

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS PONTA GROSSA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ALLAN FRANCIS DE PAULA
JEAN PAUL FLEISCHMANN RAMLOW**

**APLICATIVO WEB PARA PESSOAS FÍSICAS INTITULADAS
ATIRADORES, COLECIONADORES, CAÇADORES E CIDADÃO
COMUM QUE UTILIZAM PRODUTO CONTROLADO PELO EXÉRCITO
BRASILEIRO E POLÍCIA FEDERAL**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2015

**ALLAN FRANCIS DE PAULA
JEAN PAUL FLEISCHMANN RAMLOW**

**APLICATIVO WEB PARA PESSOAS FÍSICAS INTITULADAS
ATIRADORES, COLECIONADORES, CAÇADORES E CIDADÃO
COMUM QUE UTILIZAM PRODUTO CONTROLADO PELO EXÉRCITO
BRASILEIRO E POLÍCIA FEDERAL**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Msc Rogério Ranthum

PONTA GROSSA

2015



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Coordenação do Curso Superior de Tecnologia em
Análise e Desenvolvimento de Sistemas



TERMO DE APROVAÇÃO

APLICATIVO WEB PARA PESSOAS FÍSICAS INTITULADAS ATIRADORES, COLECIONADORES, CAÇADORES E CIDADÃO COMUM QUE UTILIZAM PRODUTO CONTROLADO PELO EXÉRCITO BRASILEIRO E POLÍCIA FEDERAL

por

ALLAN FRANCIS DE PAULA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 04 de novembro de 2015 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

ROGÉRIO RANTHUM
Prof. Orientador

VINÍCIUS CAMARGO ANDRADE
Membro titular

GERALDO RANTHUM
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso –



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Coordenação do Curso Superior de Tecnologia em
Análise e Desenvolvimento de Sistemas



TERMO DE APROVAÇÃO

APLICATIVO WEB PARA PESSOAS FÍSICAS INTITULADAS ATIRADORES, COLECIONADORES, CAÇADORES E CIDADÃO COMUM QUE UTILIZAM PRODUTO CONTROLADO PELO EXÉRCITO BRASILEIRO E POLÍCIA FEDERAL

por

JEAN PAUL FLEISCHMANN RAMLOW

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 04 de novembro de 2015 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

ROGÉRIO RANTHUM
Prof. Orientador

VINÍCIUS CAMARGO ANDRADE
Membro titular

GERALDO RANTHUM
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso –

DEDICATÓRIA

Dedicamos este Trabalho primeiramente a Deus, que nos deu todas as condições para que conseguíssemos bons resultados ao final da jornada acadêmica.

Aos nossos excelentes Professores, pela dedicação a nós alunos desta renomada Instituição.

Ao nosso Professor Orientador Rogério Ranthum, pela confiança e liderança.

E as nossas queridas e amadas famílias, pelo apoio e compreensão.

AGRADECIMENTOS

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de nossas vidas. Portanto, desde já pedimos desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte de nosso pensamento e de nossa gratidão.

Agradecemos ao nosso orientador Prof. Msc Rogerio Ranthum, pela sabedoria com que nos guiou nesta trajetória.

Aos nossos colegas de sala.

A Secretaria do Curso, pela cooperação.

Gostaríamos de deixar registrado também, o nosso reconhecimento às nossas famílias, pois acreditamos que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que, por algum motivo, contribuíram para a realização desta pesquisa.

“Algumas pessoas acham que foco significa dizer sim para a coisa em que você vai se focar. Mas não é nada disso. Significa dizer não às centenas de outras boas ideias que existem. Você precisa selecionar cuidadosamente.”

Steve Jobs, em 2008, para a revista Fortune.

RESUMO

RAMLOW, Jean Paul Fleischmann. PAULA, Allan Francis de. Aplicativo Web para Pessoas Físicas Intituladas Atiradores, Colecionadores, Caçadores e Cidadão Comum que Utilizam Produto Controlado pelo Exército Brasileiro e Polícia Federal. 2015. 94 f. Trabalho de Conclusão de Curso, Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2015.

Este trabalho apresenta a proposta de um aplicativo web para facilitar o acesso de pessoas físicas na aquisição de armas de fogo e insumos controlados e fiscalizados pelos órgãos federais Exército Brasileiro e Polícia Federal, o qual a estrutura funcional destas instituições até chega a ser atualizada, mas ainda em vias de ser adequada, executando os encargos de ordem técnica e burocrática, por meio de suas seções internas ainda em grande parte com processos físicos e não digitais. Existem dois universos distintos fiscalizados quanto a natureza da atividade: 1) Pessoas Jurídicas: fabricação, importação, exportação, desembaraço alfandegário, comercialização e tráfico; 2) Pessoas Físicas: praticantes de tiro desportivo, colecionadores de armas, munições, viaturas, aviões, etc., praticantes da caça legalizada e o cidadão que apenas deseja a posse de arma de fogo, visando a segurança própria e familiar. O aplicativo web desenvolvido neste trabalho tem como objeto o universo de pessoas físicas, haja visto que, devido a grande quantidade de legislação existente sobre o tema, gera muitas dúvidas quanto aos procedimentos. Dessa forma, a intenção deste trabalho é facilitar o acesso através de um ambiente interativo e com um mínimo esforço por parte do interessado. Para o desenvolvimento do aplicativo, optou-se pela linguagem PHP, atrelada à HTML, CSS e o *FrameWork* CodeIgniter.

Palavras-chave: Fiscalização. Aplicativo. Exército Brasileiro. Polícia Federal

ABSTRACT

Ramlow, Jean Paul Fleischmann. PAULA, Allan Francis. Web application for individuals entitled Shooters, Collectors, Hunters and Common Citizens who use controlled by the Brazilian Federal Police and Army product. 2015. 94 f. Work Course Completion, Federal Technological University of Paraná. Ponta Grossa, 2015.

This work presents a proposal of a web application to facilitate the access of individuals in the acquisition of firearms and controlled inputs and supervised by federal agencies Brazilian Army and Federal Police, where the functional structure of these institutions to reach to be updated, but still on track to be appropriate, the costs of running technical and bureaucratic order through its internal sections still largely with physical and not digital processes. There are two distinct universes supervised as the nature of the activity: 1) Corporations: manufacture, import, export, customs clearance, marketing and traffic; 2) Individuals: practicing target shooting, collectors weapons, ammunition, vehicles, airplanes, etc., practitioners of legalized hunting and citizens who just want possession of a firearm, aimed at individual and familial security. The web application developed in this work has as its object the universe of individuals, given the fact that due to the large amount of existing legislation on the subject, raises many questions about the procedures. Thus, the intention of this work is to facilitate access through an interactive environment with minimal effort by the person concerned. For application development, we opted for the PHP language, tied to HTML, CSS and the *FrameWork* CodeIgniter.

Keywords: Supervision. Application. Brazilian Army. Federal Police

LISTA DE ILUSTRAÇÕES

Figura 1. Exemplos de Atores.....	23
Figura 2. Relacionamento entre Atores.....	25
Figura 3. Exemplo de Relações entre Atores e Casos de Uso.....	26
Figura 4. Exemplo de Relações de inclusão entre Casos de Uso.....	27
Figura 5. Exemplo de Relações de extensão entre Casos de Uso.....	28
Figura 6. Exemplo de Relações de generalização entre Casos de Uso.....	29
Figura 7. Exemplo de Pacotes e dependências.....	30
Figura 8. Diagrama no Astah Community.....	32
Figura 9. Primeiro DER de um sistema para biblioteca.....	33
Figura 10. DER mais completo do sistema para bibliotecas.....	33
Figura 11. Exemplo de agregação.....	38
Figura 12. Exemplo de composição.....	39
Figura 13. Exemplo de especialização/generalização.....	40
Figura 14. Exemplo de dependência.....	41
Figura 15. Exemplo de realização.....	42
Figura 16. Exemplo de classe associativa.....	43
Figura 17. Exemplo de restrição em uma associação.....	43
Figura 18. Aplicação desenvolvida reutilizando classes de biblioteca.....	45
Figura 19. Aplicação desenvolvida reutilizando um <i>framework</i>	45
Figura 20. Visão conceitual da estrutura de um <i>framework</i>	47
Figura 21. Camadas do padrão MVC.....	51
Figura 22. O Modelo MVC.....	52
Figura 23. Diagrama Entidade Relacionamento de sistema de imobiliária.....	58
Figura 24. Diagrama de Entidade Relacionamento (variação).....	59
Figura 25. Diagrama Entidade Relacionamento (variação 2).....	59
Figura 26. Atributos apresentados como elipses.....	60
Figura 27. Diagrama com atributos nas entidades.....	60
Figura 28. Diagrama de Casos de Uso.....	63
Figura 29. Diagrama de Classes.....	64
Figura 30. Diagrama Entidade-Relacionamento.....	66
Figura 31. Modelo Relacional.....	67
Figura 32. Tela de Login.....	70
Figura 33. Tela de Cadastre-se.....	70
Figura 34. Tela de recuperação de senha.....	71
Figura 35. Tela Inicial.....	72
Figura 36. Tela alterar senha.....	72
Figura 37. Tela Processos Exército.....	73
Figura 38. Tela Processos Polícia Federal.....	73
Figura 39. Tela Notificação.....	74
Figura 40. Tela de Pagamento.....	75
Figura 41. Tela de Pagamento Efetuado com Sucesso.....	75
Figura 42. Tela Inicial para o Gerente do Sistema.....	76
Figura 43. Tela de Cadastros.....	77
Figura 44. Tela de Cadastro de Dados.....	77
Figura 45. Tela de Upload de Arquivos.....	78
Figura 46. Tela de Processos Cadastrados do Cliente.....	78
Figura 47. Tela de Gerenciamento de Clientes.....	79
Figura 48. Tela de Gerenciamento de Armas.....	79

Figura 49. Tela de Cadastro de Processos.....	80
Figura 50. Tela de Gerenciamento de Processos.....	80
Figura 51. Tela de Configuração do Sistema.....	81
Figura 52. Tela de Personalização do Sistema.....	81
Figura 53. Tela de Gerenciamento de Login.....	82
Figura 54. Tela de Relatórios.....	82
Figura 55. Sistema desenvolvido com Framework CodeIgniter.....	98
Figura 56. Diretório config em destaque.....	98
Figura 57. Diretório controllers em destaque.....	99
Figura 58. Diretório helpers de funções em destaque.....	99
Figura 59. Diretório models em destaque.....	100
Figura 60. Diretório views em destaque.....	100

LISTA DE ABREVIATURAS

CR	Certificado de Registro
CRAF	Certificado de Registro de Arma de Fogo
DFPC	Diretoria de Fiscalização de Produtos Controlados
EB	Exército Brasileiro
GIF	Graphics Interchange Format - Formato para Intercâmbio de Gráficos
HTML	Hepertext Makup Language
HTTP	Hipertext Transfer Protocol
JPEG	Joint Photographic Experts Group
MVC	Model-View-Control
OJB	Object-Relational Bridge
POO	Programação Orientada ao Objeto
PF	Polícia Federal
PNG	Portable Network Graphics
R-105	Regulamento para a Fiscalização de Produtos Controlados
ROM	Relation Object Model
SI	Sistemas de Informação
SIGMA	Sistema de Gerenciamento Militar de Armas – SIGMA
SINARM	Sistema Nacional de Armas - SINARM
SQL	Structured Query Language
SSL	Secure Socket Layer
TCP/IP	Tranfer Control Protocol/Internet Protocol
TI	Tecnologia da Informação
UML	Unified Modeling Language
URL	Uniform Resource Locator. Localizado
UTFPR	Universidade Tecnológica Federal do Paraná
VO	Value Object
W3C	World Wide Web Consortium
XML	Extensible Makup Language
XSL	Extendible Stylesheet Language

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1 OBJETIVOS.....	2
1.1.1 OBJETIVO GERAL.....	2
1.1.2 OBJETIVOS ESPECÍFICOS.....	2
1.2 JUSTIFICATIVA.....	3
2. EMBASAMENTO TEÓRICO.....	4
2.1 PRODUTO CONTROLADO.....	4
2.2 EXÉRCITO BRASILEIRO.....	4
2.2.1 Concessão de Certificado de Registro.....	5
2.2.1.1 Atirador desportivo.....	5
2.2.1.2 Caçador.....	5
2.2.1.3 Colecionador.....	6
2.2.1.4 Atirador de esporte de ação com arma de pressão.....	6
2.2.1.5 Clubes de Tiro, Clubes de Caça e Associações de Tiro.....	6
2.2.2 SIGMA.....	6
2.3 POLÍCIA FEDERAL.....	7
2.3.1 Concessão de Arma de Fogo.....	7
2.3.2 SINARM.....	9
2.4 A GESTÃO DA INFORMAÇÃO NO SETOR PÚBLICO.....	10
2.5 TECNOLOGIA DA INFORMAÇÃO.....	11
2.6 A LINGUAGEM DE PROGRAMAÇÃO.....	12
2.7 A PROGRAMAÇÃO ORIENTADA A OBJETOS – POO.....	12
2.7.1 Classes.....	14
2.7.2 Atributos.....	14
2.7.3 Métodos.....	15
2.7.4 Objetos.....	15
2.7.5 Abstração.....	16
2.7.6 Encapsulamento.....	17
2.7.7 Herança.....	18
2.7.8 Polimorfismo.....	19
2.8 LINGUAGEM UNIFICADA DE MODELAGEM (UML).....	20
2.8.1 MODELO DE CASOS DE USO.....	21
2.8.1.1 Diagramas de Casos de Uso.....	22
2.8.1.2 Atores.....	22
2.8.1.3 Casos de Uso.....	24
2.8.1.4 Relacionamentos entre Atores.....	24
2.8.1.5 Relacionamentos entre Atores e Casos de Uso.....	25
2.8.1.6 Relacionamentos entre Casos de Uso.....	26
2.8.1.7 Relacionamento de Inclusão.....	26
2.8.1.8 Relacionamento de Extensão.....	27
2.8.1.9 Relacionamento de Generalização.....	28
2.8.1.10 Decomposição de Diagramas de Casos de Uso.....	29
2.9 FERRAMENTAS CASE.....	31
2.10 DIAGRAMA DE CLASSES.....	34
2.10.1 Persistência.....	35

2.10.2 Classes, Atributos e Métodos.....	36
2.10.3 Relacionamentos.....	37
2.10.4 Associações.....	37
2.10.5 Classe Associativa.....	42
2.10.6 Restrição.....	43
2.11 FRAMEWORK.....	44
2.11.1 Framework Codeigniter.....	48
2.12 PADRÕES DE PROJETO.....	48
2.12.1 Arquitetura MVC (Model –View – Controller.....	49
2.13 MODELO ENTIDADE RELACIONAMENTO (MER) E DIAGRAMA ENTIDADE-RELACIONAMENTO (DER).....	53
2.13.1 Modelo Entidade Relacionamento.....	54
2.13.2 Entidades.....	54
2.13.3 Relacionamentos.....	55
2.13.4 Atributos do DER.....	56
2.13.5 Diagrama Entidade Relacionamento.....	57
3. DESENVOLVIMENTO.....	61
3.1 DIAGRAMA DE CASOS DE USO.....	62
3.1.1 Detalhamento dos Casos de Uso.....	63
3.2 DIAGRAMA DE CLASSES.....	63
3.3 MODELO ENTIDADE RELACIONAMENTO E MODELO RELACIONAL.....	64
3.3.1 Diagrama do Modelo Entidade Relacionamento.....	66
3.3.2 Modelo Relacional.....	67
3.4 IMPLEMENTAÇÃO.....	68
4. RESULTADOS.....	68
4.1 SISTEMA.....	69
4.1.1 Tela de Login.....	69
4.1.2 Tela de Cadastre-se.....	70
4.1.3 Tela de Recuperação de Senha.....	71
4.1.4 Tela Inicial.....	71
4.1.5 Tela Altera Senha.....	72
4.1.6 Tela Processos Exército.....	72
4.1.7 Tela Polícia Federal.....	73
4.1.8 Tela Notificação.....	74
4.1.9 Tela Pagamento.....	74
4.1.10 Tela Pagamento Efetuado com Sucesso.....	75
4.1.11 Tela Inicial para o Gerente do Sistema.....	76
4.1.12 Tela de Cadastros.....	76
4.1.13 Tela de Cadastro de Clientes.....	77
4.1.14 Tela de Gerenciamento de Cadastros.....	79
4.1.15 Tela de Cadastro de Processos.....	80
4.1.16 Tela de Gerenciamento de Processos.....	80
4.1.17 Tela de Configurações do Sistema.....	81
4.1.18 Tela de Relatórios.....	82
5. CONCLUSÃO.....	83
5.1 DIFICULDADES ENCONTRADAS.....	84

5.2 TRABALHOS FUTUROS.....	86
REFERÊNCIAS.....	87
APÊNDICE A - SCRIPTS DAS TABELAS DA APLICAÇÃO.....	90
APÊNDICE B - PRINCIPAIS TELAS DE DESENVOLVIMENTO DO APLICATIVO.....	98
APÊNDICE C - DETALHAMENTO DE TODOS OS CASOS DE USO DESENVOLVIDOS NA APLICAÇÃO.....	101

1. INTRODUÇÃO

As últimas décadas testemunharam grandes mudanças nas estruturas sociais, políticas e econômicas do mundo. Os “velhos métodos” cedem lugar a um mundo novo, ágil, e cada vez mais digital. Este movimento traz reflexos diretos nos governos e nas estruturas necessárias a atuação do Estado perante a sociedade, que por sua vez está se adaptando e aprendendo a lidar com este cenário.

Segundo Goldsmith e Eggers (2006), o modelo tradicional e hierárquico de governo já não atende o cenário complexo e em constante transformação. Eles defendem que os sistemas burocráticos são rígidos, estruturados sobre procedimentos de comando e controle, com restrições de trabalho, com a cultura e modelos operacionais introvertidos, inadequados para abordar problemas que transcendem os limites das organizações.

O advento da Internet trouxe a rápida expansão e evolução dos meios de comunicação, que ampliaram e revolucionaram o acesso à informação, provocando um processo de comunicação jamais experimentado pela humanidade. As fronteiras físicas estão desaparecendo em meio à revolução digital que evolui rapidamente. Como fruto deste processo de mudança, este cenário caminha em velocidade acelerada com novos desafios e paradigmas para o Estado. Os projetos de inclusão digital estão mais abrangentes, proporcionando a expansão do acesso à Internet e tornando-o mais democrático. Os dispositivos móveis estão cada vez mais interconectados e tornando-se acessíveis a uma camada maior da população. A informação agora torna-se um dos principais ativos das organizações e, por consequência, o uso adequado das tecnologias da informação e comunicação tornam-se fatores determinantes para o sucesso neste ambiente.

As reformas administrativas pelas quais o Estado passa, neste momento, são realizadas em todas as suas áreas, esferas e poderes, visando a flexibilização e modernização da máquina pública para adequá-la a esta nova realidade. Há um processo de transformação que é iminente e irrefutável.

A democratização e universalização do acesso aos serviços públicos desempenha papel fundamental. É neste contexto que as distâncias físicas são superadas e a tecnologia da informação torna-se uma ferramenta para viabilizar a prestação de serviços públicos de forma ágil e eficiente à milhões de brasileiros.

Estes serviços, principalmente por meios eletrônicos, tem como um de seus principais objetivos disponibilizar todos ou a maior parte dos serviços a partir de um único ponto de entrada, a qualquer hora. Portanto, não basta apenas digitalizar o serviço, é preciso agregar valor, reduzir a burocracia e simplificá-lo, (ESPÍNDOLA, 2011).

Atualmente existem inúmeros serviços públicos prestados pela Internet, mas nem sempre oferecidos de forma integrada, o que obriga o usuário a realizá-lo em etapas e em mais de um sítio ou portal. E quando se trata de produtos controlados, tais como armas de fogo, insumos para munição e outros desta natureza, a falta de suporte digital para o usuário é mais acentuada, fato este que possibilitou e embasou nossos estudos neste projeto.

1.1 OBJETIVOS

Os objetivos geral e específicos serão descritos abaixo.

1.1.1 Objetivo Geral

Desenvolvimento de software facilitador à burocracia do processo de aquisição de produtos controlados pelo Exército Brasileiro (EB) e Polícia Federal (PF), denominados armas e munições, adquiridos por importação, na Indústria Nacional, Comércio Especializado ou por transferência legalizada.

1.1.2 Objetivos Específicos

- Fazer uso das linguagens de programação orientada a objeto para web, PHP, HTML entre outras;
- Utilizar o *Framework CodeIgniter* como ferramenta de segurança e estruturação MVC e o *FrameWork* CSS e JS – Foundation, que permitem desenvolver muito mais rápido as fundações básicas de um *website*;
- Modelar e Implementar um software de gestão documental ao cidadão comum para aquisição de produtos controlados no âmbito do Exército Brasileiro e Polícia Federal.

1. Delimitação de Estudo:

O estudo de caso em questão limita-se aos seguintes itens:

- Clientes: Clube de Tiro, atiradores e cidadãos comuns;
- Atuação: controle e gestão documental;
- Foco do trabalho: Módulo operacional;
- Concentração: otimização do processo de aquisição de produtos controlados no âmbito EB e PF.

1.2 JUSTIFICATIVA

Atualmente a legislação brasileira permite ao cidadão comum adquirir e registrar armas de fogo em seu nome de duas formas: junto à Polícia Federal - PF no SINARM – Sistema Nacional de Armas, para defesa própria ou caça de subsistência (Lei nº 10.826 de 22 de dezembro de 2003) ou junto ao Exército Brasileiro - EB no SIGMA – Sistema de Gerenciamento Militar de Armas, como atirador desportivo, caçador ou colecionador (Decreto nº 3.665, de 20 de novembro de 2000. Regulamento para a Fiscalização de Produtos Controlados (R-105) e Portaria nº 51-COLOG, de 8 SET 15).

Em ambos os sistemas ainda impera procedimentos burocráticos e na forma física, ou seja, formulários, documentos e taxas necessárias devem ser impressos e entregues presencialmente nos referidos órgãos, estando sujeitos a extravios de documentos, erros de análise documental e morosidade no andamento processual.

A fim de otimizar o processo para o cidadão comum, que se torna muito complicado devido à burocracia e às incontáveis legislações existentes, evitando erros no preenchimento da documentação padrão ou mesmo a falta de algum dos documentos obrigatórios sem a necessidade da participação direta dos interessados e com o máximo de agilização nos processos documentais, foi que surgiu a ideia da implementação de uma ferramenta web acessível e de fácil uso a todos interessados na aquisição de produtos controlados pelos organismos governamentais envolvidos.

2. EMBASAMENTO TEÓRICO

Neste capítulo, veremos todo embasamento teórico que foi pesquisado e estudado para servir de base para o desenvolvimento do projeto foco deste trabalho.

2.1 PRODUTOS CONTROLADOS

Os produtos que, devido ao seu poder de destruição ou outra propriedade, deva ter seu uso restrito a pessoas físicas e jurídicas legalmente habilitadas, capacitadas técnica, moral e psicologicamente, de modo a garantir a segurança social e militar do país, R-105 (2000, p.04).

O exercício da fiscalização de produtos controlados abrange as mais variadas atividades, tais como: fabricação, importação, exportação, desembaraço alfandegário, comercialização e tráfego, prática de tiro esportivo, coleção de armas, munições, viaturas, aviões, entre outros, prática da caça legalizada ou ainda defesa própria, para as pessoas físicas e jurídicas, cada uma delas adequadas ao interesse que o produto ou a atividade desperta.

A regulamentação e o controle do comércio e circulação de armas de fogo no Brasil ocorrem por motivos óbvios: para além do contexto social de extrema violência nos meios urbanos e rurais, o rastreamento de armas de fogo é condição fundamental para a tutela do sistema de segurança pública e de persecução penal.

O Estatuto do Desarmamento (Lei Federal 10.826/2003) é a norma central no controle brasileiro sobre armas. É uma legislação ordinária federal que desvela a preocupação do Estado quanto à incolumidade física de pessoas e do patrimônio, entre outros bens jurídicos. (Queiroz e Mesquita, 2009)

O Controle de armas no país é compartilhado pela PF e pelo EB, e neste contexto entra o cidadão comum, que deseja adquirir arma de fogo, produto este controlado, conforme seu interesse e submetendo-se aos critérios dos órgãos controladores, tendo cada um deles, definições, normas e procedimentos próprios para as concessões.

2.2 EXÉRCITO BRASILEIRO

De acordo com o R-105 (2000, p. 05) produtos controlados pelo EB são todos aqueles que, devido ao seu poder de destruição ou outra propriedade, deva ter seu uso restrito a pessoas físicas e jurídicas legalmente habilitadas, capacitadas técnica, moral e psicologicamente, de modo a garantir a segurança social e militar do país. Exemplos: armas, munições, explosivos, produtos químicos, entre outros.

Em relação ao cidadão comum, o EB fiscaliza às atividades de colecionamento, tiro desportivo e caça, para tanto o mesmo necessita realizar seu credenciamento junto à Diretoria de Fiscalização de Produtos Controlados – DFPC, bem como o cadastro das armas de fogo no Sistema de Gerenciamento Militar de Armas - SIGMA.

2.2.1 Concessão de Certificado de Registro

Para se habilitar, o cidadão comum deve realizar primeiramente o credenciamento através do processo de Certificado de Registro – CR.

O CR, segundo o R-105 (2000, p .04) e Portaria nº 51-COLOG, de 8 SET 15, é o documento hábil que autoriza as pessoas físicas ou jurídicas à utilização industrial, armazenagem, comércio, exportação, importação, transporte, manutenção, recuperação e manuseio de produtos controlados.

Ao solicitar o CR, o cidadão necessita indicar as categorias nas quais deseja exercer suas atividades, nas conformidades do R-105 e Portaria nº 51-COLOG, de 8 SET 15, podendo ser:

2.2.1.1 Atirador desportivo

Pessoa física praticante do esporte de tiro, devidamente registrado na Associação competente, ambas reconhecidas e sujeitas às normas baixadas pelo Exército Brasileiro.

2.2.1.2 Caçador

Pessoa física praticante de caça desportiva, devidamente registrado na Associação competente, ambas reconhecidas e sujeitas às normas baixadas pelo Exército Brasileiro.

2.2.1.3 Colecionador

Pessoa física ou jurídica que coleciona armas, munições ou viaturas blindadas, devidamente registrado e sujeito às normas baixadas pelo Exército Brasileiro.

2.2.1.4 Atirador de esporte de ação com arma de pressão

São atividades recreativas de entretenimento, não enquadradas no art. 72 da Portaria nº 51-COLOG, de 8 SET 15, nas quais são empregadas armas de pressão.

2.2.1.5 Clubes de Tiro, Clubes de Caça e Associações de Tiro

Clubes de Tiro, Clubes de Caça e Associações de Tiro, são entidades jurídicas com espaço físico apropriado para a prática ou treinamento de tiro desportivo, devidamente registrados no órgão federal competente, neste caso, Exército Brasileiro, com autorização para sediar treinamentos e competições de nível municipal, estadual e federal.

2.2.2 SIGMA

O Sistema de Gerenciamento Militar de Armas – SIGMA, é o sistema eletrônico de registro de armas de fogo gerenciado pelo Exército Brasileiro e pautado no Decreto 5.123, de 1º de julho de 2004, da seguinte forma:

Art. 2º. O SIGMA, instituído no Ministério da Defesa, no âmbito do Comando do Exército, com circunscrição em todo o território nacional, tem por finalidade manter cadastro geral, permanente e integrado das armas de fogo importadas, produzidas e vendidas no país, de competência do SIGMA, e das armas de fogo que constem dos registros próprios.
§ 1º Serão cadastradas no SIGMA:

I.as armas de fogo institucionais, de porte e portáteis, constantes de registros próprios:

a) das Forças Armadas;

b) das Polícias Militares e Corpos de Bombeiros Militares;

c) da Agência Brasileira de Inteligência; e

d) do Gabinete de Segurança Institucional da Presidência da República;

II.as armas de fogo dos integrantes das Forças Armadas, da Agência Brasileira de Inteligência e do Gabinete de Segurança Institucional da Presidência da República, constantes de registros próprios;

III.as informações relativas às exportações de armas de fogo, munições e demais produtos controlados, devendo o Comando do Exército manter sua atualização;

IV.as armas de fogo importadas ou adquiridas no país para fins de testes e avaliação técnica; e

V.as armas de fogo obsoletas.

§ 2º Serão registradas no Comando do Exército e cadastradas no SIGMA:

I.as armas de fogo de colecionadores, atiradores e caçadores; e

II.as armas de fogo das representações diplomáticas.

2.3 POLÍCIA FEDERAL

O Estatuto do Desarmamento permite a tutela administrativa preventiva e de polícia sobre arma de fogo, decorrente do artigo 144 da Constituição Federal, que define a segurança pública como dever do Estado. Ou seja, o controle da arma de fogo está umbilicalmente atrelado ao dever previsto no artigo 144, cujo fim é a preservação, em linhas gerais, da ordem pública.

Nesse panorama, a Polícia Federal ainda assume um papel de destaque em relação aos demais órgãos de segurança pública no controle de armas, porque, por opção legislativa consolidada no Estatuto do Desarmamento, compartilha (junto com o Comando do Exército) da responsabilidade sobre a circulação de armas de fogo em território nacional, uma vez que o Sistema Nacional de Armas - SINARM está instituído no seu âmbito.

2.3.1 Concessão de Arma de Fogo

Anterior ao Estatuto do Desarmamento, era incumbência das Secretarias de Segurança Pública dos Estados e do Distrito Federal, realizar o registro e controle das armas de fogo em circulação nos estados da federação.

Com o advento do Estatuto, passa à Polícia Federal através do SINARM, conforme Art. 1º e 2º, a emissão de autorização de porte e o registro de armas de fogo de uso permitido, bem como a compra de munição, para o cidadão comum.

