA DA INFORMAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

EDUARDO ERLO

**Desenvolvimento de uma ferramenta
WEB para extração customizada de
informações de bancos de dados
relacionais**

Alexandre E. K. Nascimento
Orientador

Caxias do Sul, Dezembro de 2012

*A minha família, pelo amor e apoio.
A minha namorada, pela compreensão.
Ao meu orientador, pela ajuda e amizade.
Aos professores e colegas de curso pelas horas de convívio e amizade.*

EDUARDO ERLO

RESUMO

Este trabalho apresenta um sistema que visa atender a crescente necessidade dos sistemas de gestão por extração de dados para geração de relatórios e gráficos de análise. Neste trabalho é apresentado um sistema que se propõe a mostrar as possibilidades de dados a serem listados, criação de regras, filtros e formatação dos dados, para geração de relatórios rápidos. Para que isto aconteça, foi utilizada a metodologia de desenvolvimento ICONIX, visando obter artefatos para o processo de desenvolvimento do sistema.

Palavras-chave: Geração de Relatórios, Iconix.

Development of a web tool for customized data extraction for relational
databases

ABSTRACT

This present study approaches a system that aim to attend a growing informatics system necessity of data extraction to generating reports and analysis charts. In this study it's presented a system that purports to showing the possibilities of data to be listed, generation of rules, filters and format of data for generation of quick reports. For this to happen, was used the ICONIX development methodology, aiming to obtain artifacts to the process of system development.

Keywords: Report Generation, Iconix.

SUMÁRIO

LISTA DE ACRÔNIMOS	10
LISTA DE FIGURAS	11
LISTA DE TABELAS	15
LISTA DE TRECHOS DE CÓDIGO	16
1 INTRODUÇÃO	17
2 REFERENCIAL TEÓRICO	20
2.1 A importância da informação	20
2.2 Geração de informações a partir de sistemas de informática	21
2.3 Sistemas Geradores de Relatórios	21
2.4 Catálogo de meta-informações em bancos de dados	22
2.5 Tecnologias envolvidas	23
2.5.1 Python	23
2.5.2 Django	25
2.5.3 Django <i>Models</i>	27
2.5.4 Django-Introspection	28
2.5.5 Django db	29
2.5.6 HTTP, HTML, CSS e Javascript	30
2.5.7 Apache HTTPD	31
2.5.8 Mod_WSGI	32
2.5.9 JQuery	33
2.5.10 Matplotlib	34
2.5.11 WkHTMLtoPDF - HTML to PDF Converter	35
2.5.12 ICONIX	35
2.5.13 UML - Unified Modeling Language(Linguagem Unificada de Modelagem)	37

3	PROBLEMA	41
4	PROPOSTA DE SOLUÇÃO	45
5	PROCESSO DE DESENVOLVIMENTO DO SISTEMA	48
5.1	Requisitos Não Funcionais	48
5.2	Requisitos Funcionais	48
5.3	Diagrama de Caso de Uso do Sistema	51
5.4	Diagrama de Domínio do Sistema	52
5.5	UC-01 Autenticar Usuário	55
5.5.1	Análise de Requisitos	55
5.5.2	Análise	56
5.5.3	Projeto	58
5.6	UC-02 Manter Usuário	60
5.6.1	Análise de Requisitos	60
5.6.2	Análise	61
5.6.3	Projeto	64
5.7	UC-03 Manter Usuário(Administrador)	66
5.7.1	Análise de Requisitos	66
5.7.2	Análise	67
5.7.3	Projeto	70
5.8	UC-04 Manter Entidade Composta	72
5.8.1	Análise de Requisitos	72
5.8.2	Análise	74
5.8.3	Projeto	78
5.9	UC-05 - Manter Relatório	81
5.9.1	Análise de Requisitos	81
5.9.2	Análise	83
5.9.3	Projeto	86
5.10	UC-06 - Buscar Relatórios	89
5.10.1	Análise de Requisitos	89
5.10.2	Análise	90
5.10.3	Projeto	92
5.11	UC-07 - Gerar Relatório	94
5.11.1	Análise de Requisitos	94
5.11.2	Análise	95
5.11.3	Projeto	98
5.12	UC-08 Exportar PDF	100

5.12.1	Análise de Requisitos	100
5.12.2	Análise	101
5.12.3	Projeto	103
5.13	UC-09 Exportar XLS(planilha eletrônica)	105
5.13.1	Análise de Requisitos	105
5.13.2	Análise	106
5.13.3	Projeto	108
5.14	UC-10 Enviar Relatório por E-mail	110
5.14.1	Análise de Requisitos	110
5.14.2	Análise	111
5.14.3	Projeto	114
5.15	UC-11 - Agendar Geração de Relatório	116
5.15.1	Análise de Requisitos	116
5.15.2	Análise	117
5.15.3	Projeto	120
5.16	UC-12 Gerar Relatório Agendado	122
5.16.1	Análise de Requisitos	122
5.16.2	Análise	123
5.17	UC-13 Consultar Relatórios Salvos	126
5.17.1	Análise de Requisitos	126
5.17.2	Análise	127
5.17.3	Projeto	129
5.18	UC-14 - Salvar Relatório	131
5.18.1	Análise de Requisitos	131
5.18.2	Análise	132
5.18.3	Projeto	134
5.19	UC-15 Duplicar Relatório	136
5.19.1	Análise de Requisitos	136
5.19.2	Análise	138
5.19.3	Projeto	140
5.20	UC-16 Duplicar Entidade Composta	142
5.20.1	Análise de Requisitos	142
5.20.2	Análise	143
5.20.3	Projeto	145
5.21	UC-17 Salvar Filtros de Relatório	147
5.21.1	Análise de Requisitos	147
5.21.2	Análise	148
5.21.3	Projeto	150
5.22	UC-18 Mostrar Lista de Valores de Campo de Filtro	152

5.22.1	Análise de Requisitos	152
5.22.2	Análise	152
5.22.3	Projeto	155
5.23	UC-19 Listar Eventos de Gerações de Relatórios no Dashboard	157
5.23.1	Análise de Requisitos	157
5.23.2	Análise	158
5.23.3	Projeto	160
6	MATRIZES DE RASTREABILIDADE	162
7	IMPLEMENTAÇÃO	169
7.1	Módulos	169
7.1.1	Core	170
7.1.2	MetadadosBanco	171
7.1.3	Logs	172
7.1.4	EntidadeComposta	172
7.1.5	Relatorio	173
7.1.6	GeracaoRelatorio	174
7.1.7	Utils	174
7.2	Camada de Apresentação	175
7.2.1	Core	175
7.2.2	EntidadeComposta	176
7.2.3	Relatorio	176
7.2.4	GeracaoRelatorio	177
8	CONSIDERAÇÕES FINAIS	178
8.1	Considerações sobre o Desenvolvimento	178
8.2	Funcionalidades Adicionais	179
8.3	Trabalhos Futuros	180
	REFERÊNCIAS	181
9	ANEXOS	183

LISTA DE ACRÔNIMOS

API	Application Programming Interface	34
CSS	Cascading Style Sheets.....	30
HTML	HyperText Markup Language	30
HTTP	HyperText Transfer Protocol.....	30
PDF	Portable Document Format	47
RF	Requisito Funcional.....	48
RNF	Requisito Não Funcional	48
RUP	Rational Unified Process (Processo Unificado Racional)	36
TCP/IP	Transmission Control Protocol/Internet Protocol.....	31
UC	Use Case (Caso de Uso).....	36
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)	35
URL	Uniform Resource Locator (Localizador-Padrão de Recursos) .	27
XP	eXtreme Programing (Programação Extrema)	36

LISTA DE FIGURAS

Figura 2.1: Fluxo Entre as Fases do Processo ICONIX.	36
Figura 2.2: Fluxo Entre as Fases do Processo ICONIX.	37
Figura 2.3: Modelo Larman.	39
Figura 2.4: Modelo utilizado neste trabalho.	40
Figura 4.1: Estrutura da ferramenta.	46
Figura 5.1: Diagrama de Caso de Uso do Sistema como um Todo.	51
Figura 5.2: Diagrama de Domínio do Sistema como um Todo.	52
Figura 5.3: Tela 1 - Protótipo não funcional de interface - Autenticar Usuário.	55
Figura 5.4: Diagrama de caso de uso - Autenticar Usuário.	55
Figura 5.5: Modelo de Domínio - Autenticar Usuário.	56
Figura 5.6: Diagrama de sequência - Autenticar Usuário.	58
Figura 5.7: Diagrama de classes - Autenticar Usuário.	59
Figura 5.8: Tela 2 - Protótipo não funcional de interface - Manter Usuário.	60
Figura 5.9: Diagrama de caso de uso - Manter Usuário.	61
Figura 5.10: Diagrama de domínio - Manter Usuário.	61
Figura 5.11: Diagrama de sequência - Manter Usuário.	64
Figura 5.12: Diagrama de classes - Manter Usuário.	65
Figura 5.13: Tela 3 - Protótipo não funcional de interface - Manter Usuário(Administrador).	66
Figura 5.14: Diagrama de caso de uso - Manter Usuário.	67
Figura 5.15: Diagrama de domínio - Manter Usuário (Administrador).	67
Figura 5.16: Diagrama de sequência - Manter Usuário(Administrador).	70
Figura 5.17: Diagrama de classes - Manter Usuário (Administrador).	71
Figura 5.18: Tela 4 - Protótipo não funcional de interface - Buscar Entidades Compostas.	72
Figura 5.19: Tela 4.1 - Protótipo não funcional de interface - Manter Entidades Compostas.	73
Figura 5.20: Diagrama de caso de uso - Manter Entidade Composta.	73
Figura 5.21: Diagrama de domínio - Manter Entidade Composta.	74

Figura 5.22: Diagrama de sequência - Manter Entidade Composta.	78
Figura 5.23: Diagrama de sequência - Manter Entidade Composta(Remove).	79
Figura 5.24: Diagrama de classes - Manter Entidade Composta.	80
Figura 5.25: Tela 5 - Protótipo não funcional de interface - Manter Relatório.	81
Figura 5.26: Tela 5.1 - Protótipo não funcional de interface - Manter Relatório(condições de formatação).	82
Figura 5.27: Diagrama de caso de uso - Manter Relatório.	82
Figura 5.28: Diagrama de domínio - Manter Relatório.	83
Figura 5.29: Diagrama de sequência - Manter Relatório.	86
Figura 5.30: Diagrama de sequência - Manter Relatório.	87
Figura 5.31: Diagrama de classes - Manter Relatório.	88
Figura 5.32: Tela 6 - Protótipo não funcional de interface - Buscar Relatórios.	89
Figura 5.33: Diagrama de caso de uso - Buscar Relatórios.	89
Figura 5.34: Diagrama de domínio - Buscar Relatórios.	90
Figura 5.35: Diagrama de sequência - Buscar Relatórios.	92
Figura 5.36: Diagrama de classes - Buscar Relatórios.	93
Figura 5.37: Tela 7 - Protótipo não funcional de interface - Gerar Relatório.	94
Figura 5.38: Tela 7.1 - Protótipo não funcional de interface - Gerar Relatório(resultado).	95
Figura 5.39: Diagrama de caso de uso - Gerar Relatório.	96
Figura 5.40: Diagrama de domínio - Gerar Relatório.	96
Figura 5.41: Diagrama de sequência - Gerar Relatório.	98
Figura 5.42: Diagrama de classes - Gerar Relatório.	99
Figura 5.43: Tela 8 - Protótipo não funcional de interface - Exportar PDF.	100
Figura 5.44: Diagrama de caso de uso - Exportar PDF.	100
Figura 5.45: Diagrama de domínio - Exportar PDF.	101
Figura 5.46: Diagrama de sequência - Exportar PDF.	103
Figura 5.47: Diagrama de classes - Exportar PDF.	104
Figura 5.48: Tela 9 - Protótipo não funcional de interface - Exportar XLS.	105
Figura 5.49: Diagrama de caso de uso - Exportar XLS.	105
Figura 5.50: Diagrama de domínio - Exportar XLS.	106
Figura 5.51: Diagrama de sequência - Exportar XLS.	108
Figura 5.52: Diagrama de classes - Exportar XLS.	109
Figura 5.53: Tela 10 - Protótipo não funcional de interface - Enviar Relatório por E-mail.	110
Figura 5.54: Diagrama de caso de uso - Enviar Relatório por E-mail.	111
Figura 5.55: Diagrama de domínio - Enviar Relatório por E-mail.	111
Figura 5.56: Diagrama de sequência - Enviar Relatório por E-mail.	114
Figura 5.57: Diagrama de classes - Enviar Relatório por E-mail.	115

Figura 5.58: Tela 11 - Protótipo não funcional de interface - Agendar Geração de Relatório.	116
Figura 5.59: Diagrama de caso de uso - Agendar Geração de Relatório.	117
Figura 5.60: Diagrama de domínio - Agendar Geração de Relatório.	117
Figura 5.61: Diagrama de sequência - Agendar Geração Relatório.	120
Figura 5.62: Diagrama de classes - Agendar Geração de Relatório.	121
Figura 5.63: Tela 12 - Protótipo não funcional de interface - Gerar Relatório Agendado.	122
Figura 5.64: Diagrama de caso de uso - Gerar Relatório Agendado.	122
Figura 5.65: Diagrama de domínio - Gerar Relatório Agendado.	123
Figura 5.66: Diagrama de classes - Gerar Relatório Agendado.	125
Figura 5.67: Tela 13 - Protótipo não funcional de interface - Consultar Relatórios Salvos.	126
Figura 5.68: Diagrama de caso de uso - Consultar Relatórios Salvos.	126
Figura 5.69: Diagrama de domínio - Consultar Relatórios Salvos.	127
Figura 5.70: Diagrama de sequência - Consultar Relatórios Salvos.	129
Figura 5.71: Diagrama de classes - Consultar Relatórios Salvos.	130
Figura 5.72: Tela 14 - Protótipo não funcional de interface - Salvar Relatório.	131
Figura 5.73: Diagrama de caso de uso - Salvar Relatório.	132
Figura 5.74: Diagrama de domínio - Salvar Relatório.	132
Figura 5.75: Diagrama de sequência - Salvar Relatório.	134
Figura 5.76: Diagrama de classes - Salvar Relatório.	135
Figura 5.77: Tela 15 - Protótipo não funcional de interface - Duplicar Relatório.	136
Figura 5.78: Diagrama de caso de uso - Duplicar Relatório.	137
Figura 5.79: Diagrama de domínio - Duplicar Relatório.	138
Figura 5.80: Diagrama de sequência - Duplicar Relatório.	140
Figura 5.81: Diagrama de classes - Duplicar Relatório.	141
Figura 5.82: Tela 17 - Protótipo não funcional de interface - Duplicar Entidade Composta.	142
Figura 5.83: Diagrama de caso de uso - Duplicar Entidade Composta.	142
Figura 5.84: Diagrama de domínio - Duplicar Entidade Composta.	143
Figura 5.85: Diagrama de sequência - Duplicar Entidade Composta.	145
Figura 5.86: Diagrama de classes - Duplicar Entidade Composta.	146
Figura 5.87: Tela 18 - Protótipo não funcional de interface - Salvar Filtros de Relatório.	147
Figura 5.88: Diagrama de caso de uso - Salvar Filtros de Relatório.	147
Figura 5.89: Diagrama de domínio - Salvar Filtros de Relatório.	148
Figura 5.90: Diagrama de sequência - Salvar Filtros de Relatório.	150
Figura 5.91: Diagrama de classes - Salvar Filtros de Relatório.	151

Figura 5.92: Tela 18 - Protótipo não funcional de interface - Mostrar Lista de Valores de Campo de Filtro.	152
Figura 5.93: Diagrama de caso de uso - Mostrar Lista de Valores de Campo de Filtro.	153
Figura 5.94: Diagrama de domínio - Mostrar Lista de Valores de Campo de Filtro.	153
Figura 5.95: Diagrama de sequência - Mostrar Lista de Valores de Campo de Filtro.	155
Figura 5.96: Diagrama de classes - Mostrar Lista de Valores de Campo de Filtro.	156
Figura 5.97: Tela 19 - Protótipo não funcional de interface - Eventos de Gerações de Relatórios no Dashboard.	157
Figura 5.98: Diagrama de caso de uso - Eventos de Gerações de Relatórios no Dashboard.	158
Figura 5.99: Diagrama de domínio - Eventos de Gerações de Relatórios no Dashboard.	158
Figura 5.100: Diagrama de sequência - Eventos de Gerações de Relatórios no Dashboard.	160
Figura 5.101: Diagrama de classes - Eventos de Gerações de Relatórios no Dashboard.	161
Figura 7.1: Diagrama de Pacotes do Sistema Desenvolvido.	170

LISTA DE TABELAS

Tabela 7.1: Requisitos Não Funcionais	48
Tabela 7.2: Requisitos Funcionais	49
Tabela 7.3: Caso de Uso Detalhado Autenticar Usuário	57
Tabela 7.4: Caso de Uso Detalhado Manter Usuário	62
Tabela 7.5: Caso de Uso Detalhado Manter Usuário (Administrador)	68
Tabela 7.6: Caso de Uso Detalhado Manter Entidade Composta	75
Tabela 7.7: Caso de Uso Detalhado Manter Relatório	84
Tabela 7.8: Caso de Uso Detalhado Buscar Relatórios	91
Tabela 7.9: Caso de Uso Detalhado Gerar Relatório	97
Tabela 7.10: Caso de Uso Detalhado Exportar PDF	102
Tabela 7.11: Caso de Uso Detalhado Exportar XLS	107
Tabela 7.12: Caso de Uso Detalhado Enviar Relatório por E-mail	112
Tabela 7.13: Caso de Uso Detalhado Agendar Geração de Relatório	118
Tabela 7.14: Caso de Uso Detalhado Gerar Relatório Agendado	124
Tabela 7.15: Caso de Uso Detalhado Consultar Relatórios Salvos	128
Tabela 7.16: Caso de Uso Detalhado Salvar Relatório	133
Tabela 7.17: Caso de Uso Detalhado Duplicar Relatório	139
Tabela 7.18: Caso de Uso Detalhado Duplicar Entidade Composta	144
Tabela 7.19: Caso de Uso Detalhado Salvar Filtros de Relatório	149
Tabela 7.20: Caso de Uso Detalhado Mostrar Lista de Valores de Campo de Filtro	154
Tabela 7.20: Caso de Uso Detalhado Listar Eventos de Gerações de Relatórios no Dashboard	159
Tabela 7.20: Matriz de Rastreabilidade Telas x UC	163
Tabela 7.21: Matriz de Rastreabilidade RF x UC	165
Tabela 7.22: Matriz de Rastreabilidade Mod. Domínio x UC	167

LISTA DE TRECHOS DE CÓDIGO

2.1	Python: Imprimindo uma lista ordenada com os itens com a primeira letra em maiúscula	24
2.2	Python: Exemplo de uso de funções da biblioteca padrão	24
2.3	Django: Linguagem de templates	27
2.4	Django: urls.py	27
2.5	Django: exemplo de uso do módulo <i>models</i>	28
2.6	Django: exemplo de uso do módulo <i>models</i> 2	28
2.7	Django: exemplo de uso do módulo <i>models</i> 2	29
2.8	Exemplo de utilização do módulo Django db	29
2.9	Estrutura básica de um documento HTML	30
2.10	Exemplo de código CSS	31
2.11	Exemplo de código Javascript	31
2.12	Arquivo de configuração mod_wsgi Apache	32
2.13	Arquivo de configuração python para o Apache	33
2.14	Exemplo do uso de JQuery no HTML	34
2.15	Comparação entre javascript puro e JQuery	34
2.16	Exemplo básico de uso do utilitário WkHTMLtoPDF	35
2.17	WkHTMLtoPDF - Exemplo de uso com arquivo HTML local	35

1 INTRODUÇÃO

Este trabalho apresenta um sistema que visa atender a crescente necessidade dos sistemas de gestão por extração de dados para geração de relatórios e gráficos de análise. Desta forma, são conceituados e apresentados conteúdos sobre análise de *schema* de banco de dados para verificação das possibilidades de amostragem dos dados filtrados. A partir destes conceitos, é apresentado o sistema que se propõe a mostrar as possibilidades de dados a serem listados, geração de regras, filtros e formatação dos dados, para geração de relatórios eficientes e personalizáveis. Para que isto aconteça, foi utilizado a metodologia de desenvolvimento ICONIX, visando obter artefatos para o processo de desenvolvimento do sistema.

Este trabalho tem como objetivo o desenvolvimento de uma ferramenta web que conecta-se à bancos de dados relacionais, extraindo meta-informações de seu *schema*, que é toda a estrutura de tabelas, colunas e características da base de dados usada pelo sistema a ser integrado. Estas informações então permitirão a definição de quais dados serão mostrados, bem como filtros para estes dados, regras para amostragem, e formatação dos mesmos, com o objetivo de obter um relatório personalizado e facilmente modificável em poucos minutos. As regras, restrições e formatações escolhidas serão salvas, sendo possível em um momento posterior acessar diretamente uma página de seleção de filtros para a geração do relatório, tornando-se transparente a tecnologia que fora usada para geração do mesmo.

Esta ferramenta poderá vir a substituir as outras ferramentas e técnicas usadas no momento nas empresas de informática para criação de relatórios, uma vez que está sendo projetada para ser produtiva e roda totalmente em ambiente web. A necessidade do conhecimento de programação de sistemas será reduzida aos usuários desta ferramenta, uma vez que ela é baseada em processos padronizados de configuração, com interfaces web e sem necessidade de programação.

As empresas de informática geralmente usam a linguagem de programação padrão do sistema para a criação de seus relatórios, o que costuma ser repetitivo. Além disso, requer envolvimento de um profissional com conhecimento do modelo de dados do software, e outro que conheça a linguagem de programação utilizada pela empresa e

saiba como usar a IDE para geração do relatório.

Conforme citado em (MATTOS, 2007): "Quando se trata de softwares, a satisfação dos usuários bem como suas necessidades não está relacionada a métricas exatas, e sim a características como confiabilidade, manutenibilidade, operabilidade, extensibilidade, escalabilidade, portabilidade, entre outras."Essas características são cada vez mais determinantes para a escolha do software a ser usado, e estão relacionadas diretamente com a exigência dos usuários. Por sua vez, para que os desenvolvedores possam fazer a melhor negociação nesse mercado de oferta e procura, é necessário conhecer o que há de melhor em plataformas de desenvolvimento, bem como as principais características que cada uma delas pode oferecer. Essas plataformas incluem:

- compiladores;
- ambiente de desenvolvimento - IDE;
- servidores de aplicação;
- sistemas gerenciadores de banco de dados - SGBDs;
- APIs e frameworks.

Para que se tenha uma ideia mais ampla das tecnologias e opções existentes no mercado é necessário o conhecimento de diversos itens, em que se destacam a evolução tecnológica e suas relações, os paradigmas de programação e os ambientes de desenvolvimento."Com base nestas afirmações, pode-se notar que existe grande exigência de conhecimento dos profissionais de TI para o desenvolvimento de softwares, onde todo recurso que puder ser economizado é de grande valor. No entanto, é possível o uso de uma ferramenta para mapeamento da estrutura do banco de dados, que permita ao usuário criação de regras, filtros e formatação de dados suficientes para simplificar a criação de relatórios, com facilidade e agilidade, podendo vir a dispensar a presença de um desenvolvedor do sistema. Este trabalho apresenta uma ferramenta para este fim.

Este trabalho está organizado da seguinte forma:

Serão abordados assuntos utilizados no desenvolvimento do sistema desenvolvido. Neste capítulo foi apresentado o objetivo que pretende-se alcançar com o sistema, são ainda identificadas as motivações da implementação de um sistema para emissão de relatórios.

No capítulo 2, é apresentado o referencial teórico que visa contextualizar o ambiente utilizado no desenvolvimento do trabalho, tais como, a importância da

informação, geração de informações a partir de sistemas de informática, e as tecnologias e metodologias envolvidas.

No capítulo 3, é apresentado o problema que visa ser solucionado com a ferramenta desenvolvida, enquanto no capítulo 4, é apresentada a proposta de solução, com a estrutura que liga as tecnologias envolvidas, formando assim o sistema como um todo.

No capítulo 5 será mostrado como é o funcionamento do sistema, juntamente com seu processo de desenvolvimento, onde, serão apresentadas as telas que do mesmo e qual as funcionalidades que cada uma tem. Também devem ser apresentados os diagramas que visam demonstrar a modelagem, arquitetura e o funcionamento do sistema. Para tal será usada a metodologia ICONIX, que será apresentada no capítulo 2, nas tecnologias envolvidas. Para finalizar esta etapa do trabalho será apresentado no capítulo 6 o cronograma de implementações, correções e finalização da segunda etapa, onde o sistema foi produzido e testado. E a partir das análises feitas em todos os capítulos anteriores, será apresentado no capítulo 7 as considerações obtidas com o desenvolvimento desta fase. Após isto, serão apresentadas as referências bibliográficas que foram utilizadas para o desenvolvimento deste trabalho.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os assuntos que fundamentam este trabalho, como os conceitos de geração de relatórios, as tecnologias envolvidas e o valor do poder de pesquisa e análise de informações.

2.1 A importância da informação

As informações, além de medirem o grau de comprometimento de uma entidade com o mercado, trazem vantagens diante da competitividade que há entre as empresas do ramo. Segundo (MEDEIROS, 2007), uma empresa tem seu grau de competitividade proporcional à importância que a mesma dá a informação. O autor ainda diz que as informações são as "molas-mestras" para a tomada de decisão, e uma decisão errônea pode acarretar sérias consequências ao desempenho da empresa como um todo. A busca da qualidade da informação deve então, ser uma constante no dia-a-dia das organizações, e ao armazenamento e ao processamento adequado da informação é atribuído um papel fundamental no âmbito de um sistema de informação. A partir desta afirmação, podemos concluir que as empresas com mais poderes de informação, são as que possuem os sistemas com as melhores fontes que permitam a extração da informação desejada. O mesmo autor também afirma que é necessário que um banco de dados tenha uma representação eficiente, que possibilite acesso a informações corretas, em tempo hábil.

Segundo (MATTOS,2010), não basta fornecer ou obter informações, é necessário fazê-las compreensíveis e compreendê-las. Porém, o que é uma informação importante para uma pessoa, pode não ser para outra, pois há tipos diferentes de interesse na mesma informação, que representada de outra maneira, ou ainda juntamente com outros dados, toma um formato diferente de representação.

Levando em consideração a subjetividade da relevância de informações, que varia de pessoa para pessoa, ou ainda de empresa para empresa, é importante dar ao usuário do sistema a liberdade de visualizar suas informações, gerar gráficos e exportar para formatos diversos.

2.2 Geração de informações a partir de sistemas de informática

Além de armazenar os dados da empresa, uma das ações mais importantes dos sistemas de informática é gerar informações e indicadores sobre o andamento da empresa como um todo, de acordo com a área que o software em questão se destina. (YASMINA, 2006) afirma que, "aplicações de suporte à gestão dos recursos humanos, gestão financeira e contábilística, gestão de estoques, gestão da produção, gestão de projetos, gestão logística, gestão de clientes e fornecedores, gestão da qualidade, entre outras, encontram-se perfeitamente enraizadas nas empresas. Além do suporte aos processos operacionais, estas aplicações geram uma quantidade avultada de informação, que deve ser utilizada pela organização para aumentar o conhecimento." A autora ainda afirma que as aplicações informáticas usadas pela organização são elementos essenciais da construção e partilha de conhecimento sobre si própria, o negócio que realiza, o seu exterior e a sua possibilidade de interação com as entidades externas relevantes para a empresa. Todo este conhecimento é fundamental para apoiar o processo de tomada de decisão. A informação gerada pelas aplicações informáticas disponibiliza aos gestores um conjunto de indicadores sobre o negócio, que lhe dão indicações do que aconteceu no passado e lhe permitem traçar cenários para o futuro.

(MATOS,2010) afirma que, "A informação é um dado acrescido de significado, contexto, e entendimento dos seus padrões e relações com outros dados, geralmente se utilizando de alguma forma de organização". A partir disto então, pode-se notar o papel fundamental dos sistemas informatizados para manipular e tratar diversos dados co-relacionados para transformá-los em informações relevantes. A tarefa repetitiva de transformar todos os dados, seja com fórmulas, junções ou representações, pode ser facilmente realizada por um sistema informatizado, uma vez que seja disponibilizado ao mesmo um padrão de tratamento para estes dados.

2.3 Sistemas Geradores de Relatórios

"Ter a informação correta no menor tempo possível é hoje o grande diferencial para as empresas que querem se manter na dianteira no mundo dos negócios.", (PRIMAK, 2008).

Para que esta afirmação seja realidade nas empresas, pode-se utilizar um sistema facilitador para a obtenção de todos os tipos de informação desejadas sobre o sistema. Ferramentas como esta são chamadas de sistemas geradores de relatórios.

Um sistema de geração de relatórios trata-se de uma ferramenta onde é possível, a partir de configurações e regras, que podem ser definidas ou feitas pelo próprio usuário, extrair informações, formatá-las e apresentá-las de forma dinâmica, podendo vir a dispensar a presença de profissionais da informática durante todo o

processo de criação dos relatórios.

Pode-se obter muitos benefícios com a utilização de um software para tal propósito, a seguir são apresentados alguns destes benefícios, conforme citado em (PRIMAK, 2012), onde são citados alguns benefícios do uso de um sistema de geração de relatórios.

- Incorporar os projetos de tecnologia com as metas estabelecidas pelas empresas na busca do máximo retorno do investimento;
- Compreender as tendências dos negócios, melhorando a consistência no momento de decisão de estratégias e ações a serem tomadas;
- Facilitar a identificação de riscos;
- Planejamento corporativo mais amplo;
- Facilitar o acesso e distribuir informação de modo mais amplo para obter envolvimento de todos dentro da empresa;
- Oferecer dados estratégicos para análise com um mínimo de atraso em relação a uma transação ou evento dentro da empresa;

Além disto, o autor ainda afirma que: "os gestores poderão ter acesso às informações rapidamente e poderão abreviar o tempo de resposta melhorando assim os processos decisórios. Dessa forma, a informação será o verdadeiro capital integralizado da empresa trazendo conhecimento para as decisões imediatas e para aquelas que virão no futuro."

As ferramentas existentes que se propõem a gerar relatórios dinâmicos, como por exemplo o Microsoft Crystal Reports e o Oracle Reports Builder, não oferecem interface e usabilidade para usuários finais, muitas vezes acostumados apenas com os sistemas de gestão das empresas onde trabalham. Desta forma as empresas ficam dependentes de profissionais com maior conhecimento em informática e em desenvolvimento de softwares para poder utilizar o sistema como desejado.

Com base no que foi apresentado nesta seção, a próxima seção começa a apresentar uma visão mais técnica de como é possível que estes sistemas funcionem.

2.4 Catálogo de meta-informações em bancos de dados

Os bancos de dados relacionais disponibilizam algumas tabelas onde estão armazenadas informações gerais sobre a estrutura atual do banco de dados como um todo. Nestas tabelas é possível obter, entre outros dados, os nomes de todos os *schemas*, que são as coleções de objetos divididos por usuário ou grupo, com todas

as suas tabelas, e estas tabelas, com todas as suas colunas, com informações tais como o tamanho de cada coluna, o tipo, o número de casas decimais (dependendo do tipo), e até se esta coluna faz parte de alguma chave primária ou estrangeira ou de algum índice ou constraint. Uma chave primária em um banco de dados é definida para uma ou mais colunas possuírem valores que não se repetem em uma tabela, servindo assim de referência para acesso direto ao registro desejado. Chaves estrangeiras são a referência da chave primária de uma tabela em outra, obtendo-se então uma relação entre duas tabelas. Já índices e constraints são estruturas para melhor performance no acesso aos dados e validações de integridade dos dados, respectivamente. Com estes dados, é possível fazer desde análises rigorosas em volumes de dados e quantidades de tabelas criadas por um período definido, até consultas e alterações de dados dinâmicas. Pode-se notar o uso destes recursos em ferramentas de mais baixo nível, como frameworks de desenvolvimento e ferramentas de auxílio a consultas em bancos de dados.

As tecnologias necessárias para o andamento deste trabalho são apresentadas nas próximas seções.

2.5 Tecnologias envolvidas

Para o desenvolvimento deste trabalho é necessário um aprofundamento a cerca dos conceitos gerais de emissão de relatórios em sistemas informatizados e das linguagens de programação, metodologias e tecnologias necessárias para a implementação. Nesta seção serão apresentadas as tecnologias envolvidas, tais como Python, Django e seus módulos, WkHTMLToPdf, apache, JQuery e ModWSGI, bem como as metodologias de análise e notações empregadas, Iconix e UML.

2.5.1 Python

Python (<http://python.org>) é uma linguagem de programação multiplataforma de alto nível. Segundo (ASCHER, 2004), "De modo geral, o Python eleva em alguns pontos a marca da qualidade no mundo dos scripts". Esta linguagem foi lançada por Guido van Rossum em 1991, e possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Segundo (BORGES, 2010) "A produtividade e expressividade das linguagens dinâmicas se encaixam perfeitamente com as metodologias ágeis, que nasceram do desenvolvimento de software de código aberto e defendem um enfoque mais pragmático no processo de criação e manutenção de software do que as metodologias mais tradicionais."

Este autor também cita que, "Entre as linguagens dinâmicas, o Python se destaca como uma das mais populares e poderosas. Existe uma comunidade movimentada

de usuários da linguagem no mundo, o que se reflete em listas ativas de discussão e muitas ferramentas disponíveis em código aberto.”

O fato do Python ser uma linguagem dinâmica facilita no desenvolvimento, e o deixa mais interessante e veloz. (THIAGO GALESI, 2010) diz que Python é uma linguagem dinâmica e com uma poderosa biblioteca padrão.

(ASCHER, 2004), diz que talvez o único inconveniente do Python seja que, conforme a implementação, sua velocidade de execução pode não ser tão rápida quanto as linguagens compiladas, como C e C++.

Este trabalho foi desenvolvido utilizando esta linguagem de programação, no que diz respeito ao processamento servidor. Esta escolha foi feita levando-se em consideração a alta produtividade, facilidade de manutenção e por ser uma tecnologia *open-source*. No trecho de código 2.1, pode ser visto um exemplo simples de uso da linguagem de programação Python.

Trecho de Código 2.1: Python: Imprimindo uma lista ordenada com os itens com a primeira letra em maiúscula

```
1 lista = ['laranja', 'banana', 'uva']
2 lista.sort()
3
4 for item in lista:
5     print item.capitalize()
```

Conforme pode ser visto no trecho de código 2.1, Python é uma linguagem poderosa e produtiva, onde em poucas linhas de código escrito, pode-se realizar a tarefa necessária. Pode-se observar no exemplo que a linguagem possui funções para ordenação de listas, e para formatação de strings (cadeias de caracteres).

Conforme citado por (BORGES, 2010), "É comum dizer que o Python vem com "baterias inclusas", em referência a vasta biblioteca de módulos e pacotes que é distribuída com o interpretador." É nestas bibliotecas onde o Python tem muito de sua produtividade, pois os métodos e funções comumente utilizados já estão implementados, prontos para uso.

Trecho de Código 2.2: Python: Exemplo de uso de funções da biblioteca padrão

```
1 # somar os cem primeiros números naturais
2 sum(x for x in range(101))
3
4 # retornar o quadrado de uma sequência
5 [x*x for x in range(1,11)]
6
7 # enviando e-mail
8 import smtplib
9 server = smtplib.SMTP('mail.xx.net')
10 server.sendmail('leonardo@xy.com', 'leonardo@xx.net',
11 """To: leonardo@xx.net
12 From: leonardo@xy.com
13 Subject: Teste de envio de email
```



```
14 Estou lhe enviando este email como um teste.
15 """
16
17 server.quit()
18 print 'Envio OK'
```

2.5.2 Django

A fim de facilitar e auxiliar no desenvolvimento deste trabalho, foi utilizado um framework chamado Django. Este framework foi de fundamental importância para a ferramenta que foi projetada, pois possui muitas características e módulos que foram muito usados.

Django é um framework para desenvolvimento web, totalmente escrito em Python, que utiliza o padrão MVC (model - view - controller, <http://www.dsc.ufcg.edu.br/jacques/cursos/map/html/arqu/mvc/mvc.htm>). Foi criado originalmente como sistema para gerenciar um site jornalístico na cidade de Lawrence, no Kansas. Tornou-se um projeto de código aberto e foi publicado sob a licença BSD em 2005. O nome Django foi inspirado no músico de jazz Django Reinhardt.

Django utiliza o princípio DRY (*Don't Repeat Yourself*), onde faz com que o desenvolvedor aproveite ao máximo o código já feito, evitando a repetição. Isto, segundo (THIAGO GALES, 2010), quer dizer que, se você seguir determinadas convenções na maneira de organizar seu código, não precisará ficar configurando características específicas de seu software. Isto agiliza muito o desenvolvimento. Será necessário utilizar configurações um pouco mais complexas somente quando o comportamento padrão do Django não é o esperado, ou quando não é possível seguir convenções criadas pelo framework, algo raro de acontecer. Este framework será utilizado no desenvolvimento deste trabalho, levando em consideração algumas de suas mais importantes características, que são a produtividade, a clareza do código-fonte e o fato de ser multiplataforma.

Dentre as principais características que tornaram o Django um framework web cada vez mais usado no mundo inteiro podem-se citar:

- Mapeamento Objeto-Relacional (ORM) - Com o ORM do Django é possível definir a modelagem de dados através de classes em Python. Com isso é possível gerar suas tabelas no banco de dados e manipulá-las sem necessidade de utilizar SQL (o que também é possível). Este assunto será tratado mais detalhadamente nas seções sobre Django *Models*.
- Interface Administrativa - No Django é possível gerar automaticamente uma interface para administração para os modelos criados através do ORM.
- Sistema de Templates - O Django tem uma linguagem de templates poderosa, extensível e amigável. Com ela é possível separar design, conteúdo e código

em Python. Um exemplo da linguagem de templates do Django pode ser visto no trecho de código 2.3.

Trecho de Código 2.3: Django: Linguagem de templates

```

1 <html>
2 <head><title>Aviso de compra</title></head>
3
4 <body>
5
6 <h1>Aviso de compra</h1>
7
8 <p>Caro {{ cliente_nome }},</p>
9
10 <p>Obrigado por comprar na {{ companhia }}. Sua entrega est´a marcada para
    {{ ship_date|date:"F j, Y" }}.</p>
11
12 <p>Seguem abaixo os itens comprados:</p>
13
14 <ul>
15 {% for item in lista_de_itens %}
16     <li>{{ item }}</li>
17 {% endfor %}
18 </ul>
19
20 {% if comprou_garantia_de_entrega %}
21     <p>Sua entrega est´a assegurada por nossa companhia.</p>
22 {% else %}
23     <p>Nˆao foi contratada garantia de entrega, portanto fique ateno ao
        rastreo de seus produtos.</p>
24 {% endif %}
25
26 <p>Sinceramente,<br />{{ companhia }}</p>
27
28 </body>
29 </html>

```

- URLs Elegantes - O sistema de mapeamento de *URL* (Uniform Resource Locator (Localizador-Padrˆao de Recursos)) ´e feito atrav´es de expressˆoes regulares, o que torna as urls totalmente personaliz´aveis, pois nˆao ´e necess´ario seguir os padrˆoes de parˆametros GET, por exemplo. Um exemplo pr´atico ´e mostrado no trecho de c´odigo 2.4, onde a url que come¸a com a string “mostra_hora”, tem seu processamento para resposta direcionado para a fun¸˜ao mostra_hora.

Trecho de C´odigo 2.4: Django: urls.py

```

1 from django.conf.urls import patterns, url, include
2
3 urlpatterns = patterns('',
4     (r'^mostra_hora/', 'relogio.views.mostra_hora'),
5 )

```

2.5.3 Django *Models*

Models ´e um m´odulo do Django que possibilita criar classes de objetos, que serˆao convertidos em tabelas no banco de dados relacional, abstraindo o desenvolvedor do

uso de *SQL*. Um *model* contém a representação dos dados do projeto. Normalmente é a parte que conversa com o banco de dados e que persiste os dados.

Como pode ser visto no trecho de código 2.5, a sintaxe é de criação de uma classe de objeto qualquer, porém seguindo alguns padrões:

Trecho de Código 2.5: Django: exemplo de uso do módulo *models*

```

1 from django.db import models
2
3 class Pessoa(models.Model):
4     nome = models.CharField(max_length=30)
5     sobrenome = models.CharField(max_length=30)

```

Após isto, é possível criar, editar e remover uma ou mais instâncias da classe Pessoa do banco de dados, sem o uso de *SQL*, exemplificado no trecho de código 2.6.

Trecho de Código 2.6: Django: exemplo de uso do módulo *models* 2

```

1 from projeto.app.models import Pessoa
2 # Criando uma Pessoa
3 pessoa = Pessoa()
4 pessoa.nome = "Eduardo"
5 pessoa.sobrenome = "Silva"
6 pessoa.save()
7
8 # Alterando uma pessoa
9 pessoa = Pessoa.objects.get(nome="Eduardo")
10 pessoa.sobrenome = "Erlo"
11 pessoa.save()
12
13 # Removendo uma Pessoa
14 pessoa = Pessoa.objects.get(nome="Eduardo")
15 pessoa.remove()

```

Este módulo é utilizado para acessar apenas a estrutura das tabelas que são utilizadas pela ferramenta deste trabalho. Os dados do banco de dados do sistema a ser integrado serão lidos de tabelas já existentes, não sendo recomendado o uso deste módulo para acesso as mesmas. Além disso, o uso do *SQL* se faz necessário para executar consultas mais elaboradas e dinâmicas, que é o que se propõe a fazer a ferramenta desenvolvida neste trabalho.

2.5.4 Django-Introspection

A fim de obter-se informações sobre a estrutura geral do banco de dados a ser integrado com a ferramenta desenvolvida, foi utilizado o módulo de consulta de meta-informações sobre a estrutura de banco de dados, que é chamado de *django-introspection*. Este módulo tem a funcionalidade de retornar os nomes das tabelas e suas respectivas colunas, com seus tipos, tamanhos e *constraints*. O uso deste módulo se faz útil, pois a forma de obter estas informações varia de acordo com o banco de

dados, por exemplo: Para o caso do banco de dados PostgreSQL, segundo (GROU, 2012), para obtermos as meta-informações sobre o mesmo, deve ser usada a tabela *tables* do *schema*, onde o Postgres armazena seus meta-dados, dos quais é possível obter o nome das tabelas, sendo feita então uma junção com a tabela *columns*, para obtermos informações relativas as colunas. Utilizando o *django-introspection*, o método utilizado para obter tais informações será totalmente abstraído do trabalho, facilitando o desenvolvimento e mantendo o máximo possível de portabilidade entre bancos de dados.

No trecho de código 2.7 é apresentado um exemplo básico de como obter a lista de tabelas disponíveis no banco de dados ao qual se está conectado, e logo após isto, buscando as informações da tabela clientes.