O Estatuto do Desarmamento nos Art 4º e Art. 28, regulamenta as condições para a aquisição de arma de fogo:

Art. 4º. Para adquirir arma de fogo de uso permitido o interessado deverá, além de declarar a efetiva necessidade, atender aos seguintes requisitos:

I – comprovação de idoneidade, com a apresentação de certidões de antecedentes criminais fornecidas pela Justiça Federal, Estadual, Militar e Eleitoral e de não estar respondendo a inquérito policial ou a processo criminal;

II – apresentação de documento comprobatório de ocupação lícita e de residência certa;

III – comprovação de capacidade técnica e de aptidão psicológica para o manuseio de arma de fogo, atestadas na forma disposta no regulamento desta Lei.

(...)

Art. 28. É vedado ao menor de 25 (vinte e cinco) anos adquirir arma de fogo, ressalvados os integrantes das entidades constantes dos incisos I, II e III do art. 6º desta Lei.

A legislação ainda prevê a seguinte classificação para o porte e registro de arma de fogo:

- a) Integrantes das forças armadas (Art. 6º, I);
- b) Integrantes de órgãos referidos nos incisos do caput do Art. 144 da Constituição Federal (Art. 6º, II);
- c) Integrantes das Guardas Municipais (Art. 6º, III e IV);
- d) Agentes da Agencia Brasileira de Inteligência, do Departamento de Segurança do Gabinete de Segurança Institucional da Presidência da República (Art. 6º, V);
- e) Integrantes dos órgãos policiais referidos no Art. 51, IV e no Art. 52, XIII da Constituição Federal (Art. 6º, VI);
- f) Integrantes do quadro efetivo dos agentes e guardas prisionais, escoltas de presos e guardas portuárias. (Art. 6º, VII);

- g) Empresas de segurança privada e de transporte de valores constituídas (Art. 6º, VIII);
- h) Caçador de subsistência (Art. 6º, § 5º);
- i) Cidadão comum que comprove efetiva necessidade (Art. 10º).

Para o porte de arma de fogo, cabe ressaltar que a arma deverá ser registrada em nome da instituição, nos casos acima citados ou da pessoa que estiver conduzindo a mesma.

No caso do caçador de subsistência é necessário primeiramente o registro no Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis – IBAMA, e este somente poderá portar e registrar espingardas em locais compatíveis com a categoria.

2.3.2 SINARM

O Sistema Nacional de Armas – SINARM, é o sistema eletrônico de registro de porte e armas de fogo gerenciado pela Polícia Federal e instituído pela Lei 10.826, 22 de dezembro de 2003 – Estatuto do Desarmamento, da seguinte forma:

Art. 1º O Sistema Nacional de Armas - SINARM, instituído no Ministério da Justiça, no âmbito da Polícia Federal, tem circunscrição em todo o território nacional.

Art. 2º Ao SINARM compete:

I - identificar as características e a propriedade de armas de fogo, mediante cadastro;

II - cadastrar as armas de fogo produzidas, importadas e vendidas no País;

III – cadastrar as autorizações de porte de arma de fogo e as renovações expedidas pela Polícia Federal;

IV - cadastrar as transferências de propriedade, extravio, furto, roubo e outras ocorrências suscetíveis de alterar os dados cadastrais, inclusive as decorrentes de fechamento de empresas de segurança privada e de transporte de valores;

V - identificar as modificações que alterem as características ou o funcionamento de arma de fogo;

VI.integrar no cadastro os acervos policiais já existentes;

VII - cadastrar as apreensões de armas de fogo, inclusive as vinculadas a procedimentos policiais e judiciais;

VIII – cadastrar os armeiros em atividade no País, bem como conceder licença para exercer a atividade;

IX – cadastrar mediante registro os produtores, atacadistas, varejistas, exportadores e importadores autorizados de armas de fogo, acessórios e munições;

X – cadastrar a identificação do cano da arma, as características das impressões de raiamento e de microestriamento de projétil disparado, conforme marcação e testes obrigatoriamente realizados pelo fabricante.

XI – informar às Secretarias de Segurança Pública dos Estados e do Distrito Federal os registros e autorizações de porte de armas de fogo nos respectivos territórios, bem como manter o cadastro atualizado para consulta.

Parágrafo único. As disposições deste artigo não alcançam as armas de fogo das Forças Armadas e Auxiliares, bem como as demais que constem dos seus registros próprios.

2.4 A GESTÃO DA INFORMAÇÃO NO SETOR PÚBLICO

De acordo com os resultados da Pesquisa Sobre o Uso das Tecnologias da Informação e da Comunicação no Brasil - TIC Governo Eletrônico 2010 (CGI.BR, 2010) a principal forma de acesso aos serviços públicos é a presencial (60% de indivíduos). No entanto, quando o cidadão utiliza a tecnologia como mediadora do acesso aos serviços públicos, 35% citaram a Internet como principal forma de obtenção de algum serviço público. Entre as empresas, ao contrário do que ocorre entre cidadãos, a Internet predomina como canal de obtenção de serviços públicos: 79% utilizaram ao menos um dos serviços nos últimos 12 meses. O potencial de crescimento efetivo do e-Gov no Brasil é promissor: mais da metade da população (56% dos entrevistados) escolheria a Internet para acessar serviços de governo na próxima vez que tiver necessidade, (ESPÍNDOLA, 2011).

A estrutura da fiscalização dos órgãos federais, vem se aprimorando com o passar dos anos. Hoje a estrutura funcional até chega a ser atualizada, mas ainda em vias de ser adequada, executando os encargos de ordem técnica e burocrática por meio de suas seções internas ainda em grande parte com processos físicos e não digitais.

Toda a legislação pertinente sobre o tema, em forma de Leis, Decretos, Portarias e Normas em vigor que estabelecem todos os procedimentos para que sejam exercidas atividades com produtos controlados, são disponibilizados pelo EB e PF, em seus respectivos sítios na web.

Porém a falta de interligação dos sistemas SIGMA e SINARM, decorridos 12 (doze) anos da vigência do Estatuto do Desarmamento, aliado a falta de tecnologias de acesso a informação e serviços voltados ao segmento de produtos controlados

ao cidadão comum, foram os fatores que possibilitaram o estudo do presente trabalho.

Nesta pesquisa focaremos apenas no trato com pessoas físicas fiscalizadas pela PF e EB, tendo em vista que neste meio não encontramos nenhum serviço disponibilizado na web pra facilitar os procedimentos para este tipo de usuário, sendo que as próprias instituições fiscalizadoras também não fornecem nada neste aspecto, despertando assim, nosso interesse em desenvolver um aplicativo que supra tais necessidades, adotando para esse desenvolvimento linguagem de programação PHP, voltado à Programação Orientada a Objetos, com utilização do *framework CodeIgniter* numa arquitetura MVC e *FrameWork* CSS e JS – Foundation, que permite desenvolver muito mais rápido as fundações básicas de um *website*.

2.5 TECNOLOGIA DA INFORMAÇÃO

A metodologia para desenvolvimento de aplicações computacionais, teve início na década de quarenta, entretanto, naquela época não existiam linguagens de programação, sendo as instruções gravadas diretamente ao nível de máquina. No início da década de cinquenta, com o surgimento das linguagens montadoras, houve um grande interesse na criação de ferramentas para auxiliar no desenvolvimento de aplicações computacionais.

Segundo Carmo (2011), o surgimento do paradigma de Programação Orientada a Objetos surgiu no início da década de 60 com a Linguagem Simula, seguindo na década de 70 com a Linguagem *Smalltalk* e na década de 80 com a Linguagem C++.

O paradigma de Programação Orientação a Objetos se baseia na decomposição e classificação dos problemas, utilizando abstrações chamadas classes, sendo cada instância de uma determinada abstração a um objeto. Por sua vez, cada objeto pode ser considerado como um objeto do mundo real, sendo definido por uma interface que possua métodos que podem ser aplicados sobre o objeto (RODRIGUES, 2004).

Para Farinelli (2007), a proposta da orientação a objetos é representar o mais fielmente possível as situações do mundo real nos sistemas computacionais.

Entendemos o mundo como um todo composto por vários objetos que interagem uns com os outros. Da mesma maneira, a orientação a objetos consiste em considerar os sistemas computacionais não como uma coleção estruturada de processos, mas sim como uma coleção de objetos que interagem entre si.

2.6 A LINGUAGEM DE PROGRAMAÇÃO

O estudo e conhecimento de uma determinada linguagem de programação é, antes de mais nada, essencial para o desenvolvimento de qualquer aplicação web. Neste contexto a linguagem PHP é um conjunto de códigos com os quais possui particularidades, vantagens, desvantagens, recursos e ambientes específicos para o desenvolvimento pretendido.

Criado em 1995 por Rasmus Lerdorf, o PHP é uma linguagem *Script, Open Source*, utilizada para o desenvolvimento de aplicações web, tendo seu código embutido dentro do HTML. O PHP além de ser multiplataforma, ou seja, roda nos mais diversos tipos de Sistemas Operacionais existente na atualidade, tais como, windows, linux e MacOS X, possuem sua sintaxe semelhante à linguagem C (BARNABÉ, 2010).

2.7 A PROGRAMAÇÃO ORIENTADA A OBJETOS – POO

Segundo Meyer (1997), o termo Programação Orientada a Objetos – POO, foi criado por Alan Kay, autor da linguagem de programação *Smalltalk*. Mesmo antes da criação do *Smalltalk*, algumas das ideias da POO já eram aplicadas, sendo que a primeira linguagem a realmente utilizar estas ideias foi a linguagem Simula 67, criada por Ole Johan Dahl e Kristen Nygaard em 1967.

A POO foi criada para tentar aproximar o mundo real do mundo virtual, tendo como ideia fundamental, simular o mundo real dentro do computador. Em contra partida, a motivação da necessidade de desenvolvimento de software de qualidade, para uma maior reutilização de códigos e otimizar os níveis de manutenção e assim aumentar a produtividade e apoio para mudanças de requisitos no desenvolvimento de software, (MEYER, 1997).

A programação orientada a objetos tem como diferencial a definição de suas estruturas no conceito de herança, no qual facilita a extensão das classes já existentes na aplicação. Também se pode enfatizar a importância do polimorfismo, que permite ao programa utilizar funcionalidades dinamicamente durante a execução, (MEYER, 1997).

Baseado em uma programação modular em classes, a programação orientada a objetos disponibiliza facilmente a manutenção do sistema e reutilização de código, o que favorece o desenvolvimento de sistemas mais eficientes e complexos, (MEYER, 1997).

Atualmente os softwares estão ficando maiores e mais complexos, e assim fica cada vez mais difícil a manipulação dos modelos representados, dificultando mensurar as modificações e reutilizações das partes modeladas devido à relação contida entre os mesmos, (MEYER, 1997).

Para Soares (2004), na programação orientada a objetos existe uma dificuldade de implementação, no qual é necessária a intersecção e chamada de um código em diferentes situações no desenvolvimento, no qual a programação orientada a aspectos tem como objetivo resolver algumas ineficiências da programação orientada a objetos, aumentando a modularidade do código totalmente separado, com a característica de afetar diversas partes do sistema, conhecidos como interesses transversais.

A complexidade de uma aplicação é mensurada através da fragmentação das suas funcionalidades, determinando o comportamento do sistema, sendo a parte conhecida por requisitos gerais que fazem parte da implementação da aplicação, como por exemplo, custo, performance, confiabilidade, manutenção, portabilidade, custos operacionais, entre outros. Estes comportamentos da aplicação são chamados de requisitos não funcionais ou interesses transversais (CYSNEIROS, 2001).

A programação orientada a objetos tornou o desenvolvimento de aplicações mais flexível e reutilizável, possibilitando maior agilidade e facilidade na manutenção, principalmente modularização das funcionalidades do sistema, onde é visualizada as propriedades do sistema como componentes. Apesar de suas vantagens, a programação orientada a objeto tem apresentada algumas limitações, às vezes sendo necessário lidar com códigos que representam diferentes

características, o qual é necessário chamar códigos amarrados em uma simples solicitação, como o tratamento de interesses transversais em novas funcionalidades (SOARES, 2004).

2.7.1 Classes

Laskoski (2013), considera uma classe como a descrição de um conjunto de dados estruturados que tem como característica propriedades em comuns, também interpretada como uma estrutura modular completa que descreve de forma estática e dinâmica as propriedades dos elementos manipulados pelo programa.

Através da definição de uma classe são descritos todos os atributos dos objetos, e além da especificação destes atributos, esta definição também especifica as funcionalidades aplicadas aos objetos da classe, descrita através de métodos (RICARTE, 2001).

2.7.2 Atributos

Os atributos dos objetos são comparados com “variáveis” ou “campos” que armazenam valores diferentes, disponibilizando as características que os objetos podem conter (FARINELLI, 2007).

A descrição das propriedades de uma classe é realizada por meio de um conjunto de atributos, sendo cada atributo identificado por um nome e associado por um tipo.

O tipo associado a um atributo, no paradigma de programação orientada, é o próprio nome de uma classe, mas a maior parte das linguagens baseadas a objetos disponibilizam um grupo de tipos primitivos para serem utilizados na descrição dos atributos, como inteiro, real e caractere (RICARTE, 2001).

As mudanças dos valores dos atributos de um objeto são alterados, somente através de estímulos externos ou internos e deverão ser manipulados exclusivamente por serviços ou métodos da própria classe. Assim, a única forma de modificar os valores associados aos atributos é através de disparos de eventos que estimulam a transição desses estados no objeto.

2.7.3 Métodos

Para Farinelli, (2007), as funcionalidades de uma classe são descritas através dos métodos, definindo qual será o comportamento dos objetos dessa classe.

Pode-se comparar os métodos como “procedimentos” ou “funções” que realizam as ações próprias do objeto. Todo o comportamento de uma classe é realizado através dos métodos definidos, sendo possível que o objeto se manifeste e realize interação com outros objetos através de estímulos destes métodos.

Cada método é especificado por uma assinatura, composta por um identificador para o método (o nome do método), o tipo para o valor de retorno e sua lista de argumentos, sendo cada argumento identificado por seu tipo e nome.

A comunicação entre objetos é realizada por estímulos, sendo requisitadas as ações através de mensagens entre eles, assim são executadas as rotinas responsáveis por acessar ou alterar os atributos dos objetos ou retornar algum comportamento do objeto.

2.7.4 Objetos

Laskoski, (2013), caracteriza o objeto como uma instância de classes, possibilitando determinar quais informações um objeto contém e como serão manipuladas estas informações.

Pode-se interpretar um objeto como uma entidade capaz de armazenar as informações e oferecer várias operações para analisar ou para afetar estas informações, (LASKOSKI, 2013).

A classe é um elemento reconhecido em um texto de programa, mas um objeto é um conceito puramente dinâmico, o qual pertence à memória do computador e não ao texto do programa, ocupando espaço durante a execução, (LASKOSKI, 2013).

O estado de um objeto é determinado quando um conjunto de valores é atribuído aos seus atributos durante a execução da aplicação, o comportamento do

mesmo determina como será a ação e reação durante suas mudanças de estado e troca de mensagens com outros objetos, (LASKOSKI, 2013).

Em linguagens de programação orientada a objetos, praticamente todo o processamento é realizado através dos objetos, atribuições de valores e comportamento na troca de mensagens entre eles, (FARINELLI, 2007).

2.7.5 Abstração

A definição da palavra abstração no dicionário da língua portuguesa Michaelis, encontra-se o seguinte significado “Consideração das qualidades independentemente dos objetos a que pertencem. Processo pelo qual se isolam atributos de um objeto, considerando os que certos grupos de objetos tenham em comum”.

Considera-se abstrair o ato de separar elementos relevantes de um todo. Pode-se considerar que através da abstração são mapeadas as entidades em objetos da aplicação, sendo o produto final de um programa a representação de algum aspecto do mundo real (CARMO, 2001).

A abstração também pelo princípio que cada componente deve ocultar informações, disponibilizado somente o necessário e revelado o mínimo necessário para a realização das operações (FARINELLI, 2007).

Através do conceito de classes e objetos é possível realizar esta abstração, em que a decisão pelo conjunto correto de abstração é um fator determinante entre uma boa ou má análise orientada a objetos. (CARMO, 2001)

Considera-se que o conceito de abstração, aplicado ao desenvolvimento de sistemas, a forma de representar somente o que será utilizado na aplicação, isolando os objetos que serão representados somente com as características relevantes. Para melhor entendimento, cita-se o seguinte exemplo:

Em um sistema de agenda telefônica, não tem relevância qual é a marca e modelo do aparelho utilizado, desta forma, esta informação não é inclusa na representação do objeto, mas em um sistema de gerenciamento de aplicativos para celular, esta informação passa a ser relevante, visto que um aplicativo não funciona

corretamente em determinado aparelho, sendo necessário incluir a representação da marca e modelo na construção do objeto, (CARMO, 2001).

O recurso da abstração pode ser utilizado como uma maneira de compreender problemas complexos, e diante deste problema, procurar dividi-lo em problemas menores, resolvendo individualmente cada um deles para solucionar o problema inteiro. Desta forma, também é possível determinar maior e menor peso para as partes do problema, o que possibilita desconsiderar detalhes em determinados momentos para que outros sejam ressaltados (FARINELLI, 2007).

2.7.6 Encapsulamento

A implementação de dados correlacionados em um objeto ou entidade é conhecida, no paradigma de programação a objetos, como encapsulamento. O objetivo é possibilitar a reutilização de sistemas desenvolvidos sem dependência da implementação interna, e quando utilizado por outras entidades, existirão somente as interfaces de acesso aos membros das classes, desconsiderando como tais classes foram implementadas, (CARMO, 2001).

Segundo Farinelli (2007, p. 17), “Encapsulamento é o princípio de projeto pelo qual cada componente de um programa deve agregar toda a informação relevante para sua manipulação como uma unidade (uma cápsula)”.

O uso do encapsulamento e abstração torna a programação orientada a objetos um mecanismo poderoso, e recomenda que a representação de um objeto deve ser mantida oculta. Desta forma, cada objeto é manipulado através dos métodos públicos, pois somente a assinatura do objeto é revelada.

Na interação entre dois objetos é importante que um tenha conhecimento do conjunto de operações disponíveis do outro, criando uma interface operacional para que então envie e receba informação.

Segundo Ricarte (2001), ao criar a interface operacional, detalhes do processamento da operação do objeto não são conhecidos por outros objetos, o que possibilita ao usuário do objeto trabalhar em um nível mais alto de abstração, para que assim não haja a preocupação com os detalhes internos da classe encapsulada.

Com isso, é possível simplificar a construção de aplicações mais complexas, tais como interfaces gráficas ou aplicações distribuídas.

Ao analisar a programação estruturada, o encapsulamento é uma grande vantagem da programação orientada a objetos, devido o mesmo disponibilizar todas as funcionalidades do objeto sem que o usuário precise ter conhecimento das funcionalidades internas e de como é armazenada as informações deste objeto, (RICARTE, 2001).

Para exemplificar pode-se analisar o envio de e-mails, onde simplesmente é redigido o e-mail e enviado, onde o usuário não sabe quais os procedimentos realizados pelo provedor para que este e-mail seja enviado ao remetente, (RICARTE, 2001).

No encapsulamento, outro ponto importante é a possibilidade de realizar modificações internas em um objeto para acrescentar novos métodos sem afetar os outros componentes utilizados por sistemas que utilizam este objeto (FARINELLI, 2007).

2.7.7 Herança

O que torna a orientação a objetos única é o conceito de herança. Herança é um mecanismo que permite que características comuns a diversas classes sejam fatoradas em uma classe base, ou superclasse. A partir de uma classe base, outras classes podem ser especificadas. Cada classe derivada ou subclasse apresenta as características (estrutura e métodos) da classe base e acrescenta a elas o que for definido de particularidade para ela, (RICARTE, 2001).

Segundo Carmo (2001, p. 16), “Relação entre classes, de modo a permitir o compartilhamento de atributos e serviços semelhantes. Com a herança pode-se definir novas classes por extensão, especialização ou combinação de outras já definidas”.

O conceito de herança pode ser exemplificado através da própria genética, onde um filho herda as características genéticas dos pais e repassa suas características para seus filhos.

Para Farinelli, (2007), a programação orientada a objetos, o conceito de herança possibilita que uma classe obtenha ou transmita os atributos e métodos de outra classe para expandir ou especializar estas características de alguma forma.

Pode-se dizer também que o conceito de herança é uma forma inteligente de reaproveitamento de código, devido a possibilidade de compartilhamento de atributos e métodos, (FARINELLI, 2007).

Desta forma, uma nova classe criada pode herdar os atributos e métodos de outra classe, implementando apenas as características específicas desta classe e reutilizando todas as características da classe principal, (FARINELLI, 2007).

A classe que obtém as informações também é conhecida como subclasse, já a classe que transmite suas características é conhecida como superclasse. Existem algumas formas de realizar os relacionamentos em herança, em que:

- Extensão: neste caso a superclasse é mantida inalterada e subclasse estende as características da superclasse, implementando novos atributos e/ou métodos.
- Especificação: a subclasse não implementa suas funcionalidades, sendo a superclasse que especifica o que será oferecido pela subclasse, através de um conjunto específico de métodos públicos.
- Extensão e Especificação: a mais comum de ocorrer ao utilizar a programação orientada a objetos. Também conhecida como herança polimórfica, ao herdar as características da superclasse, a subclasse implementa alguns métodos da superclasse, redefinindo estes métodos para especializar o comportamento, sendo que os outros métodos terão o mesmo comportamento da superclasse.

Além da economia de código, a herança oferece maior integridade na manutenção da aplicação, uma vez que ao se alterar o comportamento em uma superclasse, as subclasses ligadas também serão alteradas e utilizarão os métodos atualizados sem necessidade de reprogramação. (FARINELLI, 2007)

2.7.8 Polimorfismo

Polimorfismo é a capacidade de um objeto de se adaptar a diferentes tipos de dados, desta forma, a aplicação é capaz de diferenciar, dentre os métodos

homônimos, qual será executado, devido a identificação do objeto receptor, (CARMO, 2001).

Pode-se definir também como polimorfismo a possibilidade de uma variável ser referenciada a objetos de diversas classes, durante o tempo de execução da aplicação, ou defini-lo como a capacidade que objetos distintos têm de responder a uma mesma mensagem, (FARINELLI, 2007).

Outra interpretação do polimorfismo é quando duas ou mais subclasses podem invocar métodos que têm a mesma identificação, conhecida também como assinatura do método, mas o comportamento de cada subclasse é distinto, devido a especialização realizada, (RICARTE, 2001).

Fundamental na programação orientada a objetos, o polimorfismo possibilita definir comportamentos genéricos dos objetos, abstraindo detalhes particulares quando esses não forem necessários. Desta forma, uma mesma mensagem pode conter comportamentos diferentes durante a execução, próprias de cada objeto, (RICARTE, 2001).

Para a utilização do polimorfismo, é necessário que uma classe defina os métodos exatamente com a mesma assinatura do método definido na superclasse, utilizando o mecanismo de redefinição de métodos, conhecido também como *overriding*,. (RICARTE, 2001).

Não tem como definir, através do compilador, qual método será utilizado, caso o método seja definido em outras classes. Desta forma a decisão de qual método será utilizado é definido durante o tempo de execução da aplicação, realizado pelo mecanismo de ligação tardia, conhecido também como *late binding*, *dynamic binding* ou *run-time binding*, (RICARTE, 2001).

2.8 LINGUAGEM UNIFICADA DE MODELAGEM (UML)

Na área de Engenharia de Software, a Linguagem de Modelagem Unificada (do inglês, UML - Unified Modeling Language), é uma linguagem de modelagem que permite representar um sistema de forma padronizada.

A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida ou como projetar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre os objetos.

Nesse contexto, foi utilizado a UML para o desenvolvimento dos Casos de Uso e Diagrama de Classes do sistema, que serão abordados a seguir neste capítulo.

2.8.1 MODELO DE CASOS DE USO

Nesta seção, abordaremos a forma de construção do diagrama de casos de uso e da descrição desses casos de uso, assim como sua importância no contexto geral do desenvolvimento de um sistema computacional.

Segundo Stadzisz, (2002), o modelo de Casos de Uso foi proposto por I. Jacobson como um instrumento para descrição das intenções ou requisitos para um sistema computacional. A construção do Modelo de Casos de Uso corresponde a uma das fases iniciais de um projeto de *software* pois envolve a determinação dos usos que o sistema terá, ou seja, do que ele deverá fornecer como serviços.

O modelo de Casos de Uso é diferente da visão funcional utilizada no passado nas abordagens de projeto estruturado. Ao invés de focalizar as funções (atribuições técnicas) do sistema, o modelo de Casos de Uso captura os usos ou aplicações completas do sistema. Este modelo busca responder a questão: Que usos o sistema terá? ou Para que aplicações o sistema será empregado? (STADZISZ, 2002)

Os modelos de Casos de Uso são descritos através de Diagramas de Casos de Uso na UML. De uma forma geral, cada projeto de *software* conterá um Diagrama de Casos de Uso, (STADZISZ, 2002).

Para sistemas mais extensos, é possível decompor o diagrama em um conjunto de sub-diagramas, (STADZISZ, 2002).

Uma vez construído o modelo de Casos de Uso, o restante do projeto pode ser guiado baseando-se neste modelo. Dito de outra forma, a partir do modelo de Casos de Uso, o restante do projeto irá se preocupar com a forma de realização dos casos de uso (que classes e objetos são necessários, como e quando cada um atuará, etc), (STADZISZ, 2002).

O modelo de Casos de Uso é um instrumento eficiente para determinação e documentação dos serviços a serem desempenhados pelo sistema. Ele é também um bom meio para comunicação com os clientes no processo de definição dos requisitos do sistema, (STADZISZ, 2002).

2.8.1.1 Diagramas de casos de uso

Os casos de uso de um projeto de *software* são descritos na linguagem UML através de Diagramas de Casos de Uso. Estes diagramas utilizam como primitivas Atores, Casos de Uso e Relacionamentos. Como ocorre também com outros diagramas, pode-se ainda utilizar as primitivas Pacote e Nota nos diagramas de casos de uso. As subseções a seguir descrevem estas primitivas, (STADZISZ, 2002).

2.8.1.2 Atores

Atores são representações de entidades externas, mas que interagem com o sistema durante sua execução. Basicamente, a interação de atores com o sistema se dá através de comunicações (troca de mensagens). As entidades externas representadas pelos atores podem ser:

- Pessoas: usuário, secretaria, aluno, professor, administrador, etc;
- Dispositivos: impressora, máquina ou outro equipamento externo;
- *Hardware*s: placa de modem, placa de controle, etc;
- *Software*: sistema de banco de dados, aplicativos, etc.

É importante observar que atores representam, na verdade, papéis desempenhados por pessoas, dispositivos ou outros *softwares* quando estiverem interagindo com o sistema. Por exemplo, um ator cujo identificador seja Aluno não representa um aluno específico, mas sim um aluno qualquer, ou seja, uma pessoa qualquer que esteja interagindo com o sistema na qualidade de aluno. Desta forma, um ator pode representar um entre vários indivíduos, equipamentos ou *softwares*. De forma análoga, uma entidade externa pode ser representada na forma de vários atores. Isto ocorre quando a entidade tem mais de um papel (tipo de participação ou interação) no sistema. Por exemplo, o indivíduo João da Silva poderia ser

representado em um sistema na forma do ator Usuário, pois ele interage com o sistema nesta qualidade, e também na forma do ator Administrador, pois ele também interage com o sistema para este outro fim que é a administração do *software*, (STADZISZ, 2002).

Atores são representados através de retângulos (notação genérica para classe) usando a simbologia padrão da UML ou através de ícones humanos. As duas notações são sintaticamente equivalentes, porém a segunda é seguramente mais intuitiva. A desvantagem do uso de ícones humanos ocorre quando o ator representa equipamentos, dispositivos de *hardware* ou outros *softwares*. Neste caso, a figura humana não coincide com a natureza do ator. É possível, entretanto, através de mecanismos de extensão, criar grafismos especiais ou especializados na UML para indicar tipos de atores. (STADZISZ, 2002)

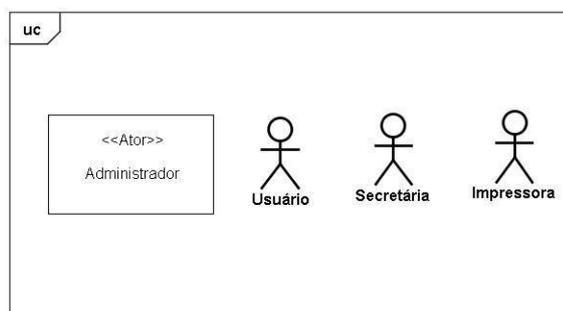


Figura 1. Exemplos de Atores

Fonte: Stadzisz, (2002)

Observe que a notação usando retângulos exige a inserção de um classificador para indicar a natureza daquilo que se está representando. No caso de atores, deve-se incluir o classificador (ou estereótipo) <<ator>> antes do nome do ator. Quando se utiliza o ícone humano, basta indicar o nome do ator abaixo do ícone, (STADZISZ, 2002).

O levantamento dos atores que interagem com um certo sistema depende de um trabalho de estudo que envolve discussões com os clientes. Procura-se neste estudo levantar as pessoas que serão beneficiadas e que usarão o sistema. Além disso, deve-se levantar os dispositivos e *softwares* com os quais o sistema devera se comunicar. Para muitos projetos, pode não ser fácil levantar corretamente o conjunto de atores na primeira tentativa. O estudo dos casos de uso e dos

relacionamentos com atores pode permitir refinar o conjunto de atores definidos. O estudo das classes do sistema, a ser feito na próxima fase, também irá contribuir para o refinamento do levantamento de atores, (STADZISZ, 2002).