Trecho de Código 2.7: Django: exemplo de uso do módulo *models* 2

```

1 from django.db import connection
2 # retorna a lista(vetor) das tabelas disponíveis no banco de dados para dentro
  da variável lista_de_tabelas
3 cursor = connection.cursor()
4 lista_de_tabelas = connection.introspection.get_table_list(cursor)
5
6 # retorna uma lista(vetor) das colunas da tabela clientes
7 connection.introspection.get_table_description(cursor,"clientes")

```

2.5.5 Django db

Para o acesso aos bancos de dados aos quais a aplicação desenvolvida irá conectar-se, será utilizado o módulo *db(database)*, incluído no pacote padrão do *django*. Desta forma, a conexão e execução de todas as consultas necessárias durante todo o uso da ferramenta será feita da mesma forma padronizada, como é mostrado no trecho de código 2.8, onde é obtida uma lista do banco de dados, com os nomes de todos os clientes cadastrados:

Trecho de Código 2.8: Exemplo de utilização do módulo Django *db*

```

1 # importa a conexão com o banco de dados
2 from django.db import connection
3
4 # obtém-se um cursor para execução de transações
5 cursor = connection.cursor()
6
7 # executa uma consulta dos nomes dos clientes
8 cursor.execute("select nome from clientes")
9
10 # faz o fetch da consulta para dentro da variável retorno
11 resultado = cursor.fetchall()
12 cursor.close()

```

Como pode ser visto no trecho de código 2.8, todos acessos e consultas necessários são totalmente abstraídos por um único padrão de escrita definido pelo framework

web Django.

2.5.6 HTTP, HTML, CSS e Javascript

O funcionamento básico das páginas Web se dá através de quatro tecnologias, são elas: *HTTP* (HyperText Transfer Protocol), *HTML* (HyperText Markup Language), *CSS* (Cascading Style Sheets) e Javascript. Abaixo será detalhado um pouco de cada uma destas tecnologias.

- **HTTP** - Hyper Text Transfer Protocol (Protocolo de Transferência de Hipertexto) é o protocolo de comunicação padrão da Web, que torna possível o funcionamento de todas as páginas da internet.
- **HTML** - Segundo (CONSORTIUM, 2012), HTML é a linguagem de publicação da World Wide Web. O HTML é um acrônimo para a expressão inglesa HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto. É uma linguagem de marcação utilizada para produzir páginas na Web. Documentos HTML são interpretados por navegadores Web, tais como Mozilla Firefox, Google Chrome, Opera e Microsoft Internet Explorer. Esta linguagem de marcação possui diversos elementos e possibilidades de combinação destes elementos. A estrutura básica de um documento HTML é mostrado no trecho de código 2.9.

Trecho de Código 2.9: Estrutura básica de um documento HTML

```
1 <html>
2 <head>
3   ...
4   ...
5 </head>
6 <body>
7   ...
8   ...
9 </body>
10 </html>
```

- **CSS** - Segundo (CONSORTIUM, 2012), as Folhas de Estilo em Cascata (CSS - Cascading Style Sheets) consistem num mecanismo simples para adicionar estilo (como sejam tipos de letra, cores ou espaçamento) a documentos web. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento. No trecho de código 2.10 é mostrado um exemplo do uso de CSS, onde está sendo modificada a fonte de todo o conteúdo do documento HTML.

Trecho de Código 2.10: Exemplo de código CSS

```

1 <style type="text/css">
2     body * {
3         font-size: 12em;
4     }
5 </style>

```

- Javascript - JavaScript é uma linguagem de script, atualmente a principal linguagem para programação em navegadores Web. Esta linguagem é baseada na linguagem ECMAScript(<http://www.ecma-international.org/>). É uma linguagem com orientação a objetos baseada em protótipos, tipagem fraca e dinâmica. Conforme pode ser visto no trecho de código 2.11, Javascript trata-se de uma linguagem de alto nível, muito poderosa quando trata-se de manipular elementos do documento HTML, adicionando-se validações, animações, eventos e muito mais.

Trecho de Código 2.11: Exemplo de código Javascript

```

1 <script type="text/javascript">
2     var tam=11;
3     function mudaFonte(tipo)
4     {
5         if (tipo=="mais")
6         {
7             if(tam<20) tam+=1;
8         }else
9         {
10            if(tam>10) tam-=1;
11        }
12        document.getElementById('tx').style.fontSize=tam+'px' ;
13    }
14 </script>

```

2.5.7 Apache HTTPD

Apache HTTPD é um servidor de aplicação web, que processa solicitações HTTP (Hyper-Text Transfer Protocol), o protocolo padrão da web. Ele possibilita servir páginas WEB de forma dinâmica.

Além de estar disponível para o Linux (e para outros sistemas operacionais baseados no Unix), o Apache também conta com versões para o Microsoft Windows.

Segundo (MARCELO, 2008), O Apache é um dos mais robustos e seguros programas desenvolvidos para ambientes *TCP/IP* (Transmission Control Protocol/Internet Protocol).

O Apache também é uma ferramenta extensível através do uso de módulos, que podem ser integrados sendo instalados no mesmo servidor onde o Apache está instalado. Isto permite que praticamente todas as funcionalidades do Apache possam ser alteradas ou ainda sejam criadas novas funcionalidades e/ou possibilidades

de integração. Para exemplificar o uso de módulos no Apache, podemos citar o módulo chamado `mod_ssl`, que é usado para garantir segurança nas transações HTTP, quando ativado, adicionando a capacidade de atender requisições utilizando o protocolo HTTPS(HTTP seguro).

(FORD, 2009) diz que o Apache é extremamente flexível, oferecendo a capacidade de um único servidor suportar diversos Websites como hosts virtuais, que são as capacidades de hospedagem de vários domínios em um único servidor.

O Apache é um serviço livre e sem custos, com possibilidade de ter sua funcionalidade estendida através de módulos. Sendo então este o principal motivo da escolha deste serviço para servir as páginas desta ferramenta. Existem módulos para servir sites desenvolvidos utilizando o framework Django, entre os mais usados, podemos citar o `MOD_PYTHON` e o `MOD_WSGI`, ambos open-source e sem custos para o uso.

2.5.8 Mod_WSGI

Para que o servidor Apache HTTPD possa servir websites desenvolvidos no framework Django, é necessário que seja instalado juntamente com o mesmo, algum módulo que possibilite esta integração. Para o desenvolvimento deste trabalho foi escolhido o módulo chamado `MOD_WSGI`.

Segundo (FOUNDATION, 2012a), o `MOD_WSGI` pode ser usado para hospedar qualquer aplicação Python que suporta a interface Python WSGI, incluindo o Django, que por sua vez funciona com qualquer versão do Apache que suporta `mod_wsgi`.

Para isto então, é usado o arquivo de configuração apresentado no trecho de código 2.12:

Trecho de Código 2.12: Arquivo de configuração `mod_wsgi` Apache

```

1 WSGIScriptAlias /minha_home_page "/path/do/meu/projeto/mais/o/arquivo/django.
   wsgi"
2
3 <Directory "/path/do/diretorio/apache/">
4     Order deny,allow
5     Allow from all
6 </Directory>
7
8 Alias /media "/path/de/um/diretorio/com/arquivos/estaticos"
9
10 <Location "/media">
11     SetHandler None
12     Order deny,allow
13     Allow from all
14 </Location>
15
16 Alias /arquivos "/path/de/outro/diretorio/com/arquivos/estaticos"
17
18 <Location "/arquivos">
```



```

19     SetHandler None
20     Order deny,allow
21     Allow from all
22 </Location>

```

Então é criado um arquivo dentro do projeto Django, com as configurações apresentadas no trecho de código 2.13:

Trecho de Código 2.13: Arquivo de configuração python para o Apache

```

1 import os, sys
2
3 PROJECT_ROOT_PATH = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
4 sys.path.insert(0, PROJECT_ROOT_PATH)
5 os.environ['DJANGO_SETTINGS_MODULE'] = 'settings'
6
7 import django.core.handlers.wsgi
8 application = django.core.handlers.wsgi.WSGIHandler()

```

Desta forma, o servidor Web Apache estará funcionando para acesso pela porta 80, que é a porta padrão da Web.

2.5.9 JQuery

Muitas vezes, o desenvolvimento de animações, validações e recursos para melhorar a experiência do usuário com a página Web na qual ele está navegando pode se tornar repetitivo. Para isto, foram criadas diversas bibliotecas javascript, que implementam funções muito usadas no dia-a-dia de um desenvolvedor, de uma maneira mais fácil de ser usada, evitando repetições de código. A fim de facilitar o desenvolvimento *client-side* desta ferramenta, foi escolhida então uma biblioteca chamada JQuery, que supriu as necessidades deste desenvolvimento por facilidade, produtividade e operabilidade nos navegadores Web mais usados. jQuery é uma biblioteca JavaScript, desenvolvida para executar nos browsers mais usados no mercado, simplificando os scripts do lado cliente que interagem com o HTML. Os scripts desenvolvidos com JQuery, são padronizados para serem escritos de uma forma apenas, sem a necessidade de preocupação de qual browser executará o script em questão. Esta biblioteca foi lançada em janeiro de 2006, sendo uma biblioteca de código aberto. A sintaxe do jQuery foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos e desenvolver aplicações AJAX. A biblioteca também oferece a possibilidade de criação de plugins sobre ela. Fazendo uso de tais facilidades, os desenvolvedores podem criar camadas de abstração para interações de mais baixo nível, simplificando o desenvolvimento de aplicações web dinâmicas de grande complexidade. Segundo (SILVA, 2012), esta biblioteca foi criada por John Resig e disponibilizada como software livre e aberto. Para que o código-fonte desenvolvido em jquery funcione, é necessário efetuar o download da biblioteca em

seu site oficial(<http://jquery.com>), e adicioná-lo como referência na página HTML desejada, conforme é mostrado no trecho de código 2.14.

Trecho de Código 2.14: Exemplo do uso de JQuery no HTML

```

1 <html>
2 <head>
3   ...
4   <script src="jquery.js"></script>
5   ...
6 </head>
7 <body>
8   ...
9   ...
10  ...
11 </body>
12 </html>

```

No trecho de código 2.15 é mostrado um exemplo de comparação do uso de apenas javascript, com o uso de JQuery.

Trecho de Código 2.15: Comparação entre javascript puro e JQuery

```

1 // MODIFICAR O VALOR DE UM INPUT NO DOM
2
3 // Em javascript puro
4 document.getElementById( 'Teste' ).value = 5;
5 // O mesmo código em JQuery
6 $( '#Teste' ).val( 5 );

```

Aproveitando-se do poder desta biblioteca javascript, existem inúmeros plugins desenvolvidos pelos programadores web, e disponibilizados na internet gratuitamente. Estes plugins vão de simples efeitos e menus, até visualizadores de fotos e manipuladores de imagens complexos, ou ainda elementos de UI ou gráficos. Durante o desenvolvimento deste trabalho, foram eventualmente utilizados plugins da biblioteca JQuery, que aumentam a usabilidade do sistema e são disponibilizados livremente na internet.

2.5.10 MatPlotLib

A fim de melhorar e facilitar a visualização dos dados gerados, tornando a visualização e a expressão das informações geradas mais rica, a ferramenta desenvolvida neste trabalho tem a opção de geração de gráficos, dependendo dos tipos de dados e cruzamentos selecionados pelo usuário na emissão do relatório. Estes gráficos ficam então anexados ao conteúdo gerado.

Matplotlib é uma biblioteca para geração de gráficos matemáticos, desenvolvida para a linguagem de programação Python, trabalhando juntamente com a biblioteca NumPy, que por sua vez é uma extensão do Python para execução cálculos matemáticos complexos. A biblioteca Matplotlib disponibiliza uma *API* (Application

Programming Interface) orientada a objetos que permite a criação de imagens que contém gráficos, podendo então ser perfeitamente acoplada nas páginas Web da ferramenta que está sendo proposta.

2.5.11 WkHTMLtoPDF - HTML to PDF Converter

Os relatórios gerados pela ferramenta terão a opção de exportação de seus resultados para o formato externo PDF. Para esta tarefa será utilizado um utilitário chamado WkHTMLtoPDF. WkHTMLtoPDF é um pequeno programa multiplataforma com interface de linha de comando para renderizar páginas HTML utilizando a engine *WebKit* (um motor de renderização utilizado nos navegadores web Google Chrome (<https://www.google.com.br/chrome/>) e Apple Safari (<http://www.apple.com/safari/>) para renderizar páginas) de uma página web em PDF.

Seu uso é feito com o seguinte comando:

Trecho de Código 2.16: Exemplo básico de uso do utilitário WkHTMLtoPDF

```
1 /caminho_completo_do_executavel_compilado http://url.com.br /caminho_completo/
  arquivo.pdf
```

No caso de um arquivo HTML local, com referências para arquivos CSS e Javascript externos locais ou remotos, usa-se da seguinte forma, mudando apenas a forma de passar-se o parâmetro url:

Trecho de Código 2.17: WkHTMLtoPDF - Exemplo de uso com arquivo HTML local

```
1 /caminho_completo_do_executavel_compilado file://caminho_completo_do_arquivo.
  html /caminho_completo/arquivo.pdf
```

Até aqui, foram apresentadas as tecnologias utilizadas para a implementação da solução, a seguir são apresentadas as tecnologias para a modelagem da solução.

2.5.12 ICONIX

O ICONIX é uma metodologia de desenvolvimento de software promovido pela empresa ICONIX Software Engineering, cujo foco de negócio reside na formação e produção de material para educação em tecnologias de objetos, em particular em CORBA, COM, Java e UML.

Conforme é mencionado por (DOUG ROSENBERG, 2012), em teoria, cada aspecto da *UML* (Unified Modeling Language (Linguagem de Modelagem Unificada)) é potencialmente útil, mas na prática, nunca parece haver tempo suficiente para fazer a modelagem, análise e design.

Segundo (CARLOS VIDEIRA, 2008), O ICONIX define-se como um processo de desenvolvimento de software prático, posicionando-se entre a complexidade e

abrangência do *RUP* (Rational Unified Process (Processo Unificado Racional)) e a simplicidade e o pragmatismo do *XP* (eXtreme Programing (Programação Extrema)) (Extreme Programming).

A Figura 2.1 ilustra a visão geral do ICONIX.

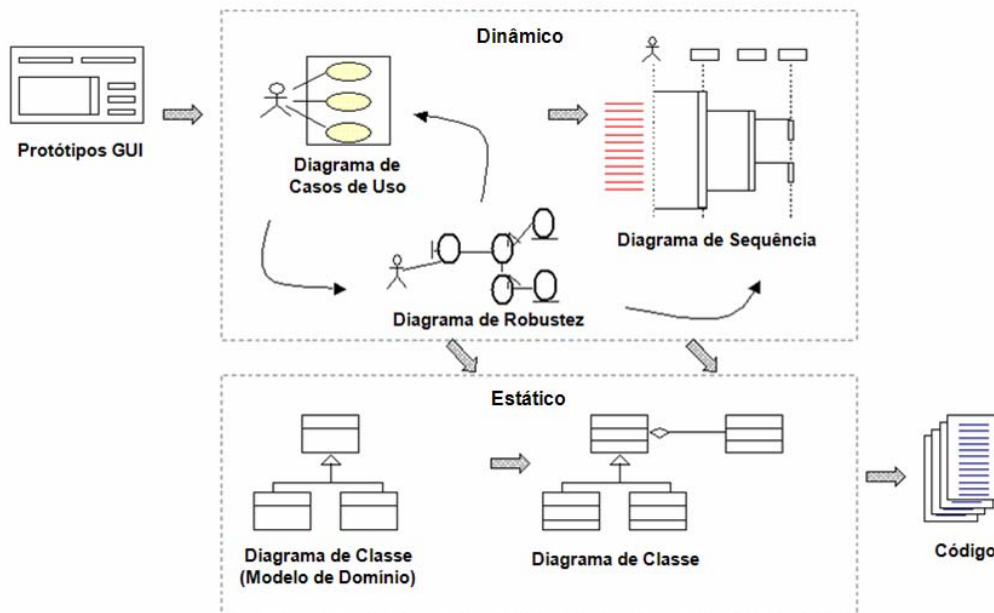


Figura 2.1: Fluxo Entre as Fases do Processo ICONIX.

Segundo (CARLOS VIDEIRA, 2008) , o ICONIX lança um conjunto de avisos relativamente a problemas e dúvidas comuns, que deverão ser levados em conta. Uma lista de forma resumida destes avisos é mostrada a seguir, e tem como principal objetivo evitar a perda de tempo com detalhes desnecessários.

- Não perder muito tempo com a inspeção gramatical.
- Endereçar a agregação e composição apenas na fase do desenho detalhado.
- Não começar a escrever os *UC* (Use Case (Caso de Uso)) até se conhecer bem como é que os atores irão atuar.
- Não passar semanas a construir modelos de casos de uso elaborados e bem desenhados, mas a partir dos quais não é possível construir-se um adequado desenho de classes.
- Não perder muito tempo em discussões sobre quando e onde usar relações "include" ou "extend".
- Não procurar fazer desenho detalhado nos diagramas de robustez

- Não começar a desenhar um diagrama de sequência antes de se ter completado o diagrama de robustez correspondente.
- Não modelar o que não é necessário modelar.

A figura 2.2 ilustra os passos necessários para a criação dos documentos de análise utilizando o processo ICONIX.

Passo	Descricao
1	Identificar os objetos do mundo real de domínio (modelagem de domínio).
2	Definir os requisitos comportamentais (casos de uso).
3	Executar análise de robustez.
4	Atribuir comportamento aos seus objetos (diagramas de sequência).
5	Concluir o modelo estático (diagrama de classes).
6	Escrever / gerar o código (código fonte).
7	Execute sistema e testes de aceitação do usuário.

Figura 2.2: Fluxo Entre as Fases do Processo ICONIX.

Por ser customizável, o processo ICONIX pode ser aplicado a equipes de desenvolvimento de qualquer tamanho. Esta metodologia será utilizada neste trabalho, com o objetivo de minimizar a complexidade e organizar o desenvolvimento.

O processo ICONIX é totalmente aberto e livre para ser usado em projetos.

2.5.13 UML - Unified Modeling Language(Linguagem Unificada de Modelagem)

”A UML (Unified Modeling Language ou Linguagem de Modelagem Unificada) é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem de software adotada internacionalmente pela indústria de Engenharia de Software.”, (GUEDES, 2008).

O autor ainda diz que ”por mais simples que seja, todo e qualquer sistema deve ser modelado antes de se iniciar a implementação, entre outras coisas, por que os sistemas de informação freqüentemente costumam possuir a propriedade de ”crescer”, isto é, aumentar em tamanho, complexidade e abrangência. Muitos profissionais costumam afirmar que sistemas de informação são ”vivos”, por que nunca estão completamente finalizados. Na verdade, o termo correto seria ”dinâmicos”, por que normalmente os sistemas de informação estão em constante mudança.”

Segundo (FOWLER, 2005), UML é uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de *software*, particularmente daqueles construídos utilizando o estilo orientado a objetos(OO).

Abaixo são mostradas as definições/objetivos de alguns dos principais itens que compõem a UML.

- Diagramas de Atividades - Segundo (CAMPINA GRANDE, 2012), o objetivo do diagrama de atividades é mostrar o fluxo de atividades em um único processo. O diagrama mostra como uma atividade depende da outra.
- Diagramas de Classes - Tem como objetivo descrever os vários tipos de objetos no sistema e o relacionamento entre eles, (CAMPINA GRANDE, 2012).
- Diagramas de Componentes - Mostra a dependência entre os componentes de software.
- Diagramas de Estruturas Compostas - Mostra o relacionamento entre os elementos, comunicação entre as classes e interface para a implementação das mesmas.
- Diagramas de Distribuição - Exibe os nós do sistema e suas conexões bem como a alocação dos processo para os nós.
- Visão Geral da Interação - Uma variação do diagrama de atividades mais o diagrama de sequência.
- Diagramas de Objetos - Como o diagrama de classes, porém mostra os objetos que foram instanciados das classes.
- Diagramas de Pacotes - Segundo (CAMPINA GRANDE, 2012), consiste no agrupamento das classes em pacotes.
- Diagramas de Sequência - Segundo (CAMPINA GRANDE, 2012), Consiste em um diagrama que tem o objetivo de mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização de uma operação.
- Diagramas de Máquinas de Estado - Procura acompanhar as mudanças de estado das instâncias em um caso de uso.
- Diagramas de Casos de Uso - Segundo (CAMPINA GRANDE, 2012), o Diagrama de Casos de Uso tem o objetivo de auxiliar a comunicação entre os analistas e o cliente. Um diagrama de Caso de Uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. O cliente deve ver no diagrama de Casos de Uso as principais funcionalidades de seu sistema

Como o ICONIX permite customização e uso somente dos artefatos necessários, optou-se por realizar a modelagem usando somente os diagramas abaixo listados:

- Protótipo de Tela - Usados para facilitar o entendimento do usuário quanto aos passos necessários para o andamento do processo.
- Diagrama de Casos de Uso - Divisão das responsabilidades entre usuário e sistema, facilitando o entendimento quanto as ações a serem tomadas para realizar as tarefas.
- Diagrama de Classes - Para facilitar o entendimento de como os objetos estão modelados e de como se relacionam.
- Diagrama de Sequência - Mostra como os objetos se comunicam para a realização das tarefas.
- Matrizes de Rastreabilidade - Facilita o entendimento de como os casos de uso atendem aos processos, requisitos e telas do sistema.

Nos casos de uso, há uma descrição dos mesmos de acordo com a metodologia e modelos apresentados em (LARMAN, 2008), conforme a figura 2.3.

Seção de caso de uso	Comentário
Nome do Caso de Uso	Começar com um verbo.
Escopo	O sistema em projeto.
Nível	"Objetivo do usuário" ou "sub-função".
Ator principal	Chama o sistema para fornecer os serviços.
Interessados e Interesses	Quem se importa com este caso de uso e o que eles desejam?
Pré-condições	O que precisa ser verdade de início e vale a pena dizer ao leitor?
Garantia de Sucesso	O que precisa ser verdade quando da finalização bem-sucedida e se vale a pena dizer ao leitor?
Cenário de Sucesso Principal	Um caminho típico, incondicional e otimista do cenário de sucesso.
Extensões	Cenários alternativos de sucesso ou fracasso.
Requisitos Parciais	Requisitos não funcionais relacionados
Lista de variantes tecnológicas e de dados	Métodos de entrada e saída e formatos de dados variáveis.
Frequência de ocorrência	Influencia a investigação, teste e oportunidade de implementação.
Diversos	Como, por exemplo, pontos em aberto.

Figura 2.3: Modelo Larman.

Na figura 2.3 pode-se observar o modelo proposto por (LARMAN, 2008). Na figura 2.4 é exibido o modelo utilizado.

Este capítulo apresentou as tecnologias que serão utilizadas no desenvolvimento deste trabalho. No capítulo 5 será exposto o problema para o qual se busca uma solução.

UC-xx	
Nome	Nome do caso de uso
Descrição	Descrição do caso de uso
Atores	Atores do caso de uso
Prioridade	Prioridade do caso de uso
Requisitos Funcionais Associados	Lista dos requisitos funcionais associados
Pré-condições	Condições para que este caso de uso possa acontecer
Pós-condições	Estado após finalizar o caso de uso
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
Ação 1	
	Ação 2
Cenário Alternativo 1	
	3.1 Verificou que um ou mais campos obrigatórios não estão preenchidos. 3.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando junto ao campo que o mesmo deve ser preenchido.
Cenário Alternativo 2	
	Alternativa 1

Figura 2.4: Modelo utilizado neste trabalho.

3 PROBLEMA

Poucos sistemas de gestão possuem funcionalidades suficientes para suprir a crescente necessidade por informação de seus clientes. Quando possuem as informações apresentadas são estáticas, tendo grande chance de não serem realmente o que o cliente deseja. A interpretação dos dados pode variar de acordo com o perfil da pessoa, pois há diferentes interesses, culturas de interpretação, gostos pessoais, etc.

Segundo (BECK, 2004), o problema básico no desenvolvimento de software é o risco. Alguns destes riscos são:

- Deslizes no cronograma;
- Projeto cancelado;
- O custo das modificações é muito elevado, e a taxa de erros cresce tanto que o sistema é substituído;
- Taxa de erros;
- Negócio mal compreendido;
- Modificações nos negócios;
- Falsa riqueza de funções;
- Rotatividade da equipe;

Em quase todas as situações, tanto no desenvolvimento de relatórios como em outras formas de visualização das informações, é o dono da informação quem decide como estas informações devem ser tratadas para que se tornem relevantes para sua necessidade. Isto leva quase sempre à customização de relatórios, fazendo com que o programador de sistemas sirva apenas como uma "ponte" entre o conhecimento do cliente sobre seu negócio, e o conhecimento técnico necessário para apresentar os dados recuperadas do banco de dados da maneira adequada. Dentro das empresas de desenvolvimento de sistemas, muitas horas-homem são usadas realizando customizações para seus clientes, ajustes de mudança de legislação,

ou ainda apenas para fazer com que as informações estejam dispostas de forma relevante para um ou mais clientes. Segundo (KOTLER, 2000), compreender as necessidades e os desejos dos clientes nem sempre é uma tarefa fácil. Alguns consumidores têm necessidades das quais não tem plena consciência, não conseguem articular essas necessidades, ou então empregam palavras que exigem alguma interpretação. Portanto muitas destas horas são dispendidas para alterar o que já estaria pronto, por não estar de acordo com o que todos realmente queriam ou estavam imaginando. Estas horas poderiam ser usadas para o desenvolvimento de novas soluções, para correção de problemas em outras partes do sistema, ou ainda investidas em atendimento e suporte para satisfação total dos clientes.

Muitas empresas de software contam com funcionários capacitados em suporte e consultoria ao negócio em que o cliente atua. Quando isto ocorre, o cliente, além de contar com uma empresa especializada que fornece o software, conta ainda com o conhecimento sobre como organizar e estruturar melhor suas informações para melhorar seus processos. Nestes casos, as definições de negócio, e de como deve ser a amostragem de dados dos relatórios é definida pelo cliente e/ou os consultores e analistas de negócio especializados naquele ramo de atuação, havendo então a passagem deste conhecimento para que os desenvolvedores possam implementar o que for necessário.

Porém esta passagem de conhecimento requer tempo de todos os envolvidos. Isto também ocorre nas mudanças de opinião que surgem durante o projeto, que por sua vez também geram informações adicionais, que devem ser passadas para os desenvolvedores. Há ainda o risco que durante e após o processo pode ocorrer diferença de idéias e falha na passagem de informações entre profissionais, o que pode gerar um produto inesperado e sem utilidade para o cliente final. É possível perceber um padrão no desenvolvimento de relatórios, por mais completos que sejam, no que diz respeito a implementação dos mesmos. Estes padrões técnicos que são utilizados no desenvolvimento dos relatórios podem ser implementados por um sistema de interface gráfica familiar aos clientes e consultores, onde fosse possível selecionar os dados a serem buscados do banco de dados, formatados, filtrados e mostrados em tela.

Segundo (IVAN LUIZIO MAGALHÃES; PINHEIRO, 2007), "a cada dia que passa, as organizações tornam-se mais dependentes da Tecnologia da Informação a fim de satisfazer seus objetivos estratégicos e para atender às necessidades do negócio em que atuam. Uma área de TI que não considerar os objetivos estratégicos da organização em que se insere como os seus próprios objetivos, será uma área de TI que deseja apenas ser um simples provedor de tecnologia, haja vista que até mesmo os provedores de tecnologia, atualmente, tendem a preocupar-se com a estratégia de

negócio de seus clientes, condição básica para a venda de serviços sob demanda”.

Esta afirmação mostra a necessidade que aos poucos se estabelece de os consultores de TI terem a possibilidade de manipular um software que gere as informações desejadas, em pouco tempo, muitas vezes, até com a presença do cliente. Porém a realidade vivida pelas empresas de informática é um pouco diferente. Os consultores ficam extremamente dependentes dos desenvolvedores de software para realizar seu trabalho. Em muitos lugares, os consultores chegam a ter acesso direto ao banco de dados para a execução de consultas *SQL*, o que torna o processo propenso a erros, visto que o consultor nem sempre tem o domínio da tecnologia de banco de dados utilizada.

A preocupação destes consultores deve ser ajudar o cliente a resolver seus problemas, portanto não devem prejudicar-se com o fato de não terem o conhecimento técnico necessário para executar algumas funções na implementação de relatórios.

O desenvolvimento de relatórios em sistemas de computação, bem como outros projetos, passa por diversas fases até sua entrega, sendo algumas delas (podendo variar):

- Análise de requisitos de software
- Especificação
- Arquitetura de software
- Implementação
- Teste
- Documentação
- Suporte e treinamento de uso do software
- Manutenção

Com uma análise simples sobre os fatos apresentados, pode-se notar que o processo atual nas empresas de informática para o desenvolvimento e manutenção de softwares mais simples, como relatórios, é muito burocrático e demorado, se comparado a complexidade envolvida com outros projetos que precisariam de mais atenção e tempo dos desenvolvedores. Consultores e clientes do sistema aguardam a finalização da implementação do desenvolvedor, em um processo relativamente simples de busca, organização e amostragem de dados que poderia ser facilmente feito pelo cliente do sistema, com a ajuda de um consultor do mesmo, que são quem mais entendem das regras de negócio, e sabem a forma na qual as informações devem ser apresentadas.

Algumas alterações simples em relatórios, que poderiam levar apenas alguns minutos, se o cliente tivesse a possibilidade de alterar a forma de amostragem dos dados, levam até dias para serem concluídas, ao passarem pelo processo de desenvolvimento de software, com análise, implementação, compilação, e atualização.

Segundo (PRIMAK, 2008), "Nos dias atuais, corporações de pequeno, médio e grande porte necessitam do BI para auxiliá-las nas mais diferentes situações para a tomada de decisão, otimizar o trabalho da organização, reduzir custos, eliminar a duplicação de tarefas, permitir previsões de crescimento da empresa como um todo e contribuir para a elaboração de estratégias."

Apesar do sistema de geração de relatórios proposto não ser um BI, pretende-se alcançar as funcionalidades necessárias para suprir as necessidades citadas.

Partindo deste cenário, o capítulo 6 propõe uma solução de software para endereçar os problemas apontados.

4 PROPOSTA DE SOLUÇÃO

A partir das necessidades apontadas, este trabalho propõe desenvolver uma ferramenta WEB para extração customizada de informações de bancos de dados relacionais de um sistema de gestão. O funcionamento desta ferramenta será em duas fases:

- Primeira Fase - Nesta fase, o ator será um profissional com conhecimento técnico sobre a estrutura de dados do sistema de gestão. Após a ferramenta ser instalada e configurada corretamente para funcionar com o sistema de gestão, a mesma identificará a estrutura do banco de dados, fornecendo para o usuário as tabelas, colunas, e outras informações necessárias para que possa ser criado o que será chamado aqui de "entidade composta". Entidades compostas são dados provindos da base do sistema, combinados da forma desejada, formando representações de objetos do mundo real.
- Segunda Fase - Os objetos criados na primeira fase serão disponibilizados para o cliente, como dono da informação(ator desta fase), ou até mesmo consultores especializados no sistema. Com estes objetos, será possível selecionar atributos para que sirvam de filtros, e quais dados serão mostrados em tela na emissão do relatório, e também a opção de geração de gráficos de análise, dependendo dos dados escolhidos. Desta forma o objetivo do software, que é abstrair a necessidade do uso de SQL, é alcançado.

Desta forma, espera-se reduzir os problemas descritos no capítulo anterior, uma vez que os dados, após transformados em entidades compostas, poderão ser manipulados para apresentação e visualização pelo próprio dono da informação, ou pelos consultores especializados no sistema, que são os profissionais que mais têm contato com o cliente, e os que mais possuem conhecimentos no ramo em que o cliente atua. Assim a falha na comunicação não seria mais problema, uma vez que não haveria mais tantos profissionais envolvidos no processo.

Para criação das entidades compostas, podem ser selecionados filtros-padrão, que são restrições-base para os dados que serão buscados do banco de dados. Por

exemplo: ao criar a entidade composta chamada Cliente, pode-se optar por usar um filtro-padrão para que sejam buscados apenas os clientes ativos.

Para o desenvolvimento desta ferramenta, as tecnologias apresentadas no referencial teórico são integradas conforme a figura 4.1.

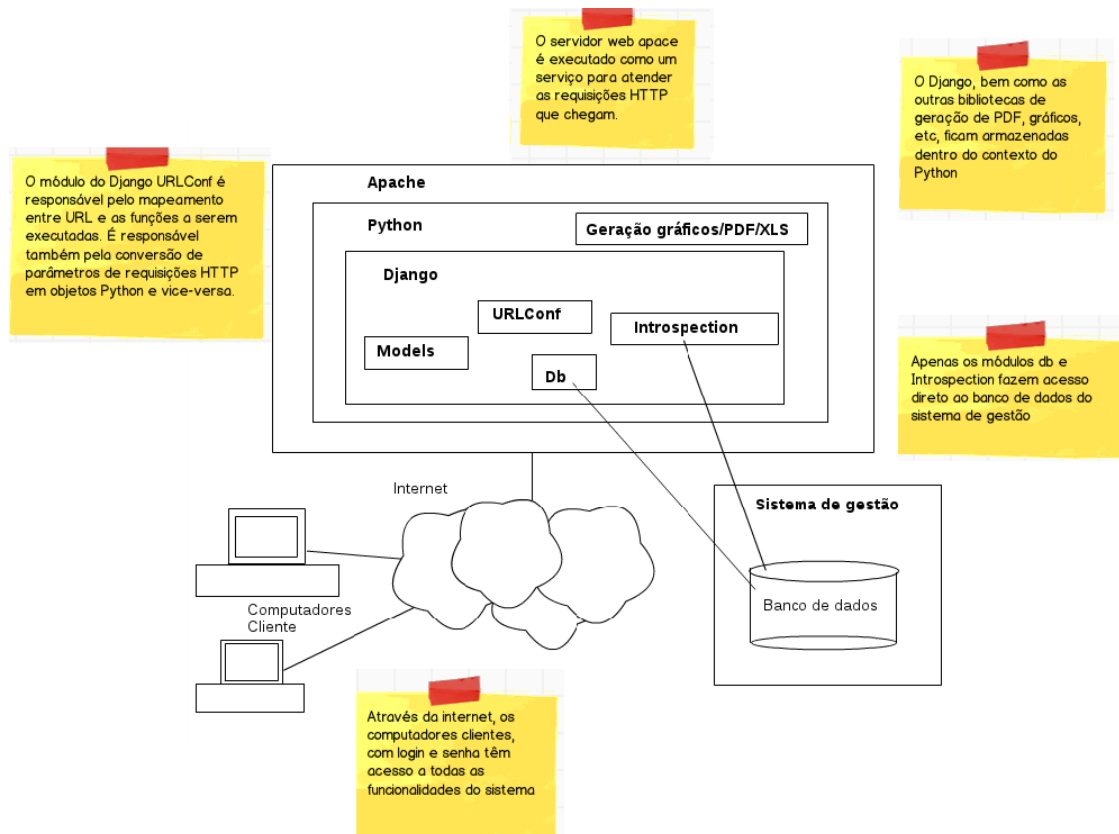


Figura 4.1: Estrutura da ferramenta.

Conforme ilustra a figura 4.1, um servidor com Apache instalado estará servindo as requisições referentes ao funcionamento do sistema. No mesmo servidor, estará instalado o interpretador da linguagem de programação python, que por sua vez possuirá instaladas bibliotecas de geração de imagens com gráficos, PDF e XLS para os relatórios. Além disso, também haverá instalado o módulo Django, que é um framework para desenvolvimento WEB com python. Os módulos internos inclusos no Django executam a comunicação necessária com os bancos de dados necessários, e também recebe requisições HTTP, transformando-as em respostas para os clientes (browsers).

Com esta ferramenta, poderá ser dispensável a presença de profissionais de programação no desenvolvimento de relatórios simples, agilizando e simplificando o processo de criação dos mesmos. Desta forma, haverá o uso de menos horas, e um maior acompanhamento de profissionais de negócio durante a criação e manutenção destes relatórios. Boa parte, talvez todos os relatórios existentes no sistema poderão ser substituídos por relatórios desenvolvidos nesta ferramenta,

ganhando assim todo o tempo que era até então dispendido na criação, manutenção e customização dos relatórios existentes no sistema.

Com isto acredita-se que o dono da informação ganhe muito mais liberdade para conhecer e manipular seus dados.

Os relatórios criados podem ser exportados para formatos externos, como *PDF* (Portable Document Format), o que possibilita que o relatório seja visto em qualquer visualizador de documentos PDF. Estes relatórios poderão ainda ser exportados em um formato específico, permitindo sua importação em outra instância desta mesma ferramenta, o que possibilita que a empresa dona do sistema de gestão possa desenvolver relatórios em comum para todos os seus clientes, ou ainda desenvolver relatórios como novos produtos, disponibilizando para venda aos seus clientes, os donos da informação. Com isto, grupos de usuários de um sistema de gestão em comum poderão compartilhar diferentes relatórios, idéias e técnicas.

Ao montar um relatório, é possível escolher quais campos serão utilizados como filtro para a geração do mesmo, desta forma, o relatório poderá ser acessado posteriormente diretamente em uma tela de filtros de dados montada dinamicamente. Visando facilitar o trabalho, ainda será possível agendar gerações de relatórios, que ao serem emitidos serão automaticamente enviados por e-mail, ou salvos na pasta de trabalho do usuário.

Por estar executando em ambiente Web, é possível acessar a partir de qualquer lugar onde exista conexão com a internet, como por exemplo, bares, restaurantes, aeroportos.

No capítulo a seguir é apresentada a modelagem do sistema.

5 PROCESSO DE DESENVOLVIMENTO DO SISTEMA

Para o desenvolvimento de um sistema faz-se necessário detalhar o problemas em pequenas partes, facilitando a compreensão. Neste capítulo serão representados a partir da metodologia ICONIX os requisitos e artefatos para o desenvolvimento deste sistema conforme os casos de uso.

5.1 Requisitos Não Funcionais

Os requisitos não funcionais, ou *RNF* (Requisito Não Funcional), são relativos às restrições sobre como o sistema deve desempenhar suas funções.

A tabela 5.1, apresenta todos os requisitos não funcionais do sistema proposto:

Tabela 5.1: Requisitos Não Funcionais

Identificação	Descrição	Prioridade
RNF-001	O sistema deve funcionar em ambiente web.	Essencial
RNF-002	A interface do sistema deve ser desenvolvida em xHTML, validando conforme especificações da W3C.	Desejado
RNF-003	O design de interface deve ser desenvolvido com o uso de CSS - Cascading Style Sheets (Folhas de Estilo em Cascata) – em um arquivo separado do xHTML. Facilitando futuras mudanças.	Desejado

5.2 Requisitos Funcionais

A tabela 7.2, apresenta os requisitos funcionais, ou *RF* (Requisito Funcional), estes são funcionalidades que devem ser obtidos durante o desenvolvimento do sistema:

Tabela 7.2 - Requisitos Funcionais

Identificação	Descrição	Prioridade
RF-001	O sistema deve controlar o acesso dos usuários.	Essencial
RF-002	O sistema deve manter usuários.	Essencial
RF-003	O sistema deve manter as entidades compostas completas (explicadas no capítulo 6).	Essencial
RF-004	O sistema deve manter relatórios, a partir das entidades compostas previamente criadas.	Essencial
RF-005	Permitir salvar resultados de relatórios e consultar posteriormente.	Essencial
RF-006	Consultar e visualizar resultados de relatórios salvos.	Desejável
RF-007	Permitir salvar filtros utilizados em relatórios para agendamento de execução automática dos mesmos.	Desejável
RF-008	Gerar os relatórios que foram agendados	Desejável
RF-009	Envio de e-mail dos relatórios gerados.	Essencial
RF-010	Geração de relatórios em formato HTML.	Essencial
RF-011	Geração de arquivo no formato PDF, a partir de um relatório gerado no formato HTML.	Importante
RF-012	Geração de arquivo no formato XLS, a partir de um relatório gerado no formato HTML.	Desejável

RF-013	Geração de imagem com gráfico, sobre campos selecionados do relatório selecionado.	Essencial
RF-014	Agendamento para geração automática de relatórios, para serem enviados por e-mail, ou salvos na pasta de trabalho do usuário.	Importante
RF-015	O sistema deve manter pastas de trabalho do usuário.	Essencial
RF-016	O sistema deve permitir duplicação de entidade compostas, para facilitar a criação de novas entidades-compostas a partir de entidades-base.	Desejável
RF-017	O sistema deve permitir duplicação de relatórios, para facilitar a criação de novos relatórios a partir de relatórios-base.	Importante
RF-018	O sistema deve exibir uma lista de valores possíveis para os campos de filtros de relatórios que sejam do formato texto.	Importante
RF-019	O sistema deve exibir as notificações de gerações automáticas de relatórios para o usuário na tela inicial, na seção Dashboard(neste sistema, são notificações ao usuário, localizadas na tela inicial).	Importante

A seguir será apresentada a análise do sistema conforme metodologia ICONIX. Onde as telas representam os casos de uso e visam atender aos requisitos listados na tabela 7.2.

5.3 Diagrama de Caso de Uso do Sistema

A figura 5.1 representa o caso de uso do sistema como um todo.

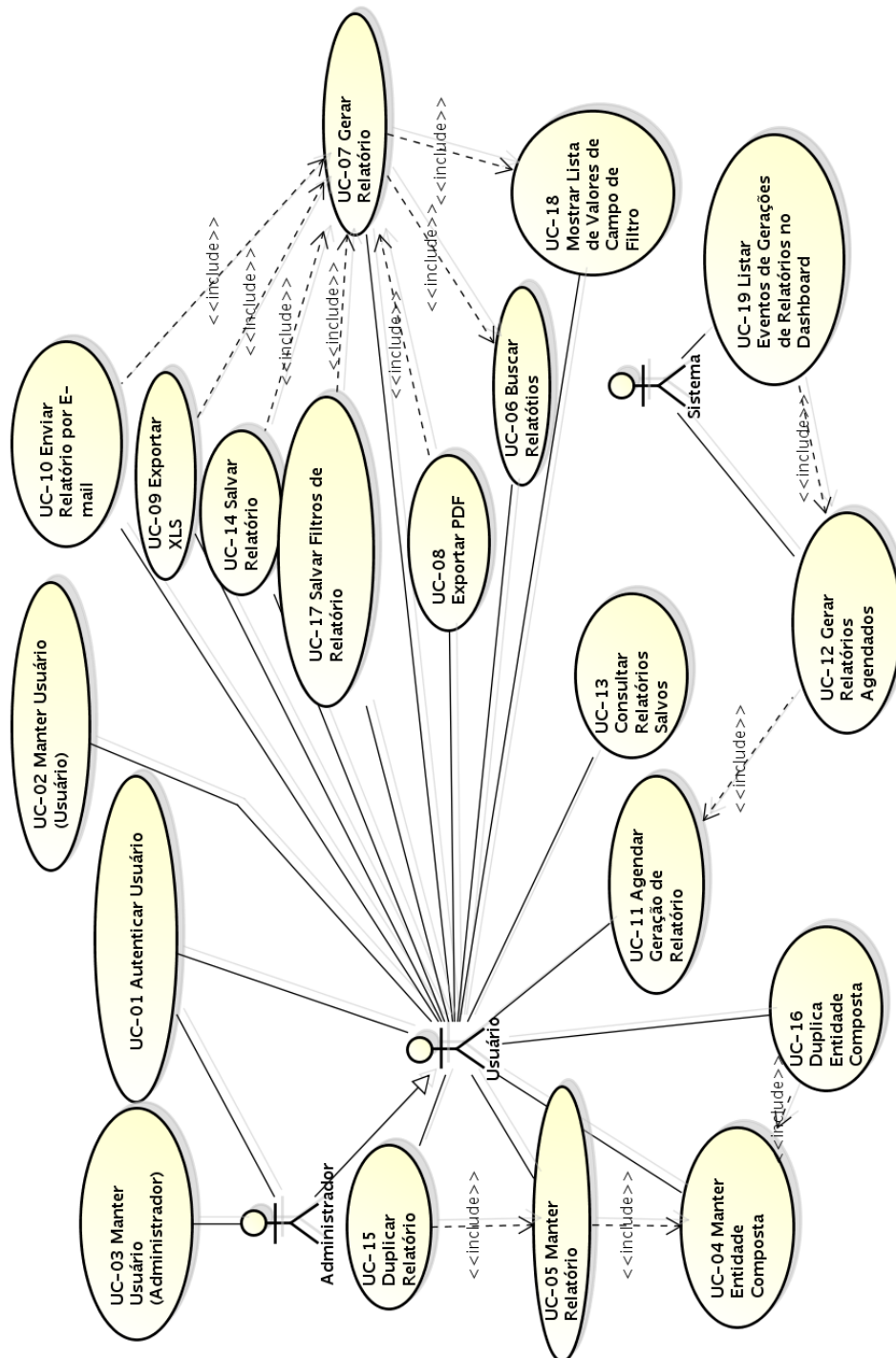


Figura 5.1: Diagrama de Caso de Uso do Sistema como um Todo.