Embora a UML não imponha restrições, costuma-se considerar determinados atores como atores implícitos. Desta forma estes atores não aparecem no diagrama de casos de uso embora eles estejam presentes e participem da execução dos casos de uso. Os atores implícitos representam essencialmente dispositivos e *softwares* que são sempre usados e que não impõem protocolos especiais de comunicação. Desta forma, a supressão destes atores não traz nenhum efeito significativo sobre os modelos e simplifica as representações. Os exemplos mais comuns de atores que se pode considerar como implícitos são: monitor de vídeo, teclado, mouse, autofalante, microfone, unidade de disco e sistema operacional. Estas entidades serão atores legítimos, mas cuja inclusão no modelo de casos de uso não traz contribuição para a modelagem, (STADZISZ, 2002).

2.8.1.3 Casos de uso

A descrição dos serviços a serem oferecidos pelo sistema é feita na UML discriminando-se os Casos de Usos do sistema. Cada Caso de Uso descreve uma aplicação ou uso completo do *software*. O conceito de caso de uso não deve ser confundido com o de módulo, já que um caso de uso não é um componente do sistema, mas sim um de seus empregos possíveis. Também não se deve confundir o conceito de caso de uso com o de função que possui um escopo muito mais limitado, traduzindo frequentemente apenas um recurso ou utilidade do sistema. Um caso de uso é muito mais abrangente, envolvendo todo um conjunto de transações que juntas constituem um serviço completo oferecido pelo sistema, (STADZISZ, 2002).

A especificação das funcionalidades de um sistema na forma de casos de uso permite uma visão mais abrangente das aplicações do sistema favorecendo um levantamento mais completo e preciso de suas atribuições, (STADZISZ, 2002).

2.8.1.4 Relacionamentos entre atores

Como os atores são entidades externas, os relacionamentos entre eles serão relações externas ao sistema. Embora estas relações possam ser desprezadas, pois não fazem parte do sistema e nem são de conhecimento do sistema, é possível inclui-las no modelo de casos de uso. De certa forma, estas relações descrevem parte do modelo de negócios da empresa, (STADZISZ, 2002).

As duas relações mais comuns entre atores são a comunicação (ou associação) e a especialização (ou generalização). Um relacionamento de comunicação indica que os dois atores, de forma uni ou bidirecional, realizam uma comunicação (troca de informação ou de mensagem) que possui um significado formal para o sistema modelado. No exemplo da Figura 1, o ator Aluno comunica-se com o ator Secretaria no sentido de transmitir uma Solicitação de Histórico Escolar. Trata-se de uma comunicação explícita cuja ocorrência deve ter alguma repercussão ou significado para o sistema. Um relacionamento de generalização, por outro lado, representa uma relação conceitual entre atores indicando que um ator é um caso especial de outro ator mais genérico. Esta relação indica que o ator especializado inclui todos os atributos do ator genérico e inclui ainda atributos adicionais. Como ilustra a Figura 2, o ator Administrador é um ator especializado do ator Usuário. Isto significa que o Administrador é um Usuário com atributos ou características extras. De certa forma, o ator especializado estende o ator genérico, (STADZISZ, 2002).

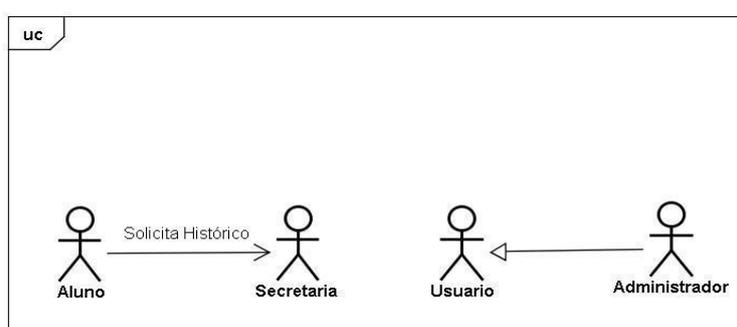


Figura 2. Relacionamento entre Atores

Fonte: Stadzisz, (2002)

2.8.1.5 Relacionamentos entre atores e casos de uso

O relacionamento entre um ator e um caso de uso expressa sempre uma comunicação entre eles, pois o ator sendo uma entidade externa não poderia

possuir qualquer relacionamento estrutural com o sistema computacional. A notação UML para este tipo de relacionamento é um segmento de reta ligando ator e caso de uso, como ilustrado na Figura 3, (STADZISZ, 2002).

Em diagramas mais complexos pode-se utilizar cadeias de segmentos de retas para se evitar cruzamentos. Como atores podem ter vários propósitos, no que diz respeito a suas interações com o sistema, podem existir mais de um relacionamento de um ator com diferentes casos de usos. De forma análoga, um mesmo caso de uso pode envolver a participação de mais de um ator. A figura 3 ilustra estas situações. O caso de uso Emitir Histórico Escolar envolve a participação de dois atores: Secretaria e Impressora. O ator Secretaria participa em dois casos de uso: Emitir Histórico e Registrar Novo Aluno.

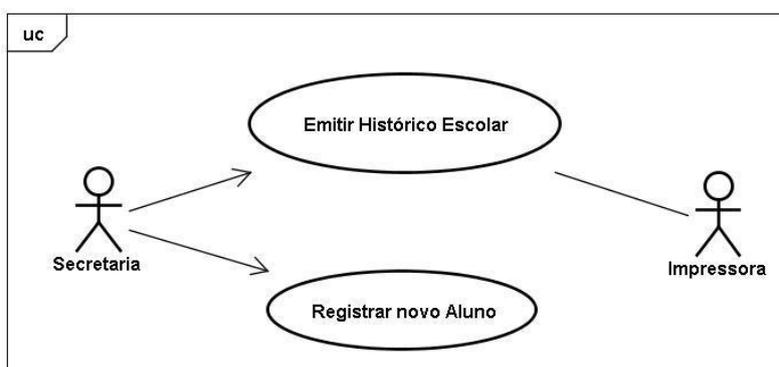


Figura 3. Exemplo de Relações entre Atores e Casos de Uso
Fonte: Stadzisz, (2002)

2.8.1.6 Relacionamentos entre casos de uso

Ao contrário do relacionamento entre ator e caso de uso, as relações entre casos de uso nunca serão do tipo comunicação. Isto ocorre porque casos de uso são aplicações completas do sistema e, por consequência, não existe sentido em fazer-se comunicar dois usos do sistema. Todas as relações entre casos de uso serão sempre estruturais. Existem três tipos de relações entre casos de uso (inclusão, extensão e generalização) conforme descritos a seguir, (STADZISZ, 2002).

2.8.1.7 Relacionamento de inclusão

Um relacionamento de inclusão é uma relação estrutural através da qual um caso de uso insere em seu interior um outro caso de uso. O caso de uso incluído (subcaso de uso) não representa um serviço completo do sistema, mas uma porção de um serviço. Isoladamente, um subcaso de uso não teria sentido. Ele será sempre um integrante de um caso de uso maior e completo, (STADZISZ, 2002).

O relacionamento de inclusão se aplica a duas situações principais. A primeira aplicação da Inclusão é para o detalhamento de casos de uso através de decomposição. Nesta situação, extraem-se partes significativas de um caso de uso criando-se subcasos de uso ligados ao caso de uso maior através de relações de inclusão. Embora raro, é possível ainda decompor subcasos de uso em outros subcasos de uso. Uma segunda aplicação do relacionamento de inclusão é a de colocar em evidência partes comuns a dois ou mais casos de uso. Nesta situação o subcaso de uso é incluído por mais de um caso de uso maior. A Figura 4 ilustra estas duas situações, (STADZISZ, 2002).

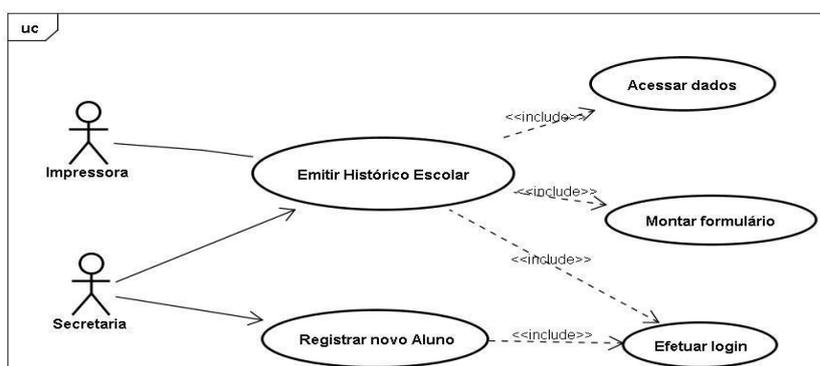


Figura 4. Exemplo de Relações de inclusão entre Casos de Uso

Fonte: Stadzisz, (2002)

2.8.1.8 Relacionamento de extensão

Um relacionamento de extensão é uma relação estrutural entre dois casos de uso através da qual um caso de uso maior é estendido por um caso de uso menor. A extensão significa que o caso de uso que estende inclui serviços especiais em um caso de uso maior. A definição de um relacionamento de extensão inclui a especificação de uma condição de extensão. Esta condição habilita a extensão, ou seja, indica quando aplicar o relacionamento. A notação para o relacionamento de extensão é ilustrada na Figura 5, (STADZISZ, 2002).

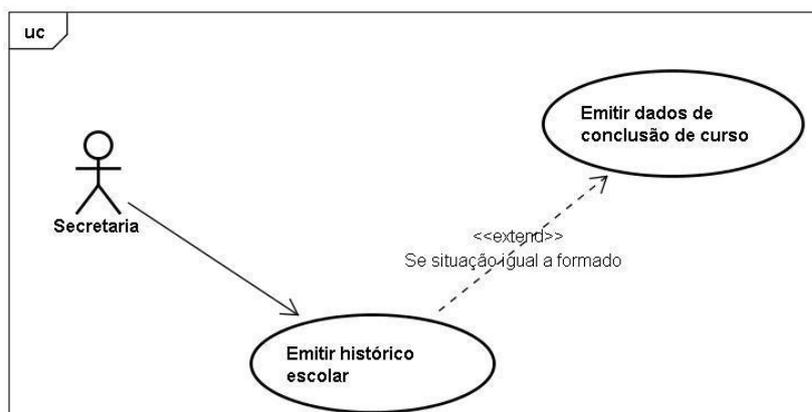


Figura 5. Exemplo de Relações de extensão entre Casos de Uso
Fonte: Stadzisz, (2002)

A diferença principal entre os relacionamentos de inclusão e extensão é o caráter de excepcionalidade da extensão. Extensões são usadas para modelar casos especiais e de exceção que ocorrem somente em certas situações (dado pela condição). Desta forma, para a maioria das ocorrências do caso de uso estendido, a extensão não se aplicará, (STADZISZ, 2002).

2.8.1.9 Relacionamento de generalização

Um relacionamento de generalização é uma relação estrutural entre um caso de uso mais geral e um caso de uso mais específico. O caso de uso mais geral representa o caso genérico cujo serviço se aplica a várias situações. O caso de uso mais específico representa a aplicação do caso de uso mais geral em uma situação particular incluindo elementos adicionais ou estendendo este caso. Visto no outro sentido, o caso de uso mais geral é uma generalização (abstração) do ou dos casos de uso mais específicos. Neste sentido, o caso de uso geral, representa as partes comuns de casos de uso especializados, (STADZISZ, 2002).

A notação UML para a generalização envolve a ligação dos dois casos de uso através de um segmento de reta e a colocação de um triângulo na extremidade do caso de uso mais geral, (STADZISZ, 2002).

A Figura 6 apresenta um exemplo de relacionamento de generalização.

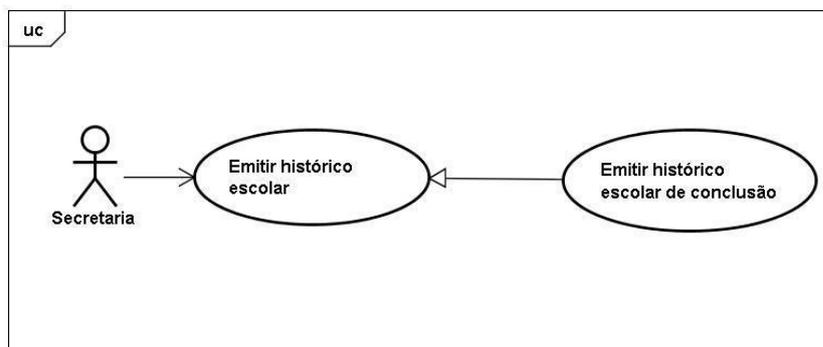


Figura 6. Exemplo de Relações de generalização entre Casos de Uso

Fonte: Stadzisz, (2002)

No exemplo da Figura 6, tem-se um caso de uso mais geral, chamado Emitir Histórico Escolar, que permite a secretaria imprimir históricos de alunos. Quando o aluno conclui na totalidade o curso, ele pode solicitar um histórico de conclusão. Este histórico é semelhante ao histórico normal, mas é mais detalhado incluindo informações adicionais. O caso de uso Emitir Histórico Escolar de Conclusão é, portanto, semelhante ao caso de uso Emitir Histórico Escolar, mas com alguns elementos adicionais. Pode-se, como feito no exemplo, estabelecer uma relação de especialização entre os dois casos de uso, (STADZISZ, 2002).

A relação estrutural definida por um relacionamento de generalização implica a incorporação (herança) dentro do caso de uso especializado de todo o serviço especificado pelo caso de uso geral, incluindo, adaptando ou excluindo alguns serviços oferecidos pelo caso de uso geral, (STADZISZ, 2002).

O relacionamento de generalização não pode ser confundido com os de inclusão e extensão pois a generalização se aplica, na maior parte dos casos, a compartilhamentos de maior dimensão. A inclusão e extensão envolvem partes menores de casos de usos. A natureza da generalização também é distinta pois trata-se de especificar modelos (casos de uso) genéricos aplicáveis a diferentes situações. A inclusão e a extensão apenas põe em evidência partes de casos de uso maiores, (STADZISZ, 2002).

2.8.1.10 Decomposição de diagramas de casos de uso

Um projeto de *software*, normalmente, conterà um único Diagrama de Casos de Uso descrevendo o conjunto de serviços oferecidos pelo sistema. Para sistemas maiores ou mais complexos, entretanto, é possível a construção de vários diagramas de casos de uso elaborados a partir da decomposição de um diagrama principal, (STADZISZ, 2002).

A decomposição de um diagrama de casos de uso pode ser feita em UML utilizando o conceito de Pacote (*package*). Um pacote é um encapsulador que não possui uma semântica específica dentro dos projetos. Ele é usado essencialmente para separar ou agrupar elementos do projeto sem criar necessariamente com isso algum vínculo entre os elementos, (STADZISZ, 2002).

Utilizando pacotes pode-se criar um primeiro diagrama contendo todos os pacotes maiores do sistema e, a seguir, tomar cada pacote e expandi-lo em um novo diagrama. Pode-se construir uma hierarquia com vários níveis de decomposição conforme a dimensão do sistema e o número de casos de uso e atores, (STADZISZ, 2002).

Os elementos (casos de uso e atores) dentro de um pacote podem ter relacionamentos com elementos de outros pacotes. Neste caso existe uma dependência entre pacotes. As dependências devem ser explicitamente definidas utilizando como notação um segmento de reta tracejado com uma seta no sentido do pacote que depende para o pacote que contém as dependências. A Figura 7 ilustra a notação utilizada para pacotes e dependências, (STADZISZ, 2002).

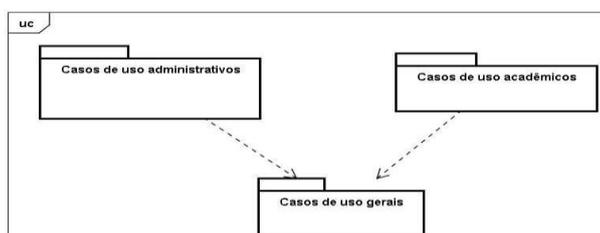


Figura 7. Exemplo de Pacotes e dependências

Fonte: Stadzisz, (2002)

Não existe uma norma para separação dos casos de uso e atores em pacotes. Pode-se, por exemplo, agrupar dentro de um pacote casos de uso de naturezas semelhantes (casos de uso de cadastro, casos de uso de emissão de relatórios, etc) ou casos de uso envolvendo os mesmos atores. De forma geral,

procura-se reunir casos de uso que possuem relacionamentos em um mesmo pacote. Quando um ator ou caso de uso tiver que aparecer em mais de um pacote, define-se este ator ou caso de uso em um pacote e copia-se o ator ou caso de uso nos demais pacotes. Neste caso, deve-se indicar nos demais pacotes qual o pacote de origem daquele ator ou caso de uso, (STADZISZ, 2002).

Concluindo, o modelo de casos de uso é uma ferramenta útil na descrição dos requisitos funcionais de um sistema computacional. Ele permite especificar o conjunto de funcionalidades ou serviços que um software deve oferecer e as relações do sistema com entidades externas (atores) necessárias para a realização destes serviços, (STADZISZ, 2002).

A notação UML para diagramas de casos de uso é em grande parte intuitiva permitindo que os modelos gerados possam ser apresentados aos clientes para discussões e revisões, (STADZISZ, 2002).

Deve-se sempre observar que o modelo de casos de uso é diferente da visão funcional no sentido que ele não apresenta encadeamentos de funções (por consequência, não descreve processos) e se limita a uma visão macroscópica dos serviços do sistema sem induzir a forma de realização (projeto) do *software*. O modelo de casos de uso oferece uma visão mais abstrata das funcionalidades do sistema favorizando um trabalho de especificação mais abrangente, (STADZISZ, 2002).

Por fim, o modelo de casos de uso pode também ser útil como ferramenta para planejamento do desenvolvimento de sistemas computacionais (estimativas por caso de uso) e como base para o desenvolvimento de projetos de *software* (projeto baseado em casos de uso), (STADZISZ, 2002).

2.9 FERRAMENTAS CASE

Do inglês, *Computer-Aided Software Engineering*, as chamadas ferramentas CASE são aquelas baseadas em computadores (softwares) utilizadas na Engenharia de Software para auxílio nas atividades desde análise de requisitos até modelagem de dados, (RODRIGUES, 2015).

No contexto desse artigo, as ferramentas CASE, permitem a criação de diagramas de forma simples em um ambiente de fácil utilização e com recursos para

incluir as principais regras de composição dos diagramas. Exemplos comuns desse tipo de ferramenta são: Star UML, *Astah Community* e *ERwin data modeler*.

A Figura 8 ilustra um exemplo de diagrama sendo construído na ferramenta Astah.

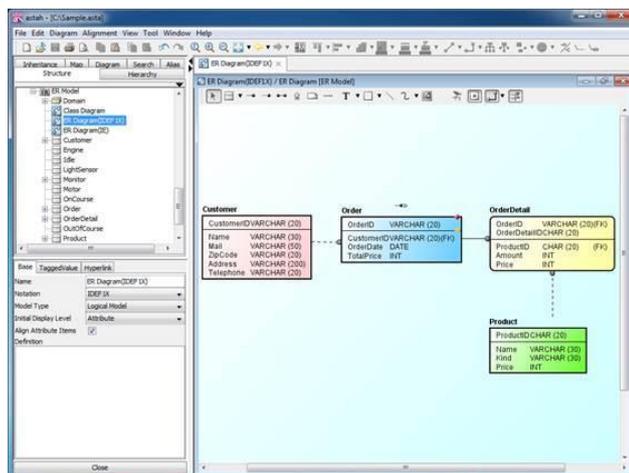


Figura 8. Diagrama no Astah Community

Fonte: Rodrigues (2015)

Além dessas ferramentas específicas, alguns IDEs (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado), como o *Visual Studio*, e ferramentas de gerenciamento de bancos de dados, como *SQL Server Management Studio*, possuem funcionalidades para criar diagramas facilmente e gerar o código equivalente (SQL para criação das tabelas, chaves e relacionamentos, por exemplo), (RODRIGUES, 2015).

Exemplo prático: Para fixar tudo que foi visto, vamos agora desenvolver um pequeno exemplo prático em que modelaremos um sistema de bibliotecas, focando especificamente no empréstimo de livros, (RODRIGUES, 2015).

Primeiramente precisamos identificar as entidades envolvidas nesse contexto. Sabemos que as entidades físicas existentes são o *Usuário* da biblioteca e o *Livro* que será emprestado. Além disso, consideraremos aqui que o livro pertence a uma *Sessão*, que ajuda na organização das obras do acervo. Em um sistema real pode haver outras informações sobre o livro, mas para esse exemplo a sessão é o bastante. Por fim, temos a entidade lógica *Empréstimo*, que tanto está relacionada com o usuário, quanto com o livro, (RODRIGUES, 2015).

Assim já podemos esboçar nosso primeiro diagrama, simples, contendo as principais entidades e o relacionamento entre elas (Figura 9).

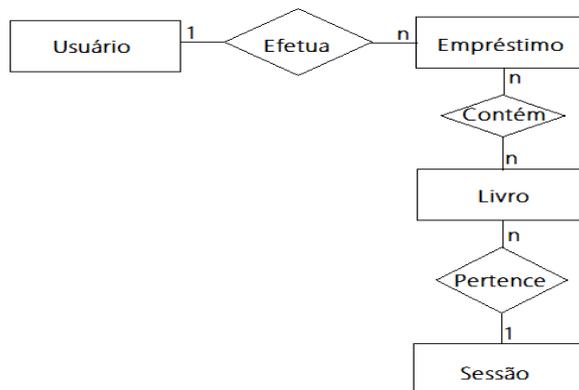


Figura 9. Primeiro DER de um sistema para biblioteca

Fonte: Rodrigues (2015)

Neste primeiro diagrama podemos identificar alguns dos conceitos vistos:

- Entidades fortes: Usuário, Livro e Sessão;
- Entidades fracas: Empréstimo;
- Relacionamentos: um Usuário *efetua* vários Empréstimos, vários Empréstimos *contêm* vários Livros, vários Livros *pertencem* a uma Sessão.

Agora que visualizamos o domínio no diagrama, podemos adicionar os atributos e outras entidades que se façam necessárias. Assim, passamos à Figura 10.

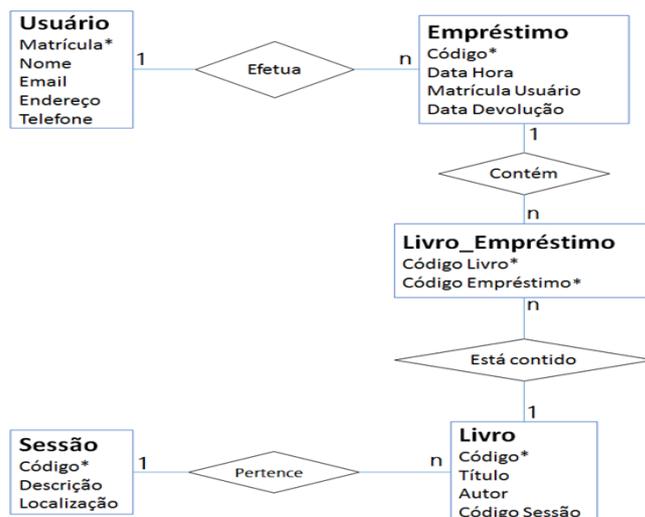


Figura 10. DER mais completo do sistema para bibliotecas

Fonte: Rodrigues (2015)

Neste ponto cabe fazer algumas observações importantes:

Especificamos os atributos de cada entidade e marcamos algumas delas com um asterisco, indicando que aquela é a chave primária da tabela, ou seja, um atributo único, que nunca poderá se repetir entre as entidades do mesmo tipo. Note que neste momento ainda não é necessário especificar o tipo de cada atributo (texto, número, data, etc.), isso só será necessário mais adiante, quando já estivermos planejando o banco de dados da aplicação, (RODRIGUES, 2015).

Surgiu a entidade associativa Livro_Empréstimo, que representa os livros contidos em um empréstimo (considerando um empréstimo contém vários livros e um livro pode estar contido em vários empréstimos). Esta entidade é composta pelas chaves das duas entidades principais. Se fosse necessário, nesta entidade também poderíamos adicionar informações complementares como quantidade (não se aplica neste caso, mas caberia em um sistema de vendas, por exemplo) e observações sobre o item. (RODRIGUES, 2015).

Na entidade associativa, o relacionamento n..n foi dividido em dois relacionamentos do tipo 1..n, agora lidos da seguinte forma: um empréstimo contém vários itens, mas um item só pode estar contido em um único empréstimo (restrito pelas chaves primárias); um livro pode estar contido em vários itens de empréstimo (ser emprestado várias vezes), mas cada item refere-se a um único livro, (RODRIGUES, 2015).

Concluindo, o Modelo Entidade Relacionamento (e principalmente o diagrama) é uma importante ferramenta durante o desenvolvimento de sistemas, principalmente aqueles mais complexos e difíceis de visualizar sem uma análise mais aprofundada.

A correta modelagem auxilia no correto desenvolvimento da base de dados e evita que várias alterações sejam necessárias para corrigir erros de concepção provenientes de falhas durante a análise, ou ainda por problemas de comunicação entre os membros da equipe, (RODRIGUES, 2015).

2.10 DIAGRAMA DE CLASSES

Segundo Guedes (2008), O Diagrama de Classes é, com certeza, o mais importante e o mais utilizado da UML. Seu principal enfoque está em permitir a

visualização das classes que comporão o sistema com seus respectivos atributos e métodos, bem como em demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si. Esse diagrama apresenta uma visão estática de como as classes estão organizadas, preocupando-se em como definir a estrutura lógica das mesmas. Este diagrama serve ainda como base para a construção da maioria dos outros diagramas da linguagem UML.

Basicamente, o Diagrama de Classes é composto por suas classes e pelas associações existentes entre elas, ou seja, os relacionamentos entre as classes. Nesse diagrama, a abstração de cada classe com seus atributos e métodos individualmente corresponde à Modelagem de Vocabulário, em que são definidas as classes que farão parte do diagrama, enquanto que a definição de como as classes se relacionam e colaboram para a execução de um determinado processo corresponde à Modelagem de Colaboração deste diagrama. Um Modelo de Colaboração normalmente não enfoca os relacionamentos entre todas as classes abstraídas no projeto, apenas as que colaboram de alguma maneira para a execução de um processo específico, (GUEDES, 2008).

2.10.1 Persistência

Em muitos casos pode ser necessário preservar de maneira permanente os objetos de uma Classe, ou seja, esta deve ser persistente. Uma Classe persistente apresenta muitas semelhanças com uma entidade como aquelas definidas no antigo Modelo Entidade-Relacionamento, utilizado para definir as estruturas de tabelas em bancos de dados relacionais. Na verdade, o Diagrama de Classes foi intencionalmente projetado para ser uma evolução do Modelo Entidade-Relacionamento e pode ser utilizado para modelar a estrutura lógica das tabelas que irão compor um banco de dados. Nas entidades que compunham o Modelo Entidade-Relacionamento, no entanto, eram definidos apenas os dados a serem preservados, os quais são representados nas classes pelos seus atributos, porém o Diagrama de Classes oferece ainda a possibilidade de definir as operações que podem ser aplicadas às tabelas, representadas pelos seus métodos, (GUEDES, 2008).

Deve ficar claro, no entanto, que nem toda classe é persistente, sendo

muitas vezes desnecessário preservar suas informações, de modo que é preciso identificar quais classes, se existirem, são persistentes, o que é feito por meio do emprego de estereótipos ou de restrições) que serão mais bem-detalhados no decorrer deste capítulo. Ao identificar uma classe como persistente, o modelador estará deixando explícito que é preciso preservar em disco, de algum modo, as instâncias da classe.

A forma como as instâncias serão preservadas irá depender da forma como o sistema for implementado, podendo estas serem atualizadas em disco, à medida que o sistema for sendo manipulado, ou mantidas em memória, por meio de uma lista encadeada, por exemplo, e registradas no disco apenas quando do encerramento do sistema ou da chamada de um método específico para isso. Muitos autores recomendam o uso de uma Camada de Persistência, ou seja, a derivação de uma subclasse a partir da classe cujas informações necessitam ser mantidas, quando seria definida a maneira como as instâncias de cada classe seriam preservadas.

Deve ficar claro também que, embora o Diagrama de Classes possa ser utilizado para modelar o projeto lógico de um banco de dados, ele não serve exclusivamente para isso, sendo comum seu uso para modelar classes não-persistentes, (GUEDES, 2008).

2.10.2 Classes, Atributos e Métodos

Classes costumam ter atributos, que armazenam os dados dos objetos da classe, além de métodos, que são as funções que uma instância da classe pode executar. Os valores dos atributos podem variar de uma instância para outra. Graças a essa característica, aliás, é possível identificar cada objeto individualmente, ao passo que os métodos são idênticos para todas as instâncias de uma classe específica.

Embora os métodos sejam declarados no Diagrama de Classes, identificando os possíveis parâmetros que são por eles recebidos e os possíveis valores por eles retomados, o Diagrama de Classes não se preocupa em definir as etapas que tais métodos deverão percorrer quando forem chamados, sendo esta uma função atribuída a outros diagramas, como os de Sequência ou Atividades,

(GUEDES, 2008).

2.10.3 Relacionamentos

As classes costumam ter relacionamentos entre si, com o objetivo de compartilhar informações e colaborarem umas com as outras para permitir a execução dos diversos processos executados pelo sistema. Nas seções seguintes estudaremos as diversas formas de relacionamento possíveis em um diagrama de classes, (GUEDES, 2008).

2.10.4 Associações

Uma associação descreve um vínculo que ocorre normalmente entre duas classes, chamado, neste caso, de associação binária, mas é perfeitamente válido que uma classe esteja vinculada a si mesma, caso conhecido como associação unária, ou que uma mesma associação seja compartilhada por várias classes, o que é conhecido como associação ternária ou N-ária, embora esta última seja o tipo mais raro e complexo, (GUEDES, 2008).

- Agregação

Agregação é um tipo especial de associação na qual se tenta demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte). Este tipo de associação tenta demonstrar uma relação Todo/Parte entre os objetos associados. Objetos-parte não podem ser destruídos por um objeto diferente do objeto-rod. O símbolo de agregação difere do de associação por conter um losango na extremidade da classe que contém os objetos-todo, (GUEDES, 2008).

A Figura 11 demonstra um exemplo de agregação, no qual há uma classe Pedido, que armazena os objetos-todo e uma classe Itens Pedido, onde são armazenados os objetos-parte.

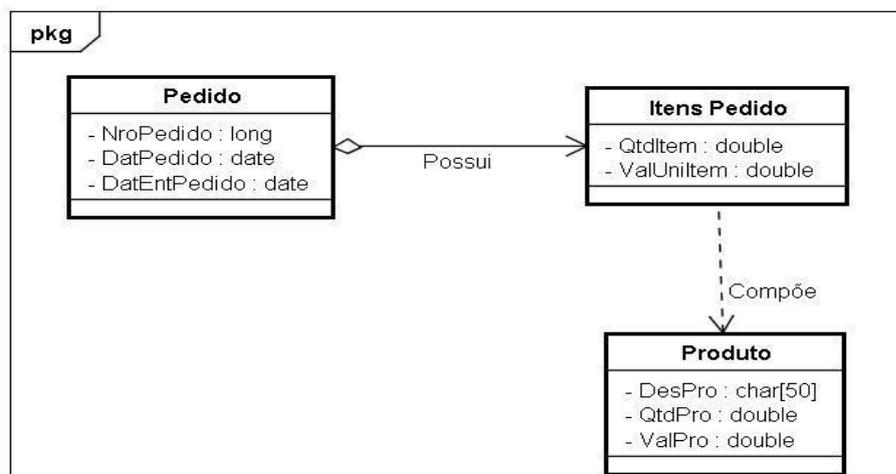


Figura 11. Exemplo de agregação

Fonte: GUEDES (2008)

Neste exemplo, a agregação existe apenas entre a classe Pedido e a classe Itens Pedido. A classe Produto possui apenas uma associação binária comum com a classe itens Pedido, incluída no exemplo acima, para torná-lo mais compreensível e completo, (GUEDES, 2008).

- Composição

Uma associação do tipo Composição constitui-se em uma variação da associação de agregação. Uma associação de composição tenta representar um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que os objetos-parte têm de pertencer exclusivamente a um único objeto-todo com que se relacionam. Em uma composição, um mesmo objeto-parte não pode se associar a mais de um objeto-todo, (GUEDES, 2008).

O símbolo de composição diferencia-se graficamente do símbolo de agregação por utilizar um losango preenchido. Da mesma forma que na agregação losango deve ficar ao lado do objeto-todo.

A Figura 12 apresenta um exemplo de composição.

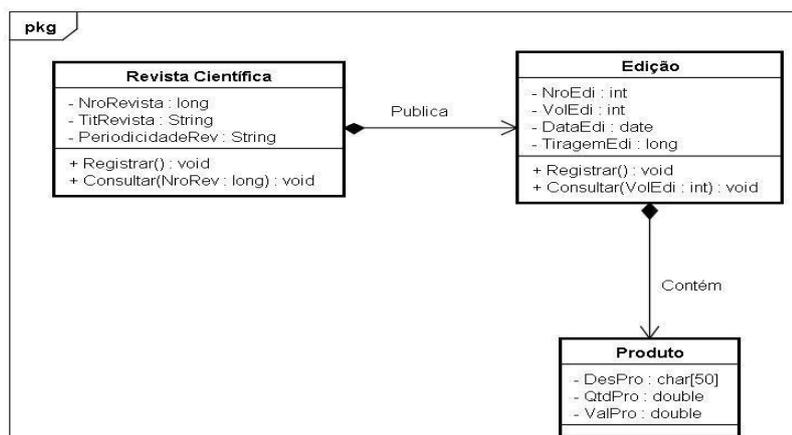


Figura 12. Exemplo de composição

Fonte: GUEDES (2008)

- **Especialização/Generalização**

Este é um tipo especial de relacionamento muito similar à associação de mesmo nome utilizado no Diagrama de Casos de Uso. Seu objetivo é identificar classes-mãe, chamadas gerais e classes-filhas, chamadas especializadas. Tal tipo de relacionamento permite também demonstrar a ocorrência de métodos polimórficos nas classes especializadas do sistema, (GUEDES, 2008).

Como no Diagrama de Casos de Uso, a especialização/generalização ocorre quando existem duas ou mais classes com características muito semelhantes. Assim, para evitar ter de declarar atributos e/ou métodos idênticos e como uma forma de reaproveitar código, cria-se uma classe geral em que são declarados os atributos e métodos comuns a todas as classes envolvidas no processo e, então, declaram-se classes especializadas ligadas à classe geral, que herdam todas as suas características, podendo ter atributos e métodos próprios, (GUEDES, 2008).

Além disso, métodos podem ser redeclarados em uma classe especializada, com o mesmo nome, mas comportando-se de forma diferente, não sendo, portanto, necessário modificar o código-fonte do sistema com relação às chamadas de métodos das classes especializadas, pois o nome do método não mudou, somente foi redeclarado em uma classe especializada e só se comporta de maneira diferente quando for chamado por objetos dessa classe. A Figura 13 apresenta um exemplo de especialização/generalização no Diagrama de Classes. O símbolo de Especialização/Generalização é o mesmo do Diagrama de Casos de Uso, (GUEDES, 2008).

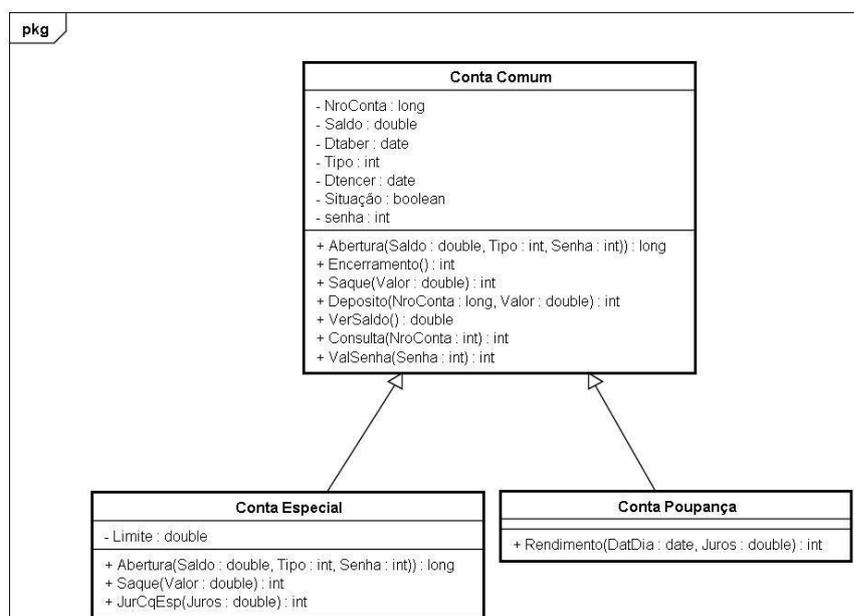


Figura 13. Exemplo de especialização/generalização

Fonte: GUEDES (2008)

Ao examinarmos a Figura 13, verificamos a existência de uma classe geral chamada Conta Comum que possui os atributos número da conta, saldo da conta, data de abertura da mesma, tipo da conta (se é comum, especial ou poupança, um atributo que será herdado), possível data de encerramento da conta, situação da conta (se está ativa ou não) e senha da conta. A classe Conta Comum possui ainda os seguintes métodos:

- Abertura - Para abrir uma conta, que neste caso atua como um método construtor.
- Encerramento - Para encerrar uma conta já existente, que não é um método destrutor, pois esse método só altera a situação da conta e possivelmente do cliente para inativa.
 - Saque - Para retirar valores da conta
 - Depósito - Para inserir valores na conta
 - Versaldo - para verificar quanto está depositado na conta
 - Consulta - Para verificar se uma determinada conta existe.
 - Valsenha - Para determinar se as senhas informadas são válidas.

Há, ainda, duas classes especializadas que herdam as características da classe geral, que são as classes Conta Especial e Conta-Poupança. Além dos

atributos e métodos herdados, a classe Conta Especial possui ainda o atributo Limite que determina quanto o cliente pode sacar além de seu saldo e os métodos Abertura, Saque e JurCheqEsp. Os dois primeiros métodos são uma redeclaração dos métodos Abertura e Saque da classe Conta Comum, pois estes precisam incluir o Limite da conta. O último método calcula o valor de juros a ser cobrado pelo uso do limite, (GUEDES, 2008).

Já a classe Conta-Poupança possui, além dos atributos e métodos herdados da classe Conta Comum), o método Rendimento, cuja função é calcular os rendimentos a serem acrescidos ao saldo da conta sempre que esta fizer aniversário, (GUEDES, 2008).

- Dependência

O relacionamento de Dependência não costuma ser encontrado com muita frequência nos diagramas de classes. Esse relacionamento, como o próprio nome já diz, quando usado no Diagrama de Classes, identifica, certo grau de dependência de uma classe em relação à outra, isto é, sempre que ocorrer uma mudança na classe da qual uma outra classe depende, esta deverá também sofrer uma mudança. O relacionamento de Dependência é representado por uma reta tracejada entre duas classes contendo uma seta apontando a classe da qual a classe posicionada na outra extremidade do relacionamento é dependente de alguma forma. A Figura 14 apresenta um exemplo de Dependência, (GUEDES, 2008).

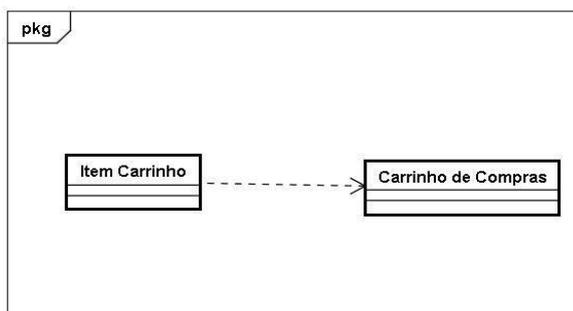


Figura 14. Exemplo de dependência

Fonte: GUEDES (2008)

A Figura 14 apresenta um exemplo de relacionamento de dependência entre as classes Item Carrinho e Carrinho de Compras, os quais representam classes utilizadas em um sistema de vendas para Internet. A classe Item Carrinho possui dependência com a classe Carrinho, porque a maioria dos eventos ocorridos nas instâncias desta classe, senão todos, afeta de alguma forma as instâncias da classe

Item Carrinho, (GUEDES, 2008).

- Realização

Uma Realização é um tipo de relacionamento especial que mistura características dos relacionamentos de generalização e dependência, sendo usada no Diagrama de Classes por identificar classes responsáveis por executar funções para classes que representam interfaces. Este tipo de relacionamento herda o comportamento de uma classe, mas não sua estrutura. O relacionamento de Realização é representado por uma seta tracejada contendo uma seta vazia que aponta para a classe de interface, enquanto na outra extremidade é definida a classe que realiza um comportamento pretendido pela classe de interface. A Figura 15 apresenta um exemplo de relacionamento de Realização. (GUEDES, 2008)

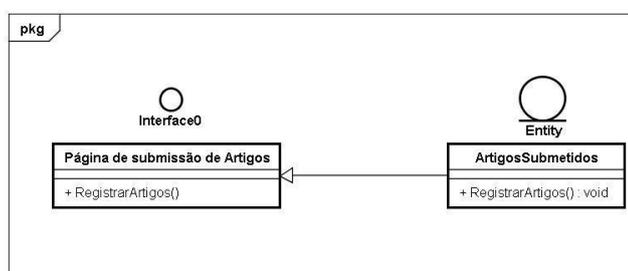


Figura 15. Exemplo de Realização

Fonte: GUEDES (2008)

2.10.5 Classe Associativa

Classes associativas são aquelas produzidas quando da ocorrência de associações que tenham multiplicidade muitos (*) em todas as suas extremidades. As classes associativas são necessárias nesses casos porque não existe um repositório que possa armazenar as informações produzidas pelas associações, pois todas as classes envolvidas apresentam multiplicidade muitos, o que obriga que seu atributo-chave seja transmitido às outras classes envolvidas e como todas tenham a mesma multiplicidade, nenhuma delas pode receber os atributos das outras, assim é preciso criar uma classe associativa para armazenar os atributos transmitidos pela associação, o que não impede que a classe associativa possua atributos próprios, além dos recebidos. Uma classe associativa é representada por uma seta tracejada partindo do meio da associação e atingindo uma classe. A Figura 16 apresenta um exemplo de classe associativa, (GUEDES, 2008).

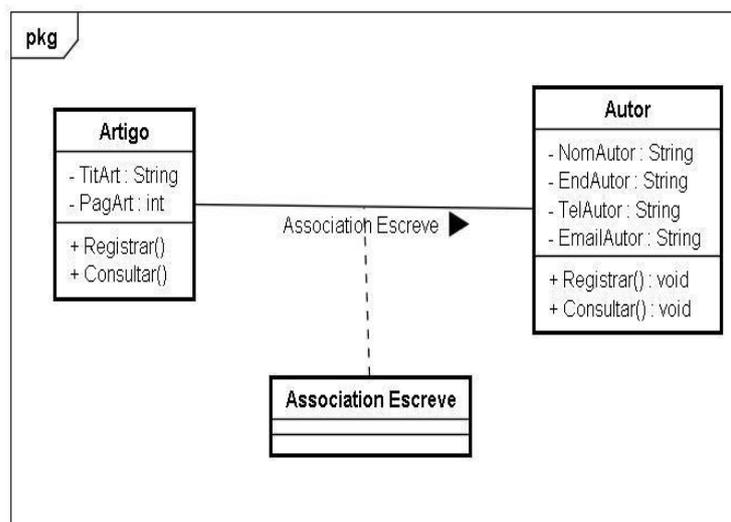


Figura 16. Exemplo de classe associativa

Fonte: GUEDES (2008)

2.10.6 Restrição

Restrições constituem-se em informações extras que definem condições a serem validadas durante a implementação dos relacionamentos entre as classes. Na verdade, as restrições podem ser utilizadas na maioria dos diagramas da UML. No entanto, são mais comumente encontradas no Diagrama de Classes. As restrições, no Diagrama de Classes, são *em geral* representadas por textos limitados por chaves.

A Figura 17 apresenta um exemplo de restrição, (GUEDES, 2008).

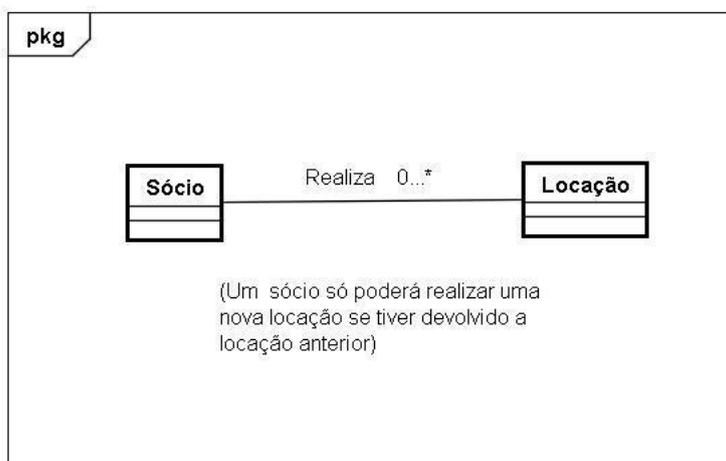


Figura 17. Exemplo de restrição em uma associação

Fonte: GUEDES (2008)

2.11 FRAMEWORK

Uma aplicação orientada a objetos é desenvolvida por classes que, através de troca de mensagens, realizam determinado processo em um sistema.

Pelas características fornecidas pelo paradigma de orientação a objetos é possível reutilizar partes da aplicação, através de alguma interface que permita acesso aos serviços (caixa-preta) ou redefinindo o comportamento de algumas subclasses (caixa-branca). Desta forma, pode-se obter diferentes aplicações utilizando a base de uma aplicação já existente, reutilizando códigos ou projetos desta aplicação, (CARMO, 1997).

Para esta reutilização pode ser necessário redefinir alguns comportamentos, em que a quantidade de modificações das classes depende do grau de semelhança entre as aplicações. No entanto, caso seja analisado a redefinição é possível identificar que existe uma parte comum de comportamento que podem ser generalizado, onde o programador pode criar superclasses que proporcionam um comportamento abstrato, (CARMO, 2001).

O conceito de classes abstratas é a representação genérica referente a determinadas entidades relacionadas, onde cada entidade representa uma abstração que será representada por uma subclasse concreta. Assim, a subclasse fornece as variações específicas do comportamento definido pela classe abstrata, (CARMO, 2001).

Como uma classe abstrata funciona como molde para subclasses, uma aplicação desenvolvida por classes abstratas pode ser utilizada como molde para a construção de outras aplicações. Neste caso, um projeto construído a partir de classes abstratas é considerado como um *framework* orientado a objetos, (CARMO, 2001).

Os *frameworks* são considerados estruturas de classes inter-relacionadas, que permitem minimizar o esforço para o desenvolvimento de aplicações, devido a definição de sua arquitetura, o que desprende o desenvolvedor de software de implementar estas características, (CARMO, 2001).

“Um *framework* é um projeto de larga escala que descreve como um programa é decomposto em um conjunto de objetos interativos. É geralmente

representado como um conjunto de classes abstratas e a maneira como suas instancias interagem”, (JOHNSON, 1997, p. 4).

A utilização de *frameworks*, invertem o modo de reutilização de classes, sendo iniciado o desenvolvimento através do entendimento do sistema contido no *framework*, implementando somente as particularidades da aplicação específica. Desta forma, a aplicação é desenvolvida pela adaptação da estrutura de classes do *framework*, (TALIGENT, 1995).

Um *framework* se difere de uma reutilização de classes de uma biblioteca, pois nesta são utilizados artefatos isolados de softwares, mas no caso do *framework*, são reutilizados um conjunto de classes inter-relacionadas.

As Figuras 18 e 19 ilustram a diferença entre a reutilização de classe de uma biblioteca e um *framework*, onde as partes sombreadas representam as classes associadas ao reuso, (SILVA, 2000).

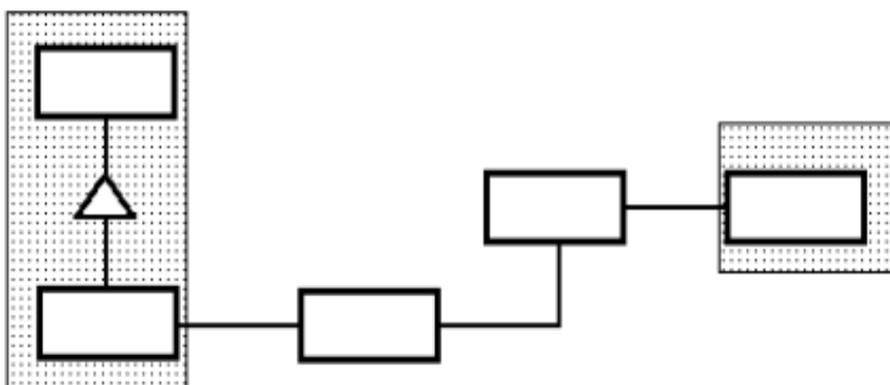


Figura 18. Aplicação desenvolvida reutilizando classes de biblioteca.

Fonte: SILVA (Ano 2000, p.31)

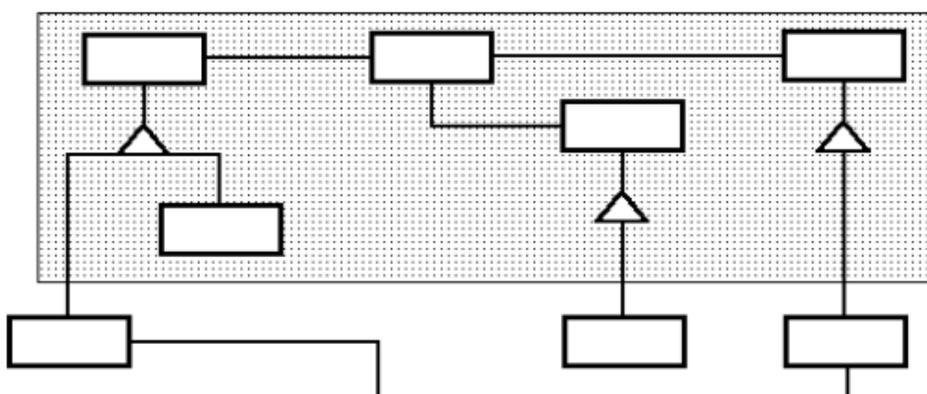


Figura 19. Aplicação desenvolvida reutilizando um *framework*.

Fonte: SILVA (Ano 2000, p.32)

Devido as facilidades que os *frameworks* oferecem, tais como: o aproveitamento de código implementado, custo de desenvolvimento reduzido e uma estrutura comum e bem definida, muitas aplicações web são desenvolvidas a partir destes *frameworks*, (SILVA, 2004).

Para Taligent, (1995), um *framework* é caracterizado por dois aspectos:

- Fornecer uma infraestrutura de projeto, onde as interconexões preestabelecidas definem a arquitetura da aplicação, o que reduz a quantidade de código a ser desenvolvido, testado e depurado. Desta forma, o desenvolvedor fica livre das responsabilidades pré-definidas pelo *framework*, que é adaptado com as necessidades específicas da aplicação.
- Fornecer um fluxo de controle da aplicação onde, em tempo de execução, as instancias das classes desenvolvidas realizam ou aguardam as chamadas das instâncias do *framework*.

Um *framework* também pode ser caracterizado pelo nível de granularidade, onde são agrupados diferentes tipos de classes mais ou menos complexas, podendo conter desde um projeto genérico completo até a construção de alto nível para solucionar situações comuns do projeto. Além disto, podem conter o projeto genérico completo para um domínio de aplicação, ou construções de alto nível que solucionam situações comuns em projetos, (SILVA, 2000).

A vantagem de utilizar um *framework* é que os desenvolvedores utilizam as mesmas convenções, classes e bibliotecas, o que torna a manutenção de uma aplicação mais fácil. Assim, independente do desenvolvedor que criou determinado *script*, não é necessário horas para o entendimento e manutenção deste *script* por outro desenvolvedor. Desta forma, o *framework* contribui para que novos desenvolvedores possam, de forma rápida, compreender a aplicação implementada, e conseqüentemente diminuir custos e tempo para treinamento, (MINETTO, 2007).

Pode-se também considerar uma vantagem do *framework* a automatização de tarefas repetitivas. Considerando o fato de que em uma aplicação é necessário a manipulação dos dados de uma tabela no banco de dados, as operações de consulta, inclusão, exclusão e alteração são praticamente idênticas para todas as tabelas. Neste caso, não faz sentido para o desenvolvedor implementar este código diversas vezes, onde estas operações poderiam ser automatizadas por ferramentas contidas em um *framework*.

Através de um *framework*, a aplicação desenvolvida pode ser interpretada por dois níveis de composição, um nível superior e outro inferior. No nível superior, representado pelas classes do *framework*, fornece a estrutura de controle da aplicação, sendo o nível inferior contextualizado pelo desenvolvedor através de subclasses. Desta forma, o nível inferior disponibiliza operações específicas de uma aplicação e a ativação destas operações é modelada através do nível superior, (CARMO, 1997).

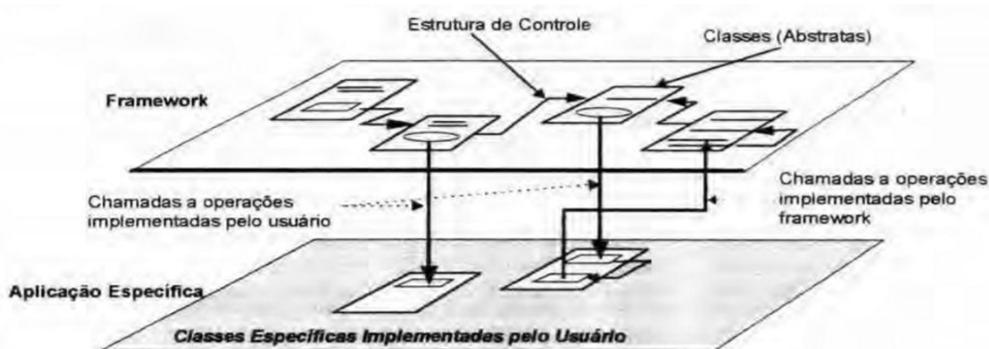


Figura 20. Visão conceitual da estrutura de um *framework*.

Fonte: CAMPOS (1997, p. 32)

Para CARMO, (1997), a partir da predominância de um projeto, o *framework* pode ser diferenciado em dois tipos:

- *Frameworks* Baseados em Herança: Denominado também como *frameworks* dirigidos pela arquitetura (*architecture-driven frameworks*) ou *framework* caixa branca, são disponibilizados para uma aplicação uma estrutura completa, sendo estendidas por códigos específicos, conforme a particularidade da aplicação. Neste caso, é necessário conhecer o funcionamento deste tipo de *framework* para a adaptação a aplicação que será desenvolvida;

- *Frameworks* Baseados em Composição: Denominados também como *frameworks* dirigidos pelos dados (*data-driven frameworks*) ou *frameworks* caixa preta, são utilizados através da composição de objetos para sua adaptação. Desta forma, o *framework* tem comportamentos diferentes dependendo dos parâmetros que são enviado pelo cliente, sendo muito utilizados como base para a construção de *toolkits* de componentes;

Estes dois tipos de *frameworks* são independentes, onde *frameworks* baseados em composição surgem de sucessivas generalizações de *frameworks* baseados em herança, devido ao desenvolvimento de um *framework* baseado em composição representar uma tarefa muito difícil e custosa, (JOHNSON, 1993).

2.11.1 Framework Codeigniter

O Codeigniter é um *framework* MVC de desenvolvimento de aplicativos para quem constrói sites e aplicações PHP. Seu objetivo é permitir que desenvolva-se projetos mais rapidamente do que poderia se estivesse escrevendo o código do zero, proporcionando um rico conjunto de bibliotecas para as tarefas mais comuns necessárias, bem como uma interface simples e estrutura lógica para acesso àquelas bibliotecas. O Codeigniter permite que o desenvolvedor mantenha o foco em seu projeto minimizando a quantidade de código necessário para uma dada tarefa, (ELLISLAB, 2014).

“Codeigniter é um *framework* PHP de alto desempenho, com características de fácil aprendizado, construído para programadores PHP que necessitam de uma ferramenta simples e elegante para criar aplicações web com recursos completos”. (ELLISLAB, 2014)

A estrutura do *framework* Codeigniter é desenvolvida para realizar a comunicação entre as camadas do padrão MVC da forma mais simples possível, onde é realizada esta comunicação enviando dados entre as camadas pelo uso de vetores, vetores multidimensionais ou objetos (GABARDO, 2012).

Codeigniter tem uma abordagem bastante flexível do padrão MVC, onde os modelos não são necessários. Alguns desenvolvedores acreditam que não precisa utilizar a camada de modelo, ou achar que a manutenção de modelos na aplicação acarreta em maior complexidade do que se deseja, desta forma, pode-se dispensar a utilização de modelos e desenvolver aplicação minimamente usando controladores e visão, (ELLISLAB, 2014).

2.12 PADRÕES DE PROJETO

Nesta Seção, veremos o padrão de projeto Model –View – Controller (MVC), que foi utilizado no desenvolvimento.

2.12.1 Arquitetura MVC (Model –View – Controller)

Segundo Macoratti (2003), com a necessidade de solucionar problemas apresentados pela programação monolítica, tais como o armazenamento de dados e códigos em uma mesma máquina, surgiu o conceito de desenvolvimento de sistemas através de camadas. Desta forma, solucionou-se a dificuldade de desenvolvimento e manutenção de aplicações, pois suas funcionalidades não ficavam em um módulo único que armazenava grandes quantidades de código.

Devido ao compartilhamento da lógica de acesso aos dados, impulsionou o desenvolvimento de aplicações em duas camadas, dividindo a base de dados da execução da aplicação, onde os armazenamentos dos dados eram realizados em uma máquina específica, e a aplicação executada em máquinas distintas. (MACORATTI, 2013)

Com a evolução da internet, surgiu uma nova visão de aplicação para os usuários e desenvolvedores, pois esta aplicação, vista como um site é disponibilizado através de um servidor web no qual apenas são realizadas requisições pelos usuários, (MACORATTI, 2013).

Em consequência da evolução no desenvolvimento de software, novos padrões de projetos surgiram para facilitar a criação e manutenção de novas aplicações. O padrão de projeto MVC, um acrônimo para *Model, View, Controller* (em português: Modelo, Visão e Controlador), teve grande destaque, sendo muito utilizado atualmente, e possui o conceito de dividir a aplicação em três camadas distintas, (MACORATTI, 2013).

“A arquitetura MVC fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. A arquitetura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário.” (MACORATTI, 2013).

O *framework* MVC surgiu através do ambiente *Smalltalk-Visual Works* com o objetivo de construir interfaces com o usuário. Sua representação na atualidade é uma das mais relevantes na estrutura de projeto adaptável dinamicamente, visto que sua fatoração de abstrações maximiza a reutilização de funcionalidades por composição de objetos. Pode-se considerar também que o grau de padronização de interfaces e encapsulamento de funcionalidade disponibilizam o desenvolvimento de interfaces de diálogos, o que possibilita ao desenvolvedor descrever a composição da interface sem a necessidade de programação (CARMO, 1997).