A figura 5.1 visa demonstrar e ligar todos os casos de uso do sistema e quais as dependências dos mesmos. A seguir será visto o diagrama de domínio, que deve exibir de forma genérica os principais objetos para estes casos de uso.

5.4 Diagrama de Domínio do Sistema

A figura 5.2 representa o caso de uso do sistema como um todo.

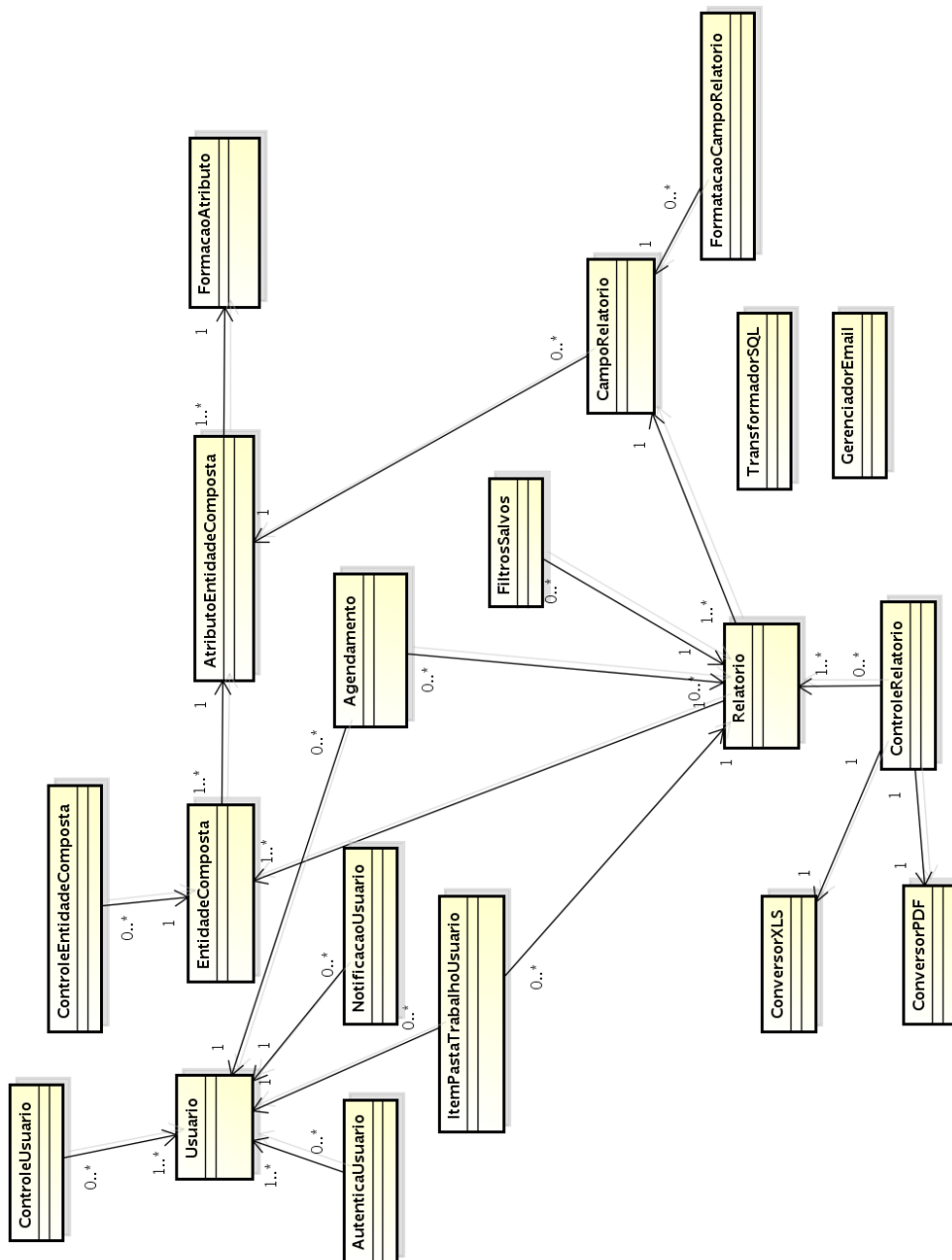


Figura 5.2: Diagrama de Domínio do Sistema como um Todo.

O diagrama da figura 5.2 apresenta os principais objetos do sistema e qual os relacionamentos entre eles.

A seguir são descritas as principais funções de cada um dos objetos do diagrama de domínio:

- Usuario - Entidade que representa os usuários do sistema.
- ControleUsuario - Contém as funções de controle de usuários necessárias para o funcionamento do sistema.
- AutenticaUsuario - Controla a autenticação de usuários no sistema e mantém a sessão dos usuários conectados.
- NotificacaoUsuario - Entidade que representa as notificações dos usuários, apresentadas no dashboard.
- EntidadeComposta - Entidade que representa as entidades compostas cadastradas no sistema.
- AtributoEntidadeComposta - Entidade que representa os atributos cadastrados para as entidades compostas.
- FormacaoAtributo - Representa as partes que formam os atributos das entidades compostas, com suas respectivas ligações.
- ControleEntidadeComposta - Contém as funções necessárias para controle das entidades compostas do sistema.
- Relatorio - Entidade que representa os relatórios cadastrados no sistema.
- CampoRelatorio - Representa os campos dos relatórios, diretamente relacionados aos atributos das entidades compostas, que são utilizados como fontes de dados para estes campos.
- FormatacaoCampoRelatorio - Representa os itens de formatação dos campos dos relatórios.
- ControleRelatorio - Contém as funções necessárias para o controle dos relatórios no sistema.
- Agendamento - Entidade que representa os agendamentos de geração automática dos relatórios dos agendados.
- FiltrosSalvos - Representa os filtros salvos de geração de relatórios.
- ItemPastaTrabalhoUsuario - Entidade que representa um relatório salvo na pasta de trabalho de um usuário.

- TransformadorSQL - Contém todas as funções necessárias para as gerações de consultas SQL para o sistema.
- ConversorXLS - Contém as funções para conversão de relatórios no formato HTML para o formato XLS.
- ConversorPDF - Contém as funções para conversão de relatórios no formato HTML para o formato PDF.
- GerenciadorEmail - Contém as funções relacionadas aos envios de emails com os resultados dos relatórios.

Nas próximas seções são descritos os casos de uso que compõem o sistema.

5.5 UC-01 Autenticar Usuário

O sistema deve verificar se o indivíduo tem permissão para acessá-lo, atendendo ao requisito RF-001 - Controle de acesso.

5.5.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.5.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.3, visa atender ao requisito RF-001.

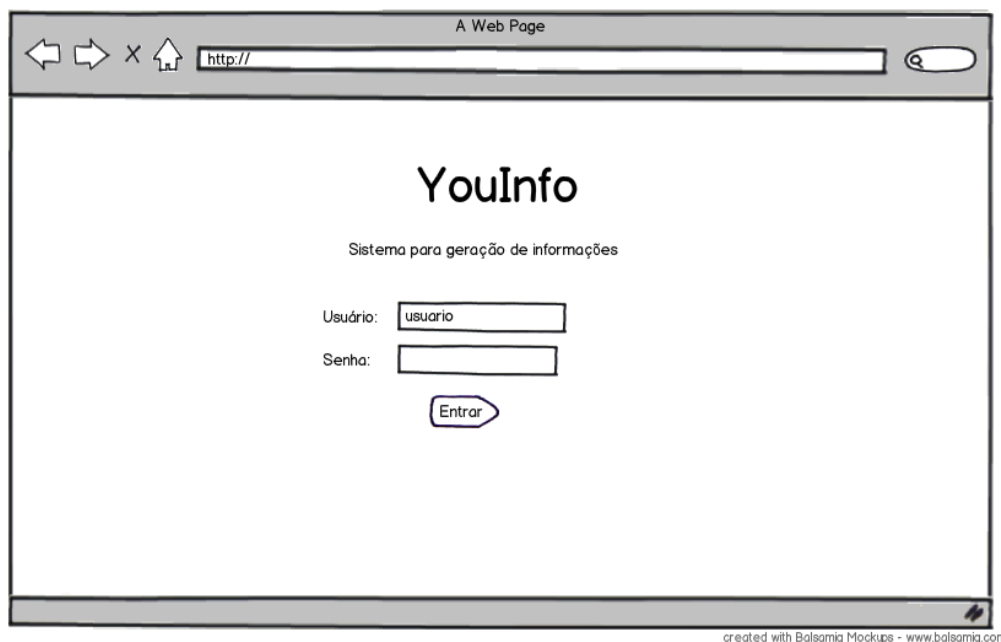


Figura 5.3: Tela 1 - Protótipo não funcional de interface - Autenticar Usuário.

5.5.1.2 Caso de Uso

A figura 5.4 demonstra o caso de uso para a tela de autenticação do usuário.

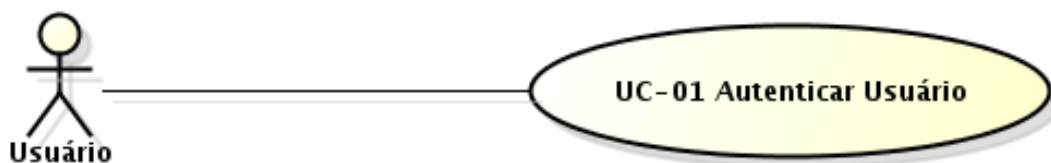


Figura 5.4: Diagrama de caso de uso - Autenticar Usuário.

5.5.1.3 Modelo de Domínio

A figura 5.5 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-001.

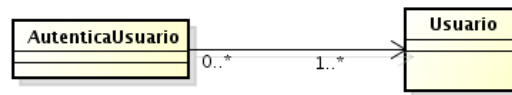


Figura 5.5: Modelo de Domínio - Autenticar Usuário.

5.5.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.5.2.1 Caso de Uso Detalhado

A tabela 5.3 apresenta o caso de uso detalhado que visa atender ao requisito RF-001.

Tabela 5.3 – Caso de Uso Detalhado Autenticar Usuário

UC-01	
Nome	Autenticar Usuário
Descrição	O sistema deve controlar o acesso dos usuários.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-001
Pré-condições	O usuário deve estar pré-cadastrado.
Pós-condições	O ator deve estar autenticado e na tela inicial do sistema.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Preenche os dados listados a seguir nos campos adequados (campos marcados com * são de preenchimento obrigatório): a. *Usuário b. *Senha 2. Clica no botão Entrar	
	3. Verifica se os campos obrigatórios estão preenchidos. 4. Verifica se usuário e senha estão corretos. 5. Autentica usuário e redireciona-o para a tela inicial do sistema.
Cenário Alternativo 1	
	3.1 Verificou que um ou mais campos obrigatórios não estão preenchidos. 3.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando junto ao campo que o mesmo deve ser preenchido.
Cenário Alternativo 2	
	4.1 Verificou que o usuário e senha estão incorretos. 4.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando que usuário ou senha está incorreto.

5.5.3 Projeto

5.5.3.1 Diagrama de Sequência

A figura 5.6 apresenta o diagrama sequência para atender ao requisito RF-001.

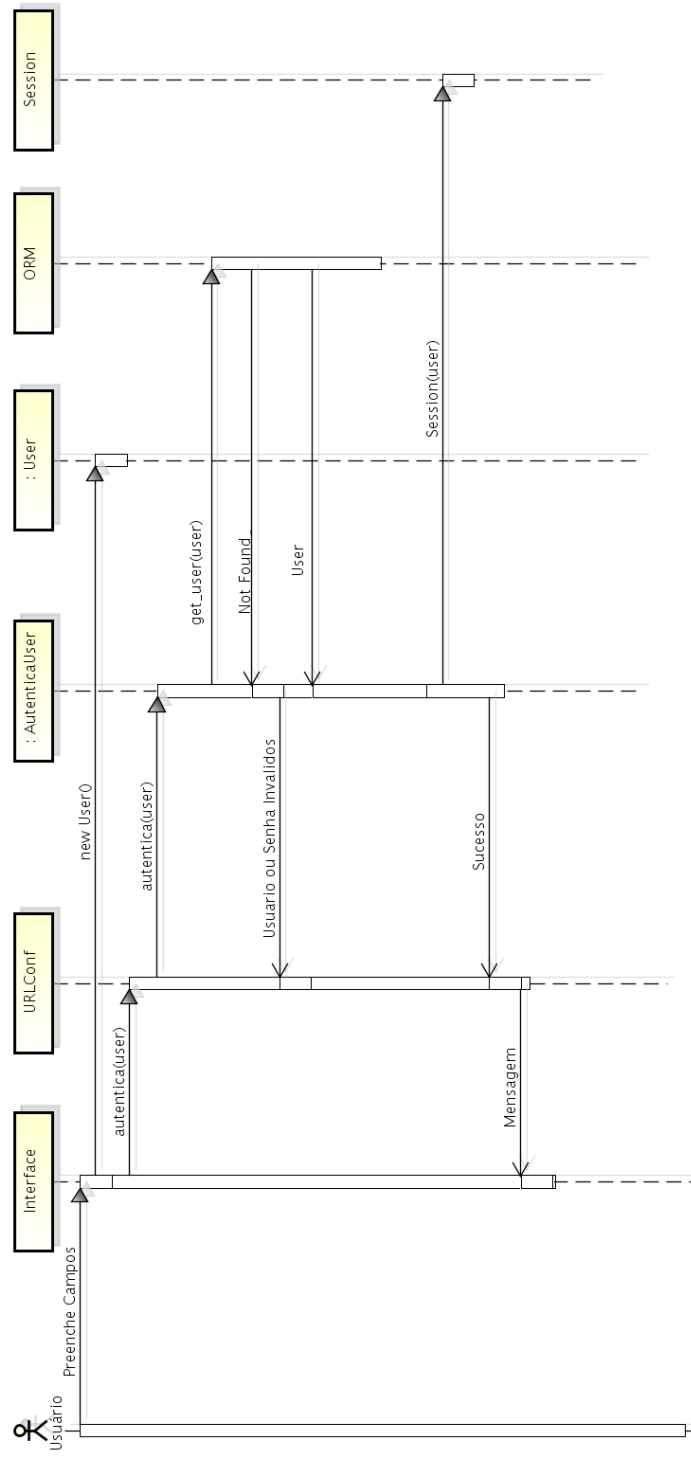


Figura 5.6: Diagrama de sequência - Autenticar Usuário.

A figura 5.6 representa a comunicação entre as classes do sistema, bem como

a vida útil dos objetos envolvidos no processo. Pode-se observar nos diagramas de sequência o padrão de fluxo de troca de mensagens onde inicia-se pelo ator executando a requisição, passando pelo interface, que envia uma mensagem para o URLConf, que a classe responsável pelo gerenciamento do protocolo de troca de mensagens entre cliente e servidor. Este, por sua vez, envia uma mensagem para a classe de controle específica do processo, que instancia os objetos de modelo necessários, faz consultas por meio da ORM, e grava dados em sessão, sempre retornando a mensagem de sucesso ou de erro adequada.

5.5.3.2 Diagrama de Classes

A figura 5.7 apresenta o diagrama classes para atender ao requisito RF-001.

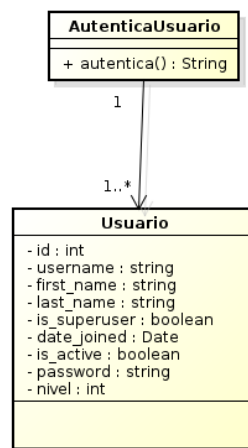


Figura 5.7: Diagrama de classes - Autenticar Usuário.

5.6 UC-02 Manter Usuário

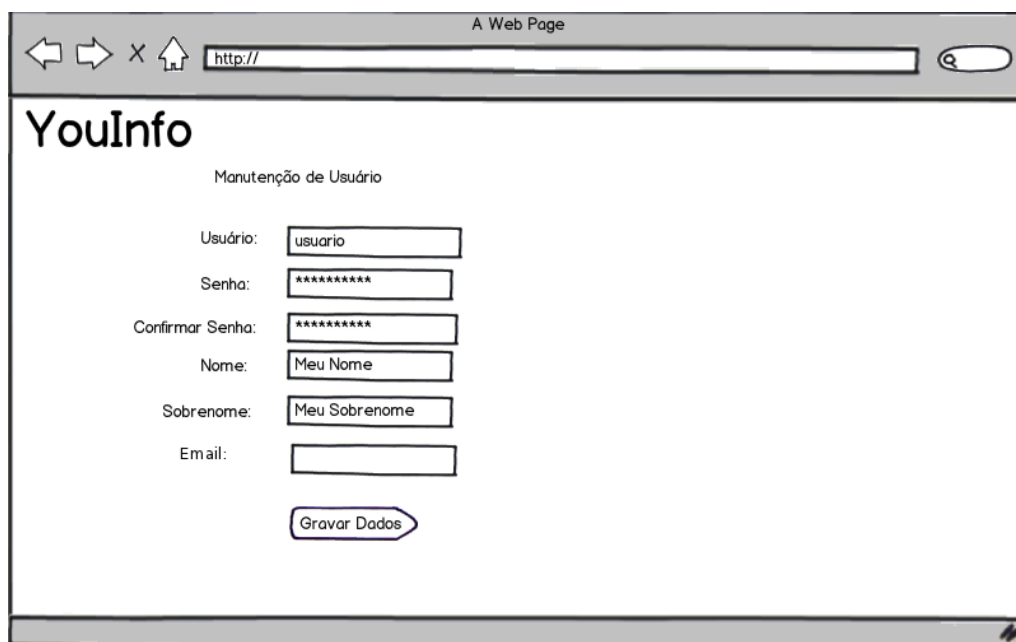
O sistema deve permitir ao usuário alterar seus dados, atendendo ao requisito RF-002 - Manter Usuários.

5.6.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.6.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.8, visa atender ao requisito RF-002.



A Web Page

http://

YouInfo

Manutenção de Usuário

Usuário:

Senha:

Confirmar Senha:

Nome:

Sobrenome:

Email:

Figura 5.8: Tela 2 - Protótipo não funcional de interface - Manter Usuário.

5.6.1.2 Caso de Uso

A figura 5.9 demonstra o caso de uso para a tela alteração de dados do usuário.

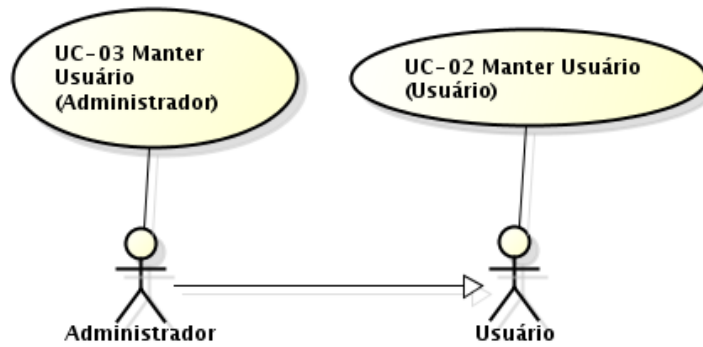


Figura 5.9: Diagrama de caso de uso - Manter Usuário.

5.6.1.3 Modelo de Domínio

A figura 5.10 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-002.

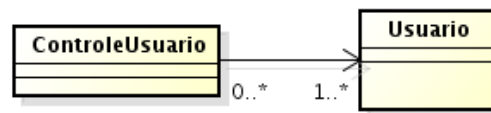


Figura 5.10: Diagrama de domínio - Manter Usuário.

5.6.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.6.2.1 Caso de Uso Detalhado

A tabela 5.4 apresenta o caso de uso detalhado que visa atender ao requisito RF-002.

Tabela 5.4 - Caso de Uso Detalhado Manter Usuário

UC-02	
Nome	Manter Usuário(Usuário)
Descrição	O sistema deve permitir que o usuário altere seus próprios dados.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-002
Pré-condições	O usuário deve estar autenticado no sistema. O usuário pode alterar somente seus próprios dados.
Pós-condições	Um usuário foi incluído ou alterado
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Preenche os dados listados a seguir nos campos adequados (campos marcados com * são de preenchimento obrigatório): a. *Usuário b. *Senha c. *Confirmar Senha d. *Nome e. *Sobrenome f. *E-mail 2. Clica no botão Salvar	
	3. Verifica se os campos obrigatórios estão preenchidos. 4. Verifica se os campos Senha e Confirmar Senha conferem. 5. Verifica se o usuário ainda não existe. 6. Persiste os dados
Cenário Alternativo 1	
	3.1 Verificou que um ou mais campos obrigatórios não estão preenchidos. 3.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando junto ao campo que o mesmo deve ser preenchido.

Cenário Alternativo 2	
	<p>4.1 Verificou que os campos senha e confirmar senha não conferem.</p> <p>4.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando que os campos senha e confirmar senha devem ser iguais.</p>
Cenário Alternativo 3	
	<p>5.1 Verificou que o usuário informado já existe.</p> <p>5.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando que o usuário informado já existe.</p>

5.6.3 Projeto

5.6.3.1 Diagrama de Sequência

A figura 5.11 apresenta o diagrama sequência para atender ao requisito RF-002.

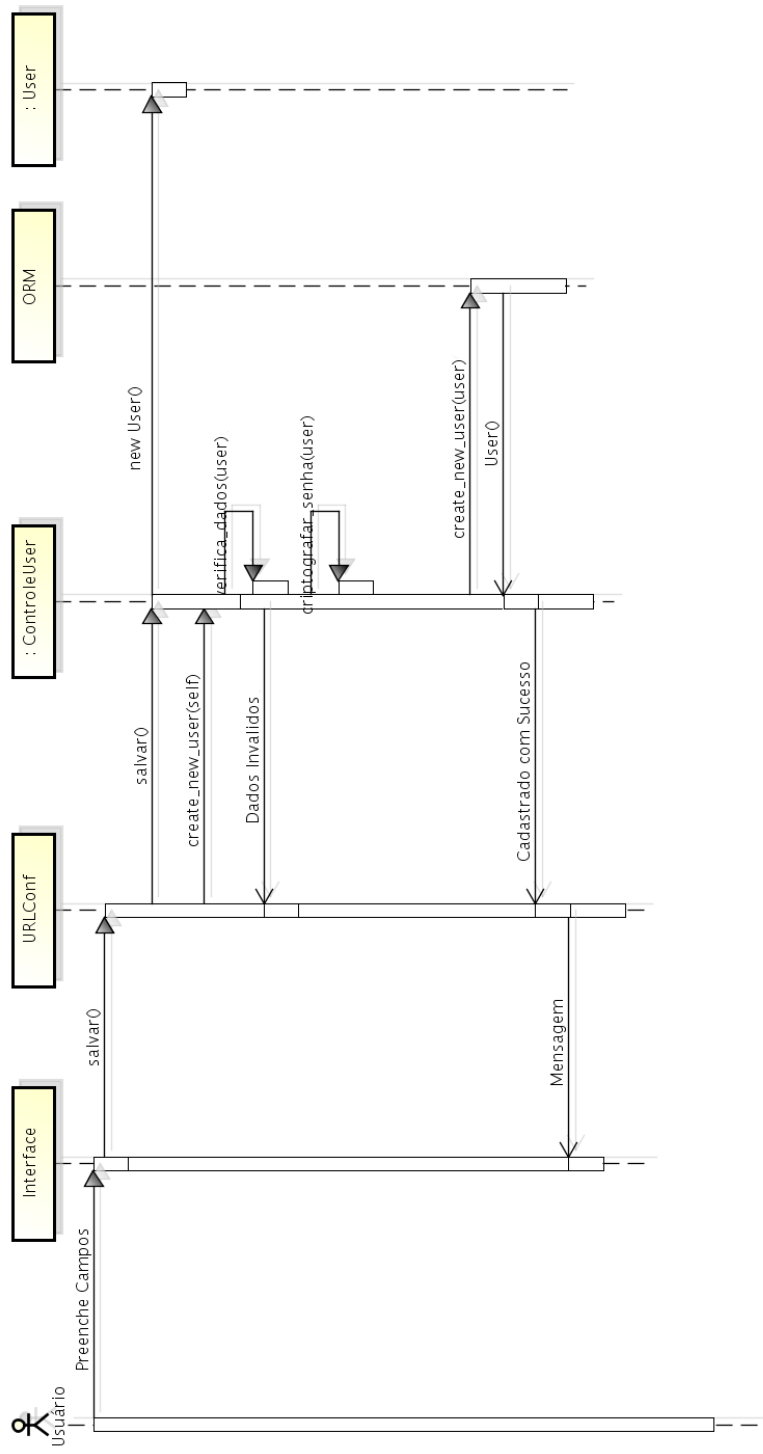


Figura 5.11: Diagrama de sequência - Manter Usuário.

5.6.3.2 Diagrama de Classes

A figura 5.12 apresenta o diagrama classes para atender ao requisito RF-002.

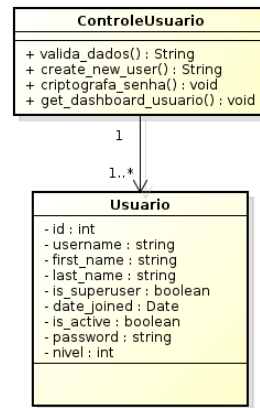


Figura 5.12: Diagrama de classes - Manter Usuário.

5.7 UC-03 Manter Usuário(Administrador)

O sistema deve permitir ao usuário administrador incluir ou alterar dados de todos os usuários do sistema, atendendo ao requisito RF-002 - Manter Usuários.

5.7.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.7.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.13, visa atender ao requisito RF-002.

A Web Page

http://

YouInfo

Manutenção de Usuário(modo administrador)

Usuário:

Senha:

Nome:

Sobrenome:

Email:

Nível:

Ativo

Figura 5.13: Tela 3 - Protótipo não funcional de interface - Manter Usuário(Administrador).

5.7.1.2 Caso de Uso

A figura 5.14 demonstra o caso de uso para a tela alteração de dados do usuário, do modo administrador.

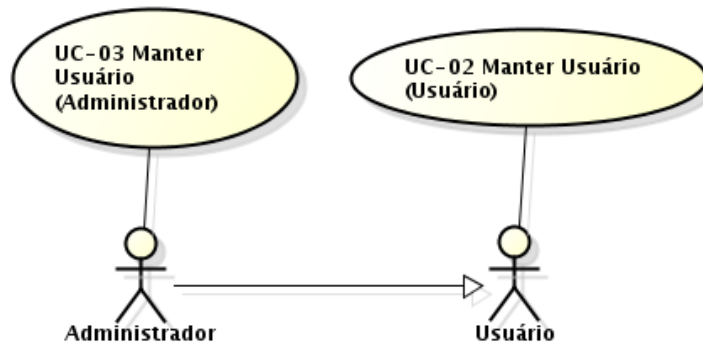


Figura 5.14: Diagrama de caso de uso - Manter Usuário.

5.7.1.3 Modelo de Domínio

A figura 5.15 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-002.

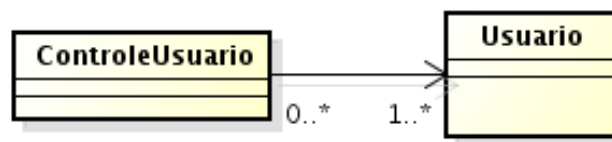


Figura 5.15: Diagrama de domínio - Manter Usuário (Administrador).

5.7.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.7.2.1 Caso de Uso Detalhado

A tabela 5.5 apresenta o caso de uso detalhado que visa atender ao requisito RF-002.

Tabela 5.5 - Caso de Uso Detalhado Manter Usuário(Administrador)

UC-03	
Nome	Manter Usuário(Administrador)
Descrição	O sistema deve permitir a inclusão e alteração de usuários
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-002
Pré-condições	O usuário deve estar autenticado no sistema como administrador.
Pós-condições	Um usuário foi incluído ou alterado
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
<p>1. Preenche os dados listados a seguir nos campos adequados (campos marcados com * são de preenchimento obrigatório):</p> <p>a. *Usuário b. *Senha c. *Confirmar Senha d. *Nome e. *Sobrenome f. *Email g *Ativo</p> <p>2. Clica no botão Salvar</p>	
	<p>3. Verifica se os campos obrigatórios estão preenchidos.</p> <p>4. Verifica se os campos Senha e Confirmar Senha conferem.</p> <p>5. Verifica se o usuário ainda não existe.</p> <p>6. Persiste os dados</p>
Cenário Alternativo 1	
	<p>3.1 Verificou que um ou mais campos obrigatórios não estão preenchidos.</p> <p>3.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando junto ao campo que o mesmo deve ser preenchido.</p>

Cenário Alternativo 2	
	<p>4.1 Verificou que os campos senha e confirmar senha não conferem.</p> <p>4.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando que os campos senha e confirmar senha devem ser iguais.</p>
Cenário Alternativo 3	
	<p>5.1 Verificou que o usuário informado já existe.</p> <p>5.2 Retorna ao passo 1, com os dados preenchidos pelo autor, informando que o usuário informado já existe.</p>

5.7.3 Projeto

5.7.3.1 Diagrama de Sequência

A figura 5.16 apresenta o diagrama sequência para atender ao requisito RF-002.

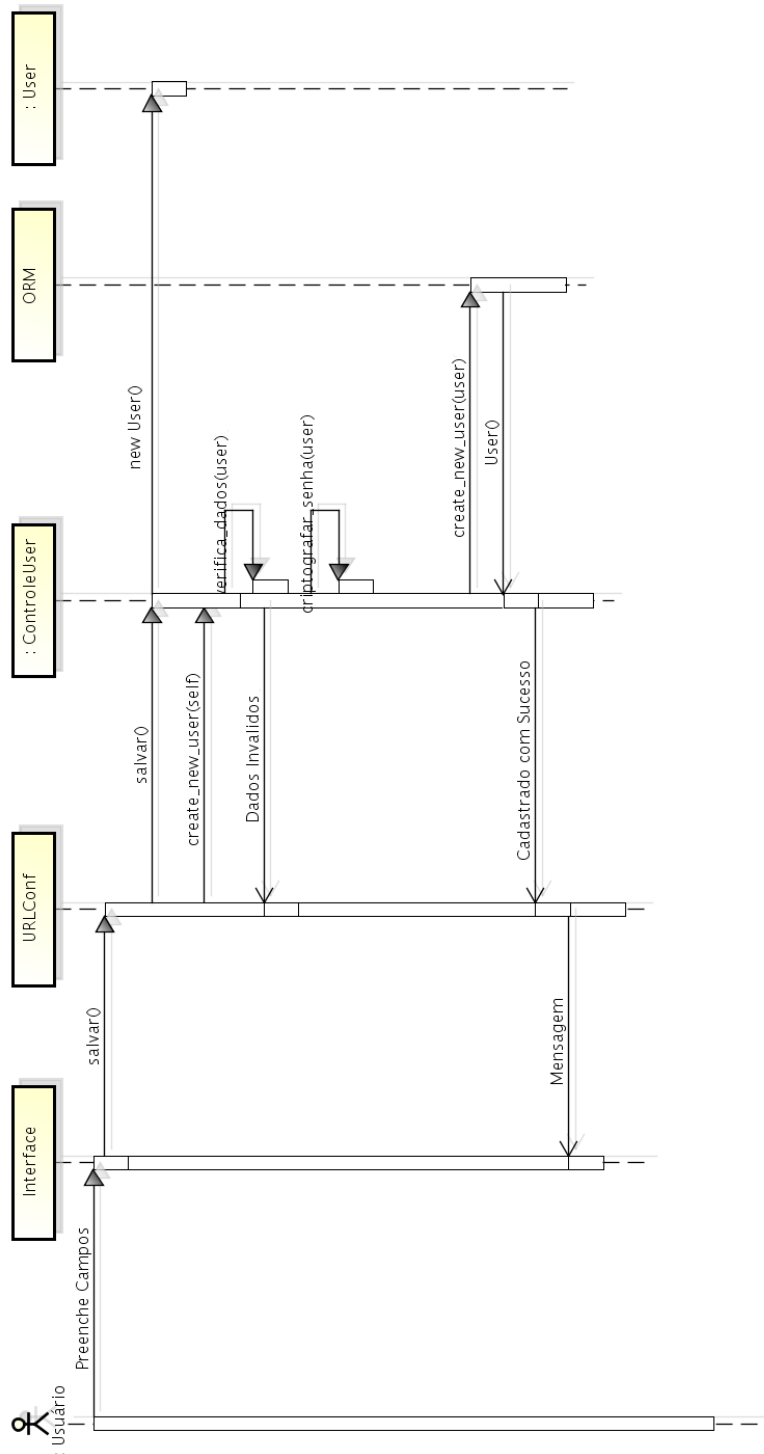


Figura 5.16: Diagrama de sequência - Manter Usuário(Administrador).

5.7.3.2 Diagrama de Classes

A figura 5.17 apresenta o diagrama classes para atender ao requisito RF-002.

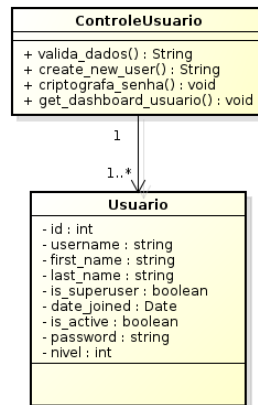


Figura 5.17: Diagrama de classes - Manter Usuário (Administrador).

5.8 UC-04 Manter Entidade Composta

O sistema manter entidades compostas, atendendo ao requisito RF-003 Manter Entidades Compostas.

5.8.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.8.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pelas figuras 7.18 e 7.19, visa atender ao requisito RF-003.

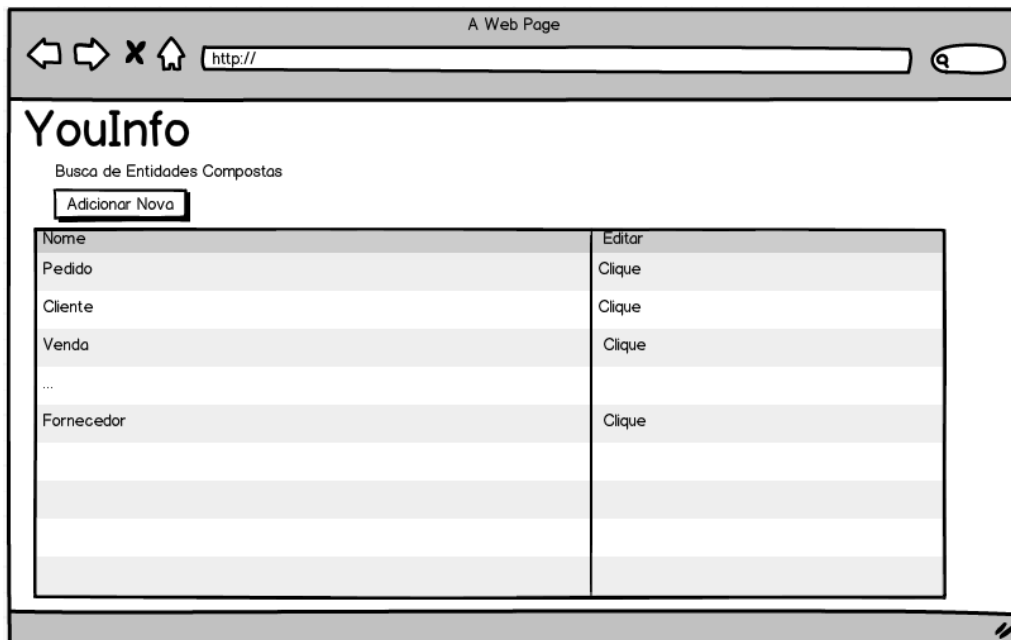


Figura 5.18: Tela 4 - Protótipo não funcional de interface - Buscar Entidades Compostas.

A seguir é mostrado o protótipo de interface da tela de manutenção de entidades compostas.

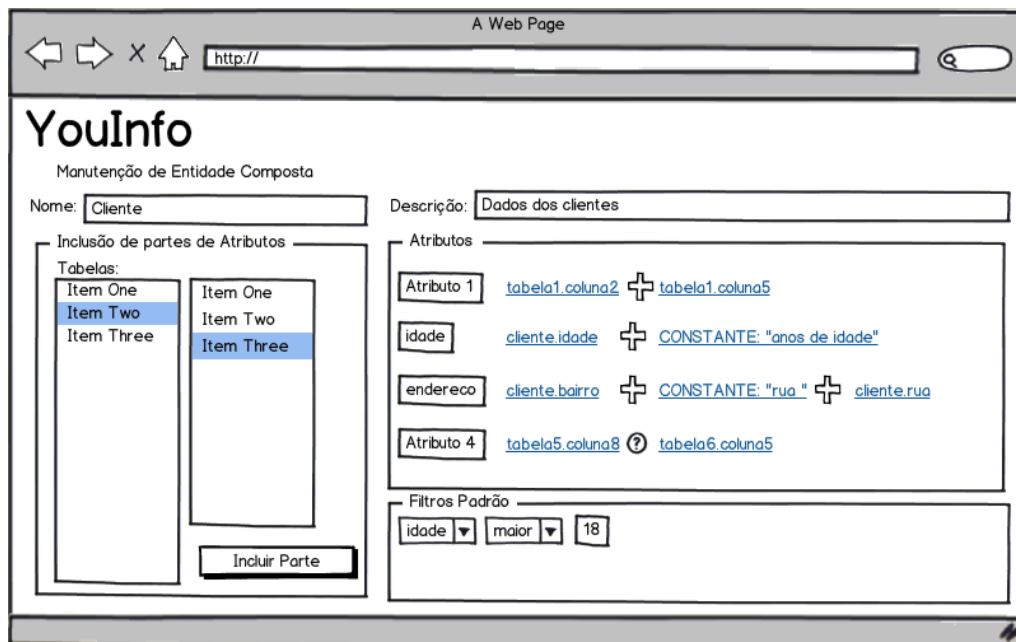


Figura 5.19: Tela 4.1 - Protótipo não funcional de interface - Manter Entidades Compostas.

5.8.1.2 Caso de Uso

A figura 5.20 demonstra o caso de uso para a tela de manutenção de entidades compostas.

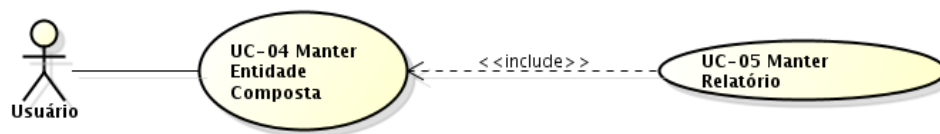


Figura 5.20: Diagrama de caso de uso - Manter Entidade Composta.

5.8.1.3 Modelo de Domínio

A figura 5.21 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-003.

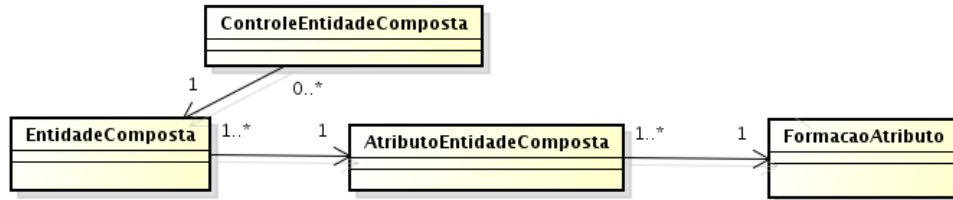


Figura 5.21: Diagrama de domínio - Manter Entidade Composta.

5.8.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.8.2.1 Caso de Uso Detalhado

A tabela 5.6 apresenta o caso de uso detalhado que visa atender ao requisito RF-003.

Tabela 5.6 - Caso de Uso Detalhado Manter Entidade Composta

UC-04	
Nome	Manter Entidade Composta
Descrição	O sistema deve permitir o cadastro, a alteração e a exclusão de entidades compostas, tratadas no capítulo 6 – Proposta de Solução.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-003
Pré-condições	O usuário deve estar autenticado no sistema. O usuário deve possuir permissão para acessar a tela de manutenção de entidades compostas.
Pós-condições	Uma entidade composta foi incluída, alterada ou removida.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Acessa a tela de manutenção de entidades compostas.	
	2. Lista as entidades compostas existentes no sistema.
3. Escolhe a entidade composta que deseja alterar ou excluir, ou clica no botão de incluir nova entidade composta.	
	4. Mostra a tela de manutenção de entidades compostas, com os campos para preenchimento, e uma lista com as tabelas disponíveis no sistema para criação da entidade composta.
5. Preenche os campos de nome e descrição da entidade composta, ambos obrigatórios.	
6. Seleciona uma tabela para formação de uma parte de um atributo, deve ser selecionada ao menos uma.	
	7. Mostra as colunas disponíveis da tabela selecionada no passo 6.
8. Seleciona uma coluna para formação de uma parte de um atributo, ou seleciona uma constante.	

9. Selecciona as relações com a próxima parte do atributo, ou selecciona fim de formação do atributo.	
	10. Executa a junção das partes.
<p>11. Repetem-se os passos 6, 7, 8, 9 e 10 até que seja seleccionado fim de formação de atributo, no passo 9.</p> <p>12. Repetem-se os passos 6,7,8,9,10 e 11 até onde o usuário não deseje mais criar atributos.</p> <p>13. Selecciona os filtros padrão que serão usados na busca dos dados da entidade composta.</p> <p>14. Clica no botão salvar.</p>	
	<p>15. Faz a validação final de campos obrigatórios , consistência entre as relações escolhidas e filtros padrão escolhidos.</p> <p>16. Persiste os dados.</p>
Cenário Alternativo 1	
	<p>2.1 Não encontrou entidades compostas cadastradas no sistema.</p> <p>2.2 Mostra uma mensagem na tela informando que não existem entidades compostas cadastradas no sistema, juntamente com o botão de inserir nova entidade composta.</p>
Cenário Alternativo 2	
	<p>10.1 Verificou que as relações entre as partes escolhidas para a formação do atributo não são possíveis devido aos tipos de dados das colunas escolhidas, ou outros fatores de validação dependendo da situação.</p> <p>10.2 Mostra uma mensagem avisando que não é possível realizar as junções selecionadas para a formação do atributo.</p> <p>10.3 Retorna a tela para que o ator possa modificar a formação do atributo.</p>
Cenário Alternativo 3	
	15.1 Verificou que as relações entre todas as tabelas escolhidas para a entidade composta não são possíveis.