Organizar o código em camadas é um dos principais objetivos do padrão de projeto MVC, o que acarreta a separação física dos componentes da aplicação, tornando essas camadas a chave para a independência entre os componentes, devido serem agrupados componentes com responsabilidades em comum, (BALTHAZAR, 2006).

A aplicação desenvolvida pelo padrão MVC fica dividida em camadas, desta forma é proporcionado uma melhor manutenção, leitura e entendimento do código desenvolvido. Pode-se considerar também que aplicações que são desenvolvidas com este padrão se tornam mais independentes, devido a separação da camada de regra de negócio, de armazenamento e de visão, possibilitando realizar alterações na regra de negócio sem modificar a interface disponível para a usuário, (BALTHAZAR, 2006).

Com isso, as camadas de negócio podem ser divididas em classes, possibilitando agrupar estas classes em pacotes ou componentes, o que reduz as dependências entre as mesmas. Desta forma, esta divisão das classes proporciona a reutilização por partes distintas da aplicação, podendo ser utilizadas até por outra aplicação, (BALTHAZAR, 2006).

As camadas são divididas, como se pode observar no nome do *framework*, em Modelo, Visão e Controlador, onde:

- Modelo (*Model*): camada que representa os dados e as regras específicas da aplicação (*business data e business logic*) para gerenciamento do acesso e a atualização desses dados.

- Visão (*View*): camada que gerencia a saída gráfica e textual da parte da aplicação visível ao usuário. Renderiza o conteúdo de um modelo acessando seus dados e especificando como eles são apresentados.
- Controlador (*Controller*): camada que interpreta as entradas de mouse e teclado do usuário, definindo o comportamento da aplicação. Envia as requisições do usuário para o modelo e seleciona as visões para apresentar os resultados destas requisições.

A Figura 21 mostra o funcionamento das camadas do padrão de projeto MVC.

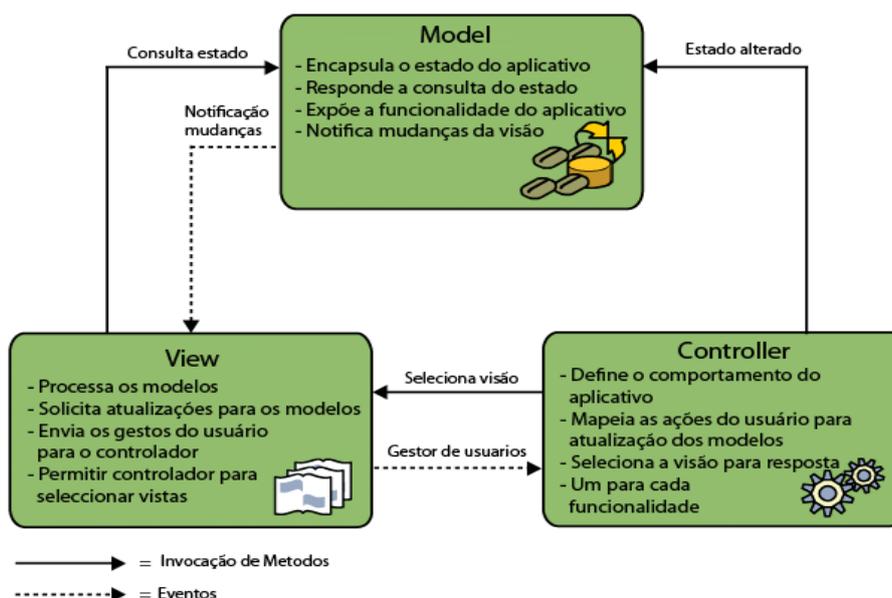


Figura 21. Camadas do padrão MVC.

Fonte: Macoratti (2013)

De forma conceitual, a instância de uma visão sempre é associada a uma representação de um objeto, através de um modelo, sendo associada opcionalmente por um controlador, (BALTHAZAR, 2006).

Através do controlador é realizada a coordenação das atividades providas na visão, as ações realizadas pelo usuário e o modelo correspondente que será instanciado. Além de representar a interface de saída da aplicação, a visão monitora as alterações produzidas através de anúncios de mudanças, gerados quando alguma modificação é realizada no modelo que interfere em seu estado interno,

ocorrendo uma atualização da visão para manter consistente os dados entre os estados interno e de apresentação. Na associação, um modelo pode ter várias visões, mas uma visão pode ter só um modelo associado (CARMO, 1997).

Pode-se observar na Figura 22, a separação física e lógica proposta pelo modelo MVC. Neste caso, é apresentada na camada de visão toda interface da aplicação que é interativa ao usuário, o que permite ao mesmo realizar requisições para a aplicação. Toda a parte de gerenciamento do comportamento da aplicação é desenvolvida na camada de controle, que realiza o intermédio na comunicação entre a visão e o modelo. A responsabilidade da camada de modelo é conter o código da requisição solicitada como, por exemplo, realizar a comunicação com outra camada para realizar uma consulta no banco de dados. (BALTHAZAR, 2006)

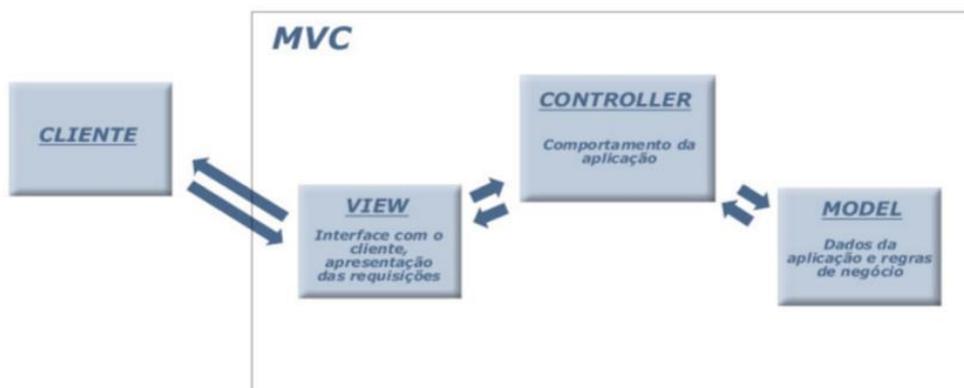


Figura 22. O Modelo MVC.

Fonte: Balthazar (2006, p. 3)

Além da vantagem de fácil manutenção, devido ao sistema ser tratado como módulos específicos, outra vantagem no desenvolvimento de aplicações com o *framework* MVC é a divisão do acesso aos dados, regra de negócio e apresentação das informações. Desta forma, é possível fornecer um trabalho em equipe, pois enquanto o designer trabalha na apresentação das informações, um desenvolvedor pode, em paralelo, implementar as regras de negócio da aplicação, e um administrador de banco de dados pode criar os modelos da aplicação, visto que qualquer alteração entre as camadas teriam pouco ou nenhum impacto entre elas (GONÇALVES, 2007).

Ainda dentro do conceito de vantagens na utilização do *framework* MVC, pode-se considerar que a reutilização é fortemente explorada, visto que a mesma

funcionalidade implementada em uma parte da aplicação pode ser reutilizada para outras partes da mesma aplicação, (BALTHAZAR, 2006).

Porém, pode-se se considerar como desvantagem, na utilização do *framework* MVC, o custo elevado no desenvolvimento de uma aplicação, devido a curva de aprendizado na utilização do mesmo, (BALTHAZAR, 2006).

Isto ocorre devido todas as classes serem organizadas em pacotes, conforme a representação de seus comportamentos. Sendo assim, o desenvolvedor é obrigado a seguir um padrão, durante os ajustes ou desenvolvimento de novas funcionalidades, aumentando o prazo para a finalização do produto final. Outra desvantagem que pode ser considerada é que o *framework* MVC precisa de um profissional especializado na equipe de desenvolvimento com domínio nos conceitos apresentados, o que pode aumentar os gastos na implementação da aplicação devido treinamentos e conscientização da efetiva adoção do padrão, (BALTHAZAR, 2006).

2.13 MODELO ENTIDADE RELACIONAMENTO (MER) E DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)

Nesta seção, veremos a importância do modelo Entidade Relacionamento e do Diagrama Entidade Relacionamento para o desenvolvimento de um sistema computacional.

Segundo Rodrigues (2015), quando se inicia o desenvolvimento de um novo sistema, ou mesmo de uma nova funcionalidade para um sistema existente, um dos primeiros passos a ser executado é o estudo e levantamento dos requisitos necessários para a construção do produto final.

Durante essa análise, identifica-se as principais partes e objetos envolvidos, suas possíveis ações e responsabilidades, suas características e como elas interagem entre si, (RODRIGUES, 2015).

A partir das informações obtidas, pode-se desenvolver um modelo conceitual que será utilizado para orientar o desenvolvimento propriamente dito, fornecendo informações sobre os aspectos relacionados ao domínio do projeto em questão, (RODRIGUES, 2015).

2.13.1 Modelo Entidade Relacionamento

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos), (RODRIGUES, 2015).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como, chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais, (RODRIGUES, 2015).

2.13.2 Entidades

Os objetos ou partes envolvidas num domínio, também chamados de entidades, podem ser classificados como físicos ou lógicos, de acordo com sua existência no mundo real. Entidades físicas: são aquelas realmente tangíveis, existentes e visíveis no mundo real, como um cliente (uma pessoa, uma empresa) ou um produto (um carro, um computador, uma roupa). Já as entidades lógicas são aquelas que existem geralmente em decorrência da interação entre ou com entidades físicas, que fazem sentido dentro de um certo domínio de negócios, mas que no mundo externo/real não são objetos físicos (que ocupam lugar no espaço). São exemplos disso uma venda ou uma classificação de um objeto (modelo, espécie, função de um usuário do sistema), (RODRIGUES, 2015).

As entidades são nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio. Exemplos práticos de entidades comuns em vários sistemas são Cliente, Produto, Venda, Turma, Função, entre outros, (RODRIGUES, 2015).

Pode-se classificar as entidades segundo o motivo de sua existência:

- Entidades fortes: são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir. Em um sistema de vendas, a entidade produto, por exemplo, independe de quaisquer outras para existir;

- Entidades fracas: ao contrário das entidades fortes, as fracas são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o mesmo exemplo, a entidade venda depende da entidade produto, pois uma venda sem itens não tem sentido;
- Entidades associativas: esse tipo de entidade surge quando há um relacionamento do tipo muitos para muitos (explicado a seguir). Nestes casos, é necessária a criação de uma entidade intermediária cuja identificação é formada pelas chaves primárias (explicado mais adiante) das outras duas entidades. No contexto de uma aplicação de vendas, como precisamos relacionar vendas e produtos numa relação muitos para muitos, a entidade produto não pode referenciar diretamente a venda, nem o inverso, pois isso caracterizaria um relacionamento um para um, ou um para muitos. Sendo assim, criamos uma entidade intermediária para representar os itens da venda, que tanto possuem a identificação do produto, quando da venda em que estão contidos. Neste caso específico, também caberiam a esta entidade informações como quantidade de itens e desconto unitário, por exemplo.

2.13.3 Relacionamentos

Uma vez que as entidades são identificadas, deve-se então definir como se dá o relacionamento entre elas. (RODRIGUES, 2015)

De acordo com a quantidade de objetos envolvidos em cada lado do relacionamento, podemos classifica-los de três formas:

- **Relacionamento 1-1 (um para um):** cada uma das duas entidades envolvidas referenciam obrigatoriamente apenas uma unidade da outra. Por exemplo, em um banco de dados de currículos, cada usuário cadastrado pode possuir apenas um currículo na base, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado.
- **Relacionamento 1-n ou 1-* (um para muitos):** uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada uma unidade da outra entidade. Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só pode estar ligado a um usuário

principal. Note que temos apenas duas entidades envolvidas: usuário e dependente. O que muda é a quantidade de unidades/exemplares envolvidas de cada lado.

- **Relacionamento n-n ou *-* (muitos para muitos):** neste tipo de relacionamento cada entidade, de ambos os lados, pode referenciar múltiplas unidades da outra. Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo título, e vice-versa.

Os relacionamentos em geral são nomeados com verbos ou expressões que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra. Essa nomenclatura pode variar de acordo com a direção em que se lê o relacionamento. Por exemplo: um autor *escreve* vários livros, enquanto um livro é *escrito* por vários autores, (RODRIGUES, 2015).

2.13.4 Atributos do DER

Atributos do DER são as características que descrevem cada entidade dentro do domínio. Por exemplo, um cliente possui nome, endereço e telefone. Durante a análise de requisitos, são identificados os atributos relevantes de cada entidade naquele contexto, de forma a manter o modelo o mais simples possível e conseqüentemente armazenar apenas as informações que serão úteis futuramente. Uma pessoa possui atributos pessoais como cor dos olhos, altura e peso, mas para um sistema que funcionará em um supermercado, por exemplo, estas informações dificilmente serão relevantes. (RODRIGUES, 2015)

Os atributos do DER podem ser classificados quanto à sua função da seguinte forma:

- **Descritivos:** representam característica intrínsecas de uma entidade, tais como nome ou cor.
- **Nominativos:** além de serem também descritivos, estes têm a função de definir e identificar um objeto. Nome, código, número são exemplos de atributos nominativos.

- Referenciais: representam a ligação de uma entidade com outra em um relacionamento. Por exemplo, uma venda possui o CPF do cliente, que a relaciona com a entidade cliente.

Quanto à sua estrutura, podemos ainda classificá-los como:

- Simples: um único atributo define uma característica da entidade. Exemplos: nome, peso.

- Compostos: para definir uma informação da entidade, são usados vários atributos. Por exemplo, o endereço pode ser composto por rua, número, bairro, etc.

Alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir. Em um cadastro de clientes, por exemplo, esse atributo poderia ser o CPF. A estes chamamos de Chave Primária, (RODRIGUES, 2015).

Já os atributos referenciais são chamados de Chave Estrangeira e geralmente estão ligados à chave primária da outra entidade. Estes termos são bastante comuns no contexto de bancos de dados. Mantendo o exemplo anterior, a entidade cliente tem como chave primária seu CPF, assim, a venda possui também um campo “CPF do cliente” que se relaciona com o campo CPF da entidade cliente.

2.13.5 Diagrama Entidade Relacionamento

Enquanto o MER é um modelo conceitual, o Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é criar de imediato sua representação gráfica, seguindo algumas regras. (RODRIGUES, 2015)

O diagrama facilita ainda a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado, (RODRIGUES, 2015).

Em sua notação original, (proposta por Peter Chen, idealizador do modelo e do diagrama), as entidades deveriam ser representadas por retângulos, seus atributos por elipses e os relacionamentos por losangos, ligados às entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n). Porém, notações mais modernas abandonaram o uso de elipses para atributos e passaram a utilizar o formato representado na UML, em que os atributos já aparecem listados na própria entidade. Essa forma torna o diagrama mais limpo e fácil de ser lido, (RODRIGUES, 2015).

A Figura 23 ilustra um exemplo simples de um diagrama para um sistema de imobiliárias.

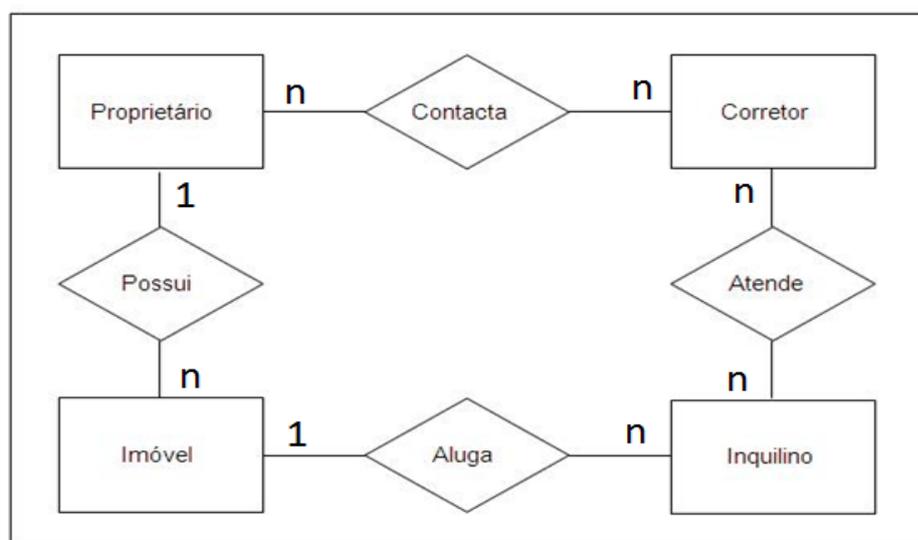


Figura 23. Diagrama Entidade Relacionamento de sistema de imobiliária

Fonte: Rodrigues (2015)

No domínio representado pelo diagrama acima temos as seguintes entidades e relacionamentos:

- Proprietário *contacta* Corretor (um proprietário pode contactar vários corretores e um corretor pode ser contactado por vários proprietários).
- Corretor *atende* Inquilino (um corretor pode atender vários inquilinos e um inquilino pode ser atendido por vários corretores).
- Inquilino *aluga* Imóvel (um inquilino aluga um imóvel e um imóvel pode ser alugado por vários inquilinos).

- Proprietário *possui* Imóvel (um proprietário possui vários imóveis e um imóvel pertence a apenas um proprietário).

Uma variante da Figura 23 pode ser vista na Figura 24, onde a cardinalidade do relacionamento é exibida junto ao losango, (RODRIGUES, 2015).

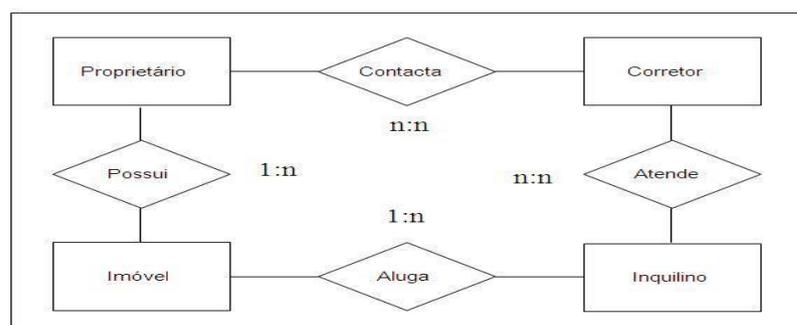


Figura 24. Diagrama de Entidade Relacionamento (variação)
Fonte: Rodrigues (2015)

Uma outra variação já mostra a cardinalidade de uma forma mais completa, deixando claro as possibilidades de números de objetos envolvidos em cada relacionamento. Nesse modelo, em cada lado do relacionamento os números aparecem no formato (X,Y) ao invés de um único número como vemos nas figuras anteriores. A Figura 25 ilustra um exemplo desse tipo. (RODRIGUES, 2015)



Figura 25. Diagrama Entidade Relacionamento (variação 2)
Fonte: Rodrigues (2015)

No diagrama ilustrado na Figura 25, se lê os relacionamentos da seguinte forma:

- *1 ou 1 grupo possui 0 ou muitos produtos.* Como de um lado temos “1 ou 1”, isso equivale a apenas “1”, pois não temos várias possibilidades. Já do lado do produto, indicamos que um grupo pode possuir nenhum produto, mas também pode possuir vários.

- *0 ou várias vendas contém 1 ou muitos produtos.* Ou seja, um produto pode nunca ser vendido (0 vendas) como também pode ser vendido várias vezes (n vendas). Já uma venda deve conter 1 ou vários produtos, pois uma venda não pode estar vazia (0 produtos).

Os atributos, como já foi dito, podem aparecer no diagrama na forma de elipses ligadas às entidades. Essa foi a notação original proposta, a Figura 26 ilustra o diagrama com muitos itens, o que pode atrapalhar um pouco a organização destes, (RODRIGUES, 2015).

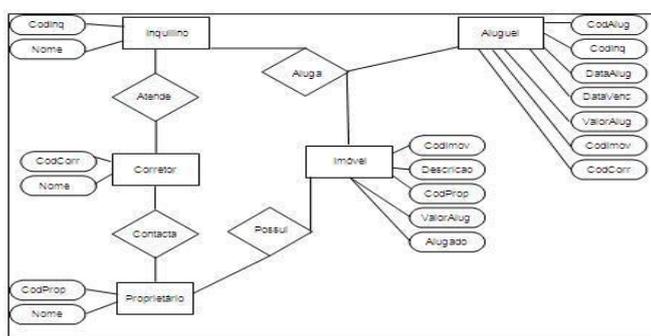


Figura 26. Atributos apresentados como elipses

Fonte: Rodrigues (2015)

Em uma notação mais atual, comumente utilizada na UML, os atributos aparecem listados dentro do próprio retângulo da entidade, enquanto o nome da entidade aparece no topo na forma de título. A Figura 27 ilustra esse exemplo, (RODRIGUES, 2015).

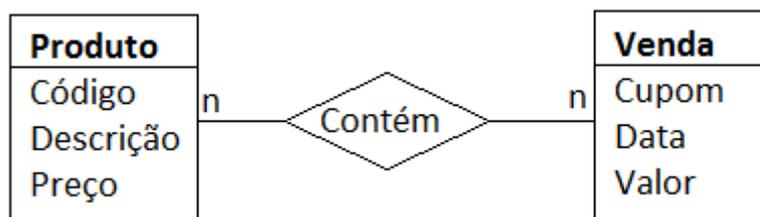


Figura 27. Diagrama com atributos nas entidades

Fonte: Rodrigues (2015)

3. DESENVOLVIMENTO

Neste capítulo, é descrito o desenvolvimento do aplicativo, foco desta pesquisa, segundo todo embasamento teórico visto no decorrer deste trabalho.

No desenvolvimento do aplicativo, foram utilizadas as seguintes ferramentas com os seguintes objetivos:

a) BRModelo: para desenvolvimento do Modelo Entidade Relacionamento com o Diagrama Entidade Relacionamento, o Modelo Relacional e os Scripts das tabelas a serem desenvolvidas no Banco de Dados (os Scripts estão anexos a este trabalho);

b) Ferramenta CASE Astah Professional: para criação dos Casos de Uso do Sistema, com o detalhamento de suas descrições, assim como o Diagrama de Classes;

c) WAMP Server 64-BIT – versão 2.5: para o gerenciamento de serviços Apache e MySQL e configurações de servidor local para testes;

d) Banco de Dados PostgreSQL – para a alocação e manipulação de dados;

e) PHP versão 5.4.12 – como linguagem base de programação orientada a objeto;

f) Apache 2.4.4 – assim como qualquer servidor do tipo, o Apache é responsável por disponibilizar páginas e todos os recursos que podem ser acessados pelo internauta. Envio de e-mails, mensagens, compras online e diversas outras funções podem ser executadas graças a servidores como o Apache. O que vale destacar no Apache é que, apesar de tudo, ele é distribuído sob a licença GNU, ou seja, é gratuito e pode ser estudado e modificado através de seu código fonte por qualquer pessoa.

g) HTML 5.5 – o código HTML está presente nas páginas da web há mais de dez anos e tem uma larga aceitação dos usuários por todo o mundo. Com certeza, qualquer pessoa que tem uma ligação um pouco mais estreita com computadores e internet já ouviu falar em HyperText Markup Language (Linguagem de marcação de hipertexto), ou seja, o HTML. A principal característica desse tipo de programação são as “tags” – as marcações para inserção de algum tipo de objeto ou característica de um objeto através de comandos em inglês e números.

h) *FrameWork* responsivo *front-end* – Foundation 3.0 - é um *framework* CSS construído com SASS, um pré-processador CSS poderoso, que permiti desenvolver muito mais rápido as fundações básicas de um *web site*, disponibilizando ferramentas para personalizar, construir e organizar os códigos CSS e JavaScript que fazem interações automáticas em diferentes resoluções de tela, chamado sistema responsivo.

O sistema baseia-se em *grids*, grades no formato tabelar de no máximo 12 colunas que variam suas larguras de acordo com a resolução da tela, tendo no CSS a codificação base para tais mudanças e adaptações.

i) *FrameWork Data Table* - é um plug-in para o *jQuery*, biblioteca *Javascript*, com base nos fundamentos da otimização progressiva e que adiciona controles avançados de interação para qualquer tabela HTML.

O *Data Table* tem dois modos diferentes de processamento de dados (ordenação, pesquisa, etc...), do lado do cliente, onde é carregado *up-front* e o processamento é feito no browser, e no servidor onde uma solicitação *Ajax* é feita para cada tabela com apenas os dados necessários e é processado no servidor.

j) *TinyMCE* é um excelente editor WYSIWYG (*What You See Is What You Get* ou "O que você vê é o que você tem" em português), principalmente para se utilizar em sistemas de administração de sites. Totalmente escrito em JavaScript, é um dos editores visuais mais leves e completos.

k) *FrameWork CodeIgniter* Versão 2.2.0 – para a estruturação dos código em formato MVC, objetivando a segurança, agilidade e eficiência do sistema;

l) *NeatBeans* versão 8.1: como editor para a escrita, organização e manipulação dos códigos fonte do sistema.

3.1 DIAGRAMA DE CASOS DE USOS

Nesta fase, foram desenvolvidos os casos de uso referentes ao sistema, ao todo 34 (trinta e quatro), conforme ilustra a Figura 28, com a descrição detalhada de todos eles, que poderá ser visualizada de forma completa no Apêndice C.

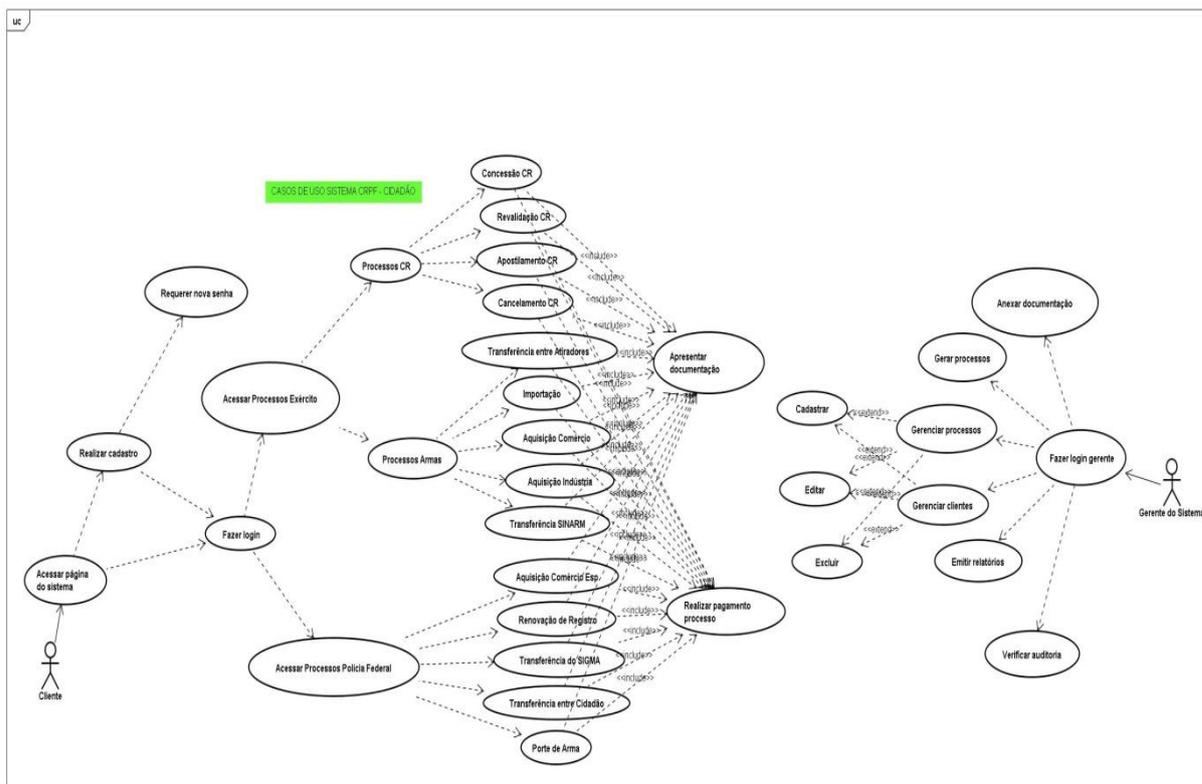


Figura 28. Diagrama de Casos de Uso

Fonte: Autoria própria

3.1.1 Detalhamento dos Casos de Uso

O detalhamento completo dos casos de uso do sistema encontram-se disponíveis para visualização no Apêndice C deste trabalho. Neste detalhamento, são descritos 34 casos de uso, conforme Diagrama de Casos de Uso acima.

3.2 DIAGRAMA DE CLASSES

Nesta fase, foi desenvolvido o diagrama de classes para o sistema, utilizando a ferramenta CASE Astah Professional, foram ao todo construídas 20 (vinte) classes para a aplicação, com todos os seus relacionamentos, atributos e procedimentos, ilustrados a seguir pela Figura 29:

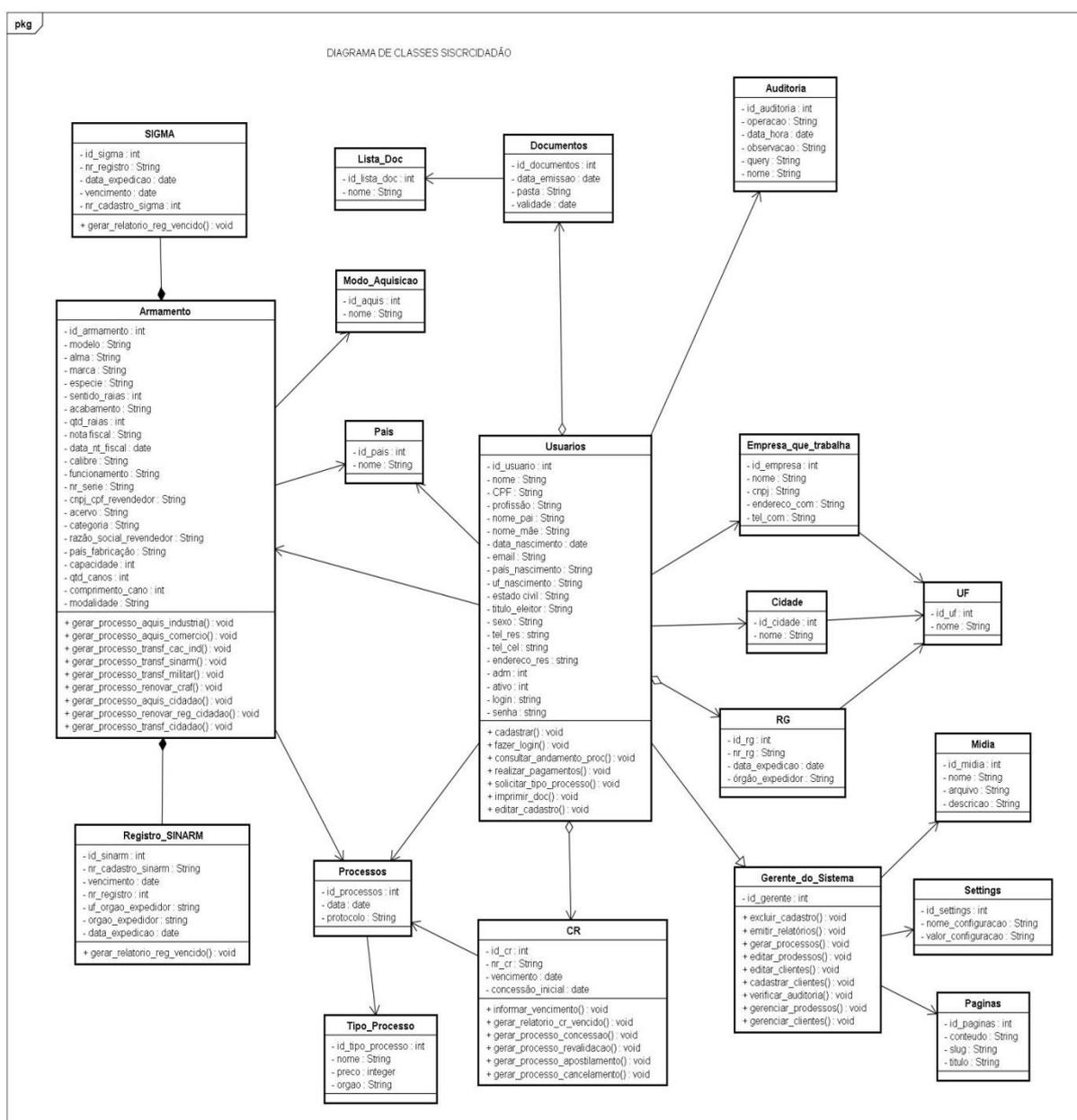


Figura 29. Diagrama de Classes

Fonte: Autoria própria

3.3 MODELO ENTIDADE RELACIONAMENTO E MODELO RELACIONAL

Com base nas informações necessárias de requisitos, foi desenvolvido o Diagrama Entidade relacionamento, com a criação de 19 (dezenove) Entidades Fortes e Fracas e 02 (duas) Entidades Associativas, feitos com a ferramenta BrModelo, sendo elas:

- | | |
|---------------------|-------------------------------------|
| 1) registro_sinarm; | 11) rg; |
| 2) registro_sigma; | 12) usuarios; |
| 3) modo_aquisicao; | 13) cidade |
| 4) armamento; | 14) uf; |
| 5) pais; | 15) Empresa_onde_Trabalha; |
| 6) lista_doc; | 16) settings; |
| 7) documentos; | 17) paginas; |
| 8) tipo_processo | 18) mídia; e |
| 9) processos | 19) auditoria. |
| 10) cr | 20) associativa_usuario_empresa |
| | 21) associativa_tipo_proc_lista_doc |

Os relacionamentos entre as entidades e suas cardinalidades estão descritas a seguir, juntamente com o Modelo Entidade Relacionamento, na forma de diagrama:

Entidade armamento x Entidade registro_sinarm: 0,1 x 0,1;

Entidade armamento x Entidade registro_sigma: 0,1 x 0,1;

Entidade armamento x Entidade processos: 1,1 x 1,n;

Entidade armamento x Entidade modo_aquisicao: 0,n x 1,1;

Entidade armamento x Entidade pais: 0,n x 1,1;

Entidade usuarios x Entidade processos: 1,1 x 1,n;

Entidade usuarios x Entidade cr: 1,1 x 0,1;

Entidade usuarios x Entidade documentos: 1,1 x 0,n;

Entidade usuarios x Entidade auditoria: 1,1 x 0,n;

Entidade usuarios x Entidade associativa usuário_empresa: 0,1 x 0,n;

Entidade usuarios x Entidade cidade: 0,n x 1,1;

Entidade usuarios x Entidade rg: 1,1 x 1,n;

Entidade usuarios x Entidade pais: 0,n x 1,1

Entidade rg x Entidade uf: 0,n x 1,1;

Entidade uf x Entidade cidade: 1,1 x 0,n;

Entidade empresa_trabalha x Entidade associativa usuário_empresa: 0,1 x 0,n;

Entidade cr x Entidade processos: 1,1 x 1,n;

Entidade processos x Entidade tipo_processo: 0,n x 1,1;

Entidade tipo_processo x Entidade associativa tipo_proc_lista_doc: 0,1 x 0,n;

Entidade documentos x Entidade lista_doc: 0,n x 1,1;

Entidade lista_doc x Entidade associativa tipo_proc_lista_doc: 0,1 x 0,n;

As Entidades paginas, mídia e settings não se relacionam.

3.3.1 Diagrama do Modelo Entidade Relacionamento

Com as Entidades definidas determinou-se o Modelo Entidade Relacionamento e o Modelo Relacional da aplicação, abordados nesta subseção e na subseção 3.1.2 a seguir. A Figura 30 ilustra o Diagrama Entidade Relacionamento da aplicação.

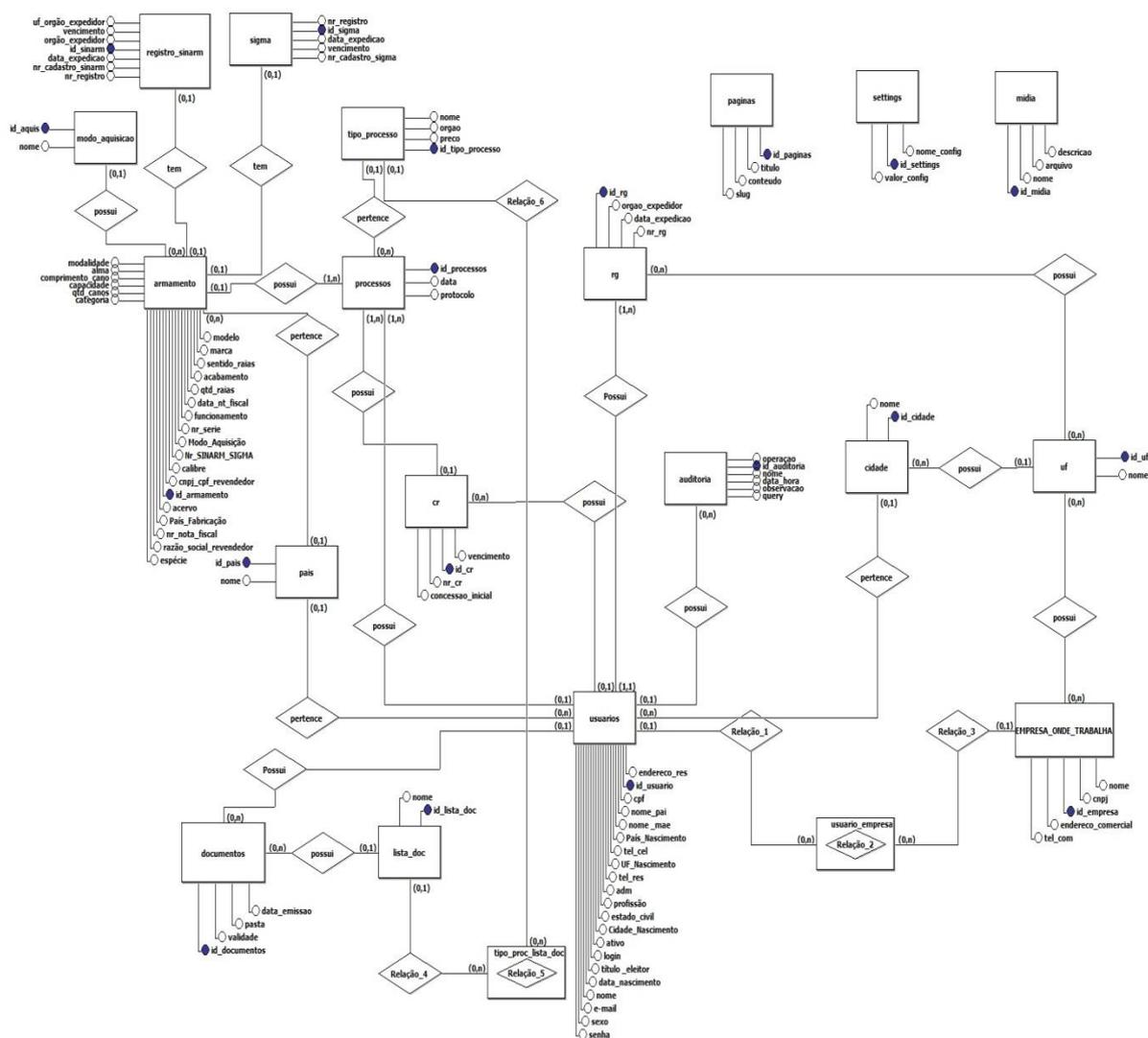


Figura 30. Diagrama Entidade-Relacionamento
Fonte: Autoria própria

3.3.2 Modelo Relacional

A Figura 31 ilustra o Modelo Relacional, que foi gerado a partir do Modelo Entidade Relacionamento pela própria ferramenta BrModelo, para isto, basta selecionar a aba do menu “Esquema Conceitual” e clicar em “Gerar Esquema Lógico”. Com o Modelo Relacional gerado, foi possível gerar também os Scripts de Tabelas SQL para o Banco de Dados, visualizado no Apêndice 1, para isto, basta selecionar a aba do menu “Esquema Lógico” e clicar em “Gerar Esquema Físico”.

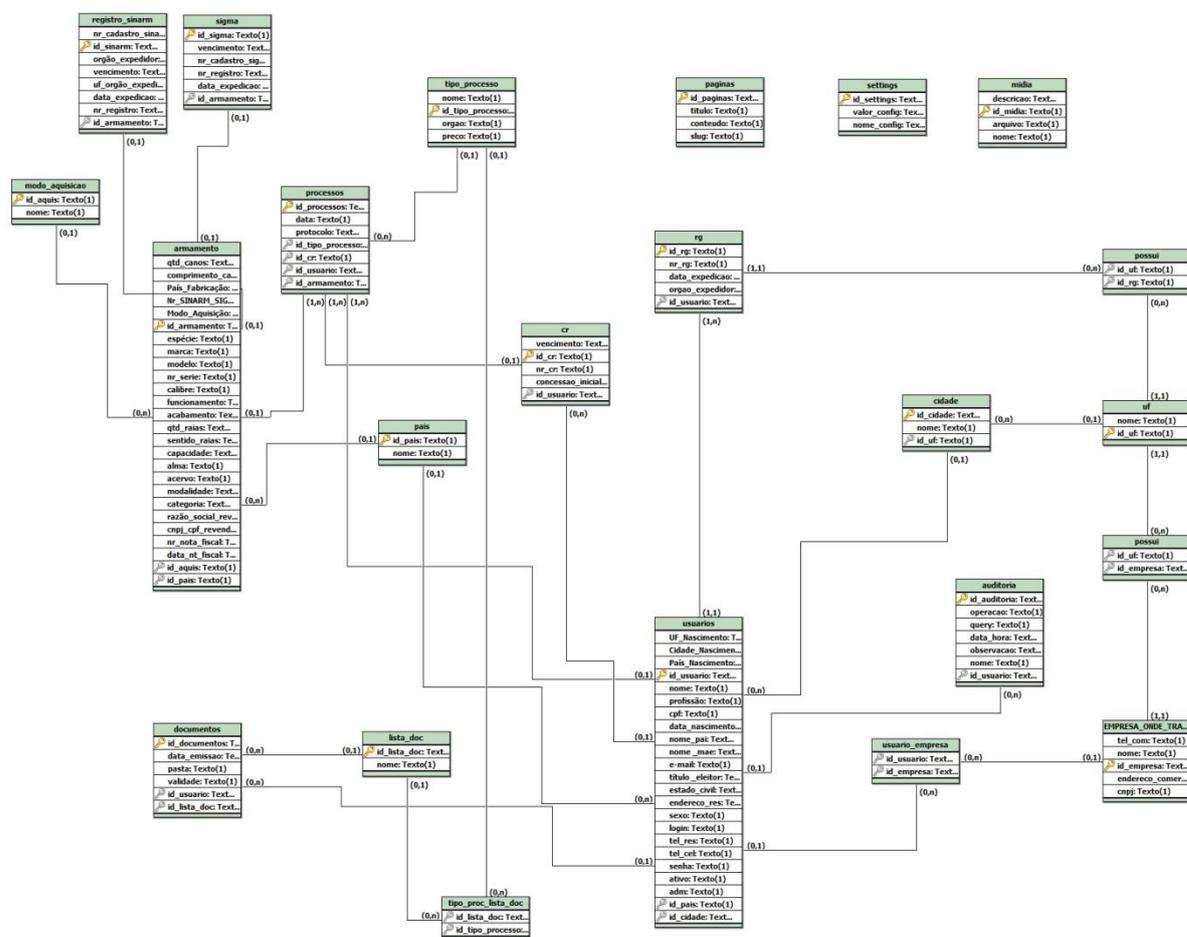


Figura 31. Modelo Relacional
Fonte: Autoria própria

3.4 IMPLEMENTAÇÃO

A implementação do projeto foi idealizada e desenvolvida, principalmente, através do *IDE NeatBeans* versão 8.1, como editor para a escrita, organização e manipulação dos códigos fonte do sistema, e do *framework CodeIgniter*, que nos permite focarmos criativamente em nossos projetos, uma vez que ele diminui o número de linhas de código necessárias para uma certa tarefa, além de associações proveitosas com outros *frameworks*, como o *FrameWork* responsivo *front-end – Foundation 3.0* e o *Data Table*, todos utilizados nesta aplicação.

Pondo em prática, esse framework, se compararmos com outros frameworks para desenvolvimento de aplicações, nos disponibiliza um conjunto de classes que podemos combinar e estender para construirmos nossas aplicações, nos poupando um considerável tempo de codificação. Estas classes que compõem a biblioteca do CodeIgniter já possuem uma estrutura de atributos e métodos que facilitam a implementação de tarefas comuns ao desenvolvimento de qualquer uma de nossas aplicações, tais como, conexão com o banco de dados, tratamento e consultas de dados retornados, construção de formulários e outros conteúdos HTML para a criação da interface visual da aplicação, entre várias outras.

O CodeIgniter, além da ampla biblioteca de classes que comentamos anteriormente, ainda dispõe de um conjunto de helpers, que podemos compreender como bibliotecas de funções, agrupadas de acordo com suas finalidades. Exemplo: um helper específico para manipulação de datas, que nos oferece um conjunto de funções que normalmente são necessárias, novamente nos poupando da tarefa de termos que codificar essas funções. Com isso ganhamos mais tempo e garantia da qualidade dos códigos utilizados, de forma que todo o código disponibilizado pelo framework é sempre testado exaustivamente antes de ser liberado. Além disso, existe toda uma comunidade de desenvolvedores colaborando com a melhora, correção e validação dos códigos que são incorporados ao CodeIgniter.

Os prints das principais telas de desenvolvimento deste aplicativo estão disponíveis no Apêndice 3.

4. RESULTADOS

Inicialmente, planejamos um sistema centralizado de controle somente ao cidadão comum, Neste sistema, teria um mínimo de fornecimento de dados e o restante seria efetuado remotamente pelo gerente do mesmo

Porém, a necessidade da abrangência dos Clubes de Tiro no sistema, se tornou preponderante, uma vez que é o local onde se dá a atividade da prática desportiva do tiro, sendo estes responsáveis pela concentração do público alvo em grande escala, ficando de fora apenas o cidadão comum com registro de arma de fogo na Polícia Federal, que também, direta ou indiretamente, participa deste universo, quer seja na forma de espectador ou ainda mais comumente quanto da realização dos testes obrigatório de tiro, ou ainda acumulando registros em ambos os sistemas federais SIGMA/SINARM.

O sistema desenvolvido contempla não somente um sistema simples de cadastro como alguns leitores podem pressupor, mas sim um sistema de controle de dados voltado a elaboração documental dos diversos tipos de processos em ambos os sistemas federais de fiscalização de produtos controlados.

Tendo em vista ainda a legislação vigente ser muito volátil, o sistema desenvolvido de forma modular, tenderá a ser de fácil alteração para suprir possíveis mudanças nos diversos tipos de processos se assim fizer necessário.

4.1 O SISTEMA

Nesta seção é descrito o sistema propriamente dito com suas telas e funcionalidades aplicado ao tema objeto deste projeto.

4.1.1 Tela de Login

A tela de Login (Figura 32), é o início do sistema, onde o usuário administrador ou comum, ambos previamente cadastrados, acessam o sistema através de um nome de usuário e uma senha individuais.

Além do login, a tela apresenta dois links, um para realizar o cadastro de acesso ao sistema (Figura 33), “Cadastre-se” e outro para resgate de senha extraviada (Figura 34), “Esqueci minha senha”.

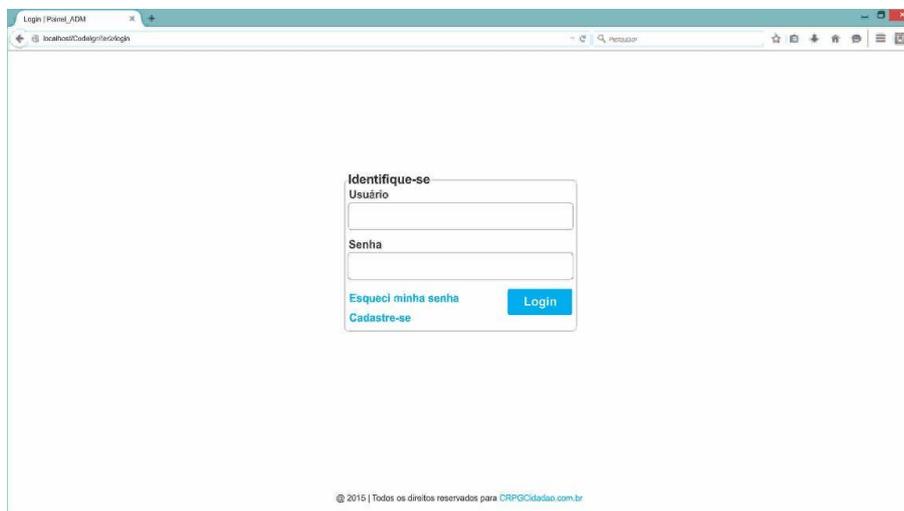


Figura 32. Tela de Login
Fonte: Autoria própria

4.1.2 Tela de Cadastre-se

A tela de Cadastre-se é a opção para o usuário que não possui acesso prévio ao sistema, mas deseja pedir autorização ao administrador para fazer uso. Esta tela contém as informações mínimas, mas necessárias para o startup do banco de dados. A Tela de Cadastre-se é ilustrada na Figura 33.

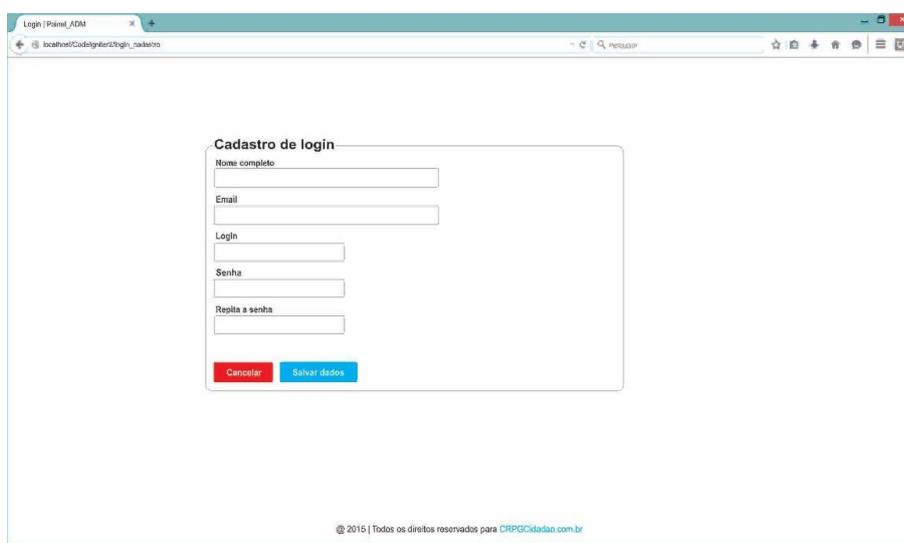


Figura 33. Tela de Cadastre-se
Fonte: Autoria própria

4.1.3 Tela de Recuperação de Senha.

A Figura 34 apresenta uma tela simples para o usuário já cadastrado, que por ventura esqueceu sua senha de acesso.

O usuário informa o email cadastrado, o sistema realiza a confirmação deste dado. Se estiver correto, envia a senha cadastrada para o usuário.

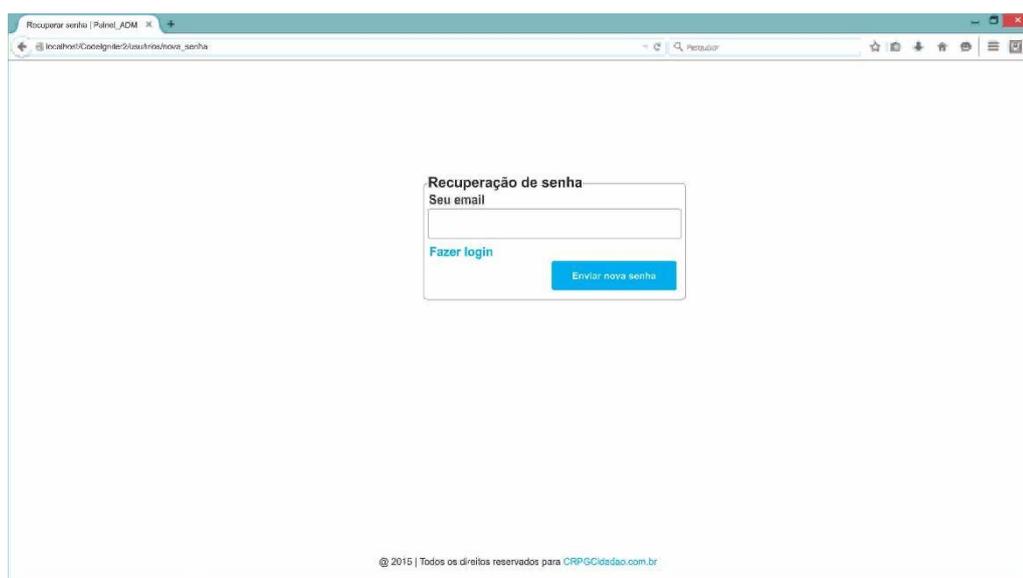


Figura 34. Tela de Recuperação de Senha
Fonte: Autoria própria

4.1.4 Tela Inicial

Após realizar Login com sucesso, o sistema redireciona o usuário à Tela Inicial, a qual apresentará duas opções de acesso, serviços junto ao Exército ou à Polícia Federal. Além dessas duas opções, a tela apresenta: (Figura 35).

- botão de saída do sistema (*logout*);
- botão para alterar senha do usuário; e
- identificação de usuário logado.

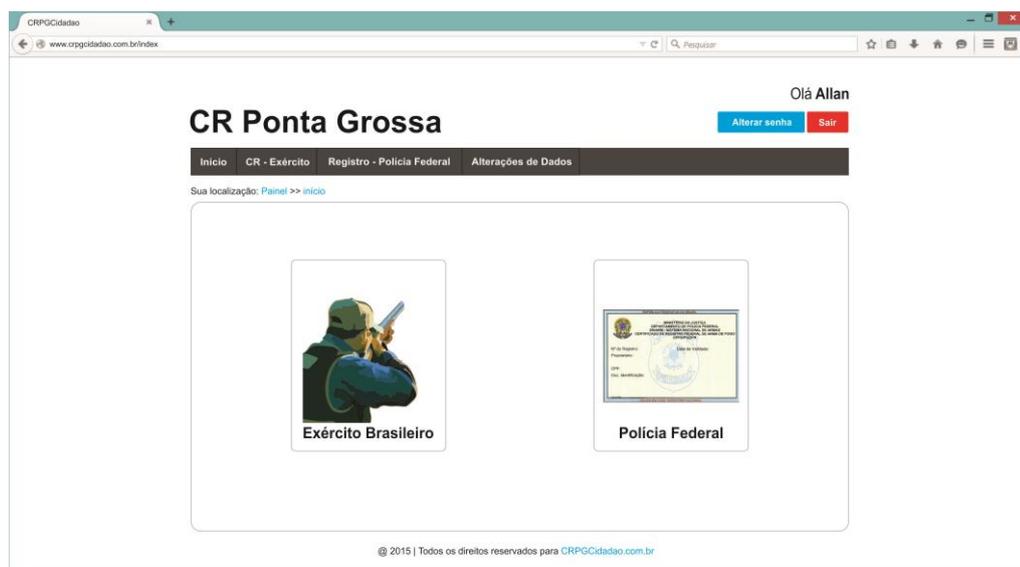


Figura 35. Tela Inicial
Fonte: Autoria própria

4.1.5 Tela Altera Senha

A tela Alterar Senha (Figura 36), possibilita o usuário cadastrado, apenas alterar sua senha, tendo os campos nome, login e email bloqueados para edição. Esta tela também pode ser acessada no menu configurações.

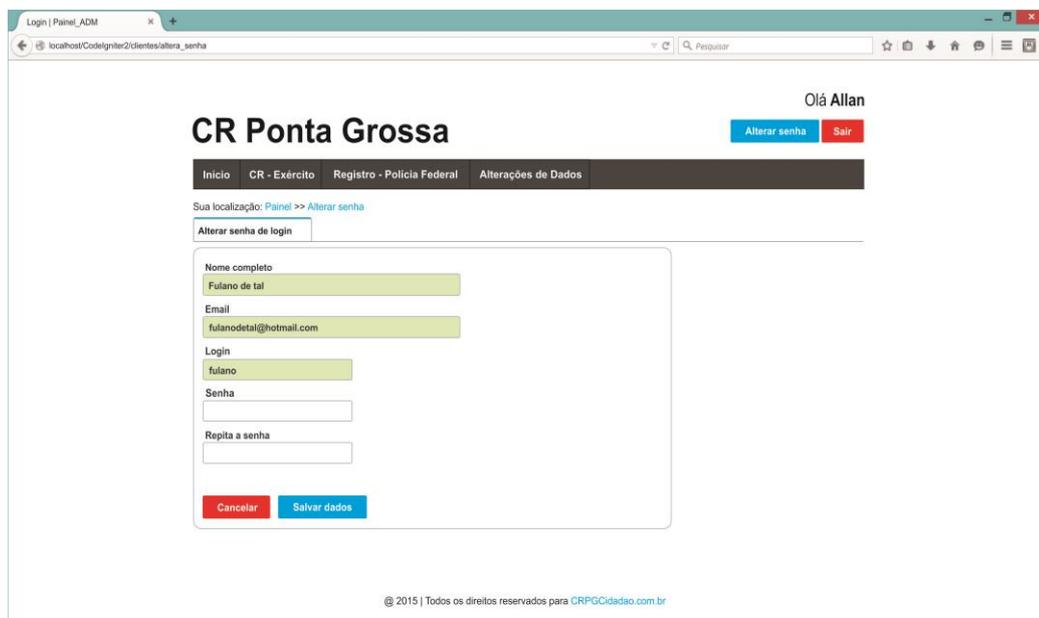


Figura 36. Tela Alterar Senha.
Fonte: Autoria própria

4.1.6 Tela Processos Exército

A tela Processos Exército (Figura 37), possibilita o usuário escolher o tipo de processo, ARMAMENTO ou CERTIFICADO DE REGISTRO (CR), através dos dois botões à disposição. Passando o cursor sobre os botões, o sistema disponibiliza as opções de processos à escolha do usuário.

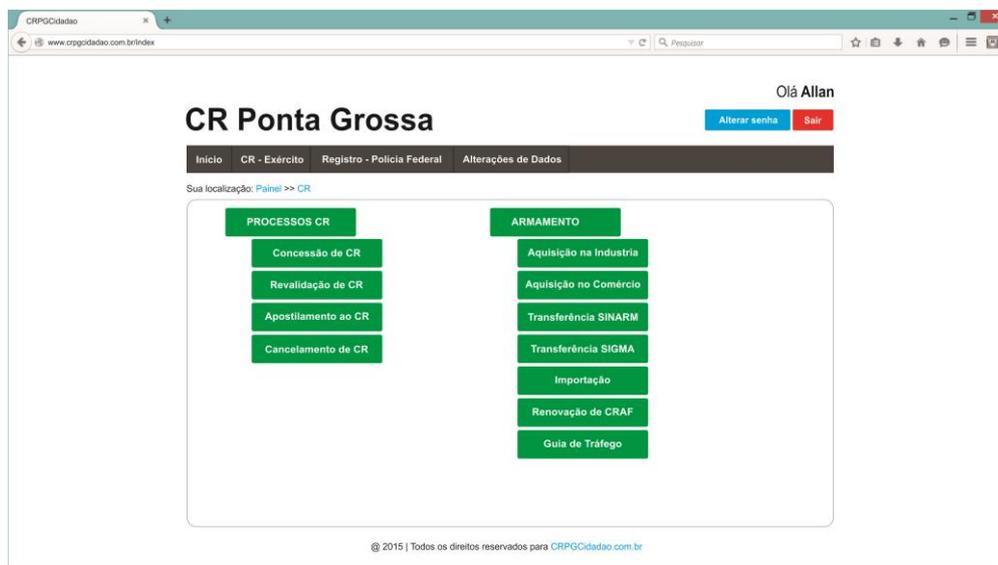


Figura 37. Tela Processos Exército.
Fonte: Autoria própria

4.1.7 Tela Processos Polícia Federal

A tela Processos Polícia Federal (Figura 38), possibilita ao usuário escolher o tipo de processo através da lista à disposição, clicando na opção desejada.