	<p>15.2 Mostra uma mensagem avisando que não é possível a junção entre as tabelas escolhidas.</p> <p>15.3 Retorna a tela para que o ator possa modificar a entidade composta.</p>
Cenário Alternativo 4	
	<p>15.1 Verificou que existe um ou mais campos obrigatórios não informados.</p> <p>15.2 Retorna a tela com os dados já preenchidos pelo ator, informando junto ao campo sua obrigatoriedade.</p>
Cenário Alternativo 5	
	<p>15.1 Verificou que nenhuma tabela foi escolhida para a formação da entidade composta.</p> <p>15.2 Retorna a tela com os dados já preenchidos pelo ator, informando que deve ser selecionada ao menos uma tabela ou view como fonte de dados para a formação da entidade composta.</p>
Observações	
<ol style="list-style-type: none"> 1. As entidades compostas possuem atributos, que por sua vez são construídos em partes, facilitando a criação e manutenção dos mesmos. 2. As partes dos atributos de entidades compostas podem ser provenientes de uma fonte de dados de colunas de tabelas, ou valores constantes, como pedaços de frases ou números. 3. O relacionamento entre as partes dos atributos são como uma parte irá unir-se com a outra parte, como por exemplo soma de dois valores numéricos, ou junção de duas palavras ou frases. 	

5.8.3 Projeto

5.8.3.1 Diagrama de Sequência

As figuras 7.22 e 7.23 apresenta os diagramas sequência alterar e remover entidade composta, para atender ao requisito RF-003.

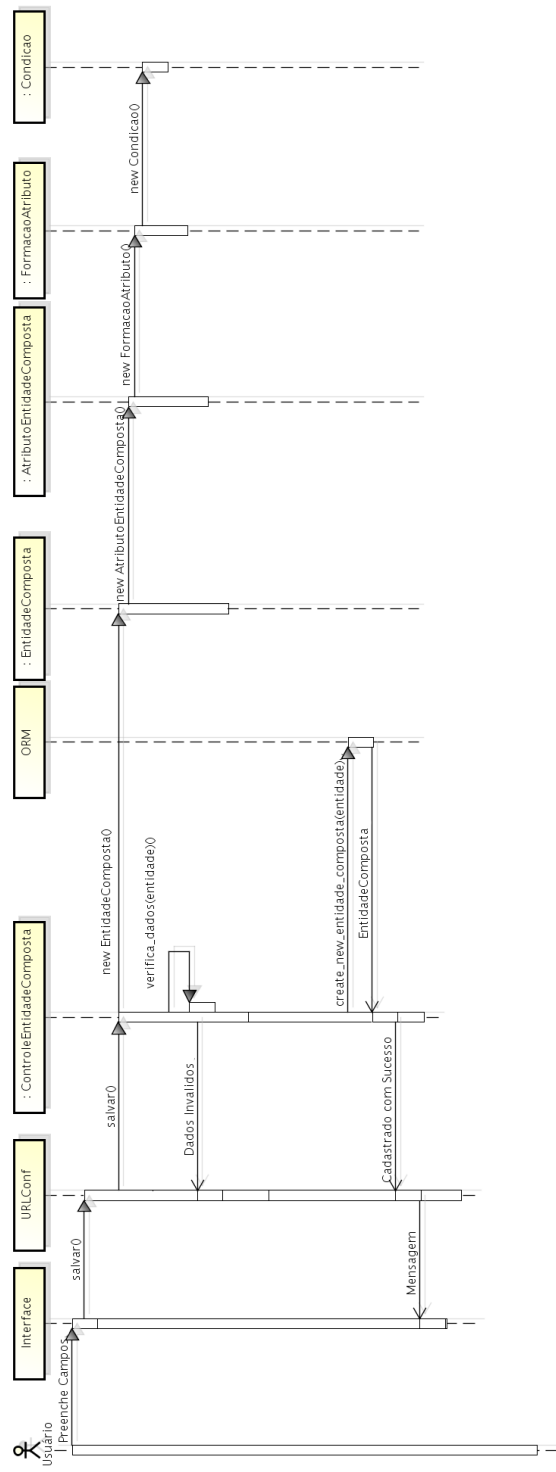


Figura 5.22: Diagrama de sequência - Manter Entidade Composta.

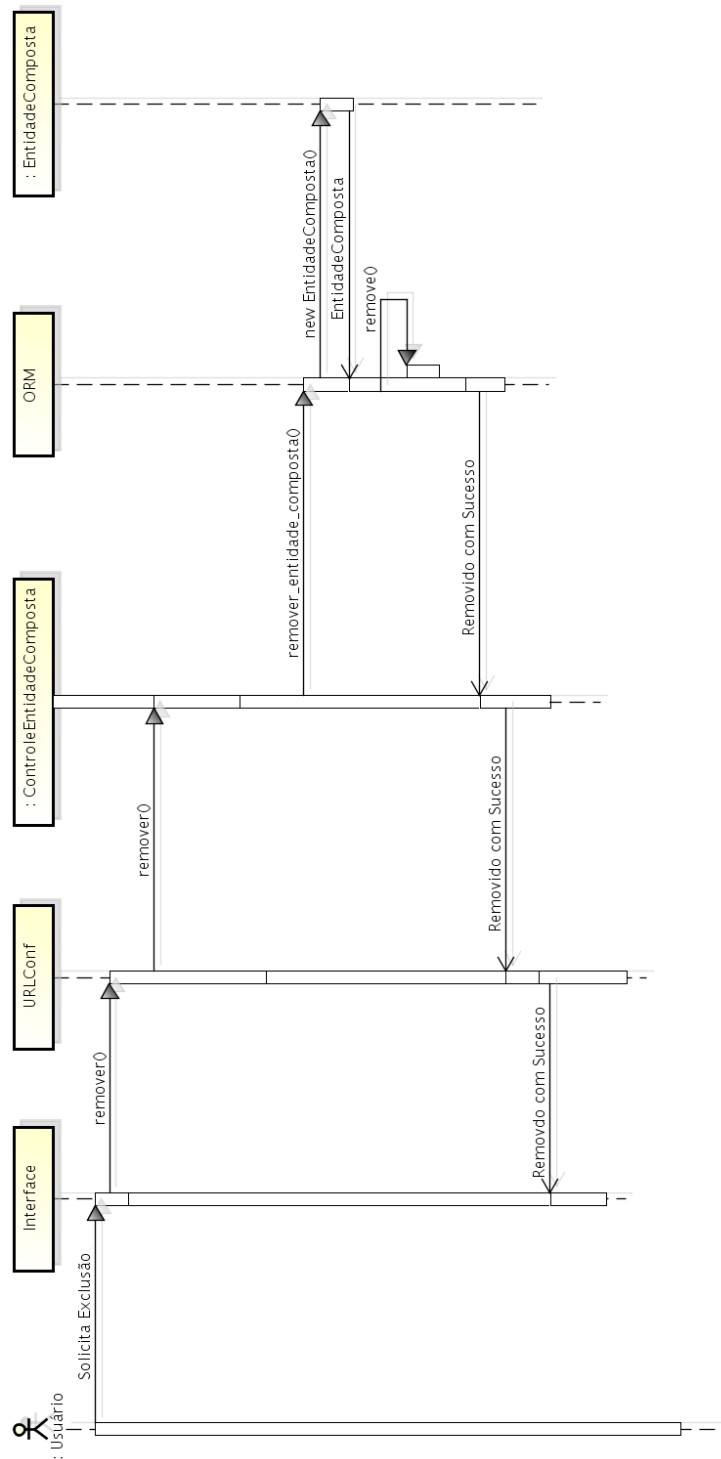


Figura 5.23: Diagrama de sequência - Manter Entidade Composta(Remover).

Para os processos onde não é necessário ter o objeto completo, como por exemplo no caso de exclusão de um objeto, onde necessita-se apenas da chave-primária do objeto, será usado um método da ORM chamado *defer*. Segundo (FOUNDATION, 2012b), em algumas situações, nossos objetos podem ter vários atributos que não iremos usar de imediato, para isto existe o método *defer*, que traz apenas os atributos

desejados, ou seja, a ORM continua enviando as instâncias dos objetos normalmente, porém apenas com os atributos solicitados, os outros atributos serão buscados separadamente no banco de dados, apenas se o código-fonte tentar acessá-los. Portanto, nos diagramas que forem apresentados somente com as classes principais, é porque não será necessário os atributos das classes filhas. Por exemplo: para buscar os relatórios, não será necessário instanciar as entidades compostas.

5.8.3.2 Diagrama de Classes

A figura 5.24 apresenta o diagrama classes para atender ao requisito RF-003.

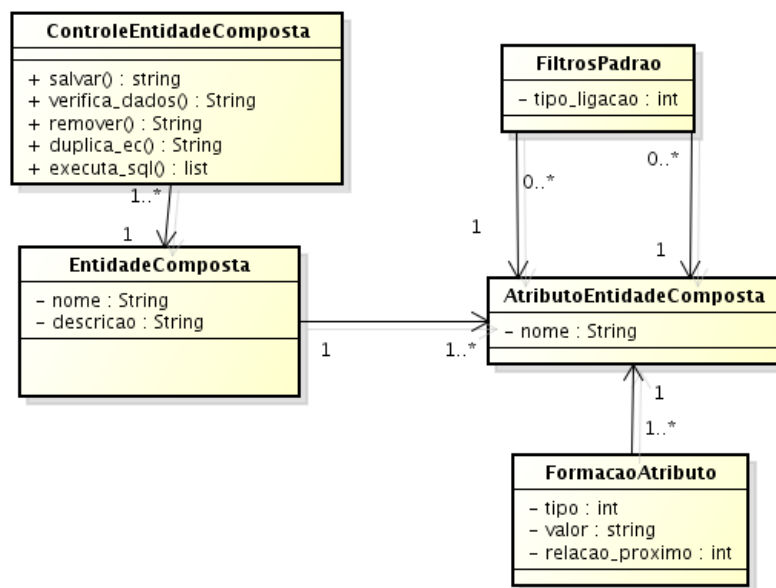


Figura 5.24: Diagrama de classes - Manter Entidade Composta.

5.9 UC-05 - Manter Relatório

O sistema permitir a manutenção de relatórios, atendendo ao requisito RF-004 - Manter Relatórios.

5.9.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.9.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.25, visa atender ao requisito RF-004.

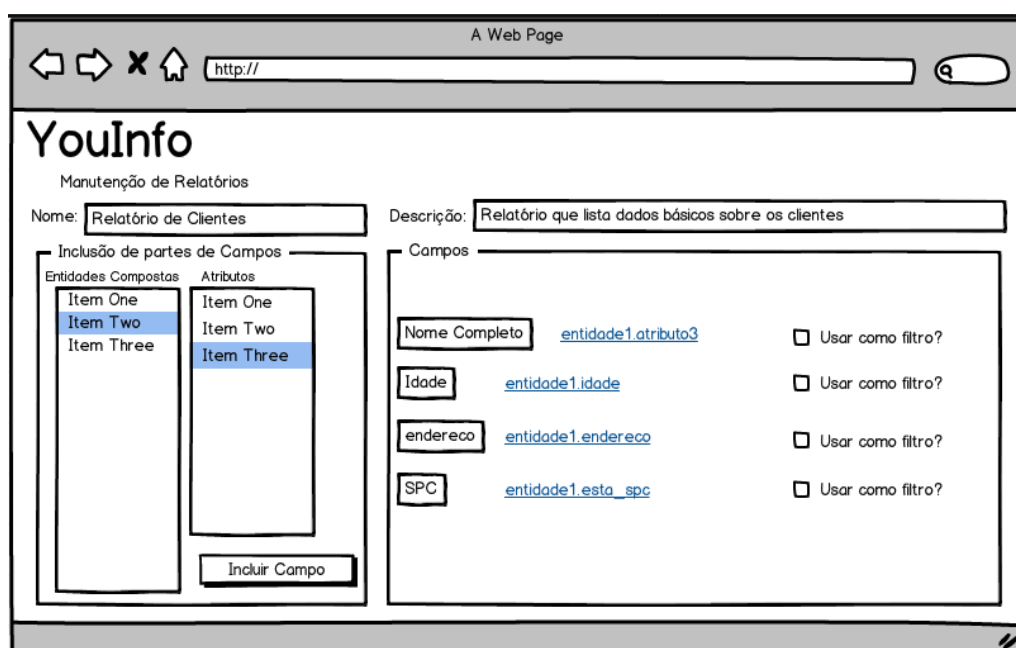


Figura 5.25: Tela 5 - Protótipo não funcional de interface - Manter Relatório.

O protótipo da figura 5.26, refere-se a escolha das condições de formatação dos campos.

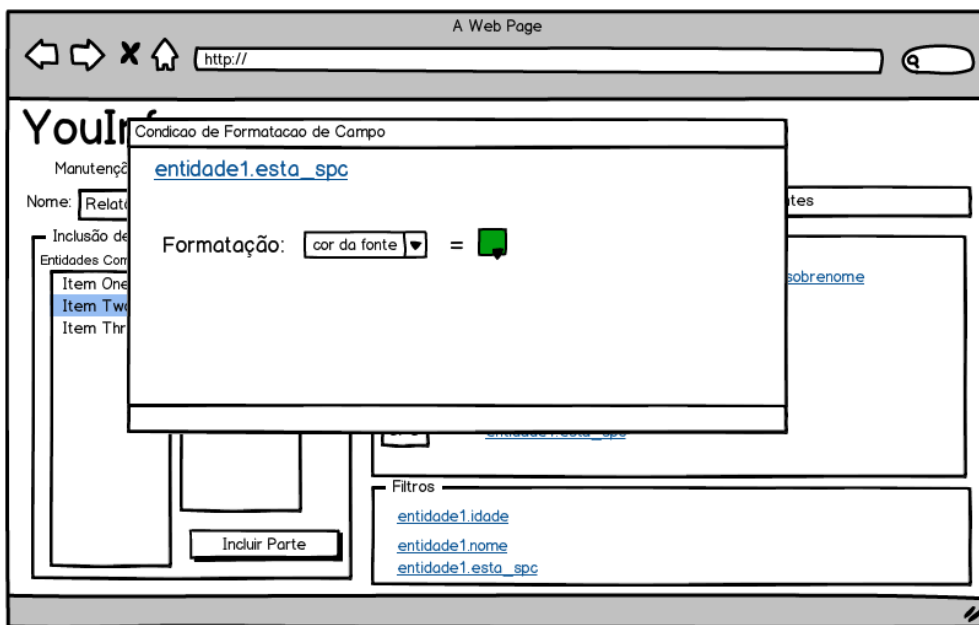


Figura 5.26: Tela 5.1 - Protótipo não funcional de interface - Manter Relatório(condições de formatação).

Nesta tela, pode-se observar a forma dinâmica que podem ser montadas as condições de formatação dos campos do relatório, onde mediante alguma condição definida pelo usuário, é feita alguma formatação visual.

5.9.1.2 Caso de Uso

A figura 5.27 demonstra o caso de uso para a tela manutenção de relatórios.

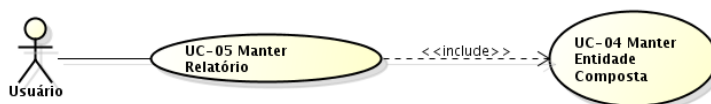


Figura 5.27: Diagrama de caso de uso - Manter Relatório.

5.9.1.3 Modelo de Domínio

A figura 5.28 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-004.

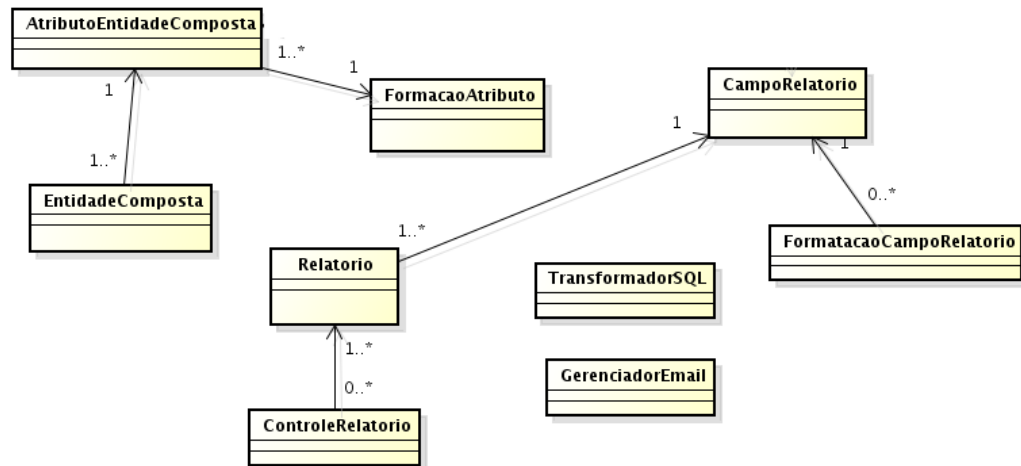


Figura 5.28: Diagrama de domínio - Manter Relatório.

5.9.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.9.2.1 Caso de Uso Detalhado

A tabela 5.7 apresenta o caso de uso detalhado que visa atender ao requisito RF-004.

Tabela 5.7 - Caso de Uso Detalhado Manter Relatório

UC-05	
Nome	Manter Relatório
Descrição	O sistema deve permitir o cadastro, a alteração e a exclusão de relatórios.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-004
Pré-condições	O usuário deve estar autenticado no sistema. Deve haver ao menos uma entidade composta previamente cadastrada.
Pós-condições	Um relatório foi incluído, alterado ou excluído.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Acessa a tela de manutenção de relatórios.	
	2. Lista os relatórios existentes no sistema.
3. Escolhe o relatório que deseja alterar ou excluir, ou clica no botão de incluir novo relatório.	
	4. Mostra a tela de manutenção de relatórios, com os campos para preenchimento, e uma lista com as entidades compostas no sistema para criação do relatório.
5. Preenche os campos de nome e descrição do relatório, ambos obrigatórios.	
	6. Verifica a relação entre as entidade compostas selecionadas. 7. Mostra os atributos das entidades compostas selecionadas.
8. Seleciona o atributo desejado de uma das entidades compostas disponíveis, para formação de um campo do relatório.	
	9. Adiciona o campo selecionado.

<p>10. Selecciona se na geração do relatório, este campo pode ser utilizado como filtro.</p> <p>11. Repetem-se os passos 8, 9 e 10 até que não sejam mais criados campos para o relatório.</p>	
	12. Persiste os dados.
Cenário Alternativo 1	
	<p>2.1 Não encontrou relatórios cadastrados no sistema.</p> <p>2.2 Mostra uma mensagem na tela informando que não existem relatórios cadastrados no sistema, juntamente com o botão de inserir novo relatório.</p>
Cenário Alternativo 2	
	<p>12.1 Verificou alguma inconsistência na montagem dos campos do relatório, como por exemplo campos obrigatórios não informados.</p> <p>12.2 Mostra uma mensagem alertando da inconsistência.</p>
Observações	
<ol style="list-style-type: none"> 1. Os relatórios possuem campos, que por sua vez são provenientes de atributos de entidades compostas. 2. Os campos dos relatórios podem ser provenientes de atributos de entidades compostas. 3. Os relacionamentos entre as entidades compostas são feitos selecionando quais campos serão usados para unir uma a outra entidade, onde para cada n entidades, devem ser selecionadas no mínimo n-1 junções. 4. As formatações dos campos referem-se a forma como eles serão apresentados na tela, no que diz respeito a tipo da fonte, cor da fonte, cor de fundo, e outros tipos de formatações, mediante condições selecionadas pelo usuário. 	

5.9.3 Projeto

5.9.3.1 Diagrama de Sequência

As figuras 7.29 e 7.30 apresentam os diagramas sequência de alterar e remover relatório, atendendo ao requisito RF-004.

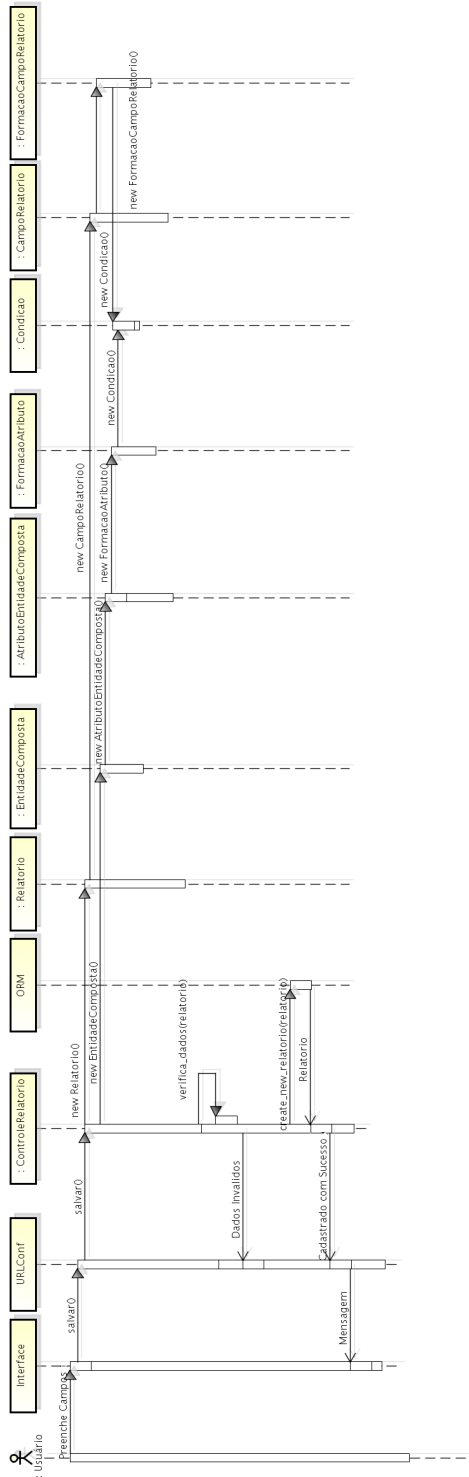


Figura 5.29: Diagrama de sequência - Manter Relatório.

A seguir é apresentado o diagrama de sequência para remover um relatório.

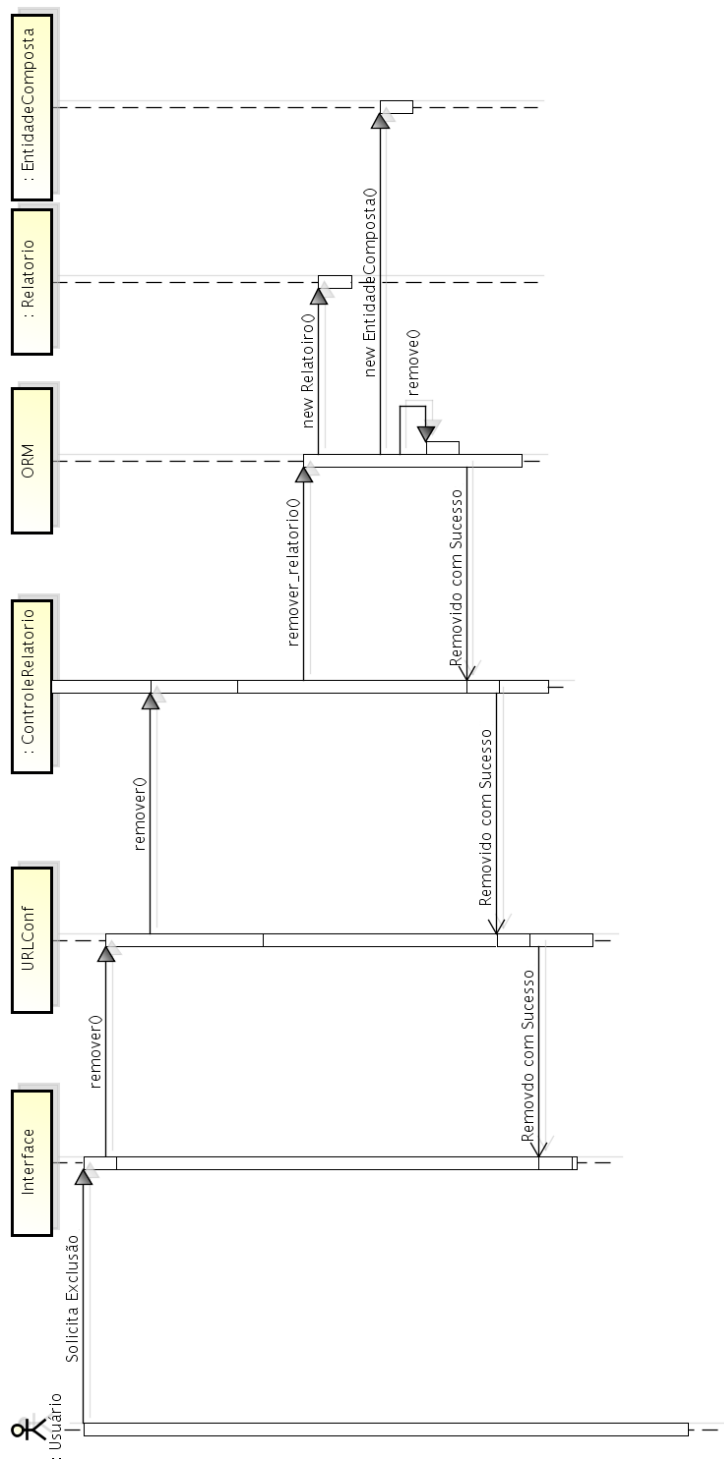


Figura 5.30: Diagrama de sequência - Manter Relatório.

5.9.3.2 Diagrama de Classes

A figura 5.31 apresenta o diagrama classes para atender ao requisito RF-004.

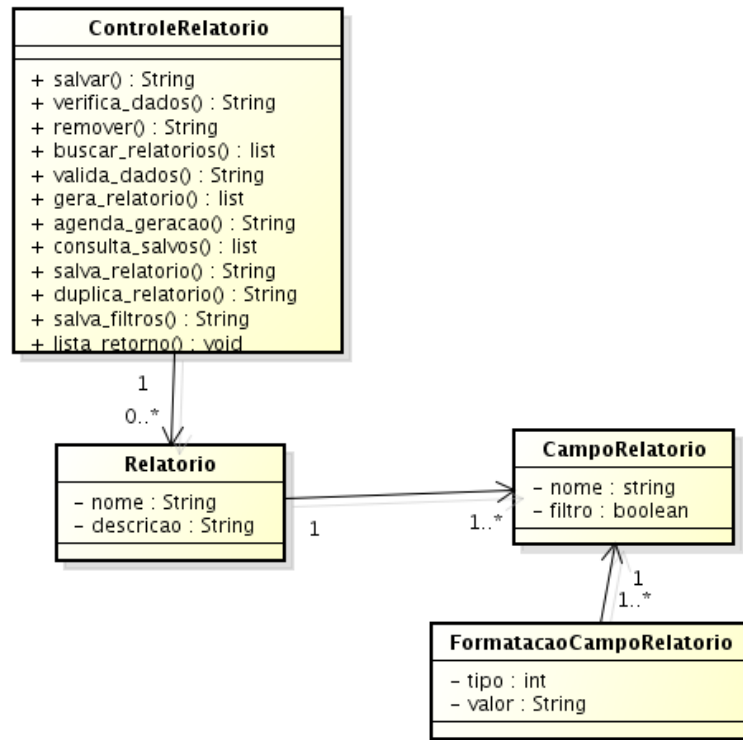


Figura 5.31: Diagrama de classes - Manter Relatório.

5.10 UC-06 - Buscar Relatórios

O sistema deve buscar os relatórios criados, sendo uma parte fundamental para atender ao requisito RF-010 - Gerar Relatório.

5.10.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.10.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.32, visa atender ao requisito RF-010.

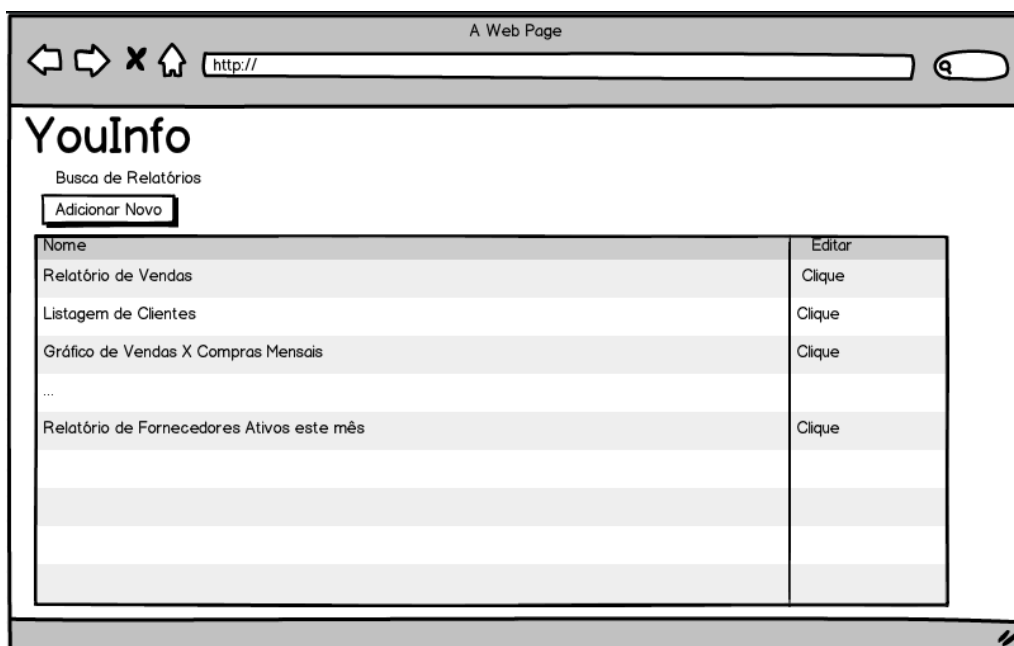


Figura 5.32: Tela 6 - Protótipo não funcional de interface - Buscar Relatórios.

5.10.1.2 Caso de Uso

A figura 5.33 demonstra o caso de uso para a tela busca de relatórios.

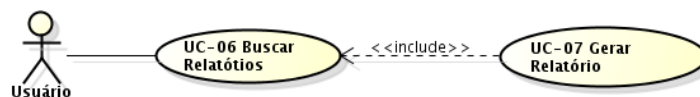


Figura 5.33: Diagrama de caso de uso - Buscar Relatórios.

5.10.1.3 Modelo de Domínio

A figura 5.34 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-010.

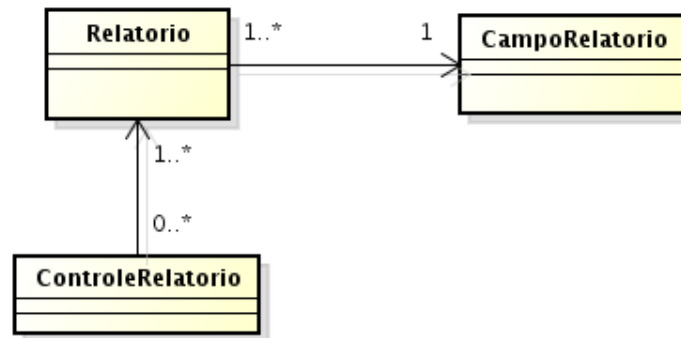


Figura 5.34: Diagrama de domínio - Buscar Relatórios.

5.10.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.10.2.1 Caso de Uso Detalhado

A tabela 5.8 apresenta o caso de uso detalhado que visa atender ao requisito RF-010.

Tabela 5.8 - Caso de Uso Detalhado Buscar Relatório

UC-06	
Nome	Buscar Relatórios
Descrição	O sistema deve permitir a busca dos relatórios criados.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	
Pré-condições	O usuário deve estar autenticado no sistema. São mostrados apenas os relatórios que o ator possui permissão de visualização.
Pós-condições	Os relatórios que o ator possui permissão de geração são listados em tela.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Acessa a tela de relatórios.	
	2. Lista todos os relatórios que o ator possui permissão de geração, se existirem.
Observações	
<p>1. A verificação da possibilidade de geração de gráficos será feita da seguinte forma: serão analisados os campos a serem listados, ao serem encontrados campos numéricos, será oferecida a possibilidade de agrupamento, dos campos vizinhos, somando os valores do campo numérico. Será também analisada a possibilidade de contagem de valores iguais nas colunas, oferecendo um gráfico com o percentual destas colunas sobre o total de registros, oferecendo assim gráficos de pizza.</p>	

5.10.3 Projeto

5.10.3.1 Diagrama de Sequência

A figura 5.35 apresenta o diagrama sequência para atender ao requisito RF-010.

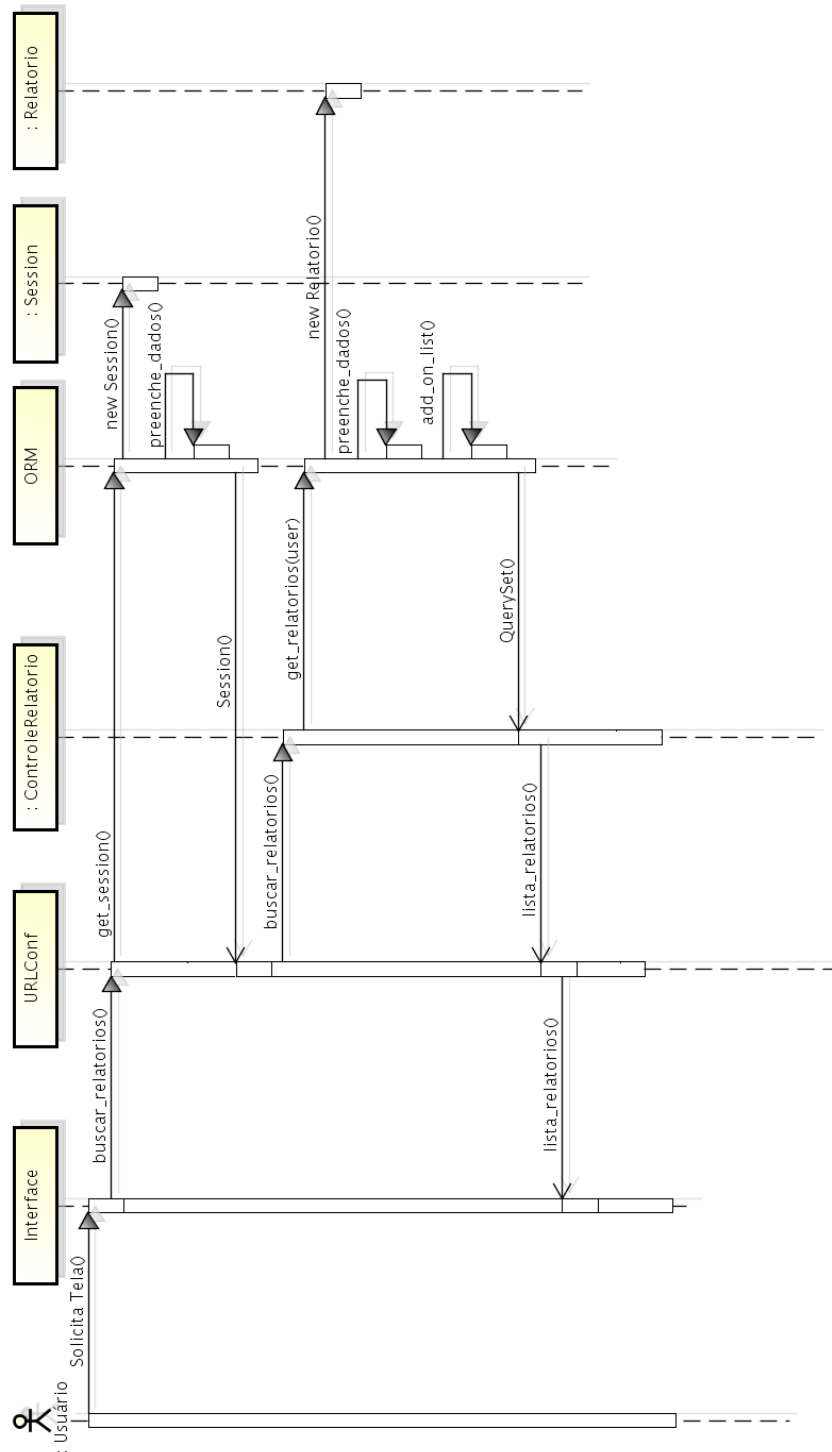


Figura 5.35: Diagrama de sequência - Buscar Relatórios.

5.10.3.2 Diagrama de Classes

A figura 5.10.3.2 apresenta o diagrama classes para atender ao requisito RF-001.

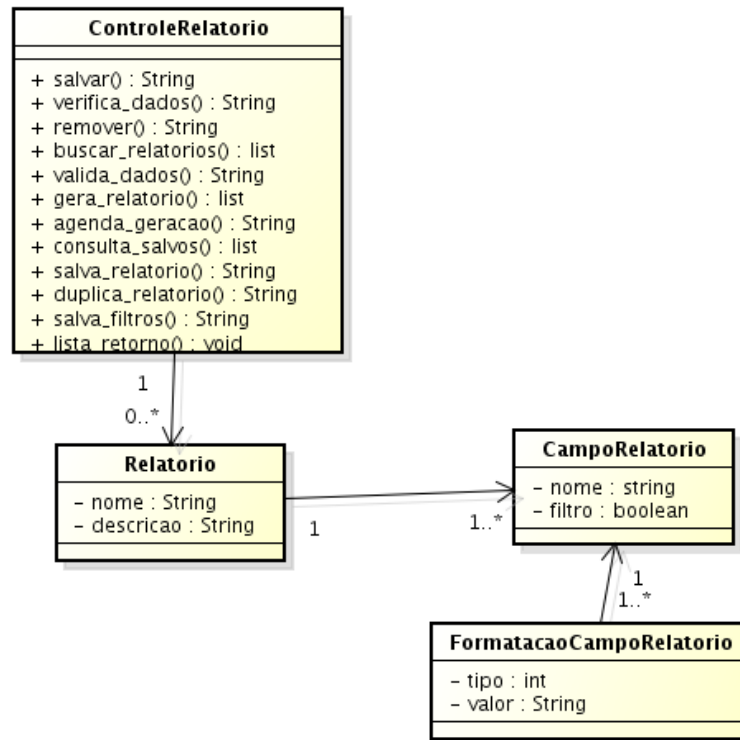


Figura 5.36: Diagrama de classes - Buscar Relatórios.

5.11 UC-07 - Gerar Relatório

O sistema deve permitir a geração de relatórios, atendendo aos requisitos funcionais RF-010, RF-013 e RF-018.

5.11.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.11.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.37, visa atender aos requisitos RF-010, RF-013 e RF-018.

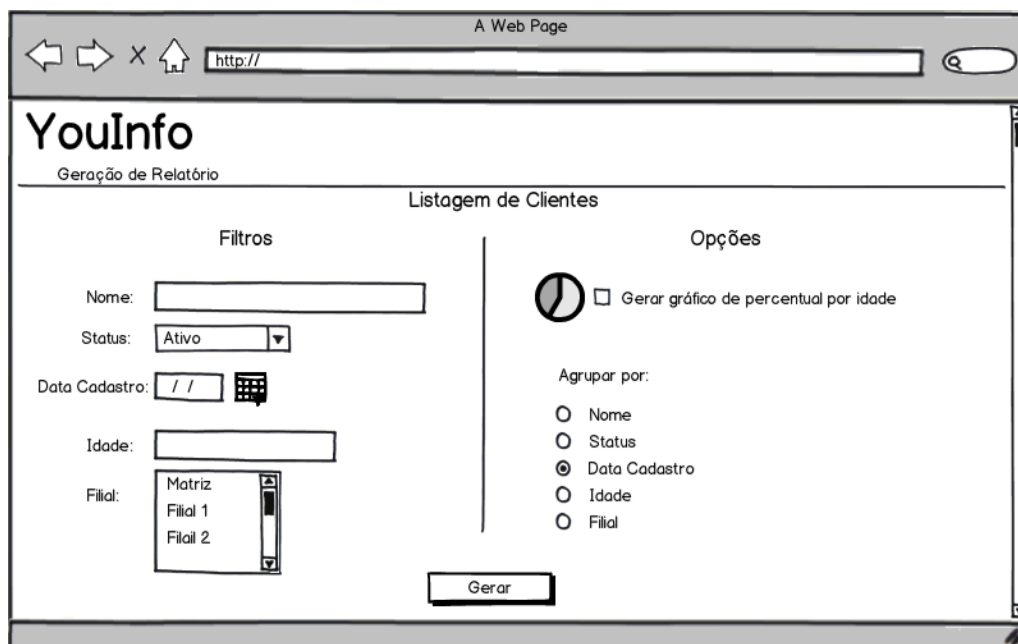


Figura 5.37: Tela 7 - Protótipo não funcional de interface - Gerar Relatório.

O protótipo da figura 5.38, mostra um exemplo de relatório após ser gerado.

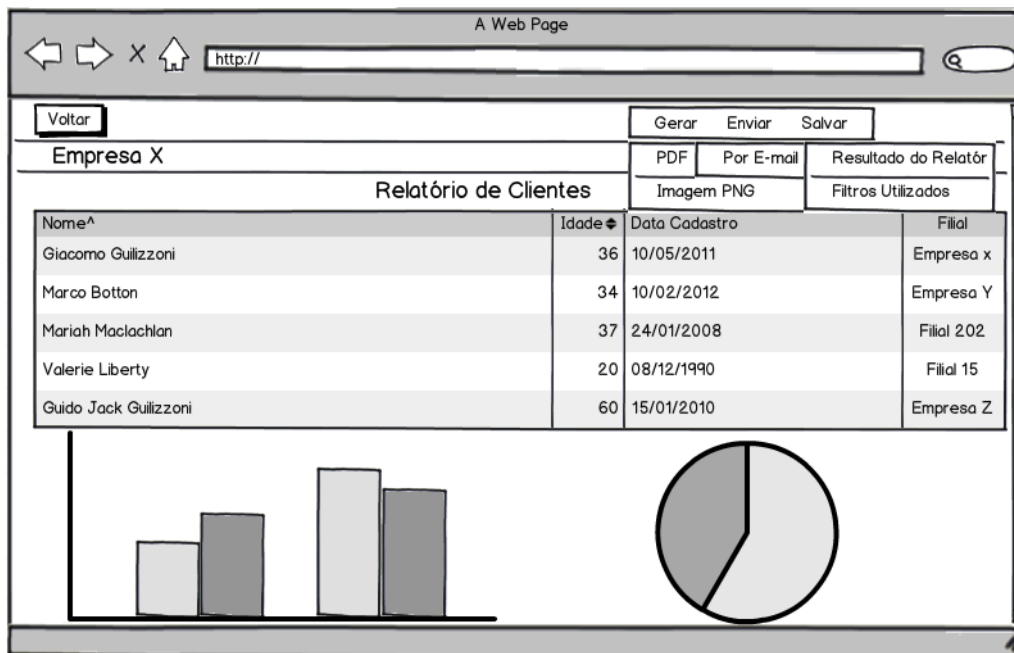


Figura 5.38: Tela 7.1 - Protótipo não funcional de interface - Gerar Relatório(resultado).

Pode-se observar na figura a existência do menu drop-down, onde são exibidas as opções possíveis após a geração do relatório, como por exemplo, exportar para PDF, PNG ou ainda enviar o relatório por e-mail.

5.11.1.2 Caso de Uso

A figura 5.39 demonstra o caso de uso para a tela de geração de relatório.

5.11.1.3 Modelo de Domínio

A figura 5.40 apresenta o modelo de domínio responsável por atender aos requisitos funcionais RF-010, RF-013 e RF-018.

5.11.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.11.2.1 Caso de Uso Detalhado

A tabela 5.9 apresenta o caso de uso detalhado que visa atender ao requisito RF-007.