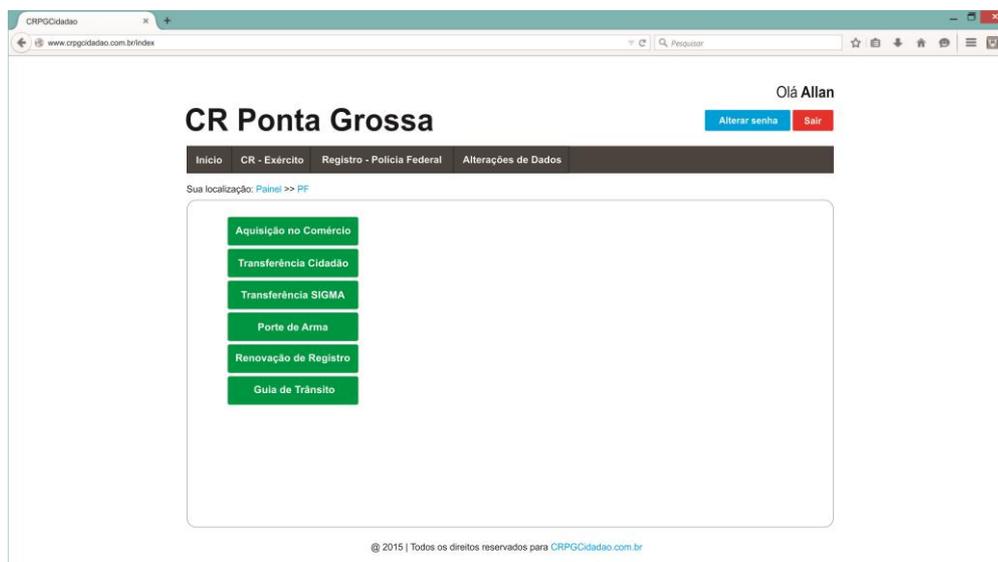


Figura 38. Tela Processos Polícia Federal.
Fonte: Autoria própria

4.1.8 Tela Notificação

A tela Notificação (Figura 39), apresenta ao usuário informações para contato e procedimentos, assim como qual documentação deverá providenciar para entrega, da seguinte forma:

- Caso seja usuário cadastrado pelo Gerente, o sistema verifica qual documento salvo está vencido e uma mensagem de notificação é apresentada apenas para que providencie os documentos vencidos para entrega.

- Caso o usuário ainda não tenha sido cadastrado pelo Gerente do Sistema, uma mensagem de notificação é apresentada para que providencie toda documentação pertinente para entrega.

Ainda, apresenta caixa para que o usuário assinale concordando com o contrato e botão de CONTINUAR PARA PAGAMENTO DO SERVIÇO.

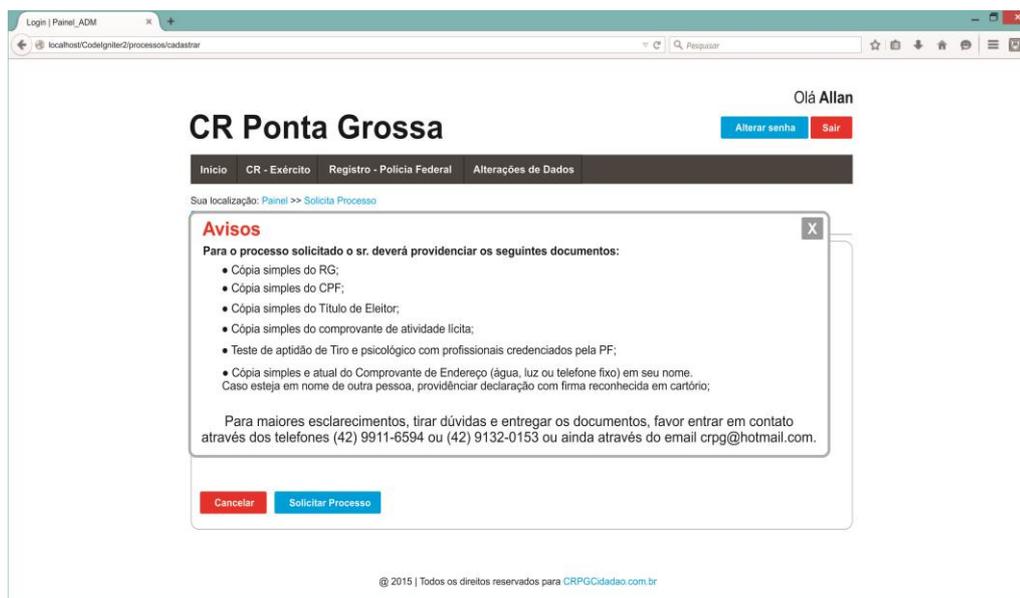


Figura 39. Tela Notificação.
Fonte: Autoria própria

4.1.9 Tela de Pagamento

Nesta tela o usuário faz o pagamento do serviço escolhido anteriormente nas Telas de Processos. O modelo escolhido foi o do “PAG SEGURO”. A Tela de Pagamento é ilustrada na Figura 40.

Olá Allan

CR Ponta Grossa

[Alterar senha](#) [Sair](#)

Início CR - Exército Registro - Polícia Federal Alterações de Dados

Sua localização: Painel >> Pagamento de Processo

Cartão de Crédito Boleto Bancário

PAGUE COM CARTÃO DE CRÉDITO

VISA [Dúvidas em 2 Cartões](#)

Número do Cartão

Nome do titular (como está gravado no Cartão)

Data de Validade
MM/YYYY

Código de Segurança

Opções de Parcelamento
R\$ 99,90x12

@ 2015 | Todos os direitos reservados para CRPGCidadao.com.br

Figura 40. Tela de Pagamento.
Fonte: Autoria própria

4.1.10 Pagamento Efetuado com Sucesso

A tela de Pagamento Efetuado com Sucesso (Figura 41), confirma o pagamento efetuado pelo usuário.

Olá Allan

CR Ponta Grossa

[Alterar senha](#) [Sair](#)

Início CR - Exército Registro - Polícia Federal Alterações de Dados

Sua localização: Painel >> Solicita Processo

Pagamento efetuado com sucesso. Aguarde contato via email onde será disponibilizado o link para acompanhamento do processo.

[OK](#)

@ 2015 | Todos os direitos reservados para CRPGCidadao.com.br

Figura 41. Pagamento Efetuado com Sucesso.
Fonte: Autoria própria

4.1.11 Tela Inicial para o Gerente do Sistema

Após realizado Login com sucesso, o sistema redireciona o usuário à Tela Inicial para o Gerente do Sistema (Figura 42), a qual apresentará o menu principal por tema e atrelados a cada um deste, um submenu específico com funções relacionadas ao tema. Além do menu, a tela apresenta:

- botão de logoff do sistema (SAIR);
- botão para alterar senha do usuário (ALTERAR SENHA);
- identificação de usuário logado; e
- *breadcrumbs*¹.

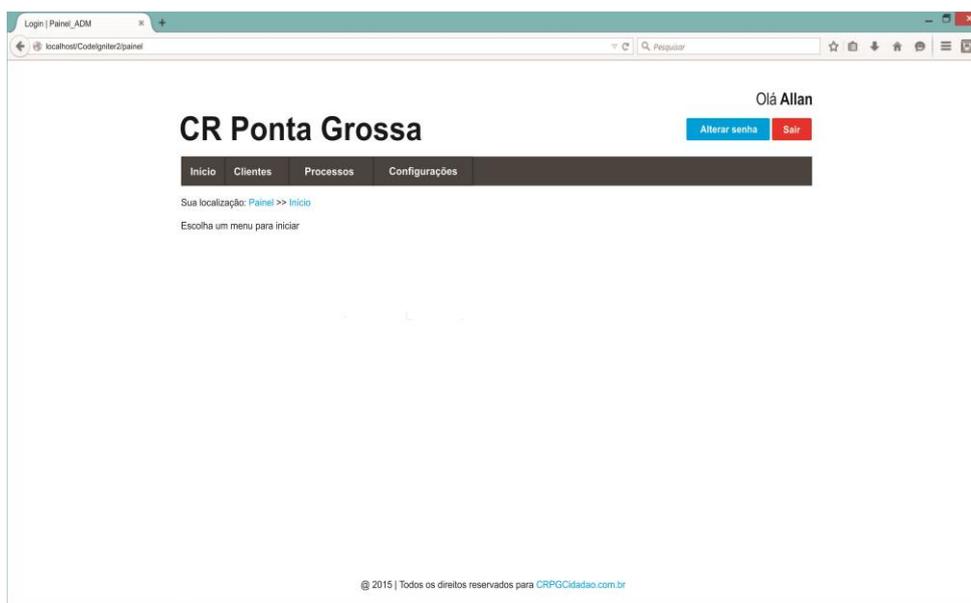


Figura 42. Tela Inicial para o Gerente do Sistema
Fonte: Autoria própria

4.1.12 Tela de Cadastros

Esta tela possui duas formas de trabalho. Se o usuário clicar no menu “cadastros” a tela apresenta dois ícones: clientes e armas (Figura 43). O primeiro listará todos os clientes cadastrados e o segundo todas as armas cadastradas.

¹ O widget Breadcrumbs ou breadcrumb trail é um auxílio à navegação usado em interfaces de usuário. Ele permite aos usuários manter o controle de suas localizações dentro de programas ou documentos.

A segunda forma de trabalho é clicando ao lado do menu “cadastrar” a qual expandirá o submenu, dando as opções de clientes, armas e gerenciar.

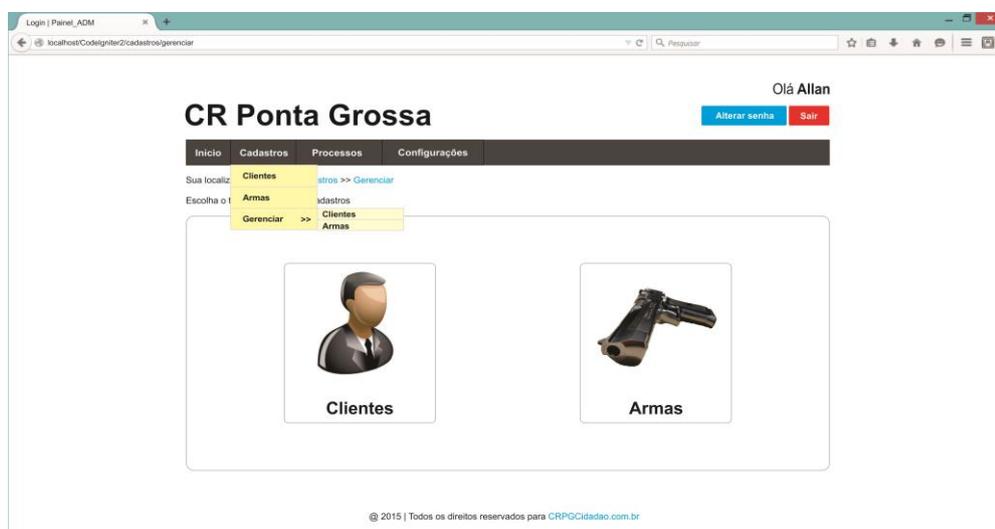


Figura 43. Tela de Cadastros
Fonte: Autoria própria

4.1.13 Tela de Cadastros de Clientes

Na Tela de Cadastro de Clientes (Figura 44), o usuário realizará o cadastro completo dos dados de um cliente, a começar dos dados básicos seguido do seu endereço e documentos. Cada etapa é dividida em abas temáticas devendo o usuário clicar no botão salvar sempre que completar a sequência de informações.

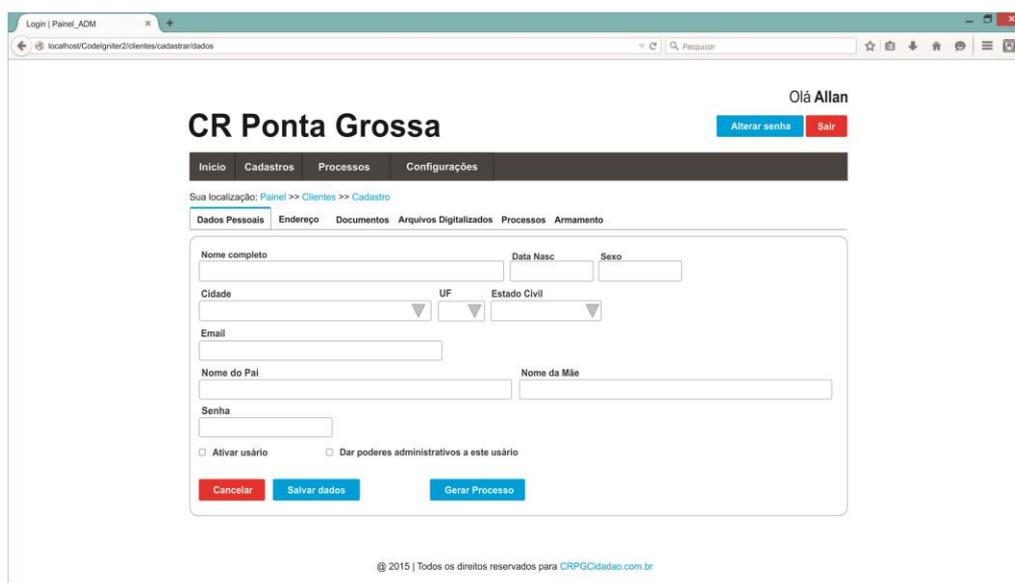
A screenshot of the 'Cadastro de Clientes' form in the CR Ponta Grossa system. The user is logged in as 'Allan'. The form is titled 'Cadastro' and has several tabs: 'Dados Pessoais', 'Endereço', 'Documentos', 'Arquivos Digitalizados', 'Processos', and 'Armamento'. The 'Dados Pessoais' tab is active, showing fields for 'Nome completo', 'Data Nasc', 'Sexo', 'Cidade', 'UF', 'Estado Civil', 'Email', 'Nome do Pai', and 'Nome da Mãe'. There is also a 'Senha' field and two checkboxes: 'Ativar usuário' and 'Dar poderes administrativos a este usuário'. At the bottom of the form are three buttons: 'Cancelar', 'Salvar dados', and 'Gerar Processo'. The footer contains the text '@ 2015 | Todos os direitos reservados para CRPGCidade.com.br'.

Figura 44. Tela de Cadastros de Dados
Fonte: Autoria própria

Ainda nesta tela o sistema apresenta duas abas, uma onde o usuário pode realizar o *upload*² dos documentos digitalizados do cliente, podendo ainda visualizar, editar ou excluir o arquivo (Figura 45), além de ser possível listar os processos existentes do cliente podendo ser editado ou excluído (Figura 46).

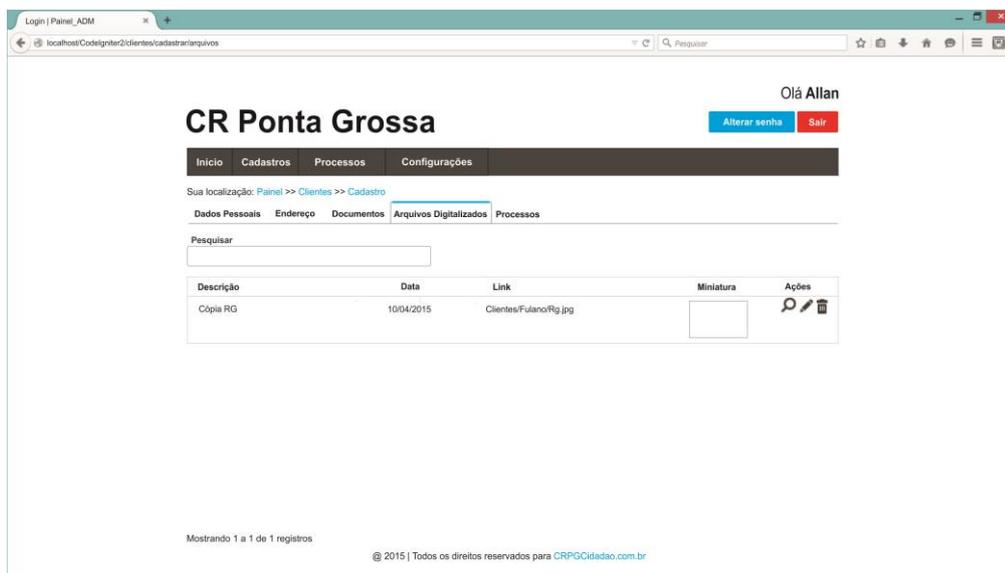


Figura 45. Tela de Upload de Arquivos
Fonte: Autoria própria

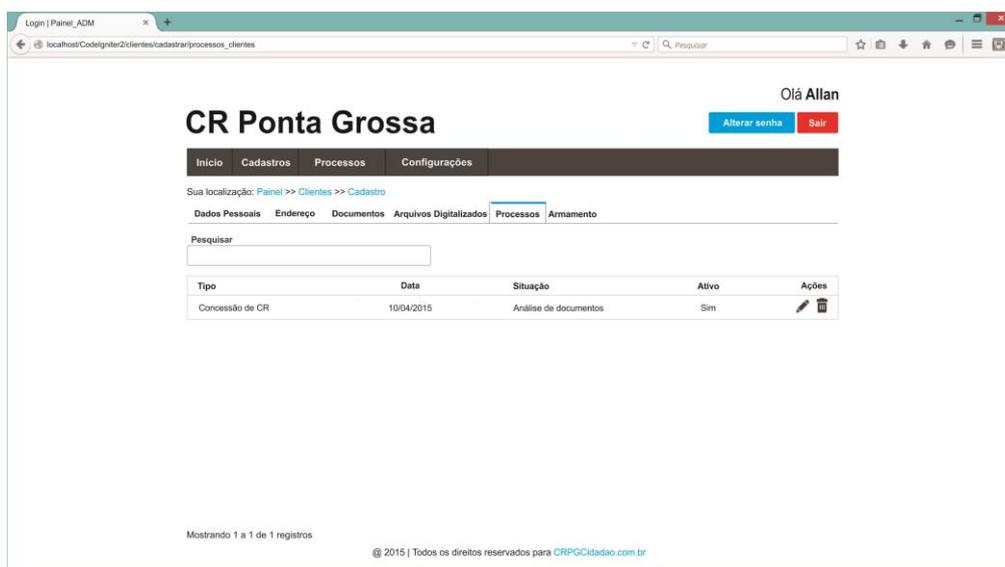


Figura 46. Tela de Processos Cadastrados do Cliente
Fonte: Autoria própria

² Upload é um termo da língua inglesa com significado referente à ação de enviar dados de um computador local para um computador ou servidor remoto, geralmente através da internet.

4.1.14 Tela de Gerenciamento de Cadastros

O gerenciamento de cadastro é dividido em duas telas:

a. Clientes com cadastro iniciado (Figura 47): os cadastros podem ser editados ou excluídos. A tela indica se o cadastro foi concluído ou se o cliente possui documentos vencidos; e

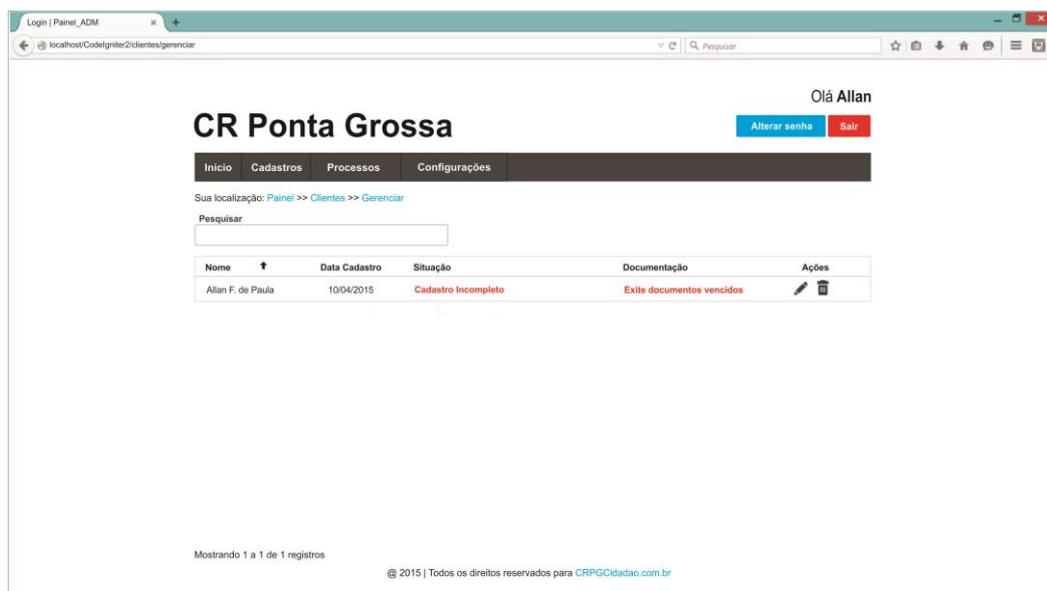


Figura 47. Tela de Gerenciamento de Clientes

Fonte: Autoria própria

b. Armas cadastradas (Figura 48): da mesma forma os cadastros podem ser editados ou excluído além da tela informar se o CRAF encontra-se vencido ou não.

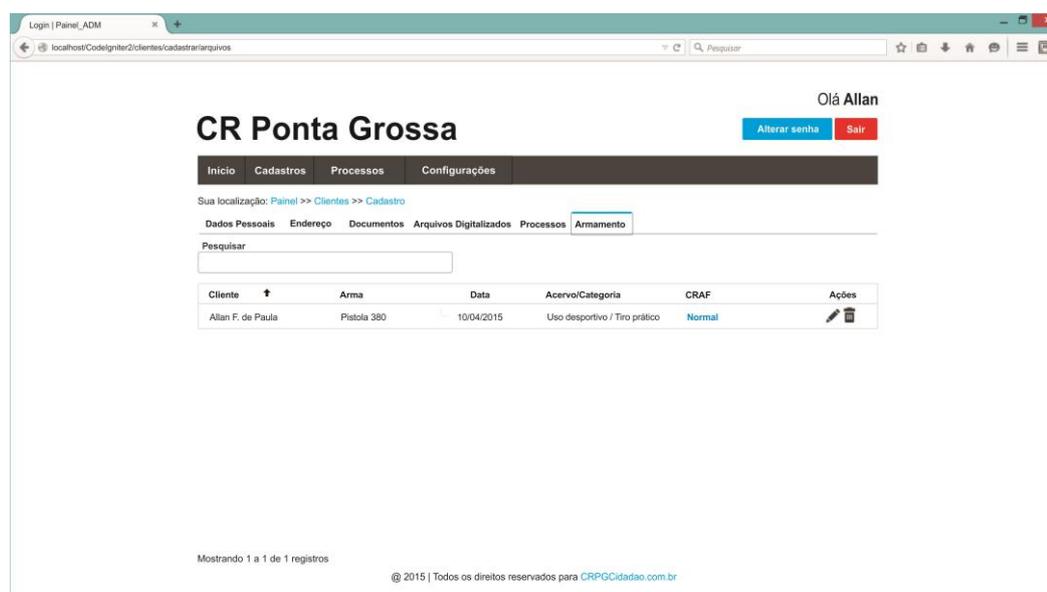


Figura 48. Tela de Gerenciamento de Armas

Fonte: Autoria própria

4.1.15 Tela de Cadastro de Processos

A Tela de Cadastro de Processos (Figura 49), possibilita ao Gerente do Sistema inserir as informações pertinentes à abertura de processos para fins de controle.

Nesta tela existe ainda opção do Gerente inserir informações referente ao histórico do processo, tais como: entrega de documentos, pagamentos realizados, entre outras.

The screenshot shows a web browser window with the URL 'localhost/CodeIgniter2/processos/cadastros'. The page title is 'CR Ponta Grossa'. The user is logged in as 'Allan' and has options to 'Alterar senha' or 'Sair'. The navigation menu includes 'Inicio', 'Cadastros', 'Processos', and 'Configurações'. The breadcrumb trail is 'Sua localização: Painel >> Processos >> Cadastro'. The form is titled 'Tipo de Processo' and contains the following fields:

- Cliente: ALLAN FRANCIS DE PAULA
- Tipo de Processo: (dropdown menu)
- Data Inicio: (text input)
- Data Término: (text input)
- Protocolo: (text input)
- Escolher Armamento: (dropdown menu)

At the bottom of the form are two buttons: 'Cancelar' (red) and 'Gerar Processo' (blue). A footer note reads: '@ 2015 | Todos os direitos reservados para CRPGCidadeao.com.br'.

Figura 49. Tela de Cadastro de Processos
Fonte: Autoria própria

4.1.16 Tela de Gerenciamento de Processos

A tela de Gerenciamento de Processos (Figura 50), lista todos os processos cadastrados classificando-os por cliente e estado de atividade. Nesta tela o Gerente pode editar ou excluir o processo.

The screenshot shows a web browser window with the URL 'localhost/CodeIgniter2/processos/gerenciar'. The page title is 'CR PG'. The user is logged in as 'Allan' and has options to 'Alterar senha' or 'Sair'. The navigation menu includes 'Inicio', 'Cadastros', 'Processos', and 'Configurações'. The breadcrumb trail is 'Sua localização: Painel >> Processos >> Cadastro'. There is a search bar labeled 'Pesquisar'. Below it is a table with the following data:

Cliente	Tipo	Data	Observação	Ativo	Ações
Allan F. de Paula	Concessão de CR	10/04/2015	Falta entregar foto 3x4	sim	 

A footer note reads: '@ 2015 | Todos os direitos reservados para CRPGCidadeao.com.br'.

Figura 50. Tela de Gerenciamento de Processos
Fonte: Autoria própria

4.1.17 Tela de configurações do sistema

A partir da tela de Configuração do Sistema (Figura 51), o Gerente pode, de forma simples, personalizar seu sistema com os dados da sua empresa, logo, marca e e-mail de administrador, alterar senha de usuários como já explanado no item 3.4.5 (Figura 36), excluir login de usuários ou apenas torná-los inativos, impossibilitando assim, o acesso do mesmo ao sistema (Figura 52).

A exclusão do login do usuário não irá excluir o cadastro de informações, armas e processos do cliente. Apenas excluirá as informações de login do usuário.