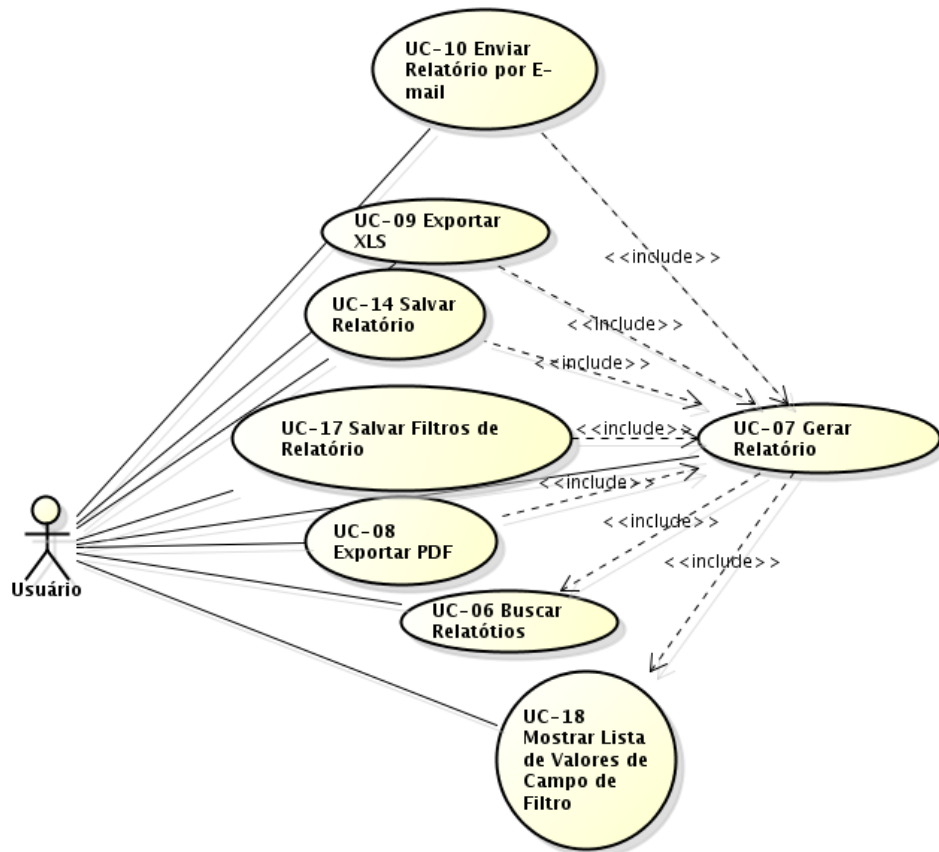


Figura 5.39: Diagrama de caso de uso - Gerar Relatório.

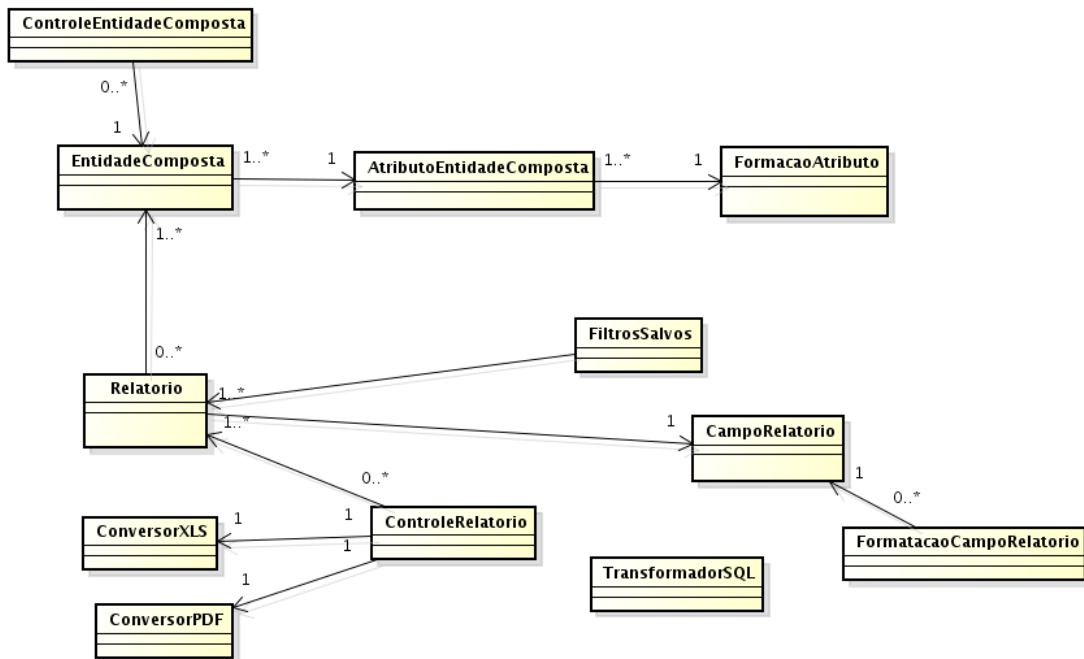


Figura 5.40: Diagrama de domínio - Gerar Relatório.

Tabela 5.9 - Caso de Uso Detalhado Gerar Relatório

UC-07	
Nome	Gerar Relatório
Descrição	O sistema deve permitir a geração do relatório criado, mediante seleção de opções e filtros.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-010, RF-013
Pré-condições	O usuário deve estar autenticado no sistema. O usuário deve possuir permissão para a geração do relatório.
Pós-condições	Um relatório foi gerado e visualizado, exportado para PDF, exportado para XLS, enviado por e-mail ou salvo.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Preenche os filtros desejados. 2. Seleciona as opções desejadas.	
	3. Verifica se os campos obrigatórios foram preenchidos. 4. Executa a geração do relatório. 5. Exibe o relatório na tela, com as opções de exportação para PDF ou XLS, envio de e-mail com o relatório, salvar o relatório ou salvar os filtros e opções utilizados na geração do relatório. 6. Salva um histórico de geração do relatório.
Cenário Alternativo 1	
	3.1 Verificou que um ou mais campos(filtros) marcados como obrigatórios não foram preenchidos. 3.2 Retorna para a tela, com os dados preenchidos pelo autor, informando junto ao campo a obrigatoriedade do mesmo.

5.11.3 Projeto

5.11.3.1 Diagrama de Sequência

A figura 5.41 apresenta o diagrama sequência para atender ao requisito RF-007.

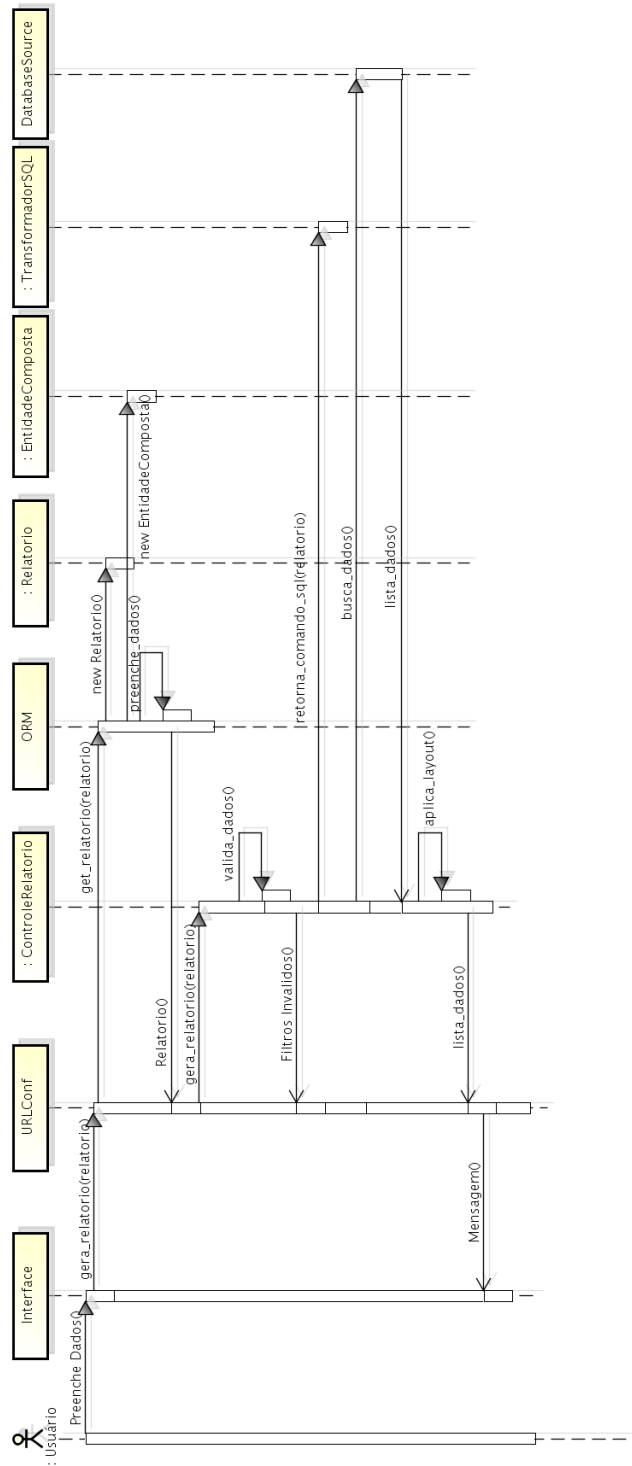


Figura 5.41: Diagrama de sequência - Gerar Relatório.

5.11.3.2 Diagrama de Classes

A figura 5.42 apresenta o diagrama classes para atender aos requisitos RF-010, RF-013 e RF-018.

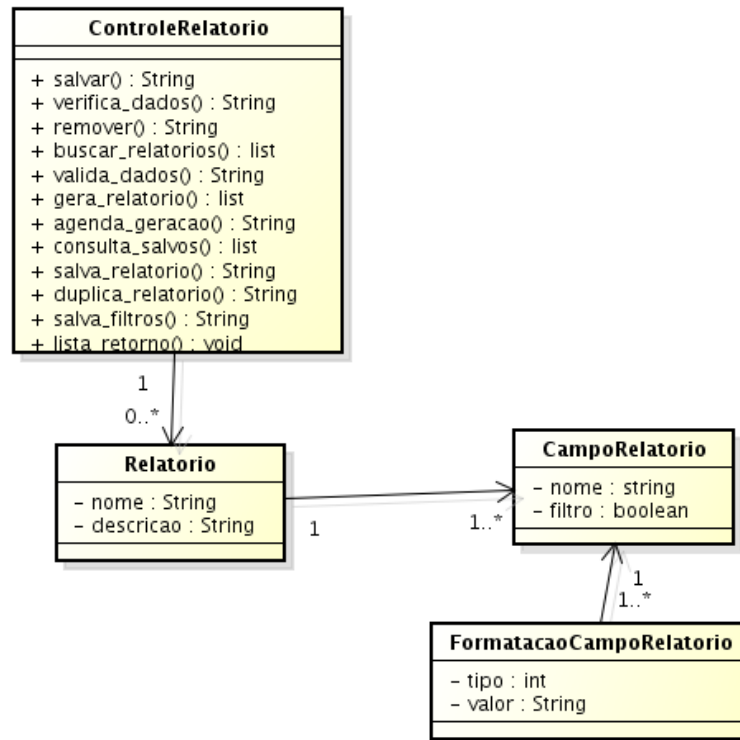


Figura 5.42: Diagrama de classes - Gerar Relatório.

5.12 UC-08 Exportar PDF

O sistema deve permitir a exportação do resultado dos relatórios para PDF, atendendo ao requisito RF-011 - Exportar PDF.

5.12.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.12.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.43, visa atender ao requisito RF-011.

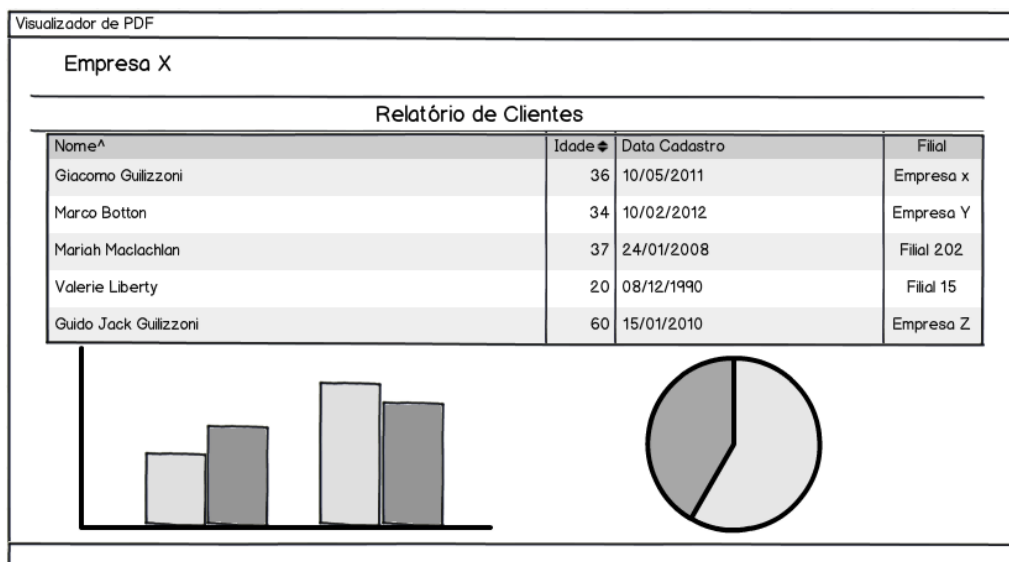


Figura 5.43: Tela 8 - Protótipo não funcional de interface - Exportar PDF.

Embora visualmente a figura 5.43 seja muito parecida com a tela 7.1, a tela 8 é uma representação do relatório gerado no formato PDF.

5.12.1.2 Caso de Uso

A figura 5.44 demonstra o caso de uso para a tela de geração de PDF.

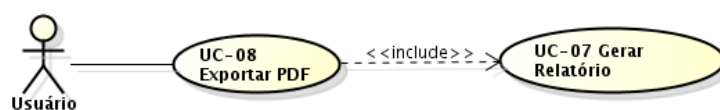


Figura 5.44: Diagrama de caso de uso - Exportar PDF.

5.12.1.3 Modelo de Domínio

A figura 5.45 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-011.



Figura 5.45: Diagrama de domínio - Exportar PDF.

5.12.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.12.2.1 Caso de Uso Detalhado

A tabela 5.10 apresenta o caso de uso detalhado que visa atender ao requisito RF-011.

Tabela 5.10 - Caso de Uso Detalhado Exportar PDF

UC-08	
Nome	Exportar PDF
Descrição	O sistema deve permitir a exportação dos relatórios gerados para o formato PDF.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-011
Pré-condições	O usuário deve estar autenticado no sistema. O usuário estar com um relatório em formato HTML gerado na tela.
Pós-condições	Um relatório foi exportado para o formato PDF.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. No menu superior da tela de visualização do relatório, clica em exportar para PDF	
	3. Converte os dados gerados em tela para o formato PDF. 4. Disponibiliza para o usuário como download.

5.12.3 Projeto

5.12.3.1 Diagrama de Sequência

A figura 5.46 apresenta o diagrama sequência para atender ao requisito RF-011.

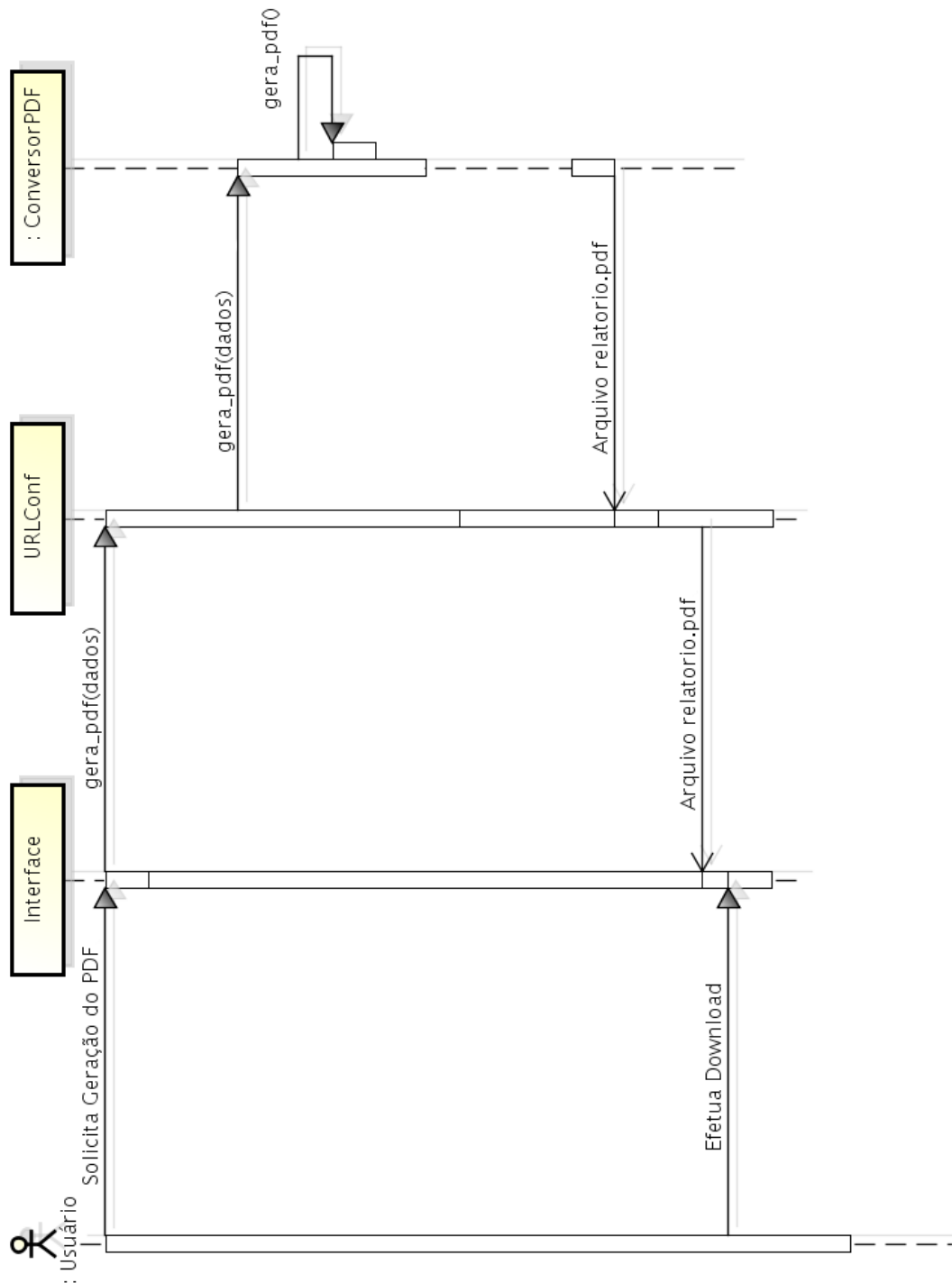


Figura 5.46: Diagrama de sequência - Exportar PDF.

Neste diagrama de sequência, os dados, já gerados em tela, são passados para a classe responsável pela conversão dos mesmos para um arquivo no formato PDF.

5.12.3.2 Diagrama de Classes

A figura 5.47 apresenta o diagrama classes para atender ao requisito RF-011.

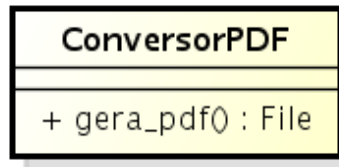


Figura 5.47: Diagrama de classes - Exportar PDF.

5.13 UC-09 Exportar XLS(planilha eletrônica)

O sistema deve permitir a exportação do resultado dos relatórios para XLS(planilha eletrônica), atendendo ao requisito RF-012 - Exportar Imagem.

5.13.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.13.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.48, visa atender ao requisito RF-012.

Name	Age	Nickname
Giacomo Guilizzoni Founder & CEO	36	Peldi
Marco Botton Tuttofare	34	
Mariah Maclachlan Better Half	37	Patata
Valerie Liberty Head Chef	:)	Val
Guido Jack Guilizzoni	6	The Guids

Figura 5.48: Tela 9 - Protótipo não funcional de interface - Exportar XLS.

5.13.1.2 Caso de Uso

A figura 5.49 demonstra o caso de uso para a tela de geração de XLS.

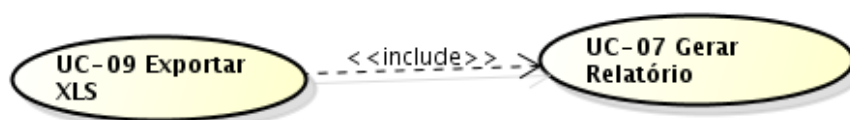


Figura 5.49: Diagrama de caso de uso - Exportar XLS.

5.13.1.3 Modelo de Domínio

A figura 5.50 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-012.



Figura 5.50: Diagrama de domínio - Exportar XLS.

5.13.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.13.2.1 *Caso de Uso Detalhado*

A tabela 5.11 apresenta o caso de uso detalhado que visa atender ao requisito RF-012.

Tabela 5.11 - Caso de Uso Detalhado Exportar XLS

UC-09	
Nome	Exportar XLS
Descrição	O sistema deve permitir a exportação dos relatórios gerados para o formato XLS.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-011
Pré-condições	O usuário deve estar autenticado no sistema. O usuário estar com um relatório em formato HTML gerado na tela.
Pós-condições	Um relatório foi exportado para o formato XLS.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. No menu superior da tela de visualização do relatório, clica em exportar para XLS	
	3. Converte os dados gerados em tela para o formato XLS(planilha eletrônica). 4. Disponibiliza para o usuário como download.

5.13.3 Projeto

5.13.3.1 Diagrama de Sequência

A figura 5.51 apresenta o diagrama sequência para atender ao requisito RF-012.

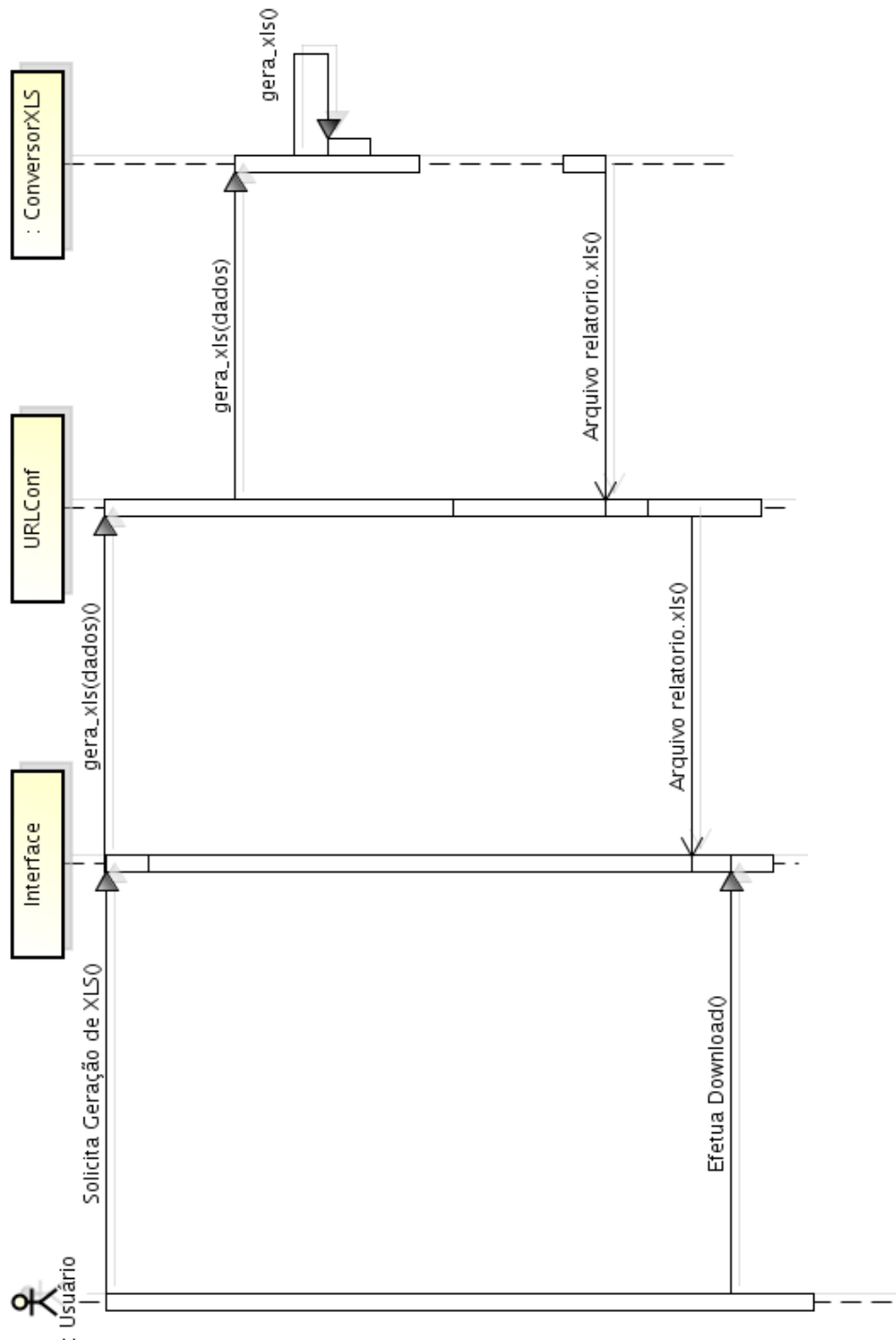


Figura 5.51: Diagrama de sequência - Exportar XLS.

Neste diagrama de sequência, os dados, já gerados em tela, são passados para a classe responsável pela conversão dos mesmos para o formato de planilha eletrônica(XLS).

5.13.3.2 Diagrama de Classes

A figura 5.52 apresenta o diagrama classes para atender ao requisito RF-012.



Figura 5.52: Diagrama de classes - Exportar XLS.

5.14 UC-10 Enviar Relatório por E-mail

O sistema deve permitir enviar os relatórios gerados por e-mail, atendendo ao requisito RF-009 - Enviar Relatório por E-mail.

5.14.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.14.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.53, visa atender ao requisito RF-009.

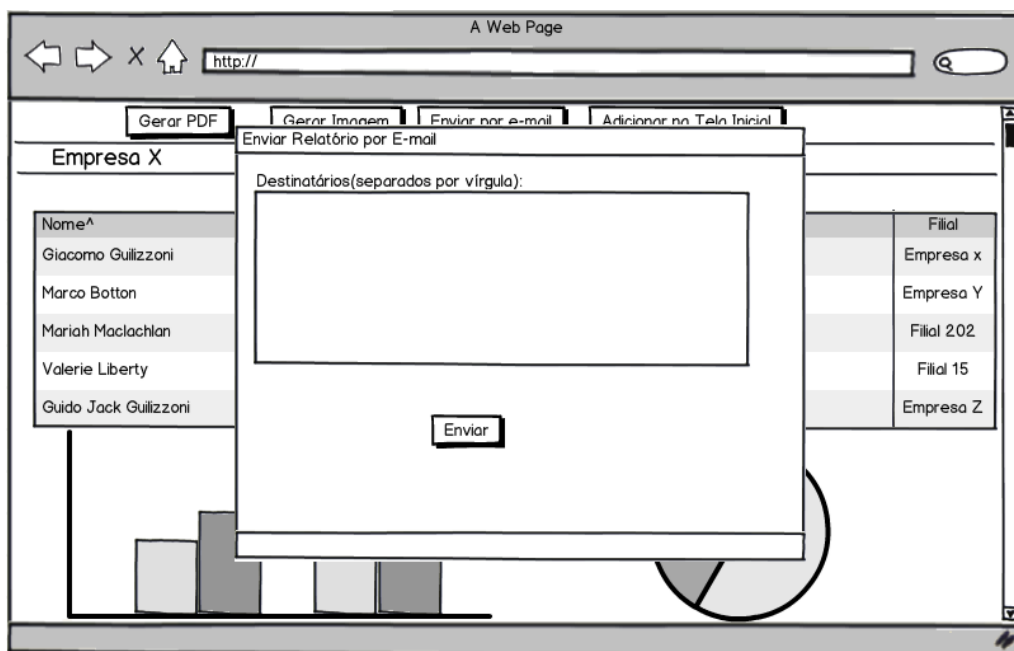


Figura 5.53: Tela 10 - Protótipo não funcional de interface - Enviar Relatório por E-mail.

Nesta tela, com os dados do relatório já gerados em tela, é selecionada a opção do menu drop-down de enviar relatório por e-mail, assim é aberta uma tela modal, solicitando ao ator que informe os destinatários do e-mail em questão, e o assunto do e-mail.

5.14.1.2 Caso de Uso

A figura 5.54 demonstra o caso de uso para a tela de envio de e-mail do relatório.

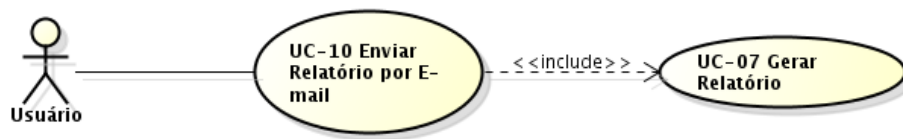


Figura 5.54: Diagrama de caso de uso - Enviar Relatório por E-mail.

5.14.1.3 Modelo de Domínio

A figura 5.55 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-009.

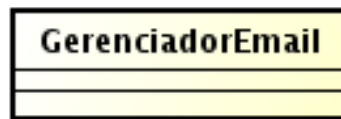


Figura 5.55: Diagrama de domínio - Enviar Relatório por E-mail.

5.14.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.14.2.1 Caso de Uso Detalhado

A tabela 5.12 apresenta o caso de uso detalhado que visa atender ao requisito RF-009.

Tabela 5.12 - Caso de Uso Detalhado Enviar Relatório por E-mail

UC-10	
Nome	Enviar Relatório por e-mail
Descrição	O sistema deve permitir o envio de um relatório gerado por e-mail
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-009
Pré-condições	O usuário deve estar autenticado no sistema. O usuário estar com um relatório em formato HTML gerado na tela.
Pós-condições	Um relatório foi enviado por e-mail.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. No menu superior da tela de visualização do relatório, clica em enviar por e-mail.	
	2. Mostra uma tela para preenchimento dos campos de envio de e-mail.
3. Preenche os campos adequados(campos obrigatórios estão marcados com *): a. *Destinatários O campo destinatários é informado separando os destinatários por vírgula.	
	4. Verifica se os campos obrigatórios estão preenchidos. 5. Verifica se o campo destinatários está correto, ou seja, emails dos destinatários, separados por vírgula. 6. Envia o e-mail.
Cenário Alternativo 1	
	4.1 Verificou que um ou mais campos obrigatórios não foram preenchidos. 4.2 Retorna para a tela, com os dados preenchidos pelo autor, informando junto ao campo a obrigatoriedade do mesmo.
Cenário Alternativo 2	

	<p>5.1 Verificou que o campo destinatários não está corretamente preenchido.</p> <p>5.2 Retorna para a tela, com os dados preenchidos pelo autor, informando junto ao campo que o mesmo deve ser preenchido da seguinte forma: email@gmail.com, email2@provedor.com.</p>
Cenário Alternativo 3	
	<p>6.1 Ocorreu erro ao enviar o e-mail, relativo a algum erro no servido de e-mails ou falha temporária na conexão com a internet.</p> <p>6.2 Retorna para a tela, com os dados preenchidos pelo autor, informando sobre o erro, solicitando ao usuário tentar novamente mais tarde, ou agendar uma geração para mais tarde.</p>

5.14.3 Projeto

5.14.3.1 Diagrama de Sequência

A figura 5.56 apresenta o diagrama sequência para atender ao requisito RF-009.

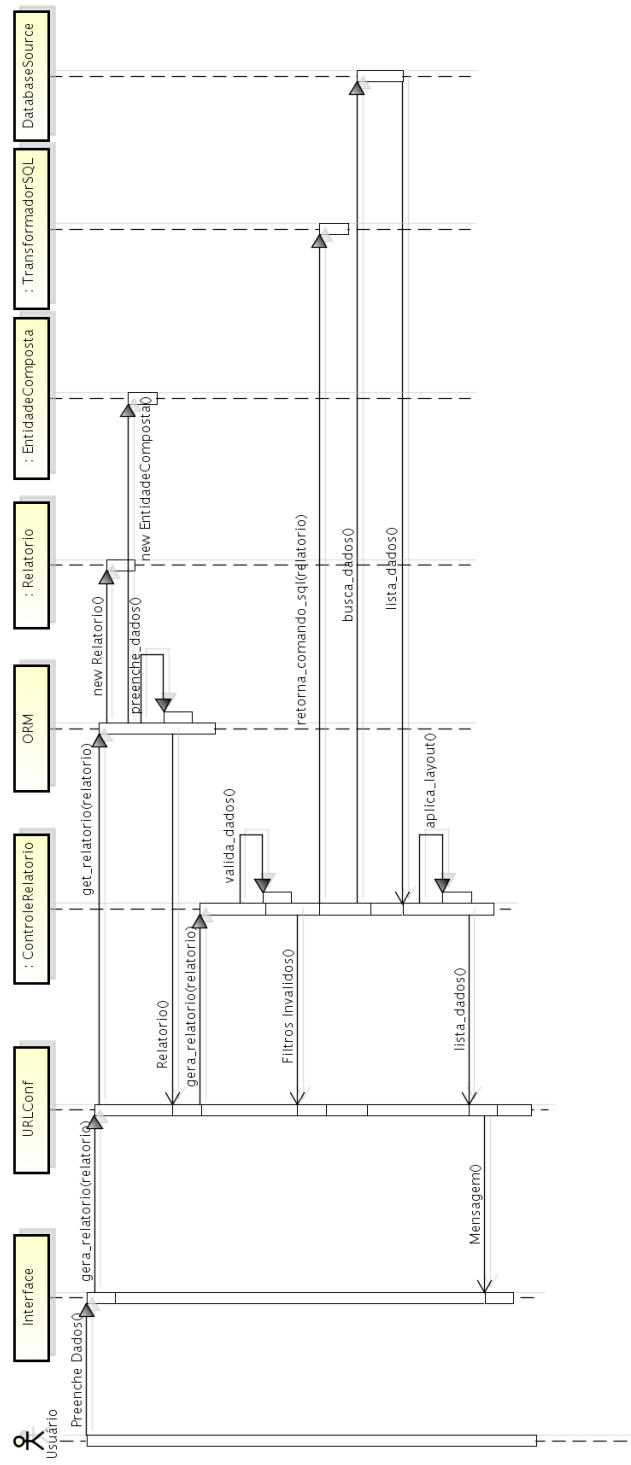


Figura 5.56: Diagrama de sequência - Enviar Relatório port E-mail.

5.14.3.2 Diagrama de Classes

A figura 5.57 apresenta o diagrama classes para atender ao requisito RF-009.

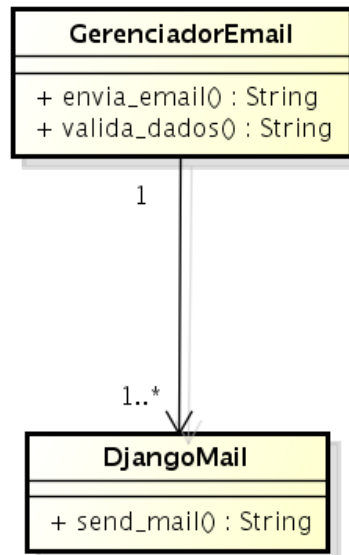


Figura 5.57: Diagrama de classes - Enviar Relatório por E-mail.

5.15 UC-11 - Agendar Geração de Relatório

O sistema deve verificar se o indivíduo tem permissão para acessá-lo, atendendo ao requisito RF-014 - Agendar Geração de Relatório.

5.15.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.15.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.58, visa atender ao requisito RF-014.

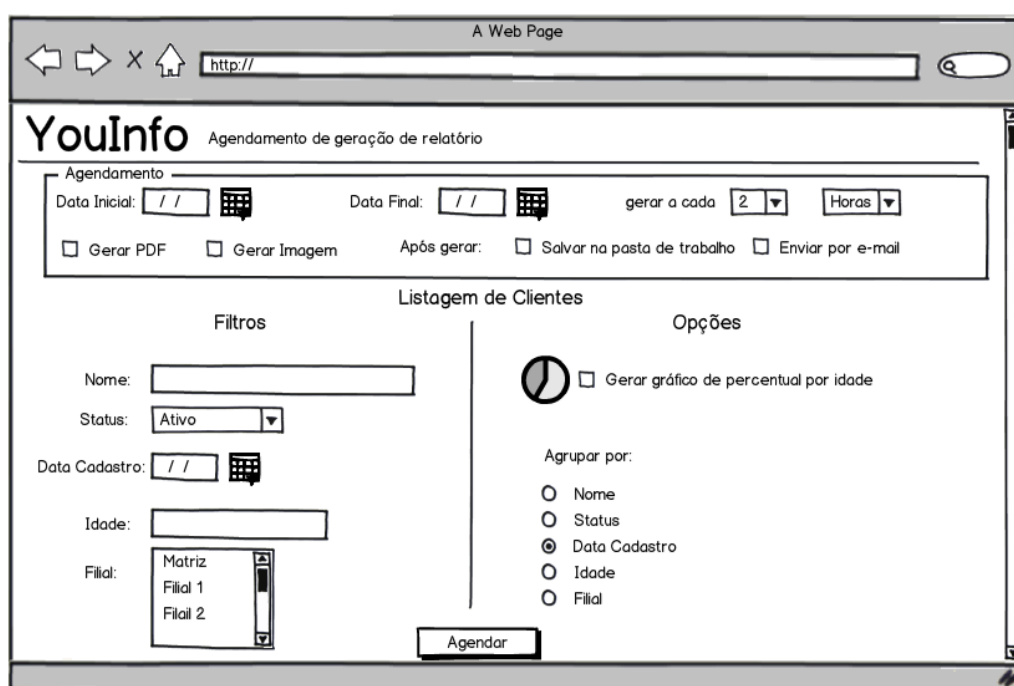


Figura 5.58: Tela 11 - Protótipo não funcional de interface - Agendar Geração de Relatório.

Esta tela é uma tela normal de geração de relatórios, porém com alguns campos adicionais para que seja possível agendar datas e frequência para a geração automática dos relatórios.

5.15.1.2 Caso de Uso

A figura 5.59 demonstra o caso de uso para a tela de agendamento de geração de relatório.

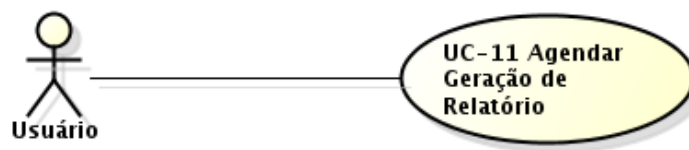


Figura 5.59: Diagrama de caso de uso - Agendar Geração de Relatório.

5.15.1.3 Modelo de Domínio

A figura 5.60 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-014.

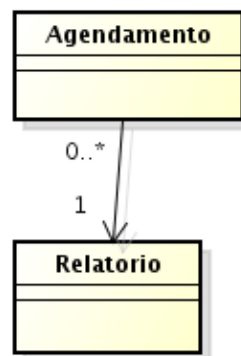


Figura 5.60: Diagrama de domínio - Agendar Geração de Relatório.

5.15.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.15.2.1 Caso de Uso Detalhado

A tabela 5.13 apresenta o caso de uso detalhado que visa atender ao requisito RF-014.

Tabela 5.13 - Caso de Uso Detalhado Agendar Geração de Relatório

UC-11	
Nome	Agendar Geração de Relatório
Descrição	O sistema deve permitir o agendamento de geração automática de relatórios.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-014
Pré-condições	O usuário deve estar autenticado no sistema. O usuário deve possuir permissão para a geração do relatório.
Pós-condições	Um relatório foi agendado para geração automática, para ser salvo na pasta de trabalho do usuário, ou enviado por e-mail.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
<p>1. Preenche os campos de agendamento de geração adequados(campos marcados com * são obrigatórios):</p> <p>a. *Data Inicial</p> <p>b. *Data Final</p> <p>c. *Intervalo de Geração</p> <p>d. *Ação após Gerar</p> <p>2. Preenche os filtros desejados.</p> <p>3. Seleciona as opções desejadas.</p>	
	<p>4. Verifica se os campos obrigatórios foram preenchidos.</p> <p>5. Verifica se a data inicial de agendamento é menor ou igual que a data final de agendamento.</p> <p>6. Persiste os dados, agendando a(s) geração(ões) do relatório.</p>
Cenário Alternativo 1	
	<p>4.1 Verificou que um ou mais campos ou filtros marcados como obrigatórios não foram preenchidos.</p> <p>4.2 Retorna para a tela, com os dados</p>

	preenchidos pelo autor, informando junto ao campo a obrigatoriedade do mesmo.
Cenário Alternativo 2	
	5.1 Verificou que a data inicial de agendamento é maior que a data final . 5.2 Retorna para a tela, com os dados preenchidos pelo autor, informando que a data inicial deve ser menor ou igual a data final de agendamento.

5.15.3 Projeto

5.15.3.1 Diagrama de Sequência

A figura 5.61 apresenta o diagrama sequência para atender ao requisito RF-014.

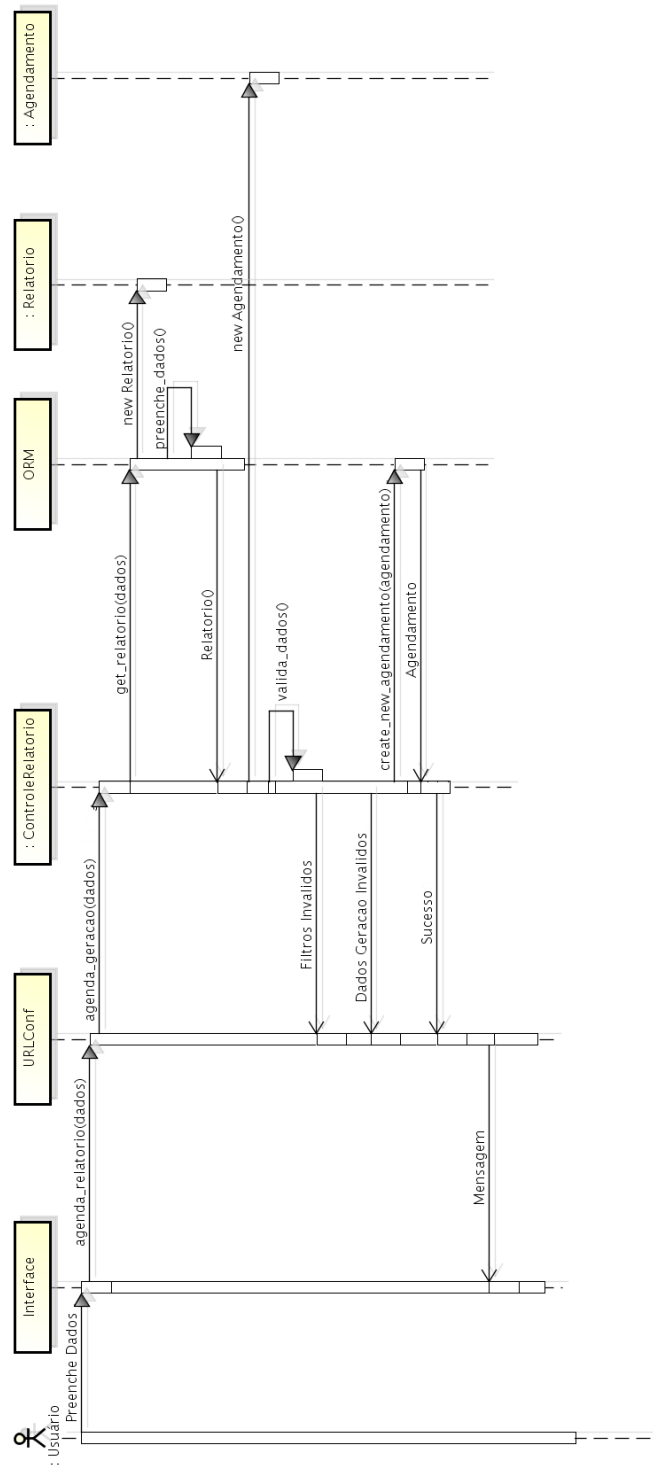


Figura 5.61: Diagrama de sequência - Agendar Geração Relatório.

5.15.3.2 Diagrama de Classes

A figura 5.62 apresenta o diagrama classes para atender ao requisito RF-014.

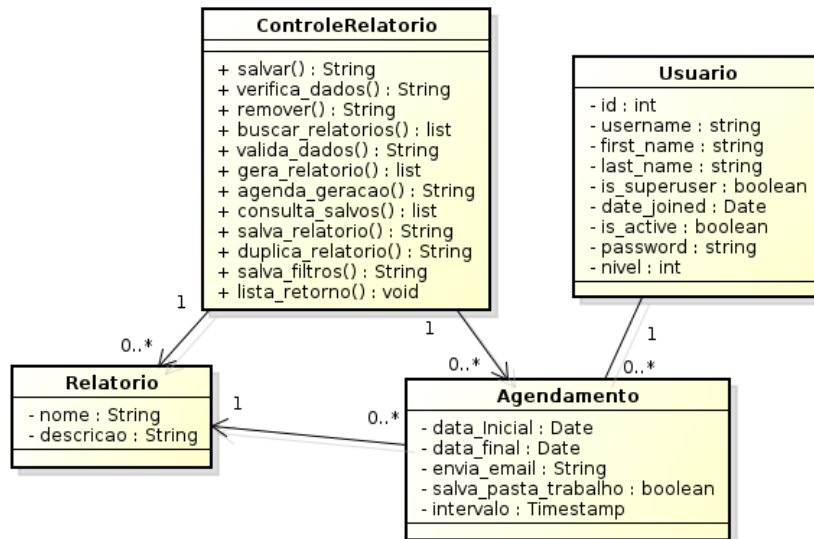


Figura 5.62: Diagrama de classes - Agendar Geração de Relatório.

5.16 UC-12 Gerar Relatório Agendado

O sistema deve gerar os relatórios agendados, atendendo ao requisito RF-008.

5.16.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.16.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.63, visa atender ao requisito RF-008.

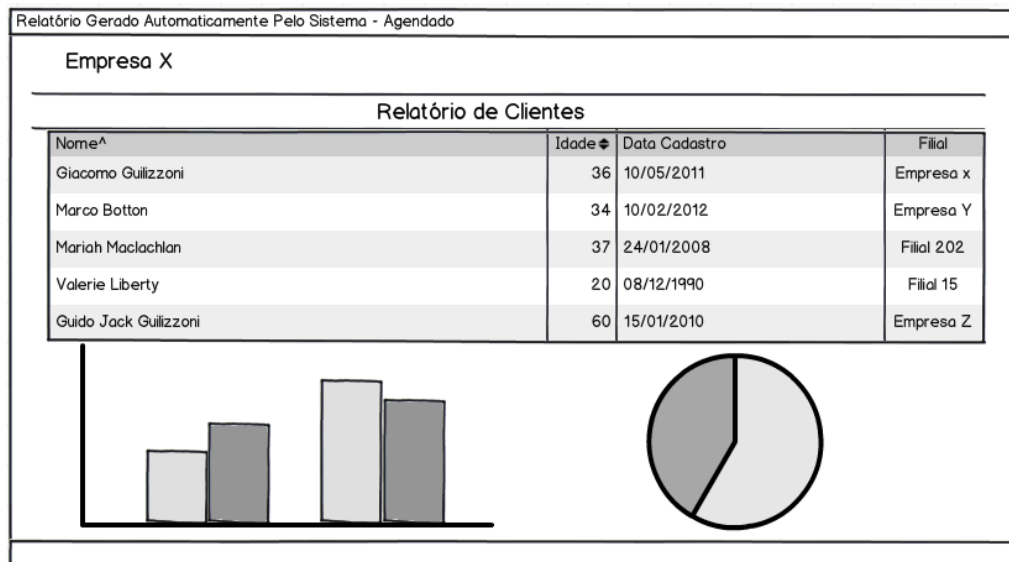


Figura 5.63: Tela 12 - Protótipo não funcional de interface - Gerar Relatório Agendado.

5.16.1.2 Caso de Uso

A figura 5.64 demonstra o caso de uso Gerar Relatório Agendado.



Figura 5.64: Diagrama de caso de uso - Gerar Relatório Agendado.

5.16.1.3 Modelo de Domínio

A figura 5.65 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-008.

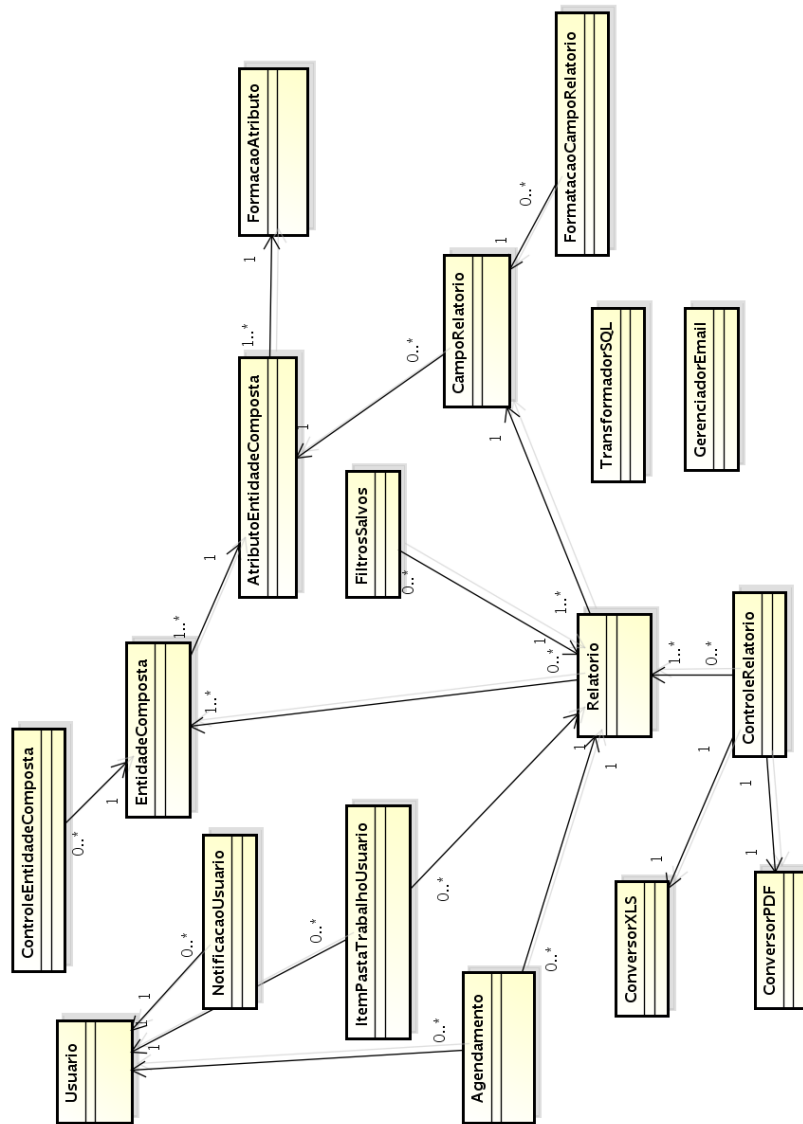


Figura 5.65: Diagrama de domínio - Gerar Relatório Agendado.

5.16.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.16.2.1 Caso de Uso Detalhado

A tabela 5.14 apresenta o caso de uso detalhado que visa atender ao requisito RF-008.

Tabela 5.14 - Caso de Uso Detalhado Gerar Relatório Agendado

UC-12	
Nome	Gerar Relatório Agendado
Descrição	O sistema deve permitir o envio de um relatório gerado por e-mail
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-008
Pré-condições	
Pós-condições	Um relatório agendado foi gerado e enviado por e-mail ou salvo na pasta de trabalho do usuário responsável pelo agendamento.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
	1. De tempos em tempos, verifica todos os relatórios que estão agendados para geração.
	2. Os relatórios cuja hora de geração seja igual a data atual, são gerados e enviados para os e-mails configurados, ou salvos na pasta de trabalho dos respectivos usuários responsáveis pelo agendamento das gerações.
	3. Cadastra uma notificação para o usuário avisando que o relatório agendado foi gerado com sucesso e está salvo na sua área de trabalho ou enviado para seu e-mail.
Cenário Alternativo 1	
	2.1 Ocorreu erro ao enviar o e-mail, relativo a algum erro no servido de e-mails ou falha temporária na conexão com a internet. 2.2 Cancela o envio do e-mail, cadastra uma mensagem para o usuário avisando o mesmo que o envio falhou.

5.16.2.2 Diagrama de Sequência

O diagrama de sequência deste caso de uso possui o mesmo fluxo de mensagens do diagrama de sequência do caso de uso 7 - Gerar Relatório.

5.16.2.3 Diagrama de Classes

A figura 5.66 apresenta o diagrama classes para atender ao requisito RF-008.

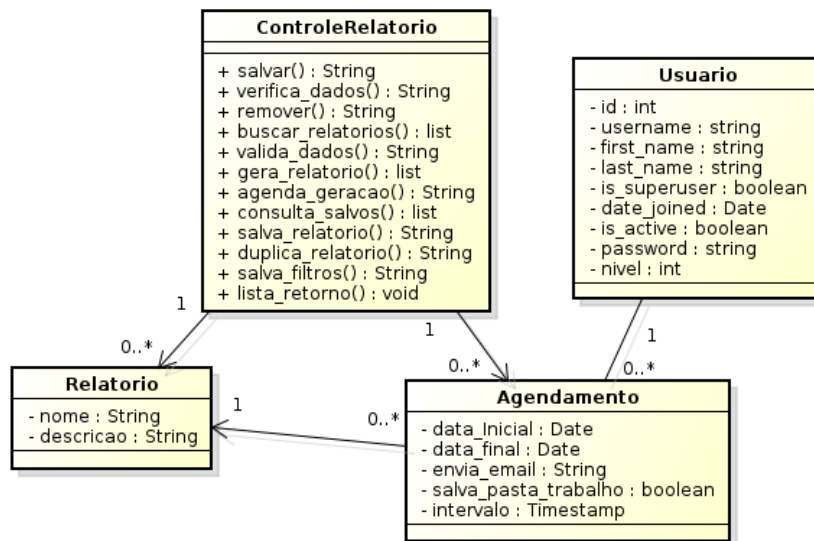


Figura 5.66: Diagrama de classes - Gerar Relatório Agendado.

5.17 UC-13 Consultar Relatórios Salvos

O sistema permitir ao ator consultar relatórios salvos, atendendo ao requisito RF-006 - Consultar Relatórios Salvos.

5.17.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.17.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.67, visa atender parcialmente ao requisito RF-006.

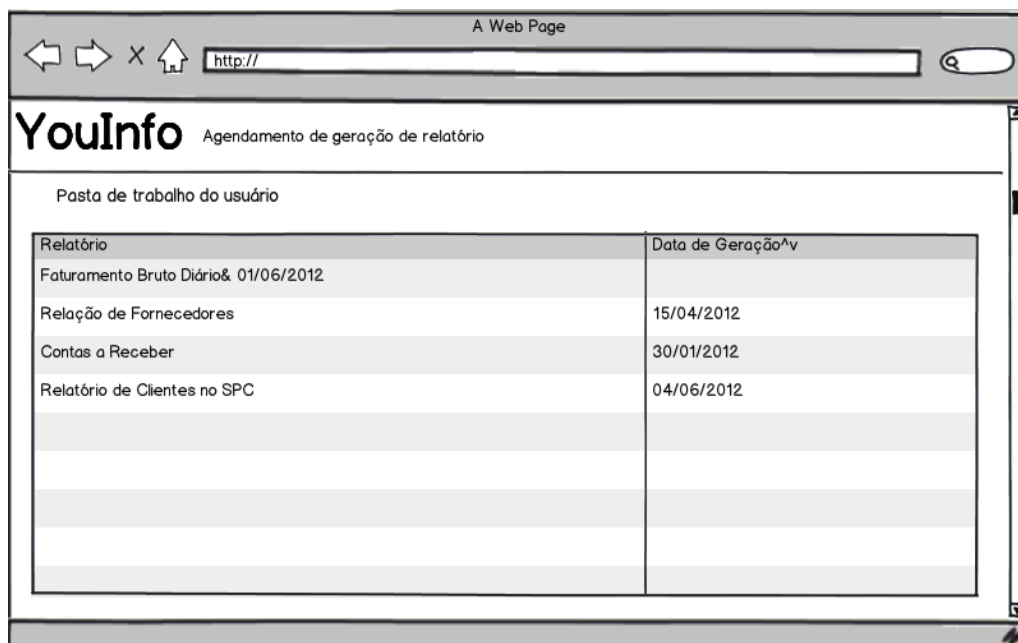


Figura 5.67: Tela 13 - Protótipo não funcional de interface - Consultar Relatórios Salvos.

5.17.1.2 Caso de Uso

A figura 5.68 demonstra o caso de uso para a tela de consulta de relatórios salvos.

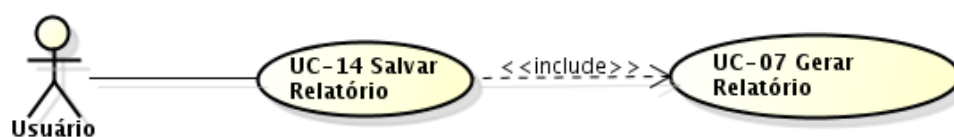


Figura 5.68: Diagrama de caso de uso - Consultar Relatórios Salvos.

5.17.1.3 Modelo de Domínio

A figura 5.69 apresenta o modelo de domínio responsável por atender parcialmente ao requisito funcional RF-006.

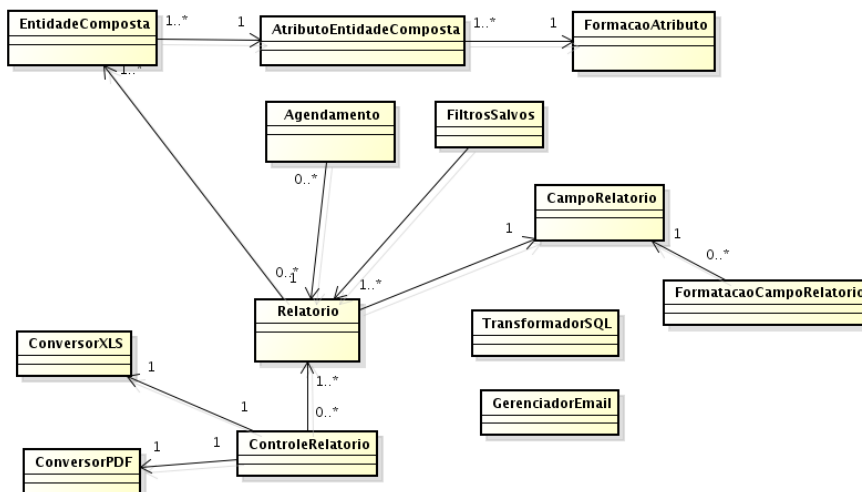


Figura 5.69: Diagrama de domínio - Consultar Relatórios Salvos.

5.17.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.17.2.1 Caso de Uso Detalhado

A tabela 5.15 apresenta o caso de uso detalhado que visa atender parcialmente ao requisito RF-006.

Tabela 5.15 - Caso de Uso Detalhado Consulta Relatórios Salvos

UC-13	
Nome	Consultar Relatórios Salvos
Descrição	O sistema deve permitir o ator consultar os relatórios salvos por gerações agendadas, ou salvos pelo próprio ator.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-006
Pré-condições	O usuário deve estar autenticado no sistema. O usuário pode visualizar apenas os relatórios salvos em sua pasta de trabalho.
Pós-condições	Os relatórios salvos na pasta de trabalho do ator foram visualizados.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Acessa a tela de sua pasta de trabalho.	
	2. Lista todos os relatórios salvos na pasta de trabalho do ator.
3. Seleciona o relatório desejado, clicando sobre ele.	
	4. Exibe o relatório selecionado na tela.
Observações	
1. A visualização do relatório são sobre mesmos dados apresentado no momento em que o ator salvou o relatório.	

5.17.3 Projeto

5.17.3.1 Diagrama de Sequência

A figura 5.70 apresenta o diagrama sequência para atender ao requisito RF-006.

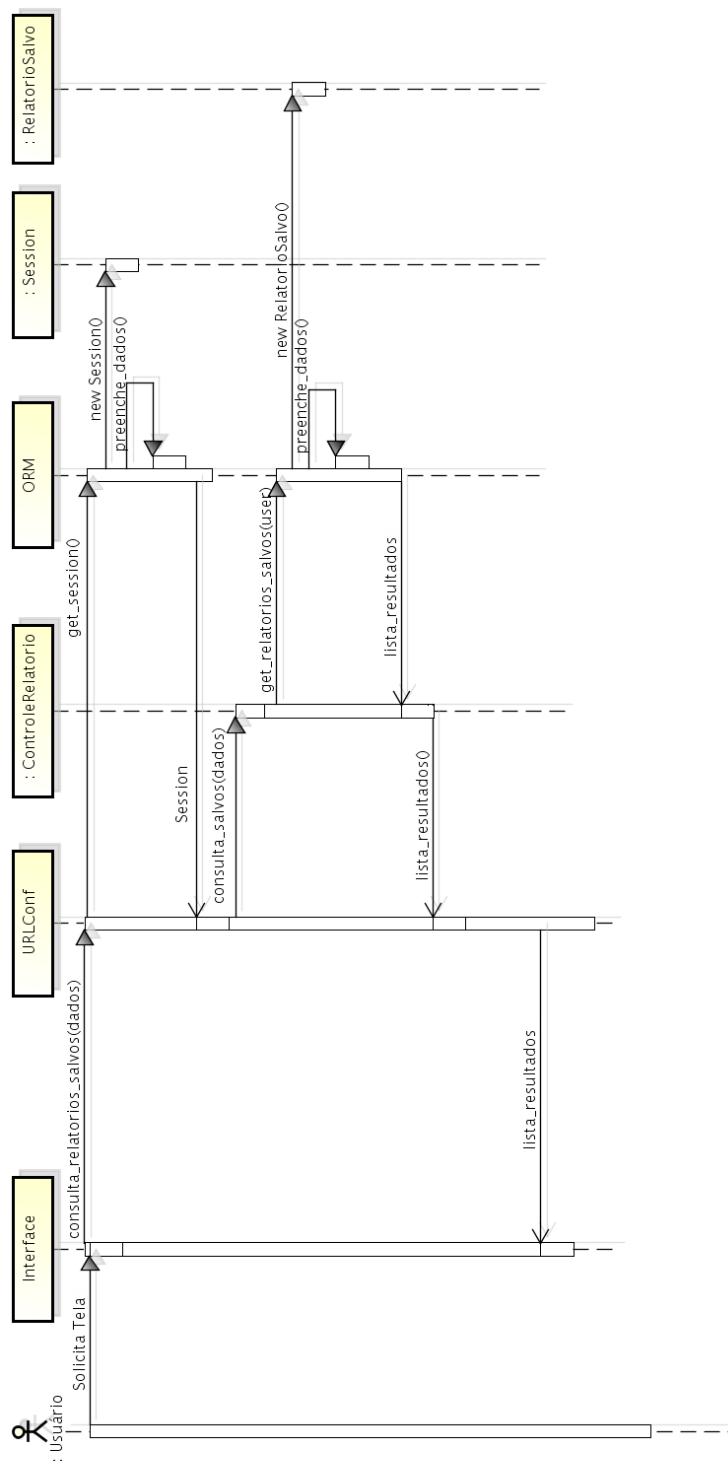


Figura 5.70: Diagrama de sequência - Consultar Relatórios Salvos.

5.17.3.2 Diagrama de Classes

A figura 5.71 apresenta o diagrama classes para atender parcilmente ao requisito RF-006.

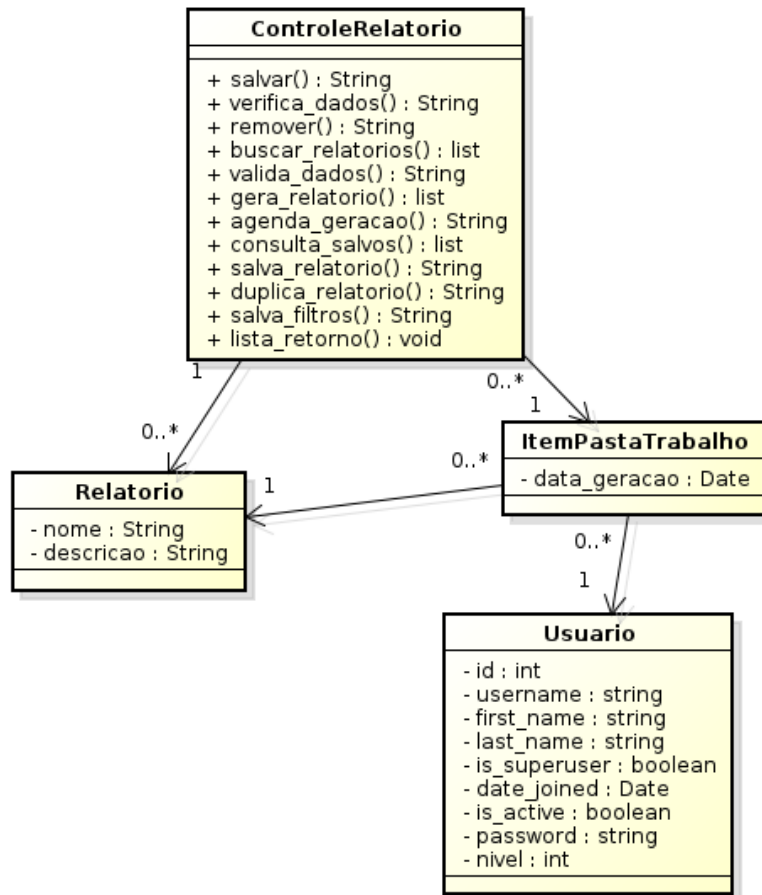


Figura 5.71: Diagrama de classes - Consultar Relatórios Salvos.

5.18 UC-14 - Salvar Relatório

O sistema deve permitir ao autor salvar resultados de relatórios, atendendo ao requisito RF-005 - Salvamento de Resultados de Relatórios.

5.18.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.18.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.72, visa atender ao requisito RF-005.



Figura 5.72: Tela 14 - Protótipo não funcional de interface - Salvar Relatório.

5.18.1.2 Caso de Uso

A figura 5.73 demonstra o caso de uso para a tela de salvar relatório.

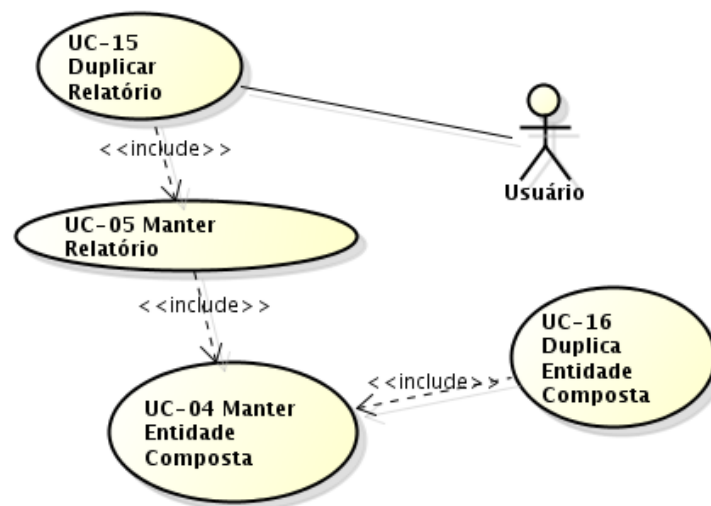


Figura 5.73: Diagrama de caso de uso - Salvar Relatório.

5.18.1.3 Modelo de Domínio

A figura 5.74 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-005.

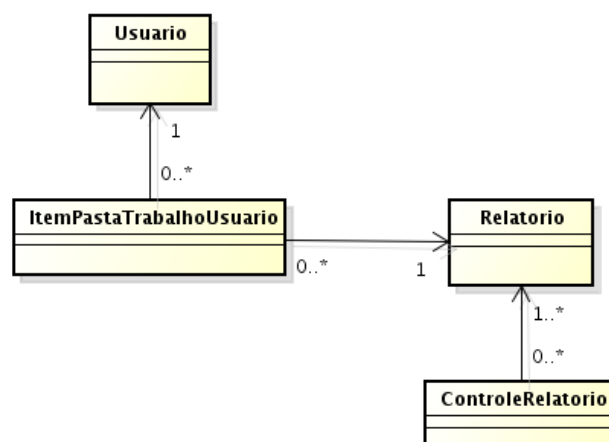


Figura 5.74: Diagrama de domínio - Salvar Relatório.

5.18.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.18.2.1 Caso de Uso Detalhado

A tabela 5.16 apresenta o caso de uso detalhado que visa atender ao requisito RF-005.

Tabela 5.16 - Caso de Uso Detalhado Salvar Relatório

UC-14	
Nome	Salvar Relatório
Descrição	O sistema deve permitir ao ator salvar o relatório que está sendo visualizado.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-005
Pré-condições	O usuário deve estar autenticado no sistema. O usuário estar com um relatório em formato HTML gerado na tela.
Pós-condições	Um relatório foi salvo na pasta de trabalho do ator.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. No menu superior da tela de visualização do relatório, clica em salvar na pasta de trabalho.	
	2. Salva o resultado do relatório na pasta de trabalho do ator.

5.18.3 Projeto

5.18.3.1 Diagrama de Sequência

A figura 5.75 apresenta o diagrama sequência para atender ao requisito RF-005.

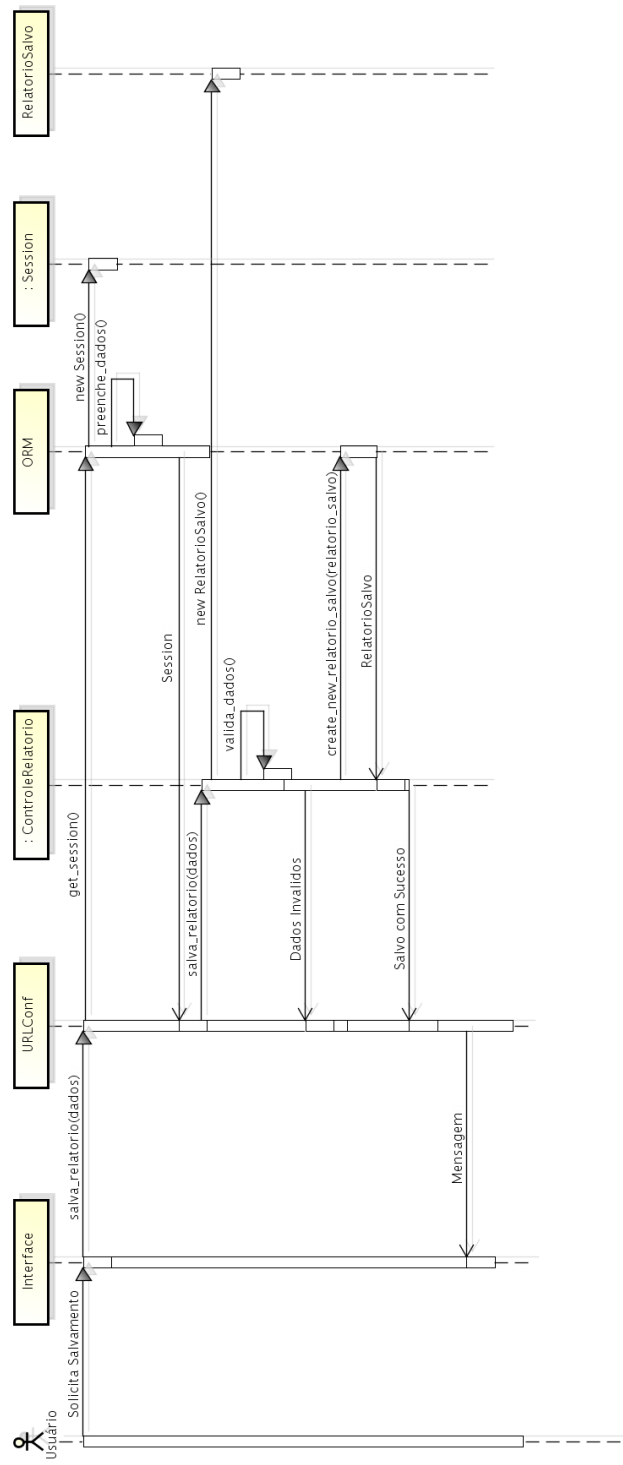


Figura 5.75: Diagrama de sequência - Salvar Relatório.

5.18.3.2 Diagrama de Classes

A figura 5.76 apresenta o diagrama classes para atender ao requisito RF-005.

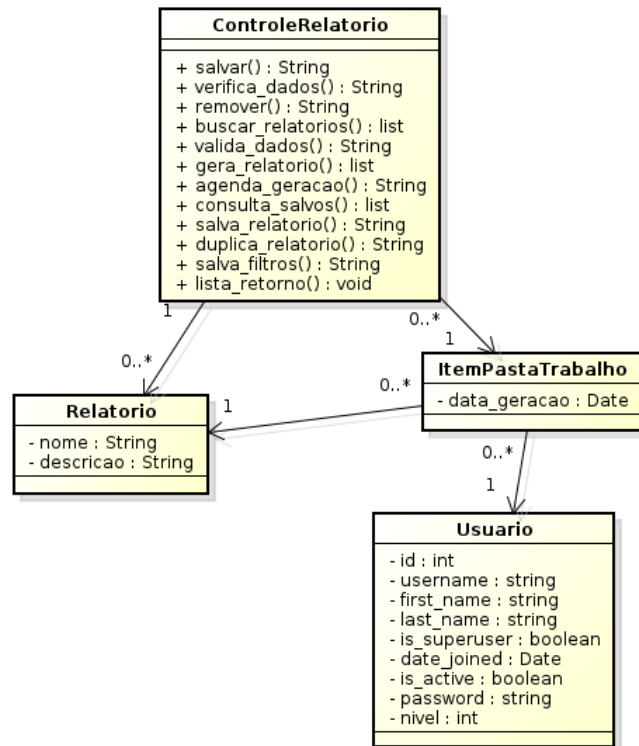


Figura 5.76: Diagrama de classes - Salvar Relatório.

5.19 UC-15 Duplicar Relatório

O sistema deve permitir a duplicação de relatórios, atendendo ao requisito RF-017 - Duplicar Relatório.

5.19.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.19.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.77, visa atender ao requisito RF-017.

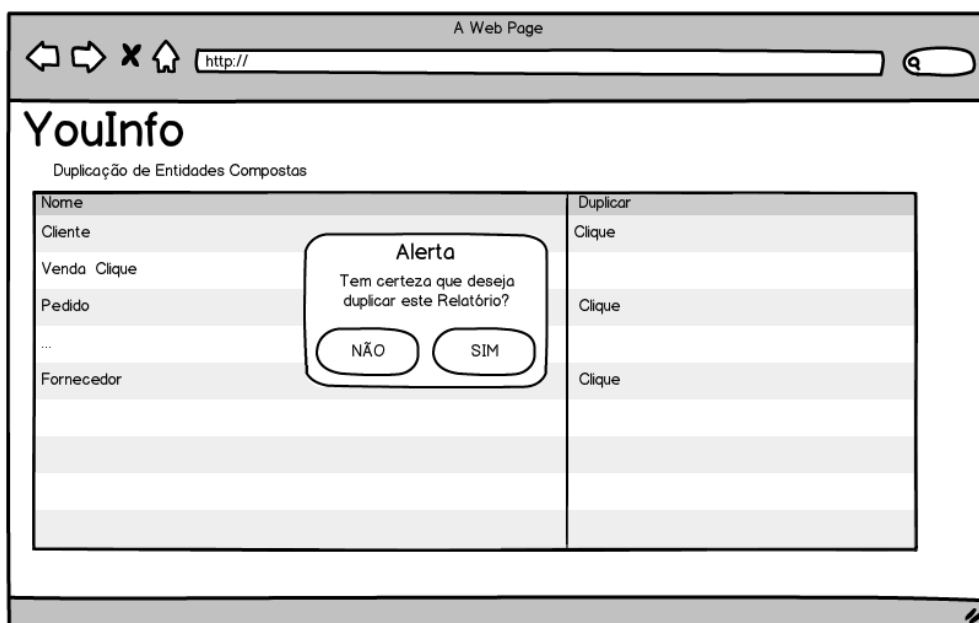


Figura 5.77: Tela 15 - Protótipo não funcional de interface - Duplicar Relatório.

5.19.1.2 Caso de Uso

A figura 5.78 demonstra o caso de uso para a tela de duplicação de relatórios.

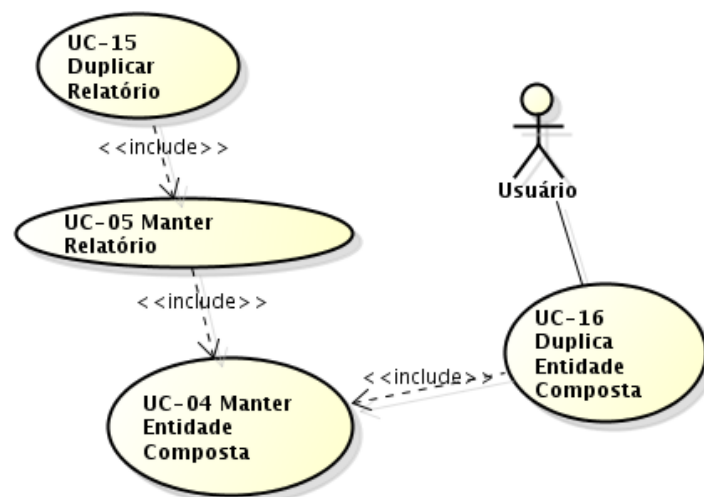


Figura 5.78: Diagrama de caso de uso - Duplicar Relatório.

5.19.1.3 Modelo de Domínio

A figura 5.79 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-017.

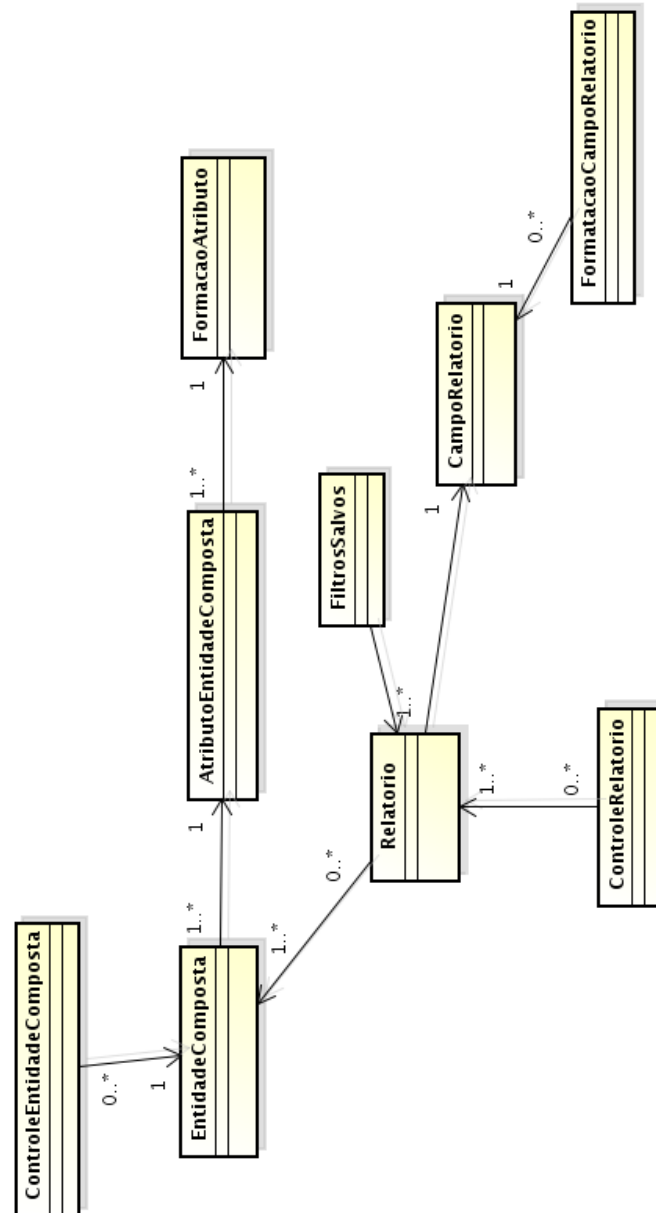


Figura 5.79: Diagrama de domínio - Duplicar Relatório.

5.19.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.19.2.1 Caso de Uso Detalhado

A tabela 5.17 apresenta o caso de uso detalhado que visa atender ao requisito RF-017.

Tabela 5.17 - Caso de Uso Detalhado Duplicar Relatório

UC-15	
Nome	Duplicar Relatório
Descrição	Duplica um relatório, criando uma cópia idêntica.
Atores	Usuário
Prioridade	Desejável
Requisitos Funcionais Associados	
Pré-condições	O usuário deve estar autenticado no sistema. O usuário deve ter permissão para geração do relatório a ser duplicado.
Pós-condições	O relatório foi duplicado, sendo criada uma cópia idêntica do mesmo.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Acessa a tela de relatórios.	
	2. Lista todos os relatórios que o ator possui permissão de geração.
3. Seleciona o relatório desejado, clicando no botão duplicar, ao lado do relatório.	
	4. Cria uma cópia idêntica do relatório, com o nome de "Cópia de <RELATORIO>", e a descrição "Cópia do relatório <RELATORIO>".
Observações	
1. Onde se lê <RELATORIO> deve ser apresentado o nome do relatório corrente.	

5.19.3 Projeto

5.19.3.1 Diagrama de Sequência

A figura 5.80 apresenta o diagrama sequência para atender ao requisito RF-017.

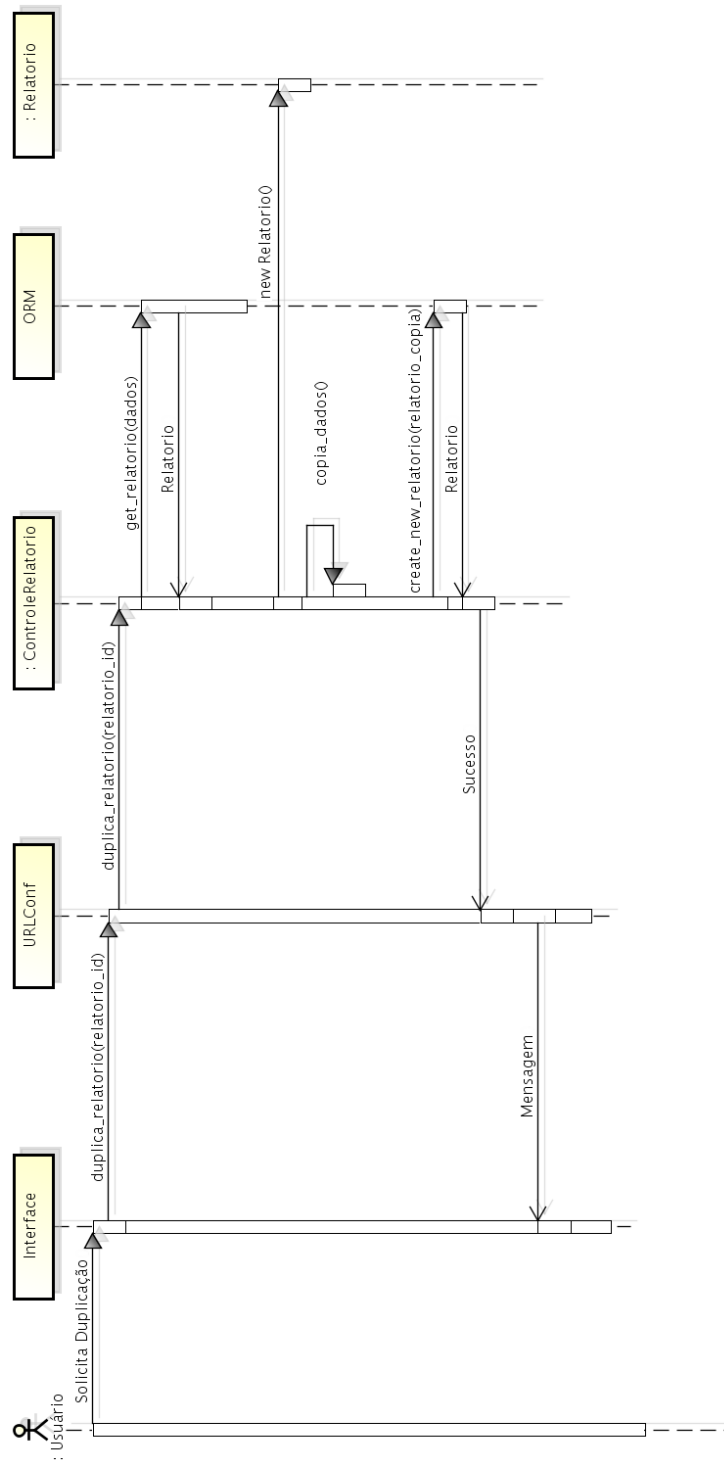


Figura 5.80: Diagrama de sequência - Duplicar Relatório.

5.19.3.2 Diagrama de Classes

A figura 5.81 apresenta o diagrama classes para atender ao requisito RF-017.

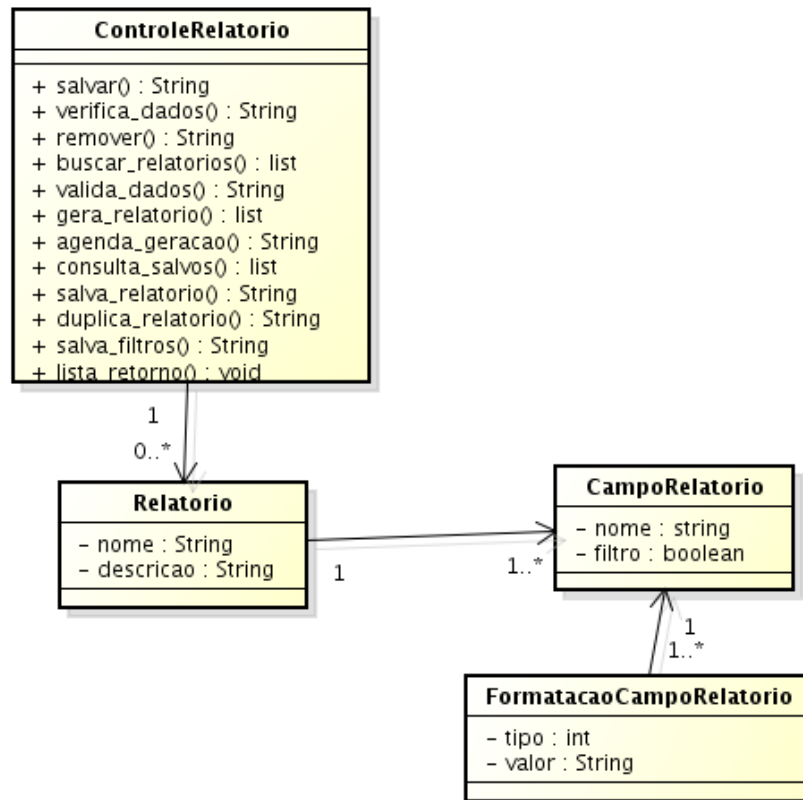


Figura 5.81: Diagrama de classes - Duplicar Relatório.

5.20 UC-16 Duplicar Entidade Composta

O sistema permitir a duplicação de entidades compostas, atendendo ao requisito RF-016 - Duplicar Entidade Composta.

5.20.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.20.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.82, visa atender ao requisito RF-016.

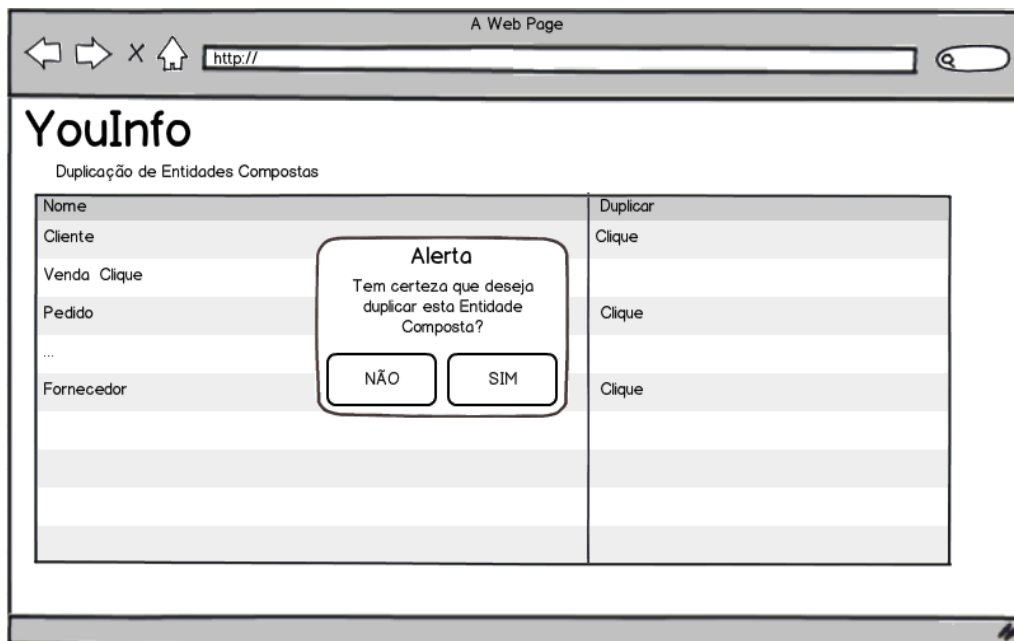


Figura 5.82: Tela 17 - Protótipo não funcional de interface - Duplicar Entidade Composta.

5.20.1.2 Caso de Uso

A figura 5.83 demonstra o caso de uso para a tela de duplicação de entidades compostas.



Figura 5.83: Diagrama de caso de uso - Duplicar Entidade Composta.

5.20.1.3 Modelo de Domínio

A figura 5.84 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-016.

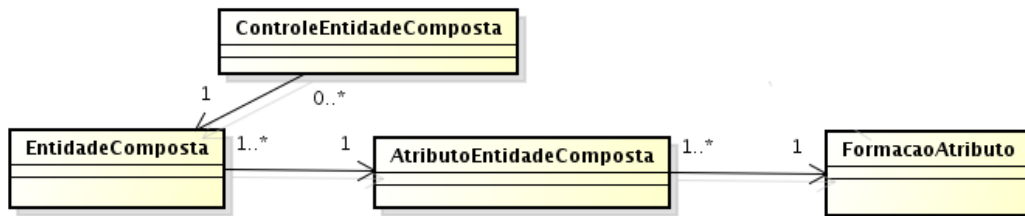


Figura 5.84: Diagrama de domínio - Duplicar Entidade Composta.

5.20.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.20.2.1 Caso de Uso Detalhado

A tabela 5.18 apresenta o caso de uso detalhado que visa atender ao requisito RF-016.

Tabela 5.18 - Caso de Uso Detalhado - Duplicar Entidade Composta

UC-16	
Nome	Duplicar Entidade Composta
Descrição	Duplica uma entidade composta, criando uma cópia idêntica.
Atores	Usuário
Prioridade	Desejável
Requisitos Funcionais Associados	
Pré-condições	O usuário deve estar autenticado no sistema.
Pós-condições	A entidade composta foi duplicada, sendo criada uma cópia idêntica da mesma.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Acessa a tela de entidades compostas.	
	2. Lista todas as entidades compostas cadastradas no sistema.
3. Seleciona a entidade composta desejada, clicando no botão duplicar, ao lado da entidade composta.	
	4. Cria uma cópia idêntica da entidade composta, com o nome de "Cópia de <ENTIDADE>", e a descrição "Cópia da entidade composta <ENTIDADE>"
Observações	
1. Onde se lê <ENTIDADE> deve ser apresentado o nome da entidade composta corrente.	

5.20.3 Projeto

5.20.3.1 Diagrama de Sequência

A figura 5.85 apresenta o diagrama sequência para atender ao requisito RF-016.

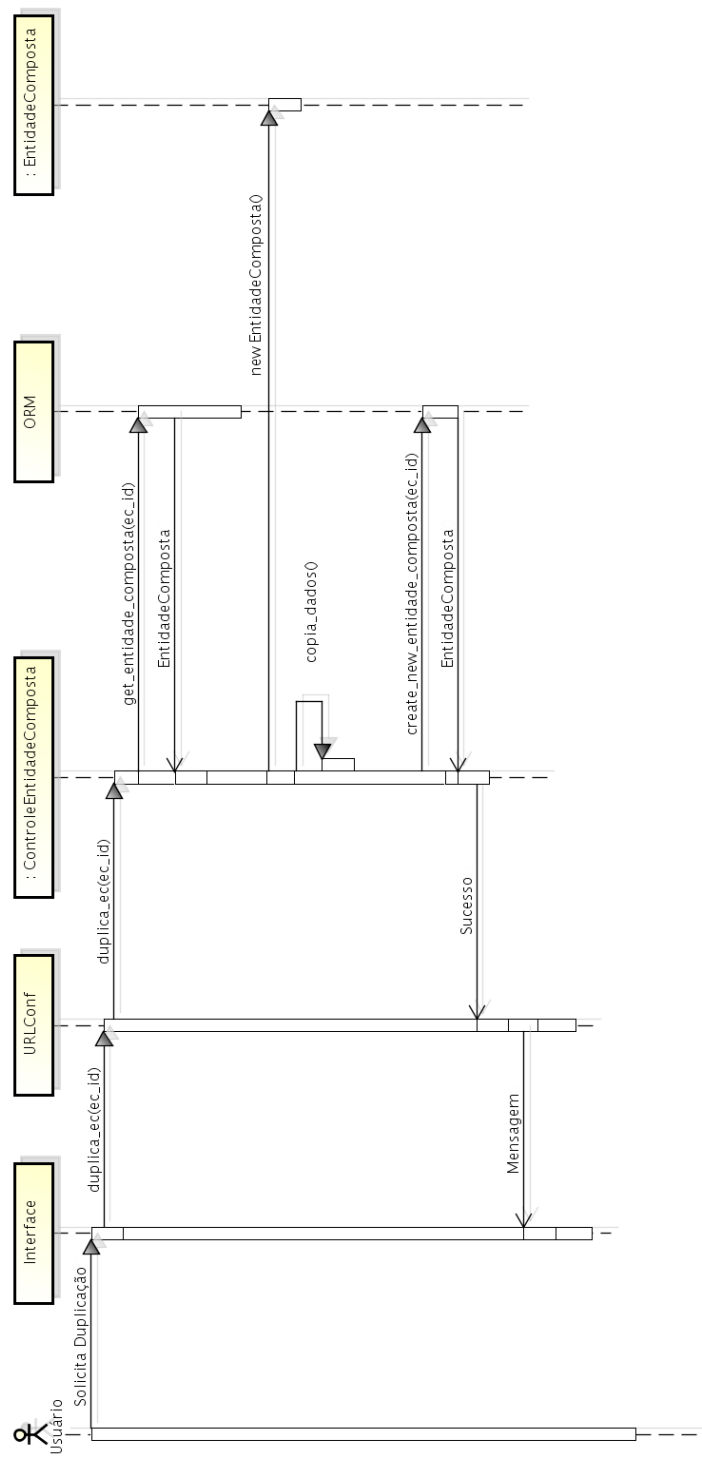


Figura 5.85: Diagrama de sequência - Duplicar Entidade Composta.

5.20.3.2 Diagrama de Classes

A figura 5.86 apresenta o diagrama classes para atender ao requisito RF-016.

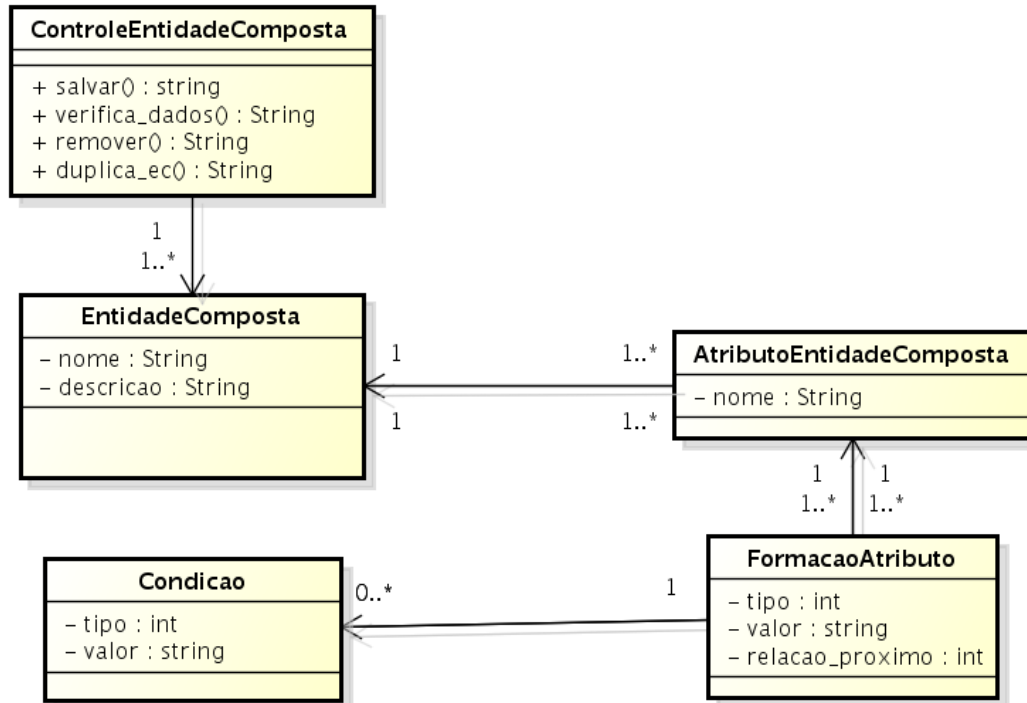


Figura 5.86: Diagrama de classes - Duplicar Entidade Composta.

5.21 UC-17 Salvar Filtros de Relatório

O sistema deve permitir o salvamento de filtros de relatórios, atendendo ao requisito RF-007 - Salvar Filtros de Relatório.

5.21.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.21.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.87, visa atender ao requisito RF-007.

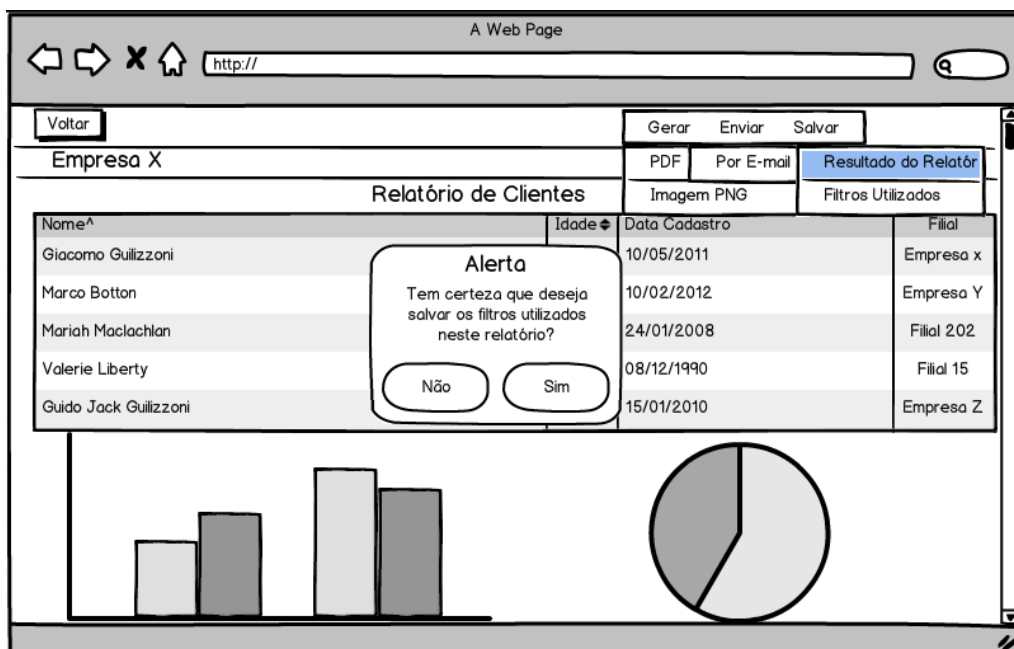


Figura 5.87: Tela 18 - Protótipo não funcional de interface - Salvar Filtros de Relatório.

5.21.1.2 Caso de Uso

A figura 5.88 demonstra o caso de uso para a tela de salvar filtros de relatório.



Figura 5.88: Diagrama de caso de uso - Salvar Filtros de Relatório.

5.21.1.3 Modelo de Domínio

A figura 5.89 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-007.



Figura 5.89: Diagrama de domínio - Salvar Filtros de Relatório.

5.21.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.21.2.1 Caso de Uso Detalhado

A tabela 5.19 apresenta o caso de uso detalhado que visa atender ao requisito RF-007.

Tabela 5.19 - Caso de Uso Detalhado Salvar Filtros de Relatório

UC-17	
Nome	Salvar Filtros de Relatório
Descrição	O sistema deve permitir ao autor salvar os filtros utilizados em um relatório gerado.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-007
Pré-condições	O usuário deve estar autenticado no sistema. O usuário estar com um relatório em formato HTML gerado na tela.
Pós-condições	Os filtros e opções utilizados na geração do relatório foram salvos.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. No menu superior da tela de visualização do relatório, clica em salvar filtros utilizados	
	2. Mostra uma tela para preenchimento de nome e descrição dos filtros, visando uma melhor identificação futura dos mesmos.
3. Preenche o campo de nome dos filtros, obrigatório.	
	4. Verifica se o campo está preenchido. 5. Verifica se ainda não existe nenhum filtro salvo para aquele relatório com o mesmo nome. 6. Salva os filtros e opções utilizados.
Cenário Alternativo 1	
	5.1 Verificou que um ou mais campos não estão preenchidos. 5.2 Retorna para a tela, com os dados preenchidos pelo autor, informando junto ao campo a obrigatoriedade do mesmo.
Cenário Alternativo 2	
	6.1 Verificou já existem filtros para o mesmo relatório com o mesmo nome. 6.2 Retorna para a tela, com os dados preenchidos pelo autor, informando junto ao campo que já existe filtros salvos para aquele relatório, com o mesmo nome.

5.21.3 Projeto

5.21.3.1 Diagrama de Sequência

A figura 5.90 apresenta o diagrama sequência para atender ao requisito RF-007.

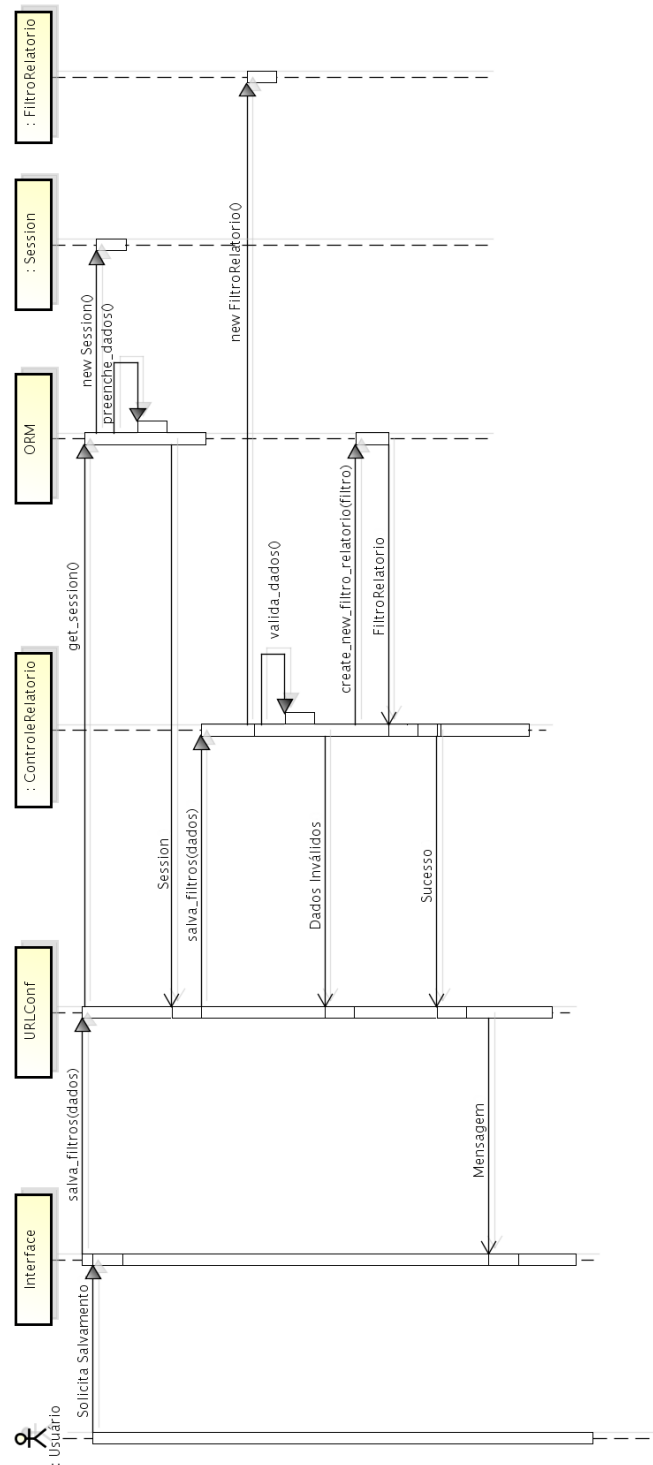


Figura 5.90: Diagrama de sequência - Salvar Filtros de Relatório.

5.21.3.2 Diagrama de Classes

A figura 5.91 apresenta o diagrama classes para atender ao requisito RF-007.

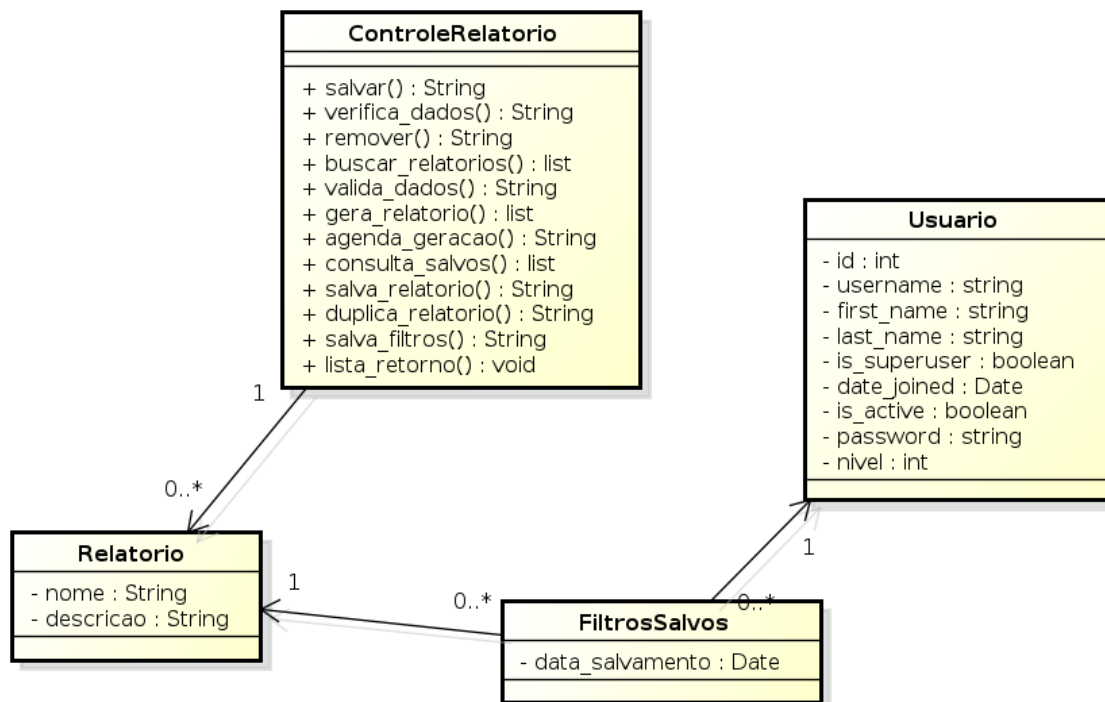


Figura 5.91: Diagrama de classes - Salvar Filtros de Relatório.

5.22 UC-18 Mostrar Lista de Valores de Campo de Filtro

O sistema deve exibir uma lista de valores possíveis para os campos de filtros de relatórios que sejam do formato texto, atendendo ao requisito RF-018.

5.22.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.22.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.92, visa atender ao requisito RF-018.

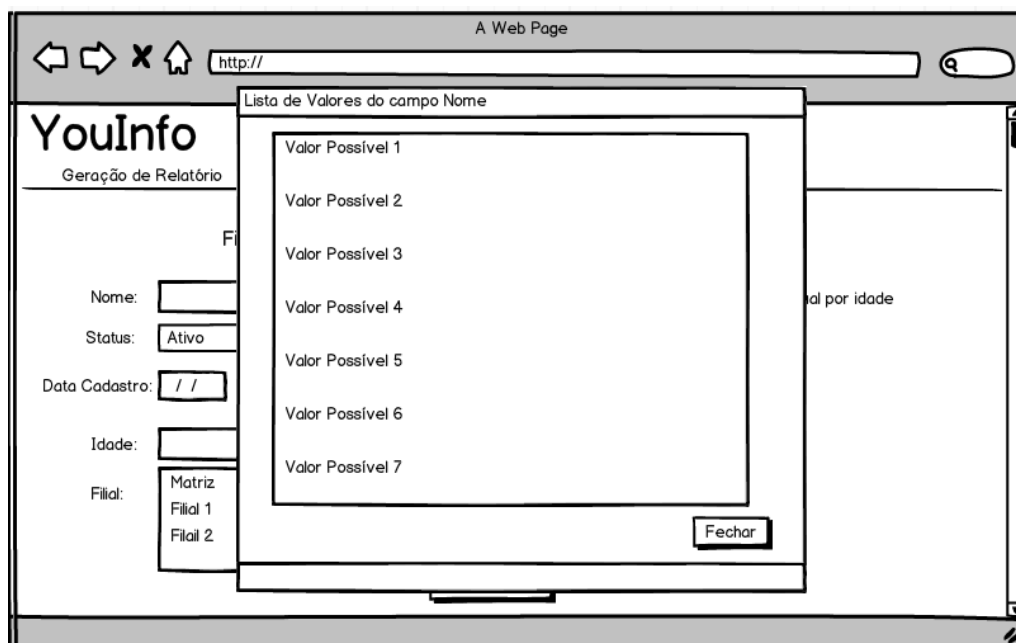


Figura 5.92: Tela 18 - Protótipo não funcional de interface - Mostrar Lista de Valores de Campo de Filtro.

5.22.1.2 Caso de Uso

A figura 5.93 demonstra o caso de uso para a tela de lista de valores de campo de filtro

5.22.1.3 Modelo de Domínio

A figura 5.94 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-018.

5.22.2 Análise

A seguir serão vistos os artefatos da fase de análise.

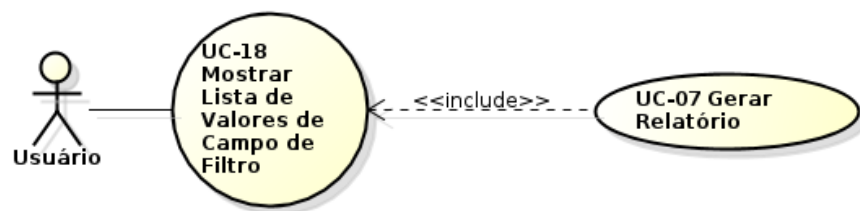


Figura 5.93: Diagrama de caso de uso - Mostrar Lista de Valores de Campo de Filtro.

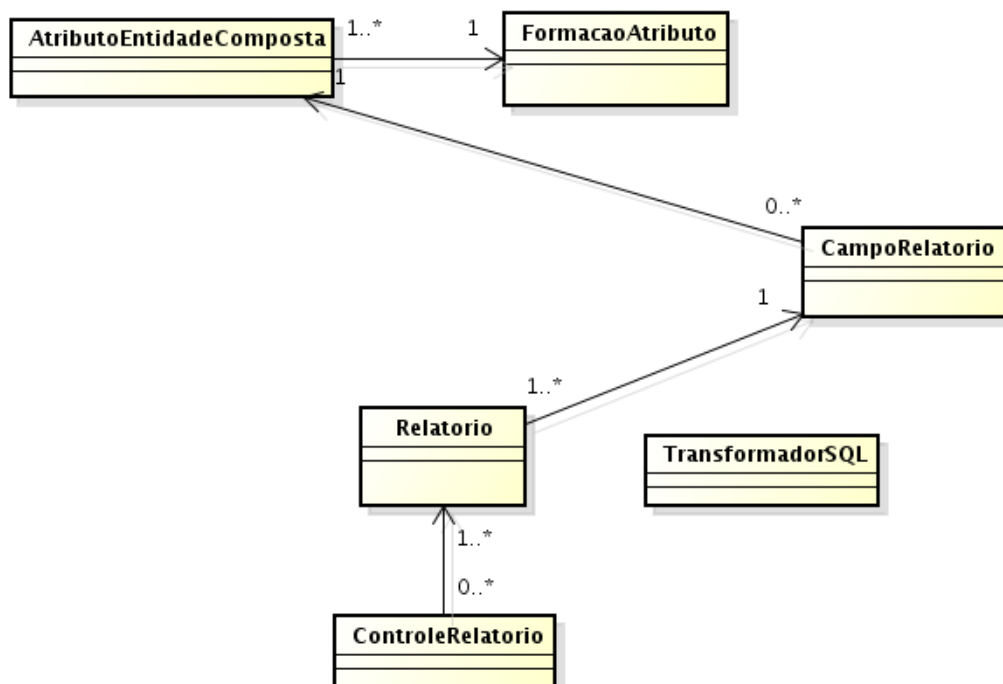


Figura 5.94: Diagrama de domínio - Mostrar Lista de Valores de Campo de Filtro.

5.22.2.1 Caso de Uso Detalhado

A tabela 5.20 apresenta o caso de uso detalhado que visa atender ao requisito RF-018.

Tabela 5.20 - Caso de Uso Detalhado Mostrar Lista de Valores de Campo de Filtro

UC-18	
Nome	Mostrar Lista de Valores de Campo de Filtro
Descrição	O sistema deve exibir uma lista de valores para os campos de filtro de relatório que sejam do formato texto.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-018
Pré-condições	O usuário deve estar autenticado no sistema. O usuário deve estar na tela de filtros de dados do relatório.
Pós-condições	A lista de valores foi exibida e o campo de filtro correspondente foi populado com o valor selecionado pelo ator.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Clica sobre o botão da lista de valores, localizado ao lado do campo.	
	2. Mostra uma janela modal com os valores disponíveis para aquele campo.
3. Seleciona o valor desejado na lista de valores.	
	4. Fecha a janela modal com a lista de valores e popula o campo de filtro correspondente com o valor selecionado pelo usuário.
Observações	
1. Somente os campos do tipo texto, ou seja, campos que não possuam seletor de data, ou outros componentes associados, devem possuir o botão de lista de valores(LOV).	

5.22.3 Projeto

5.22.3.1 Diagrama de Sequência

A figura 5.95 apresenta o diagrama sequência para atender ao requisito RF-018.

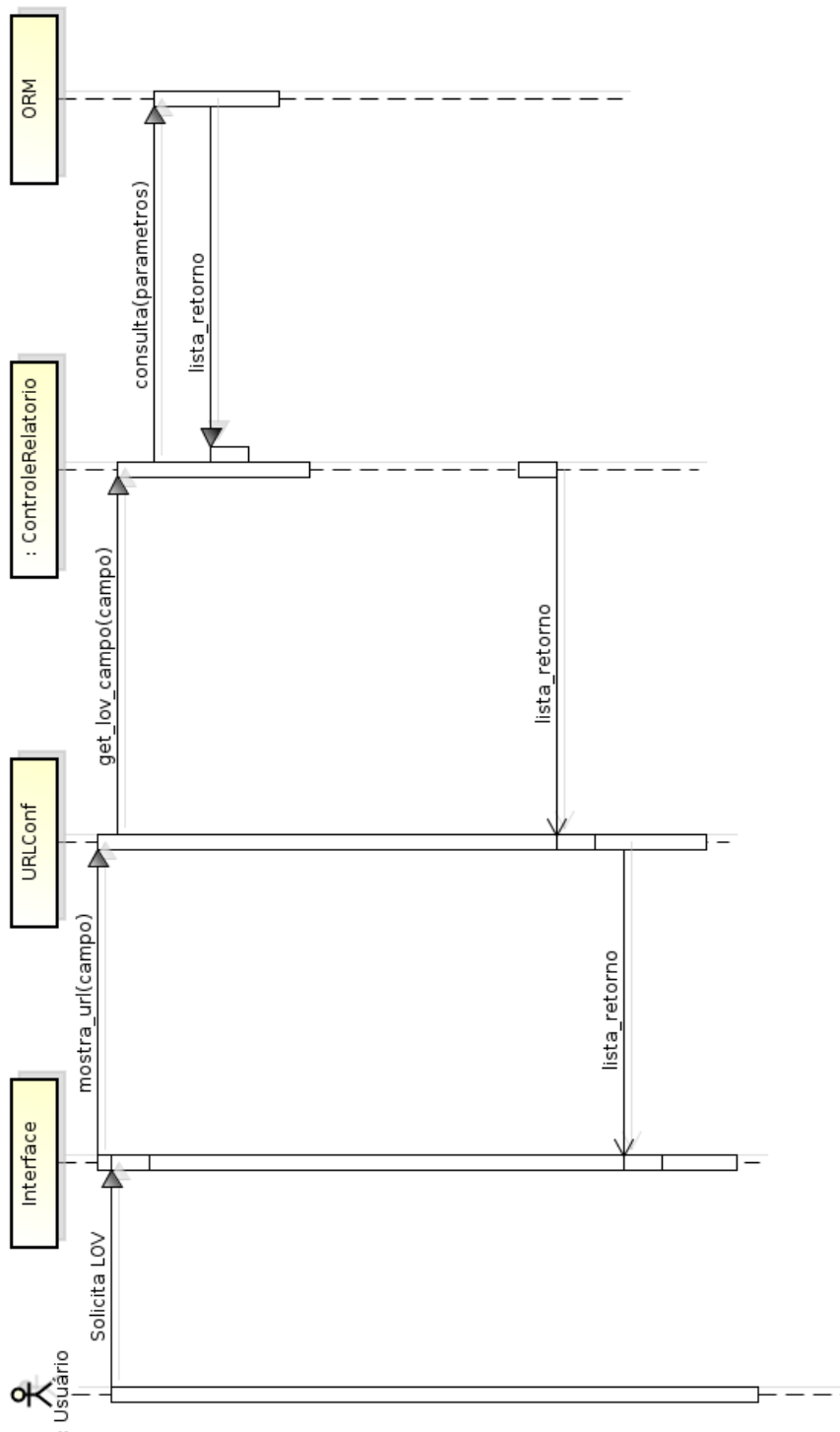


Figura 5.95: Diagrama de sequência - Mostrar Lista de Valores de Campo de Filtro.

5.22.3.2 Diagrama de Classes

A figura 5.96 apresenta o diagrama classes para atender ao requisito RF-018.

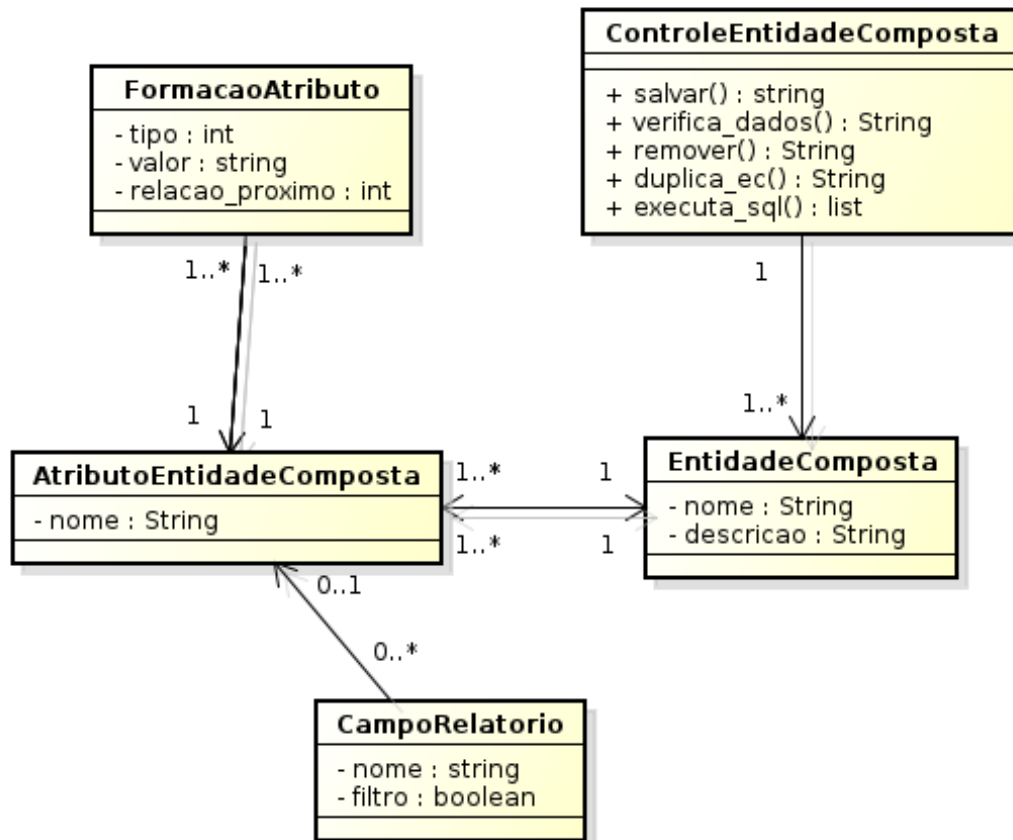


Figura 5.96: Diagrama de classes - Mostrar Lista de Valores de Campo de Filtro.

5.23 UC-19 Listar Eventos de Gerações de Relatórios no Dashboard

O sistema deve listar as notificações de gerações automáticas de relatórios relativas ao usuário, atendendo ao requisito RF-019 - Listar Eventos de Gerações de Relatórios no Dashboard.

5.23.1 Análise de Requisitos

A seguir serão vistos os artefatos referentes a análise de requisitos.

5.23.1.1 Protótipo Não Funcional de Interface Humano-Computador

O protótipo de interface humano-computador, representado pela figura 5.97, visa atender ao requisito RF-019.

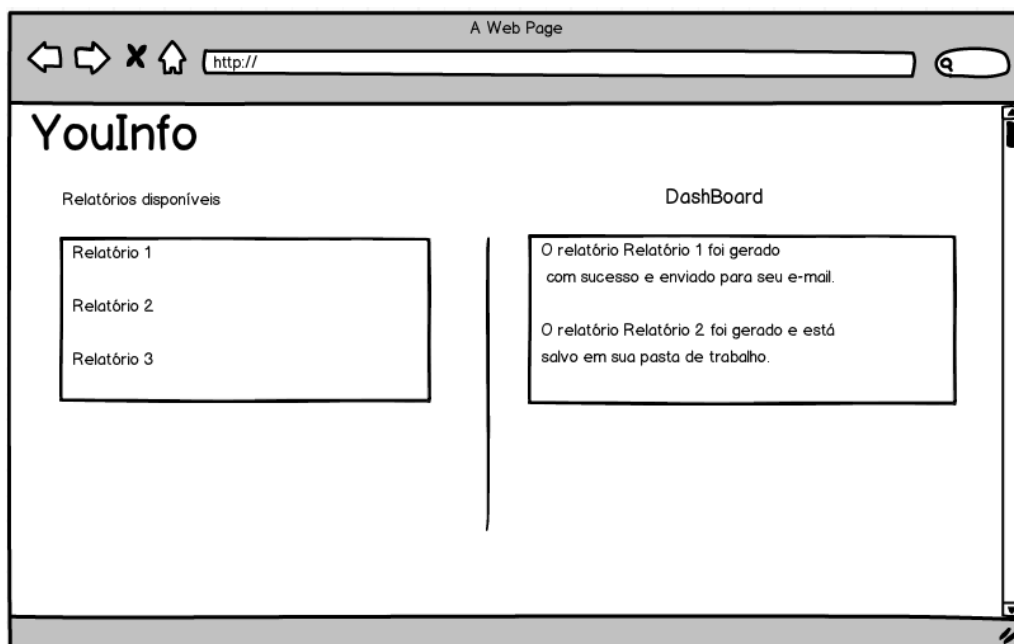


Figura 5.97: Tela 19 - Protótipo não funcional de interface - Eventos de Gerações de Relatórios no Dashboard.

5.23.1.2 Caso de Uso

A figura 5.98 demonstra o caso de uso para a tela de salvar filtros de relatório.

5.23.1.3 Modelo de Domínio

A figura 5.99 apresenta o modelo de domínio responsável por atender ao requisito funcional RF-007.

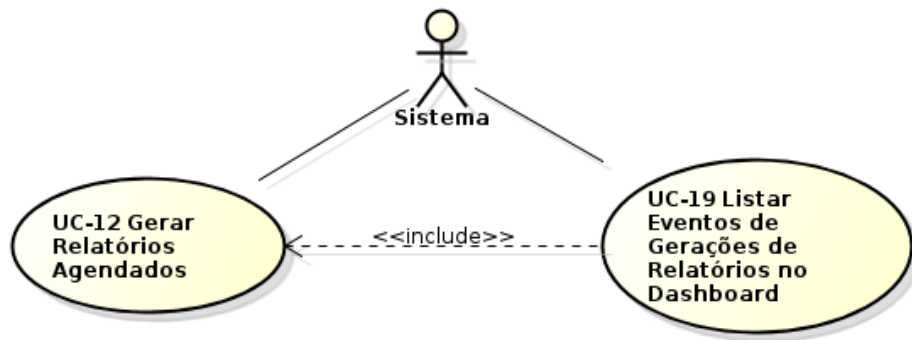


Figura 5.98: Diagrama de caso de uso - Eventos de Gerações de Relatórios no Dashboard.

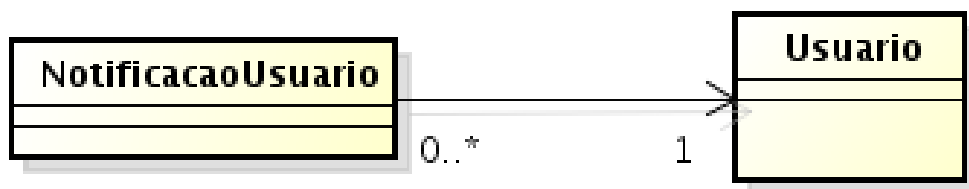


Figura 5.99: Diagrama de domínio - Eventos de Gerações de Relatórios no Dashboard.

5.23.2 Análise

A seguir serão vistos os artefatos da fase de análise.

5.23.2.1 Caso de Uso Detalhado

A tabela 5.21 apresenta o caso de uso detalhado que visa atender ao requisito RF-007.

Tabela 5.21 - Caso de Uso Listar Eventos de Gerações de Relatórios no Dashboard

UC-19	
Nome	Listar Eventos de Gerações de Relatórios no Dashboard
Descrição	O sistema deve listar as notificações para o usuário na tela inicial.
Atores	Usuário
Prioridade	Essencial
Requisitos Funcionais Associados	RF-019
Pré-condições	O usuário deve estar autenticado no sistema.
Pós-condições	O sistema exibiu na tela inicial as notificações sobre gerações de relatórios automáticas pertencentes ao usuário.
Cenário de Sucesso Principal	
Ação do Usuário(ou intenção)	Responsabilidade do Sistema
1. Acessa a tela inicial do sistema.	
	2. Busca e lista as notificações pertencentes ao ator, relativas às gerações automáticas de relatórios agendadas pelo mesmo.

5.23.3 Projeto

5.23.3.1 Diagrama de Sequência

A figura 5.100 apresenta o diagrama sequência para atender ao requisito RF-007.

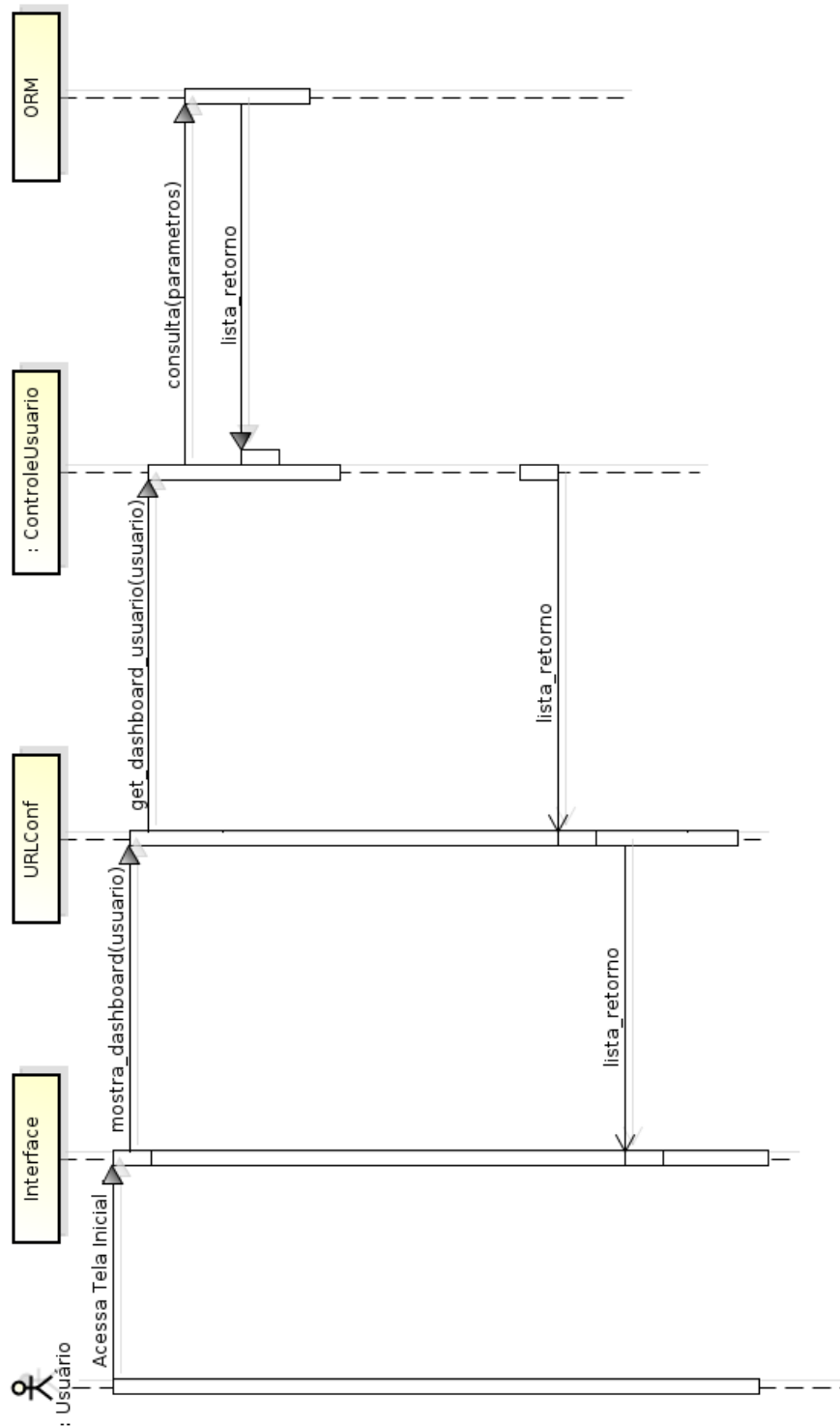


Figura 5.100: Diagrama de sequência - Eventos de Gerações de Relatórios no Dashboard.

5.23.3.2 Diagrama de Classes

A figura 5.101 apresenta o diagrama classes para atender ao requisito RF-019.

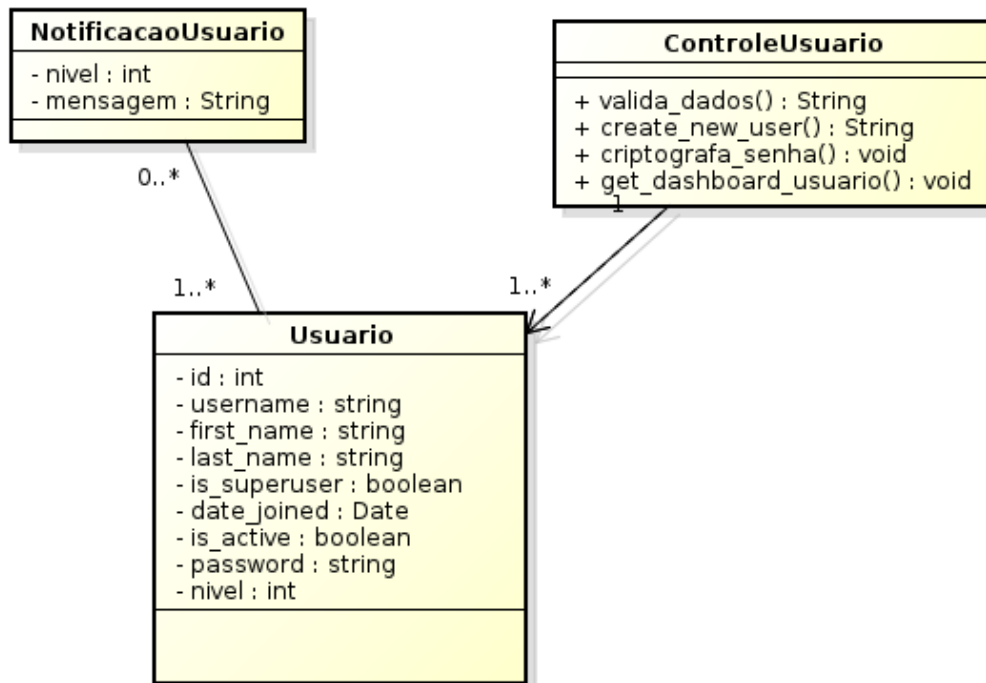


Figura 5.101: Diagrama de classes - Eventos de Gerações de Relatórios no Dashboard.

6 MATRIZES DE RASTREABILIDADE

Segue as matrizes que tem como função ligar os casos de uso com os requisitos, telas e domínios.

Segundo (BELO HORIZONTE , PRODABEL S/A), A rastreabilidade pode ajudar a:

- garantir que os requisitos especificados são associados as necessidades dos clientes;
- garantir que todo produto de trabalho está associado aos requisitos identificados.

A tabela 6.22, tem como função ligar os casos de uso com os domínios. A tabela foi dividida em duas partes, para melhor adaptação com o layout da página.

Tabela 6.22 - Matriz de Rastreabilidade Mod. Domínio x UC

Requisitos	UC-01	UC-02	UC-03	UC-04	UC-05	UC-06	UC-07	UC-08	UC-09
Usuario	X	X	X				X		
EntidadeComposta				X			X		
AtributoEntidadeComposta				X			X		
FormacaoAtributo				X			X		
Agendamento									
FiltrosSalvos									
FiltroRelatorio					X		X		
RelatorioSalvo									
PastaTrabalhoUsuario									
PermissoesUsuarioRelatorio					X	X	X		
Relatorio					X	X	X		
CampoRelatorio					X		X		
AutenticaUser	X								
ControleUser		X	X						
ControleEntidadeComposta				X	X				
ControleRelatorio				X	X	X			
GerenciadorEmail									
GeradorPDF								X	
GeradorImagem									X
TransformadorSQL							X		
NotificacaoUsuario							X		

7 IMPLEMENTAÇÃO

Este trabalho foi desenvolvido de forma a facilitar mudanças futuras e utilizando-se uma ORM para mapeamento de tabelas, o que o torna independente do banco de dados a ser utilizado para armazenamento dos dados da ferramenta desenvolvida.

7.1 Módulos

Os módulos neste sistema foram criados para divisão e organização dos arquivos-fonte presentes, afim de dividi-los por responsabilidade e fluxo de uso da ferramenta. A figura 7.1 representa o diagrama de pacotes do sistema desenvolvido.

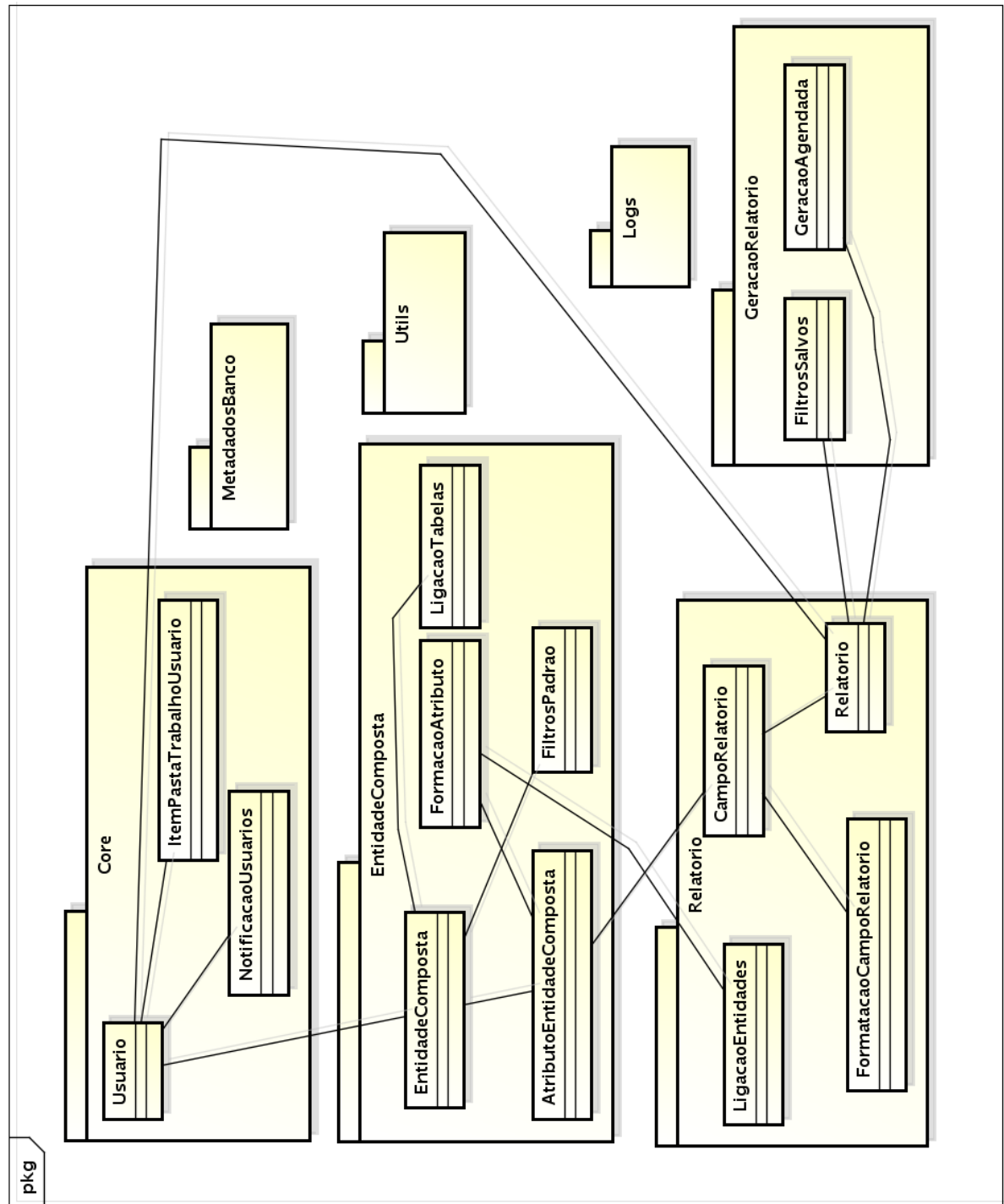


Figura 7.1: Diagrama de Pacotes do Sistema Desenvolvido.

A seguir são listados e descritos os módulos presentes no sistema.

7.1.1 Core

O módulo do core tem a função de controle principal da estrutura do sistema. Entre suas responsabilidades estão:

- Controle dos modelos Usuario e NotificacaoUsuarios;

- Autenticação dos usuários;
- Controle de acesso às páginas que só podem ser acessadas por usuários autenticados;
- Controle das permissões pelos níveis dos usuários;
- Controle da transição das páginas(mapeamento de urls do sistema);
- Controle das páginas de erro do sistema, e apresentação de mensagens aos usuários;
- Notificações aos usuários de acordo com eventos que acontecem no sistema(dashboard, na tela inicial);

Abaixo são listados os modelos pertencentes ao módulo Core:

- Usuario - A classe que representa um usuário com acesso ao sistema. Esta classe também armazena as informações cadastradas na manutenção de usuários.
- ItemPastaTrabalhoUsuario - Classe que representa um item da pasta de trabalho do usuário. Cada usuário pode ter vários registros desta classe vinculados a ele. Entre os itens de pasta de trabalho pode-se citar o resultado de um relatório salvo para posterior visualização.
- NotificacaoUsuarios - Classe que representa uma notificação para um ou mais usuários. Estas notificações são exibidas na tela inicial, na parte do dashboard. Esta classe de modelo se faz necessária para armazenar estas informações.

7.1.2 MetadadosBanco

Este módulo tem como responsabilidade a conexão com o banco de dados a serem criadas as entidades compostas e gerados os relatórios. Ao iniciar-se o sistema no servidor, ele faz tentativas de conexão ao banco de dados configurado no arquivo de configuração, e em caso de sucesso, ele busca todas as meta-informações do banco de dados necessárias para uso do sistema. Meta-informações para este sistema podem ser definidas como a estrutura das tabelas, views, colunas, etc.. A partir destas informações buscadas e gravadas em um objeto salvo em uma variável global, todo o sistema busca as informações necessárias conforme a necessidade. A fim de evitar a minimização de performance devido à situação de concorrência, cada processo que o servidor de aplicação(neste caso, o Apache) estiver rodando, terá uma instância de meta-informações do banco de dados. A performance que é ganha com esta técnica é vantajosa em relação ao consumo maior de memória causado por cada processo ter uma instância separada para guardar estas informações.

7.1.3 Logs

O módulo de logs possui funções que gravam logs internos do sistema, conforme configurado em arquivo de configurações. Estas informações gravadas pelo LOG são de importância apenas para os administradores do sistema, em caso de problemas, portanto não é visível aos usuários do sistema.

7.1.4 EntidadeComposta

O módulo EntidadeComposta existe com o objetivo de centralizar todo o controle de entidades compostas em apenas um módulo. Entre estes controles, estão:

- Controle dos modelos de EntidadeComposta, AtributoEntidadeComposta, FormacaoAtributo, LigacaoTabelas e FiltrosPadrao;
- Páginas de listagem de entidades compostas e de manutenção de entidades compostas;
- Validação correta de todos os formulários de controle de entidades compostas, antes da gravação no banco de dados;
- Gravação das entidades compostas no banco de dados da aplicação;
- Geração de toda consulta SQL dinâmica necessária a partir das entidades compostas. Entre estas gerações, pode-se citar a geração das consultas para serem exibidas nos relatórios e a geração das consultas por atributo, que servem como uma lista de valores possíveis para o campo na tela de filtragem de dados do relatório.

Abaixo são apresentados os modelos de dados pertencentes ao módulo EntidadeComposta:

- EntidadeComposta - Classe para representar uma entidade composta, que pode ser criada por usuário de nível administrador. Com esta classe, representa-se a parte mais alta da estrutura de uma entidade composta, onde as outras classes de modelo encontram-se como atributos desta classe, isto no banco de dados reflete-se em no mínimo uma tabela para cada classe de modelo.
- AtributoEntidadeComposta - Classe para representar um atributo de uma entidade composta. Armazena informações exclusivas de um atributo.
- FormacaoAtributo - Classe para representar uma parte da formação de um atributo de uma entidade composta. Com uma ou mais instâncias desta classe, obtém-se uma formação completa de um atributo de uma entidade composta, que necessita ao menos de um atributo para ser criada.

- **FiltrosPadrao** - Esta classe representa um filtro padrão aplicado sobre a entidade composta. Estes filtros padrão são restrições iniciais que ficam vinculadas as entidades compostas, e são considerados ao executar a busca de dados necessária para a entidade composta.
- **LigacaoTabelas** - Esta classe representa uma ligação entre duas tabelas, de forma análoga aos comandos *INNER JOIN* da linguagem de banco de dados chamada SQL.

7.1.5 Relatorio

Assim como o módulo EntidadeComposta, o módulo Relatorio tem como principal função centralizar o controle dos relatórios criados no sistema. Entre suas responsabilidades estão:

- Controle dos modelos de Relatorio, CampoRelatorio, FormatacaoCampoRelatorio e LigacaoEntidades;
- Páginas de listagem e manutenção de relatórios;
- Validação correta de todos os formulários de controle de relatório, antes da gravação no banco de dados;
- Gravação dos relatórios no banco de dados da aplicação;

Abaixo são apresentados os modelos de dados referentes ao módulo Relatorio:

- **Relatorio** - Classe que representa um relatório, criado a partir de entidades compostas existentes. Esta classe de modelo é a base da estrutura de formação de um relatório dentro do sistema, ficando todas as outras classes com o mesmo propósito como atributos desta classe. Os relatórios criados no sistema são armazenados na tabela de banco de dados representado por esta classe, podendo ser acessado para edição ou geração a qualquer momento.
- **CampoRelatorio** - Classe para um campo de um relatório. Um campo de relatório fica vinculado a um atributo de uma entidade composta, sendo possível selecionar se o campo pode ou não ser utilizado como filtro de dados para o relatório.
- **LigacaoEntidades** - Classe para representar a ligação de dados entre duas entidades compostas pelos seus atributos. Uma instância desta classe deve ser criada para cada valor que deseja-se utilizar de comparação entre dos atributos de entidades compostas, formando uma restrição no momento da geração do relatório.

- `FormatacaoCampoRelatorio` - Esta classe representa uma formatação de um campo de relatório. Esta formatação define como o campo vai ser apresentado no resultado do relatório, com opções tais como: cor da fonte, cor de fundo, negrito, itálico, etc...

7.1.6 `GeracaoRelatorio`

Todo assunto relativo à geração de dados a partir das entidades compostas e relatórios criados, são de responsabilidade deste módulo. Entre suas responsabilidades, estão:

- Controle dos modelos de dados `FiltrosSalvos`, `ItemPastaTrabalhoUsuario` e `GeracaoAgendada`;
- Geração dos campos de filtros dinâmicos para as telas de filtragem de dados, antes da geração efetiva do relatório;
- Geração do relatório;
- Controle das opções de pós-geração do relatório, tais como geração de PDF, geração de XLS, envio de e-mail, salvar relatório, entre outras disponíveis;

Abaixo são apresentados os modelos de dados referentes ao módulo `GeracaoRelatorio`:

- `FiltrosSalvos` - Esta classe representa a filtragem de dados de um relatório, onde ficam armazenados os nomes dos campos de filtro, e os valores utilizados, afim de facilitar futuras gerações do mesmo relatório, sem a necessidade de digitação dos filtros manualmente. Pode-se perceber notável vantagem do uso de objetos desta classe em relatórios com um número grande de filtros, ou com filtros extremamente complexos e/ou longos.
- `GeracaoAgendada` - Esta classe possui a função de representar um agendamento de geração automática de um relatório. Após a geração de um relatório, pode ser agendada uma geração automática para o mesmo, com a última combinação de filtros utilizada.

7.1.7 `Utils`

Este módulo destina-se as funções de uso comum entre os demais módulos. Entre elas, pode-se citar:

- Geração do PDF dos relatórios;
- Geração de XLS dos relatórios;

- Geração das imagens com os gráficos dos relatórios, conforme seleção do usuário;
- Envio de e-mail com o PDF e XLS dos relatórios em anexo;
- Controle da geração automática dos relatórios, conforme agendamentos dos usuários;

7.2 Camada de Apresentação

A camada de visualização, para este sistema, manipula as informações para que elas possam ser exibidas de uma maneira agradável aos usuários, além de conter alguma lógica na manipulação dos dados. Para este sistema, cada módulo possui um arquivo-fonte com as funções necessárias para a apresentação de dados, invocando, quando necessário, outras funções e métodos de outros módulos do sistema que possuam a responsabilidade de controle de lógica de negócio.

Essas funções são chamadas a partir de urls, as quais são descritas a seguir, separadas por módulo:

7.2.1 Core

Neste módulo, as funções de visualização contemplam a tela de login, tela inicial, tela de administração de usuários e pastas de trabalho.

- `/login/` - Sempre que um usuário tenta acessar a uma página, a parte de controle de módulo Core, verifica se o usuário está autenticado. Caso o usuário não tenha sessão ativa no sistema, o mesmo é redirecionado para a tela de login. Esta função então renderiza a tela de login ao usuário.
- `/` - Renderiza a tela inicial para o usuário, considerando as permissões do mesmo para os links que serão mostrados na barra de menu.
- `/alterar_meus_dados/` - Renderiza a tela de alteração de dados pessoais do usuário com sessão ativa. Caso o usuário seja de nível administrador, é possível alterar as informações de todos os usuários.
- `/alterar_minha_senha/` - Mostra a tela de alteração da senha ao usuário com sessão ativa no sistema.
- `/administrar_usuarios/criar_novo_usuario/` - Caso o usuário possua permissões de administrador, ele terá permissão para a tela de criação de um novo usuário. Esta função de visualização tem como responsabilidade mostrar o formulário de criação de um novo usuário.
- `/minha_pasta_de_trabalho/` - Mostra o conteúdo da pasta de trabalho do usuário com sessão ativa no sistema.

7.2.2 EntidadeComposta

Neste módulo, as funções de visualização são todas que se propõem ao controle de entidades compostas, conforme é listado a seguir:

- `/administrar_entidades_compostas/` - Mostra a listagem das entidades compostas disponíveis no sistema, cada uma com um link para edição, um link para deleção e um link para duplicação. Também é mostrado um link no topo da página, mostrando ao usuário a opção de criação de uma nova entidade composta.
- `/modificar_entidade_composta/<codigo>/` - Renderiza a tela de manutenção de entidade composta. Caso seja a edição de uma entidade existente, os dados existentes relativos à mesma, são exibidos na tela, e, caso seja a criação de uma nova entidade composta, é mostrada a tela com os formulários vazios.
- `/duplica_entidade_composta/<codigo>/` - Mostra a tela de confirmação de duplicação de entidade composta, com as opções ao usuário de confirmar a ação ou cancelar.
- `/deleta_entidade_composta/<codigo>/` - Mostra a tela de confirmação de exclusão da entidade composta, com as opções ao usuário de confirmar ou cancelar a ação..

7.2.3 Relatorio

Neste módulo, as funções de visualização são todas que se propõem ao controle dos relatórios criados no sistema, conforme é listado a seguir:

- `/administrar_relatorios/` - Mostra a listagem dos relatórios disponíveis no sistema, cada um com um link para edição, um link para deleção e um link para duplicação. Também é mostrado um link no topo da página, mostrando ao usuário a opção de criação de um novo relatório.
- `/modificar_relatorio/<codigo>/` - Renderiza a tela de manutenção de um relatório. Caso seja a edição de um relatório existente, os dados existentes relativos ao mesmo, são exibidos na tela, e, caso seja a criação de um novo relatório, é mostrada a tela com os formulários vazios.
- `/duplica_relatorio/<codigo>/` - Mostra a tela de confirmação de duplicação de um relatório, com as opções ao usuário de confirmar a ação ou cancelar.
- `/deleta_relatorio/<codigo>/` - Mostra a tela de confirmação de exclusão do relatório, com as opções ao usuário de confirmar ou cancelar a ação..

7.2.4 GeracaoRelatorio

Aqui concentra-se funções de visualização responsáveis pela geração de relatórios.

- `/gerar_relatorio/<codigo>/` - Chama as funções necessárias para a montagem dos campos de filtros dinâmicos, e renderiza para o usuário a tela de filtragem dos dados do relatório. Após isto, esta mesma função trata a geração do relatório, com a chamada das funções de controle necessárias para a geração da consulta, geração dos gráficos e formatação dos campos.
- `/mostra_lov/<codigo>/<pagina>/` - Mostra a tela da lista de valores possíveis para um campo de filtro do relatório.
- `/filtros_salvos/<codigo>/` - Mostra a tela com os filtros salvos para este relatório, onde o usuário pode selecionar um filtro que foi salvo anteriormente, e os mesmos serão preenchidos na tela. Este recurso facilita ao usuário a digitação repetitiva de filtros iguais para cada geração do mesmo relatório.
- `/salva_relatorio/` - Mostra a tela de confirmação de salvamento do relatório, pedindo ao usuário que informe um nome para o mesmo.
- `/gera_pdf/` - Mostra a tela de confirmação de geração do pdf do relatório que está na tela.
- `/gera_xls/` - Mostra a tela de confirmação de geração de um arquivo xls com o conteúdo dos dados do relatório que está na tela.
- `/envia_email/` - Mostra a tela que solicita ao usuário a digitação dos e-mails desejados para que o relatório atual seja enviado, então envia um e-mail com o relatório em formatos pdf e xls gerados.
- `/agendar_geracao/` - Mostra a tela que solicita ao usuário os dados necessários para que seja configurado uma série de agendamentos para gerações agendadas do relatório que está em tela, com os mesmos filtros que foram utilizados.
- `/salva_filtros/` - Mostra a tela que pede ao usuário que o mesmo informe um nome para a série de filtros dos dados que ele estará salvando para que sejam mostrados em uma geração futura do mesmo relatório.

8 CONSIDERAÇÕES FINAIS

A complexidade do desenvolvimento de uma ferramenta com o fim de geração dinâmica de relatórios, justifica o alto valor de mercado das ferramentas com o mesmo propósito, assim como justifica a procura das empresas por manterem sistemas como estes.

Esta fase proporcionou contatos com novas áreas, como conhecimento avançado em banco de dados e processo de desenvolvimento de software.

Após estes estudos e implementações, foi possível entender a importância da informação para as empresas, sob o ponto de vista que as mesmas consideram correto para seu ramo de atuação, além da necessidade cada vez maior de informação que as empresas encontram.

8.1 Considerações sobre o Desenvolvimento

Ao longo do desenvolvimento deste software, foram sendo amadurecidos conceitos e teorias que só consegue-se deixar bem claro durante a prática dos mesmos. Desta forma, houveram situações ao longo do desenvolvimento em que fizeram-se necessárias abordagens diferentes das previstas. Além disso, ocorreram novas necessidades para que o sistema ficasse tão ou mais completo quanto o esperado inicialmente.

Entre estas situações, estão:

- Requisito funcional/Caso de uso - Exportar relatório para imagem PNG: Foi identificado durante o desenvolvimento do sistema, após o desenvolvimento completo do caso de uso que propunha a exportação do relatório gerado para o formato PDF, que era desnecessária a exportação para o formato de imagem PNG. Isto ocorre porque a saída final ao olhos do usuário ficaria praticamente igual, mudando apenas o formato binário da saída. Outro motivo determinante foi que o formato de imagens não é muito utilizado hoje para representar dados de relatórios, principalmente pelo fato de ocupar mais espaço em disco se comparado aos outros formatos que se propõem ao mesmo fim. Em compensação

da falta do recurso de exportação para imagem PNG, foi desenvolvida uma nova funcionalidade considerada mais importante, que é a exportação para o formato de planilha eletrônica(XLS - Microsoft Excel). Desta forma,todos os dados dos relatórios gerados pelo sistema, podem ser exportados para o formato de planilha eletrônica, facilitando cálculos e apresentações de dados que exigem formatos mais tabulares. Este formato também é enviado por anexo juntamente com os e-mails enviados após a geração dos relatórios, seja manualmente ou provindos de gerações automáticas agendadas.

- Envio de relatórios por e-mail no formato *HTML* - Foi identificada durante o desenvolvimento a dificuldade de desenvolver um formato de código *HTML* que seja corretamente interpretado em todos os clientes de e-mail existentes. Alguns clientes de e-mail possuem uma proteção contra span que eliminam o código *HTML* contido nas páginas, ou ainda por questões de segurança, não exibem corretamente códigos que devem ser interpretados pelo navegador *WEB*. Portanto os envios de e-mails com o resultado dos relatórios foram alterados para conterem dois anexos nos mesmos, sendo eles o relatório no formato *PDF*, e o mesmo relatório no formato de planilha eletrônica(*XLS*).

8.2 Funcionalidades Adicionais

Assim como ao longo do desenvolvimento do sistema houveram situações em que não foi possível atender ao que estava previsto, seja por impossibilidade ou por opção, houveram também situações em que foram descobertas possibilidades que agregariam valor ao sistema, e que foram desenvolvidas.

Estas situações são listadas abaixo:

- Lista de valores nos campos de filtros - Durante os testes do sistema desenvolvido, notou-se certa dificuldade em filtrar os dados da maneira desejada, nos casos em que não existisse o conhecimento mínimo necessário dos valores existentes, ou nos casos em que não sabia-se o que digitar nos campos. Para estas situações foi detectada a necessidade do desenvolvimento de uma lista de valores dinâmica, que usaria da mesma fonte de dados que o relatório a ser gerado, para mostrar as possibilidades de valores para cada campo de filtro no formato texto. Esta funcionalidade foi então desenvolvida com o objetivo de melhorar a usabilidade e agregar valor ao produto, mesmo não tendo sido prevista no período de análise do sistema.
- Dashboard(notificações) na tela inicial - Para que as gerações automáticas agendadas pelos usuário não sejam esquecidas ou passem despercebidas, foi

criado um dashboard na tela inicial, onde são exibidos avisos ao usuário, sobre as gerações automáticas dos relatórios agendados pelo o mesmo.

8.3 Trabalhos Futuros

Durante o completo desenvolvimento do sistema proposto, abriram-se novas possibilidades, linhas de pensamento e idéias, que são consideradas para serem realizadas como trabalhos futuros, visando a constante evolução da ferramenta no mercado de software. Entre estas evoluções, podem-se citar:

- Tradução para outros idiomas - Pelo motivo da ferramenta desenvolvida ser em sua maioria dinâmica, e trabalhar com o padrão de alguns dos bancos de dados mais utilizados no mundo, considera-se uma excelente evolução a tradução da mesma para outros idiomas. Desta forma a mesma poderá ser utilizada em mais locais do mundo, inclusive por empresas multinacionais que precisem acessar a mesma informação em mais de um país com diferentes idiomas.
- Dashboard como um sistema de mensagens completo - O dashboard que foi criado para alertar aos usuários sobre as gerações automáticas dos relatórios, pode ser evoluída para um sistema de mensagens completo. Entre estas possibilidades, pode-se citar um sistema de chat para os usuários da ferramenta se comunicarem, e um sistema de agenda de compromissos, funcionando como um organizador de tarefas para auxiliar os usuários a lembrarem-se de seus compromissos.
- Geração de arquivos texto ou XML para integração - Incluir mais uma opção na barra de opções pós-geração de relatório, que seria a configuração de um sistema de geração de arquivos no formato texto ou XML para serem utilizados como integração de sistemas. Como exemplo pode-se citar a geração de arquivos texto para que a empresa cumpra com suas obrigações fiscais com o governo. Esta nova funcionalidade, aliada à funcionalidade já existente de geração automática dos relatórios, pode servir para geração e envio automático de arquivos para um órgão definido, como por exemplo a Receita Federal.

REFERÊNCIAS

- ASCHER, M. L. . D. **Aprendendo Python - 2 edicao.** [S.l.]: Bookman, 2004.
- BECK, K. **PROGRAMAÇÃO EXTREMA EXPLICADA:** acolha as mudanças. [S.l.]: BOOKMAN COMPANHIA ED, 2004.
- BELO HORIZONTE (PRODABEL S/A), E. de Informatica e Informacao de. **Processo de Software da PBH/Prodabel PSP.** 2012.
- BORGES, L. E. **Python para Desenvolvedores - 2 edicao.** Edicao do Autor.ed. [S.l.: s.n.], 2010.
- CAMPINA GRANDE, U. F. de. **Departamento de Informatica da UFCG.** 2012.
- CARLOS VIDEIRA, A. S. **UML, Metodologias e Ferramentas CASE.** [S.l.]: Edicoes Centro Atlantico, 2008. 332p.
- CONSORTIUM, W. W. W. **World Wide Web Consortium.** 2012.
- DOUG ROSENBERG, M. S. e. **ICONIX Process.** 2012.
- FORD, A. **Apache 2 - Guia de bolso.** [S.l.]: OReilly/Alta Books, 2009. 189p.
- FOUNDATION, D. S. **Documentacao oficial do Django 1.3 sobre o modulo MODWSGI.** 2012.
- FOUNDATION, D. S. **Django, site oficial.** 2012.
- FOWLER, M. **UML ESSENCIAL:** um breve guia para a linguagem-padrao. [S.l.]: Bookman, 2005.
- GROUP, P. G. D. **PostgreSql, site oficial.** 2012.
- GUEDES, G. **UML - Uma Abordagem Pratica.** [S.l.]: Novatec, 2008.

IVAN LUIZIO MAGALHÃES, W.; PINHEIRO, W. **GERENCIAMENTO DE SERVIÇOS DE TI NA PRÁTICA: uma abordagem com base na itil.** [S.l.: s.n.], 2007.

KOTLER, P. **Administração de marketing.** São Paulo: Pearson, 2000.

LARMAN, C. **Utilizando Uml E Padroes 3 Ed.** [S.l.]: BOOKMAN COMPANHIA ED, 2008.

MARCELO, A. **Apache - Configurando o Servidor WEB para Linux.** [S.l.]: Brasport, 2008. 132p.

MATTOS, É. de. **Programação de Softwares em Java.** [S.l.]: Universo dos Livros Editora LTDA, 2007.

MEDEIROS, L. F. de. **Banco de Dados: princípios e prática.** [S.l.: s.n.], 2007.

PRIMAK, F. **DECISOES COM B.I.: business intelligence.** [S.l.]: Ciência Moderna, 2008.

PRIMAK, F. **ARTIGO PUBLICADO PARA AULA ONLINE DE MBA EM SISTEMAS DE INF. GERENCIAIS.** 2012.

SILVA, M. **JQUERY - A BIBLIOTECA DO PROGRAMADOR JAVASCRIPT.** [S.l.]: NOVATEC, 2012.

THIAGO GALES, O. S. e. **Python e Django, Desenvolvimento Agil de Aplicacoes Web.** Edicao do Autor.ed. [S.l.]: Novatec, 2010. 280p.

YASMINA, M. **Business Intelligence - Tecnologias da Informaçao na Gestao de Conhecimento.** [S.l.]: FCA - Editora de Informatica, 2006. v.Sistemas de Informacao.

9 ANEXOS

Segue o manual que tem como objetivo auxiliar o usuário na utilização do sistema.

Documentação do sistema

Sobre o sistema

Este é um sistema que tem como objetivo a obtenção de informações de forma customizada. Como Funciona:

1. O sistema de gestão existente gera as informações;
2. Este sistema conecta-se ao banco de dados e permite que sejam configuradas as diversas formas de apresentação dos dados.
3. O resultado final são relatórios bonitos e bem estruturados, com gráficos, e outros recursos, como agrupamento e algumas definições de layout.

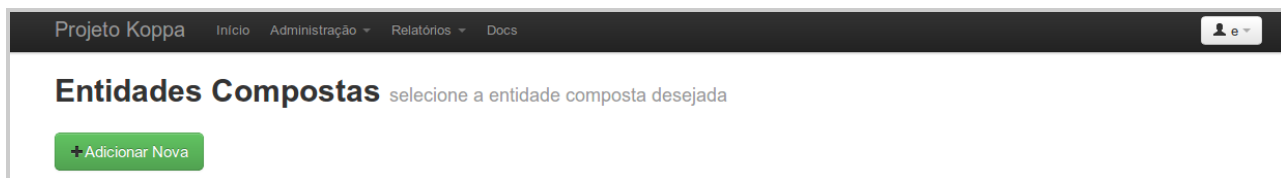
Processo de funcionamento:

Abaixo é apresentado o fluxo resumido do funcionamento do sistema:

1. Criação de entidades compostas
2. Criação de relatórios
3. Geração dos relatórios
4. Opções de pós-geração dos relatórios

Criação de entidades compostas:

- No menu superior, clicar em "Relatórios", e no submenu "Entidades Compostas";
- Será exibida uma tela com as entidades compostas disponíveis no sistema;



- Se a intenção for alterar alguma entidade composta já existente, basta clicar sobre o nome da mesma, caso contrário, clique no botão "Adicionar Nova"
- Será exibida uma tela semelhante a mostrada abaixo



- Esta tela tem como objetivo a manutenção de uma entidade composta, é possível adicionar

atributos complexos, com junção de dados de tabelas ou dados estáticos. Também é possível cadastrar filtros-padrão, que são restrições iniciais e obrigatórias para a 185 apresentação dos dados da entidade composta.

Se o objetivo for utilizar mais de uma tabela do banco de dados para a entidade composta, pode-se cadastrar as junções das tabelas, formando assim uma consulta de dados consistente.

- Para adicionar novos atributos, basta clicar no botão "Novo", localizado ao lado do texto "Atributos". Ao mesmo tempo que os atributos vão sendo adicionados, vão sendo mostradas informações dos atributos na tela, como a formação dos mesmos, e as tabelas utilizadas. Neste mesmo momento, as tabelas utilizadas já são disponibilizadas nas abas de filtros-padrão e de ligação de tabelas.
- Para salvar as informações da entidade composta, clique no botão "Salvar Entidade Composta".

Criação de relatórios:

- As entidades compostas criadas, são imediatamente disponibilizadas para serem utilizadas como fonte de informações para a criação de relatórios.
- Para acesso a tela de listagem dos relatórios disponíveis, clique no menu "Relatórios", na barra superior, e no submenu "Relatórios".
- Será exibida uma tela com os relatórios disponíveis no sistema;

Nome	Descrição	Criador	Ativo	Duplicar	Deletar
Autores	a	e	✓	Duplicar	Deletar

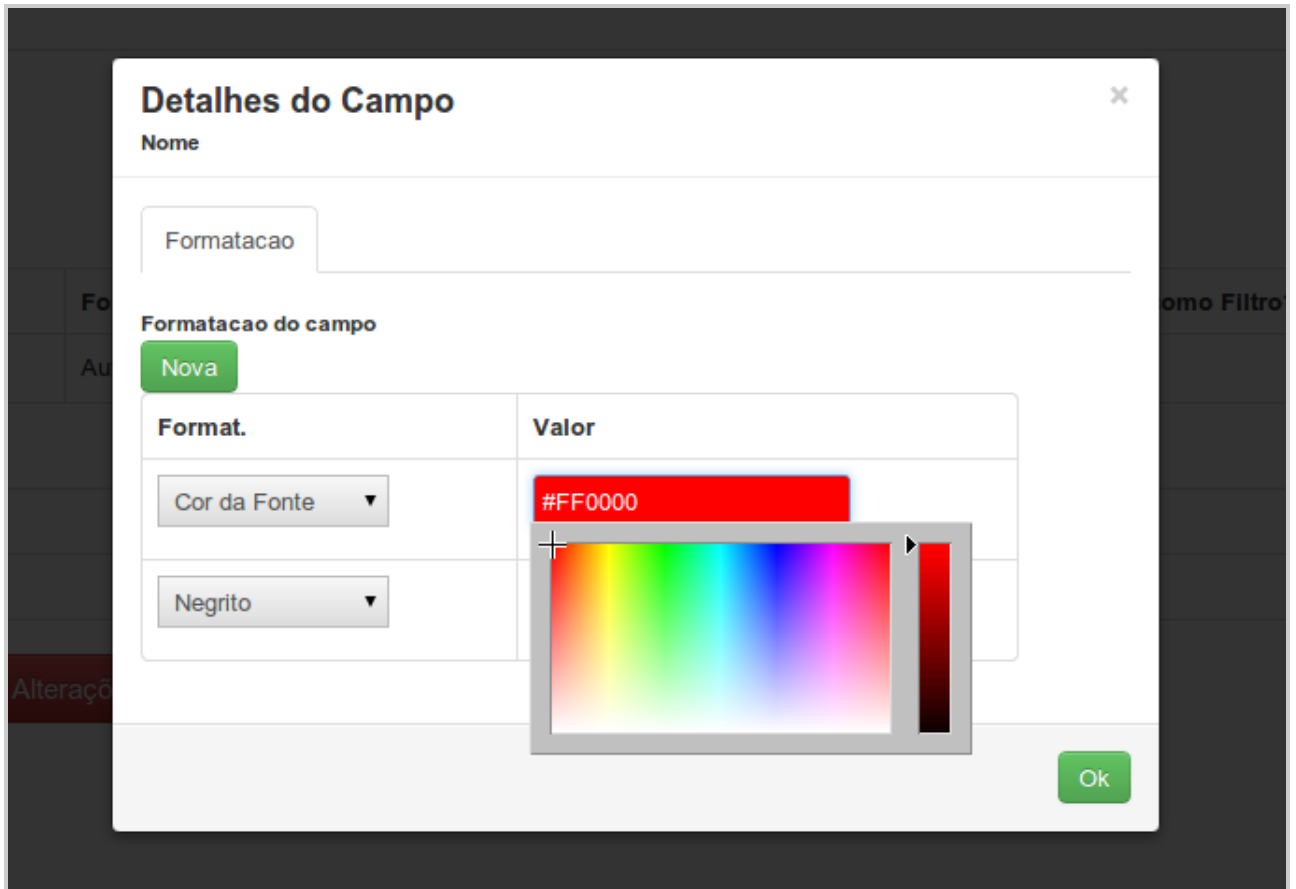
- Se a intenção for alterar algum relatório já existente, basta clicar sobre o nome do mesmo, caso contrário, clique no botão "Adicionar Novo"
- Será exibida uma tela semelhante a mostrada abaixo

Nome	Formação	Usar como Filtro?	Excluir

- Esta tela tem como objetivo a manutenção de um relatório, é possível adicionar campos, porém, diferentemente das entidades compostas, são utilizados como fonte de dados para os campos, as entidades compostas cadastradas previamente.
- Para adicionar novos campos, basta clicar no botão "Novo", localizado ao lado do texto "Campos". Ao mesmo tempo que os campos vão sendo adicionados, vão sendo

mostradas informações dos campos na tela, como a formação dos mesmos, e as entidades compostas utilizadas. Neste mesmo momento, as entidades compostas utilizadas já são disponibilizadas na aba de ligação de entidades.

- Ao adicionar um novo campo, é possível clicar sobre o mesmo, e configurar sua formatação visual. A tela de detalhes do campo é mostrada abaixo:



- Para cada um dos campos, ainda é possível selecionar a opção de permitir agrupamento pelo campo.
- Para salvar as informações da entidade composta, clique no botão "Salvar Relatório".

Informações da Tela Inicial:

- Na tela inicial são mostrados os relatórios disponíveis para a geração, e também o dashboard, que são notificações ao usuário sobre algum evento relacionado aos relatórios, como por exemplo, uma geração de relatório automática efetuada com sucesso..
- A tela inicial pode ser acessada a qualquer momento clicando-se no link início, na barra de menu principal. A tela inicial deve ser semelhante a tela da figura abaixo:



- Se a intenção for alterar algum relatório já existente, basta clicar sobre o nome d mesmo, caso contrário, clique no botão "Adicionar Novo"

- Será exibida uma tela semelhante a mostrada abaixo

Projeto Koppa Início Administração Relatórios Docs 187

Relatório manutenção de relatórios

Principal [Ligações Entidades](#)

Nome Descrição

Campos [+ Novo](#)

Nome	Formação	Usar como Filtro?	Excluir
------	----------	-------------------	---------

Entidades

Nome

[✓ Salvar Relatório](#) [✗ Cancelar Alterações](#)

Geração de relatórios:

- Para a geração dos relatórios, basta selecionar o relatório desejado, na tela inicial, e clicar sobre o nome do mesmo.
- Será mostrada a tela com os filtros possíveis para a exibição do relatório em tela. A tela deve ser semelhante a tela mostrada na figura abaixo:

Projeto Koppa Início Administração Relatórios Docs

Relatório de Autores listagem dos autores cadastrados no sistema

Nome:

Pk:

Tem contrato:

Agrupar por:

Gráfico de pizza por ocorrência:

Gráfico de barras por ocorrência:

Zebraado:

[Gerar](#)

- Basta selecionar os filtros desejados e clicar no botão "Gerar", caso ocorra algum problema nos filtros, o sistema mostrará novamente a tela de filtros, apontando os problemas ao lado de cada campo.
- Será então mostrado o resultado do relatório em tela, com as opções disponíveis.

Projeto Koppa Início Administração Relatórios Docs

188

Relatório de Autores

Nome	Pk
Abraham Bennet	409-56-7008
Marjorie Green	213-46-8915
Cheryl Carson	238-95-7766
Albert Ringer	998-72-3567
Anne Ringer	899-46-2035
Michel DeFrance	722-51-5454
Sylvia Panteley	807-91-6654
Heather McBadden	893-72-1158
Dirk Stringer	724-08-9931
Dean Stralght	274-80-9391
Livia Karsen	756-30-7391
Stearns MacFeather	724-80-9391
Ann Dull	427-17-2319
Akiko Yokomoto	672-71-3249
Michael O'Leary	267-41-2394
Burt Gringlesby	472-27-2349
Morningstar Greene	527-72-3246

- Após a geração do relatório, basta escolher uma das opções da barra de menu. Existem as seguintes opções:
 - Salvar o resultado do relatório na pasta de trabalho do usuário.
 - Voltar para a tela de filtragem dos dados.
 - Exportar o resultado do relatório para o formato PDF.
 - Exportar o resultados dos dados do relatório para o formato XLS(planilha eletrônica).
 - Enviar este relatório para qualquer e-mail, com o resultado em PDF e XLS como anexo.
 - Salvar os filtros utilizados na geração deste relatório, para poder ser utilizado em gerações futuras.
 - Agendar uma geração automática deste relatório, com datas de início e fim determinadas juntamente com o intervalo entre cada geração, com a opção de salvar o resultado na área de trabalho do usuário, ou enviar por e-mail ao finalizar cada geração.