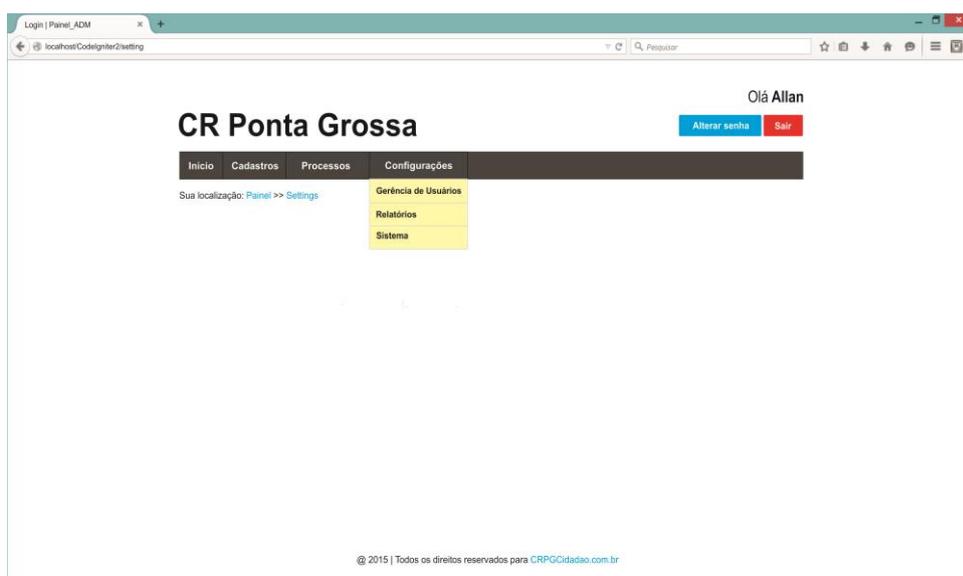


Figura 51. Tela de Configurações do Sistema
Fonte: Autoria própria

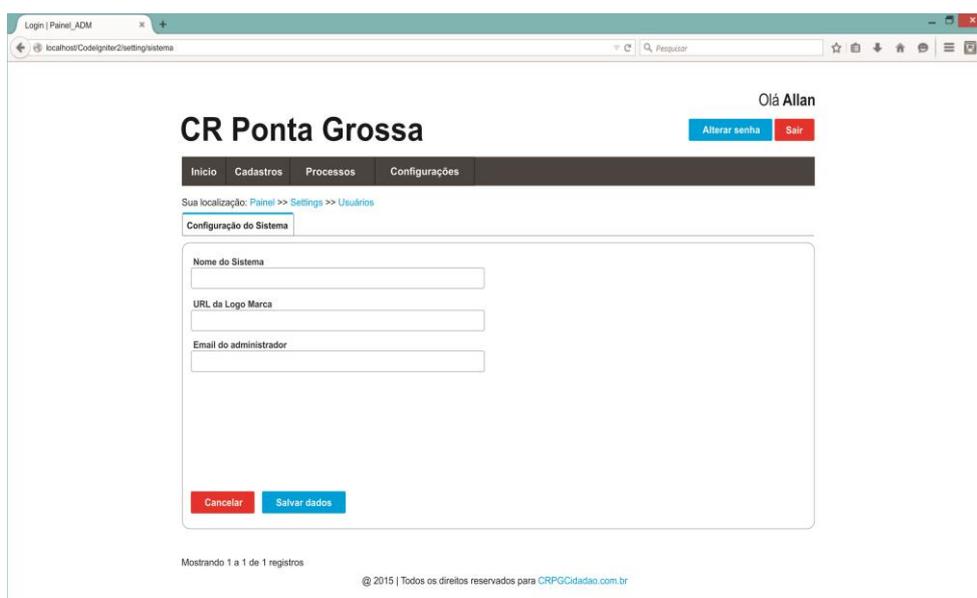


Figura 52. Tela de Personalização do Sistema
Fonte: Autoria própria

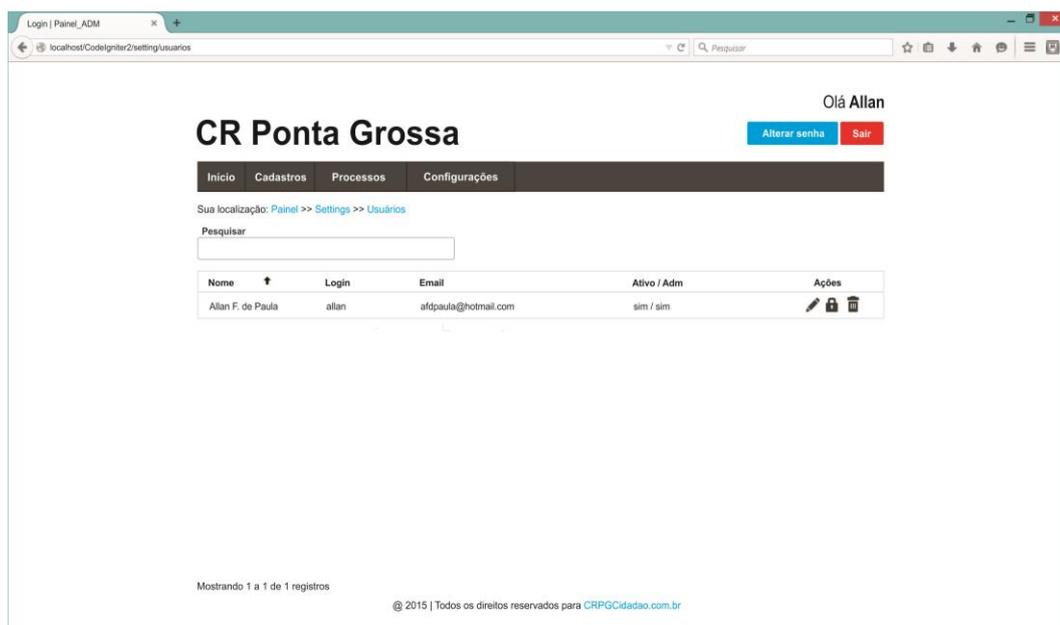


Figura 53. Tela de Gerenciamento de Login
Fonte: Autoria própria

4.1.18 Tela de Relatórios

Nesta tela, o Gerente pode gerar em pdf todo tipo de relatório. Aparece os links para escolha e com apenas um click o sistema gera o relatório escolhido. A Figura 54 ilustra a Tela de Relatórios

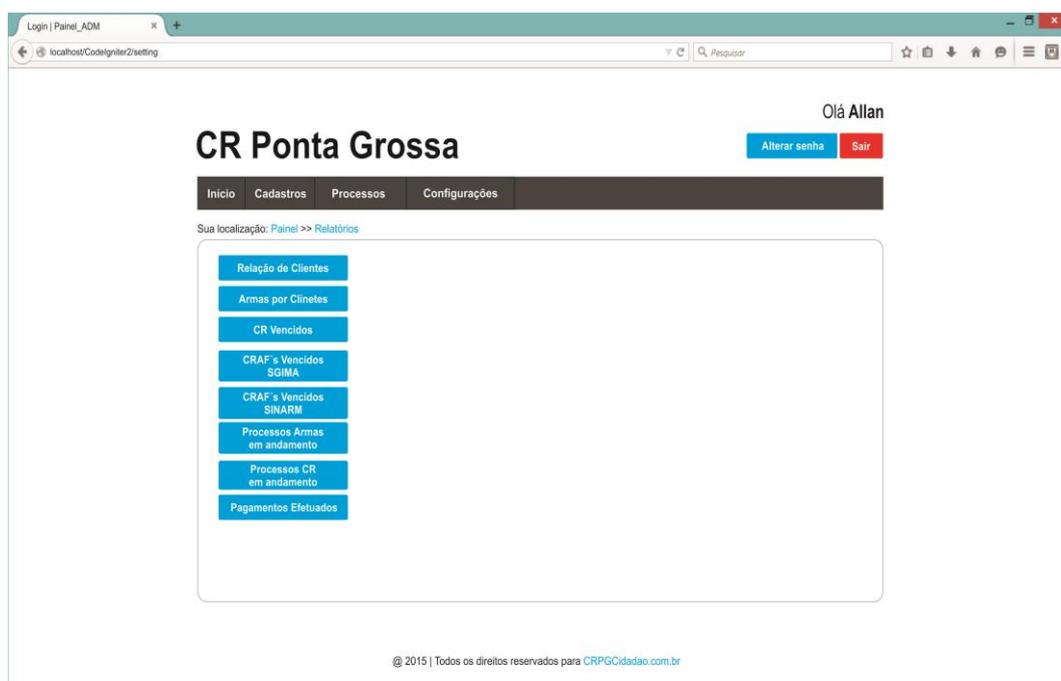


Figura 54. Tela de Relatório
Fonte: Autoria própria

5. CONCLUSÃO

Gates (1999) afirma que " tornando-se conscientes do poder da Web, os cidadãos não mais aceitarão a ideia de que serviços públicos devem ser apenas quase bons." Se o governo, que é geralmente a maior "empresa" de qualquer país, for um líder no uso da tecnologia, automaticamente melhorará as aptidões técnicas da nação e impulsionará a mudança para a sociedade da informação.

Em nossas diversas pesquisas, não obtivemos conhecimento de nenhum sistema público ou privado disponibilizado ao cidadão comum que queira solicitar ou fazer-se valer dos serviços dos órgãos de fiscalização federal no tocante a produtos controlados.

Não há interesse por parte governamental de se implementar facilidade para tais tipos de serviços, uma vez que a política desarmamentista no Brasil é imperativa, apesar das inúmeras pressões socioeconômicas que os legisladores federais tem sofrido.

A viabilidade do desenvolvimento deste projeto, durante nossas pesquisas, nos mostrou o enorme potencial não só financeiro, mas também a necessidade de uma ferramenta para organização básica e extremamente necessária aos Clubes de Tiro e cidadãos participantes de uma classe desportiva utilizadora de equipamentos e produtos controlados primordiais para a prática individual e coletiva.

O atirador desportivo competidor não fica enclausurado em sua cidade, há uma enorme mobilidade deste agora, uma vez que as guias de tráfego fornecidas outrora, que eram divididas em locais, estaduais e nacionais, conforme a abrangência da participação do atirador em competições, hoje só existe a de âmbito nacional, facilitando assim sua participação em competições nível nacional.

Dentro desse contexto outra ideia de viabilização para o sistema foi pré-comprovada através de contatos realizados com despachantes do ramo e oriundos de diversas regiões do país. A ideia que surgiu foi de disponibilizar o sistema, na forma de franquia, onde o gerenciamento do sistema ficaria a cargo do cliente, neste caso o despachante, e a manutenção caberia aos desenvolvedores, tendo o cliente a incumbência de pagar um percentual do movimento de processos realizados num período de tempo pré-determinado e os desenvolvedores fornecerem todo suporte

técnico remotamente.

Porém, depois de tanta pesquisa de campo para que este projeto pudesse ter sucesso, nosso grande desafio estava mesmo focado nas superações pessoais dos conhecimentos obtidos somente em sala de aula.

O anseio de colocar em prática os conhecimentos, na forma de uma ferramenta que poderá ser utilizada por diversas pessoas, ao mesmo tempo que nos motivou, nos preocupou pela falta de experiência no mercado como programadores.

A habilidade do trabalho em dupla, ora desenvolvendo juntos ora desenvolvendo remotamente, mostrou o quão ainda temos que estudar e nos aperfeiçoar neste ramo.

Porém, na busca de atender um tipo de usuário esquecido pelos órgãos federativos de controle, mas que possui uma importância tal que pode atentar contra a segurança da sociedade desportiva e brasileira como um todo, mostra o quão necessário se faz com que mais pessoas invistam e insistam no desenvolvimento das mais diversas tecnologias da informação e contribuindo assim para o crescimento de nosso país.

Com um mínimo esforço um sistema que armazena todas as informações pertinentes, que gera todos os tipos de relatórios necessários ao controle das informações e produz de forma automática toda a documentação segundo modelos pré-estabelecidos de forma a facilitar, integrar e assegurar ao cidadão comum e aos diversos órgãos de controle a eficiência e eficácia da gestão da informação dos produtos controlados no meio desportivo e da segurança pessoal.

5.1 DIFICULDADES ENCONTRADAS

Neste tópico, discorreremos sobre as dificuldades encontradas a partir do pré-projeto até a elaboração do projeto final, tendo para tanto, elencados os principais itens desafiadores para melhor entendimento do resultado final.

1. Público Alvo

Dentre as maiores dificuldades encontradas nas pesquisas para o pré-projeto, foi a definição exata do público alvo.

Conforme já explanado na introdução, há neste setor dois públicos alvos distintos, cidadão comum com arma registrada na polícia federal e o cidadão comum

com certificado de registro no Exército Brasileiro, ambos com sistemas de controle distintos, desconexos e burocracias internas, que fogem muitas vezes ao previsto na legislação vigente, ficando à mercê do poder discricionário da autoridade local responsável.

Dessa forma, o trabalho de pesquisa foi amplo, tanto na parte do cidadão interessado no serviço fornecido pela instituição federal, quanto ao *modus operandi*, cobrado por esta, do cidadão.

Foram realizadas visitas aos Clubes de Tiro locais, onde foram levantadas as ansiedades e necessidades dos interessados neste setor, com relação à agilidade na elaboração dos processos desejados, facilidade no trato de informações pessoais e satisfação no atendimento e êxito dos seus pedidos.

Esta demanda advém hoje, da complexidade que os organismos federais de fiscalização impõe para a efetivação dos diversos protocolos em seus sistemas, levando o cidadão leigo em legislação e sistemas informatizados, a procurar os serviços de terceiros para seus diversos anseios.

2. Legislação

A legislação que envolvem os produtos controlados, mais especificamente, as armas e seus insumos, vem sofrendo grande pressão social e econômica dos diversos segmentos de nossa sociedade, a fim de facilitar o acesso aos mesmos.

Como a discussão sobre o Estatuto do Desarmamento é muito grande e tem a demanda de tramitação no poder legislativo maior da nação, o Congresso Nacional, os órgãos fiscalizadores federais, no uso de suas atribuições discricionárias realizam diversas alterações, ora propiciando agilidade em seus processos e procedimentos, ora dificultando e criando entraves nos mesmos.

Somente no Exército Brasileiro, a legislação mudou 03 (três) vezes durante os anos de 2014 e 2015, tendo os modelos e condutas dos diversos tipos de processos, sofrido alterações significativas e assim alterando as rotinas previstas e estabelecidas no planejamento inicial.

Já a Polícia Federal, onde seus processos são de certa forma simples, a opção vislumbrada de mudança foi aumentar o valor das taxas para os diversos serviços.

Nestes aspectos a dificuldade encontrada, foi trabalhar as informações de

maneira adequada para abstraí-las para um universo de desenvolvimento de sistemas.

3. Framework's

Tendo estabelecido o Codeigniter como *Framework* principal e voltado à programação orientada ao objeto e no padrão MVC, o outro fator importante foi definir outras ferramentas que propiciassem facilidade, agilidade e segurança interagindo de forma direta e conexa com o Codeigniter.

Diversas foram as ferramentas pesquisadas durante o pré-projeto, criando de começo uma enorme dúvida em quais utilizar, levando em consideração praticidade, simplicidade e leveza.

Primeiramente e mais importante foi a definição de usar ferramentas *open source* e em segundo plano e não menos importante, ferramentas que estivessem acessíveis dentro do nosso grau de conhecimento devido ao tempo estimado para a elaboração do projeto final, uma vez que o próprio Codeigniter já trazia uma complexidade considerável para nós.

5.2 TRABALHOS FUTUROS

Como trabalho futuros, pretendemos:

1. Ampliar os módulos do sistema para que este possa controlar ou dar suporte à empresas que necessitam utilizar, transportar ou manipular os produtos controlados;
2. Desenvolver um controle centralizado dos sites comercializados, com o objetivo de controle financeiro; e
3. Agregar nos módulos, um sistema para competição, uma vez que é incumbência dos Clubes de Tiro realizarem provas de tiro em suas dependências a fim de classificar e justificar suas existências e também controlar o nível dos atiradores a eles filiados, conforme normas legais.

REFERÊNCIAS

BARNABÉ, Grasiela, Um Estudo Comparativo entre as linguagens de programação PHP, ASP e JSP. Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí, Rio do Sul, 2010.

BALTHAZAR, Glauber da Rocha, GUIMARÃES, Fabio Mendes Ramos, PAULA, Melise Maria Veiga de, FILHO, Elio Lovisi, Uma Abordagem Prática sobre a Aplicação do Padrão MVC com o *Framework* Struts, Faculdade Metodista Granbery, 2006.

CARMO, Carla Soraia Leandro do, Automação de detalhamento de peças padronizadas em concreto armado via CAD e programação orientada a objetos, Departamento de Engenharia de Estruturas, Universidade Federal de Minas Gerais, 2001.

CYSNEIROS, Luiz Marcio, Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, 2001.

DECRETO Nº 3.665, DE 20 DE NOVEMBRO DE 2000.REGULAMENTO PARA A FISCALIZAÇÃO DE PRODUTOS CONTROLADOS (R-105).

ELLISLAB, INC. CodeIgniter User Guide Version 2.1.4. CodeIgniter, 2014. Disponível em: http://codeigniter.com/user_guide/. Acesso em: 30 Outubro 2014.

ESPÍNDOLA, Carlos Eduardo; DE OLIVEIRA, João Batista Ferri, FORMIGA, Maurício Marinho, A Tecnologia da Informação como meio para facilitar o acesso do cidadão aos Serviços Públicos, IV Congresso CONSAD de Gestão Pública, 2011.

FARINELLI, Fernanda, Conceitos Básicos de Programação Orientada a Objetos, Instituto Federal Sudeste de Minas Gerais 2007.

GABARDO, Ademir Cristiano, PHP e MVC com Codeigniter, Editora Novatec, 2012.

GONÇALVES, Edson, Desenvolvendo Aplicações Web com JSP Servlet, JavaServer Faces, Hibernate, EJB 3 Persistence e Ajax, Editora Ciência Moderna Ltda., 2007.

GUEDES, Gilleanes T. A., UML – Uma Abordagem Prática, 2008.

JOHNSON. Ralph E., Components, *Frameworks*, Patterns, 1997, White paper.
KICZALES, Gregor, LAMPING John, MENDHEKAR , Anurag, MAEDA, Chris, LOPES, Cristina Videira, LOINGTIER, Jean-Marc, and John Irwin. Aspect-Oriented Programming, em European Conference on Object-Oriented Programming, 1997.

JOHNSON, Rod, How to Design *Frameworks*, em Conference on Object-Oriented Programming, Languages and Applications, 1993.

LASKOSKI, Jackson, Programação Orientada a Objetos, apostila, Disponível em: <<http://www.jack.eti.br/>>. Acesso em 19 de Nov. de 2013.

LEI Nº 10.826, DE 22 DE DEZEMBRO DE 2003.

MACORATTI, José Carlos. Padrões de Projeto : O modelo MVC.Model View Controller, Disponível em: <http://www.macoratti.net/vbn_mvc.htm> .Acesso em 01 de Out. de 2013.

MEYER, Bertrand. Object-Oriented Software Construction. Editora Prentice-Hall 1997.

RICARTE, Ivan Luiz Marques, Programação Orientada a Objetos: Uma Abordagem com Java, Departamento de engenharia de computação e automação industrial, Universidade Estadual de Campinas 2001.

RODRIGUES, Francisco Aparecido, Técnicas de orientação ao objetos para computação científica paralela, Departamento de Física e Informática, Instituto de Física de São Carlos, Universidade de São Paulo, 2004.

RODRIGUES, Joel, 2015. Disponível em <http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>.

Acesso em 03 de abril de 2015.

STADZISZ, Paulo César, 2002. Projeto de Software usando a UML, Centro Federal de Educação Tecnológica do Paraná, Departamento Acadêmico de Informática.

SILVA, Elaine Quinteiro da, MOREIRA, Dilvan de Abreu, Um *Framework* de Componentes para o Desenvolvimento de Aplicações Web Robustas de Apoio à Educação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2004.

SOARES. Sérgio, BORBA. Paulo, Desenvolvimento de Software Orientados a Aspectos Utilizando RUP e AspectJ, Centro de Informática, Universidade Federal de Pernambuco, 2004.

TALIGENT, Leveraging. Object-Oriented *Frameworks*, White Paper, 1995.

The NetBeans E-commerce Tutorial.Designing the Application, Disponível em: <<https://netbeans.org/kb/docs/javaee/ecommerce/design.html>>. Acessado em 30 de Out de 2014

APÊNDICE A - SCRIPTS DAS TABELAS DA APLICAÇÃO

```
-- Geração de Modelo físico  
-- Sql ANSI 2003 - brModelo.
```

```
CREATE TABLE armamento (  
  qtd_canos Texto(1),  
  comprimento_cano Texto(1),  
  País_Fabricação Texto(1),  
  Nr_SINARM_SIGMA Texto(1),  
  Modo_Aquisição Texto(1),  
  id_armamento Texto(1) PRIMARY KEY,  
  espécie Texto(1),  
  marca Texto(1),  
  modelo Texto(1),  
  nr_serie Texto(1),  
  calibre Texto(1),  
  funcionamento Texto(1),  
  acabamento Texto(1),  
  qtd_raias Texto(1),  
  sentido_raias Texto(1),  
  capacidade Texto(1),  
  alma Texto(1),  
  acervo Texto(1),  
  modalidade Texto(1),  
  categoria Texto(1),  
  razão_social_revendedor Texto(1),  
  cnpj_cpf_revendedor Texto(1),  
  nr_nota_fiscal Texto(1),  
  data_nt_fiscal Texto(1),  
  id_aquis Texto(1),  
  id_pais Texto(1)  
)
```

```
CREATE TABLE uf (  
nome Texto(1),  
id_uf Texto(1) PRIMARY KEY  
)
```

```
CREATE TABLE possui (  
id_uf Texto(1),  
id_empresa Texto(1),  
FOREIGN KEY(id_uf) REFERENCES uf (id_uf)  
)
```

```
CREATE TABLE modo_aquisicao (  
id_aquis Texto(1) PRIMARY KEY,  
nome Texto(1)  
)
```

```
CREATE TABLE registro_sinarm (  
nr_cadastro_sinarm Texto(1),  
id_sinarm Texto(1) PRIMARY KEY,  
orgão_expedidor Texto(1),  
vencimento Texto(1),  
uf_orgão_expedidor Texto(1),  
data_expedicao Texto(1),  
nr_registro Texto(1),  
id_armamento Texto(1),  
FOREIGN KEY(id_armamento) REFERENCES armamento (id_armamento)  
)
```

```
CREATE TABLE sigma (  
id_sigma Texto(1) PRIMARY KEY,  
vencimento Texto(1),  
nr_cadastro_sigma Texto(1),
```

```
nr_registro Texto(1),  
data_expedicao Texto(1),  
id_armamento Texto(1),  
FOREIGN KEY(id_armamento) REFERENCES armamento (id_armamento)  
)
```

```
CREATE TABLE processos (  
id_processos Texto(1) PRIMARY KEY,  
data Texto(1),  
protocolo Texto(1),  
id_tipo_processo Texto(1),  
id_cr Texto(1),  
id_usuario Texto(1),  
id_armamento Texto(1),  
FOREIGN KEY(id_armamento) REFERENCES armamento (id_armamento)  
)
```

```
CREATE TABLE documentos (  
id_documentos Texto(1) PRIMARY KEY,  
data_emissao Texto(1),  
pasta Texto(1),  
validade Texto(1),  
id_usuario Texto(1),  
id_lista_doc Texto(1)  
)
```

```
CREATE TABLE EMPRESA_ONDE_TRABALHA (  
tel_com Texto(1),  
nome Texto(1),  
id_empresa Texto(1) PRIMARY KEY,  
endereco_comercial Texto(1),  
cnpj Texto(1)  
)
```

```
CREATE TABLE usuario_empresa (  
id_usuario Texto(1),  
id_empresa Texto(1),  
FOREIGN KEY(id_empresa) REFERENCES EMPRESA_ONDE_TRABALHA  
(id_empresa)  
)
```

```
CREATE TABLE auditoria (  
id_auditoria Texto(1) PRIMARY KEY,  
operacao Texto(1),  
query Texto(1),  
data_hora Texto(1),  
observacao Texto(1),  
nome Texto(1),  
id_usuario Texto(1)  
)
```

```
CREATE TABLE cidade (  
id_cidade Texto(1) PRIMARY KEY,  
nome Texto(1),  
id_uf Texto(1),  
FOREIGN KEY(id_uf) REFERENCES uf (id_uf)  
)
```

-- Erro: Nome de tabela duplicado (este erro compromete a integridade referencial)!

```
CREATE TABLE possui (  
id_uf Texto(1),  
id_rg Texto(1),  
FOREIGN KEY(id_uf) REFERENCES uf (id_uf)  
)
```

```
CREATE TABLE paginas (  

```

```
id_paginas Texto(1) PRIMARY KEY,  
titulo Texto(1),  
conteudo Texto(1),  
slug Texto(1)  
)
```

```
CREATE TABLE settings (  
id_settings Texto(1) PRIMARY KEY,  
valor_config Texto(1),  
nome_config Texto(1)  
)
```

```
CREATE TABLE midia (  
descricao Texto(1),  
id_midia Texto(1) PRIMARY KEY,  
arquivo Texto(1),  
nome Texto(1)  
)
```

```
CREATE TABLE usuarios (  
UF_Nascimento Texto(1),  
Cidade_Nascimento Texto(1),  
País_Nascimento Texto(1),  
id_usuario Texto(1) PRIMARY KEY,  
nome Texto(1),  
profissão Texto(1),  
cpf Texto(1),  
data_nascimento Texto(1),  
nome_pai Texto(1),  
nome _mae Texto(1),  
e-mail Texto(1),  
título _eleitor Texto(1),  
estado_civil Texto(1),
```

```
endereco_res Texto(1),
sexo Texto(1),
login Texto(1),
tel_res Texto(1),
tel_cel Texto(1),
senha Texto(1),
ativo Texto(1),
adm Texto(1),
id_pais Texto(1),
id_cidade Texto(1),
FOREIGN KEY(id_cidade) REFERENCES cidade (id_cidade)
)
```

```
CREATE TABLE rg (
id_rg Texto(1) PRIMARY KEY,
nr_rg Texto(1),
data_expedicao Texto(1),
orgao_expedidor Texto(1),
id_usuario Texto(1),
FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
)
```

```
CREATE TABLE cr (
vencimento Texto(1),
id_cr Texto(1) PRIMARY KEY,
nr_cr Texto(1),
concessao_inicial Texto(1),
id_usuario Texto(1),
FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
)
```

```
CREATE TABLE tipo_processo (
nome Texto(1),
```

```
id_tipo_processo Texto(1) PRIMARY KEY,  
orgao Texto(1),  
preco Texto(1)  
)
```

```
CREATE TABLE lista_doc (  
id_lista_doc Texto(1) PRIMARY KEY,  
nome Texto(1)  
)
```

```
CREATE TABLE tipo_proc_lista_doc (  
id_lista_doc Texto(1),  
id_tipo_processo Texto(1),  
FOREIGN KEY(id_lista_doc) REFERENCES lista_doc (id_lista_doc),  
FOREIGN KEY(id_tipo_processo) REFERENCES tipo_processo (id_tipo_processo)  
)
```

```
CREATE TABLE pais (  
id_pais Texto(1) PRIMARY KEY,  
nome Texto(1)  
)
```

```
ALTER TABLE armamento ADD FOREIGN KEY(id_aquis) REFERENCES  
modo_aquisicao (id_aquis)  
ALTER TABLE armamento ADD FOREIGN KEY(id_pais) REFERENCES pais  
(id_pais)  
ALTER TABLE possui ADD FOREIGN KEY(id_empresa) REFERENCES  
EMPRESA_ONDE_TRABALHA (id_empresa)  
ALTER TABLE processos ADD FOREIGN KEY(id_tipo_processo) REFERENCES  
tipo_processo (id_tipo_processo)  
ALTER TABLE processos ADD FOREIGN KEY(id_cr) REFERENCES cr (id_cr)  
ALTER TABLE processos ADD FOREIGN KEY(id_usuario) REFERENCES usuarios  
(id_usuario)
```

```
ALTER TABLE documentos ADD FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
```

```
ALTER TABLE documentos ADD FOREIGN KEY(id_lista_doc) REFERENCES lista_doc (id_lista_doc)
```

```
ALTER TABLE usuario_empresa ADD FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
```

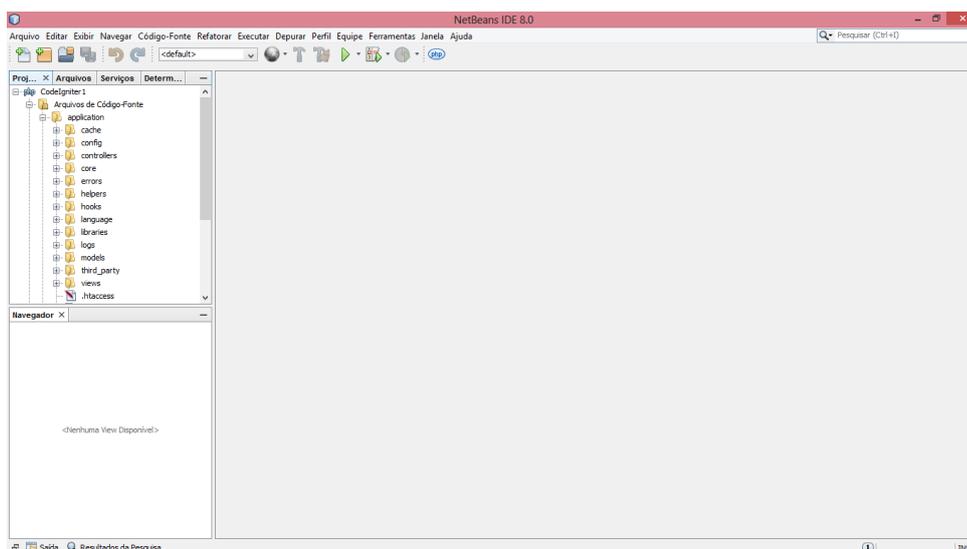
```
ALTER TABLE auditoria ADD FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
```

```
ALTER TABLE possui ADD FOREIGN KEY(id_rg) REFERENCES rg (id_rg)
```

```
ALTER TABLE usuarios ADD FOREIGN KEY(id_pais) REFERENCES pais (id_pais)
```

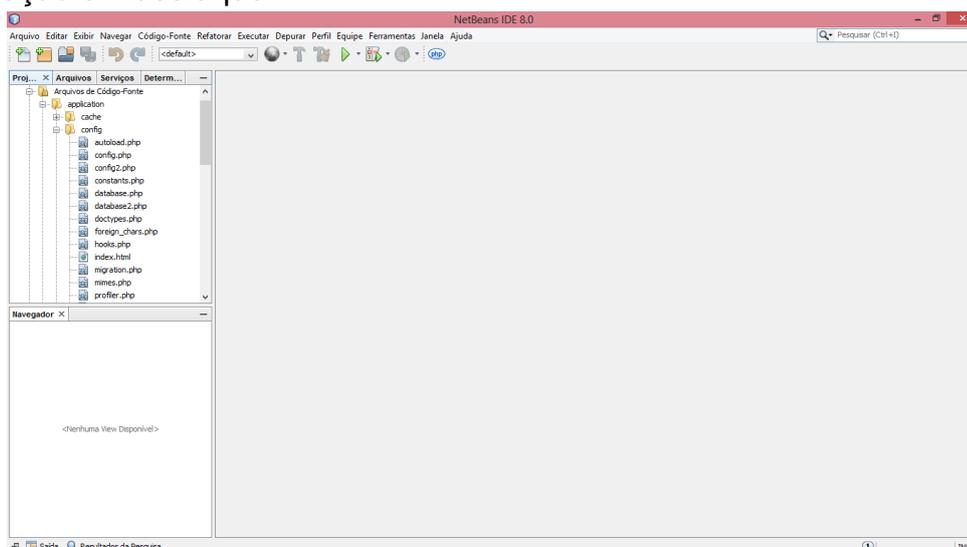
APÊNDICE B - PRINCIPAIS TELAS DE DESENVOLVIMENTO DO APLICATIVO

A figura 55 demonstra o IDE Netbeans 8.0 com o projeto CodeIgniter1 aberto, com todas as suas modulações da aplicação em destaque, do diretório cache ao diretório views.



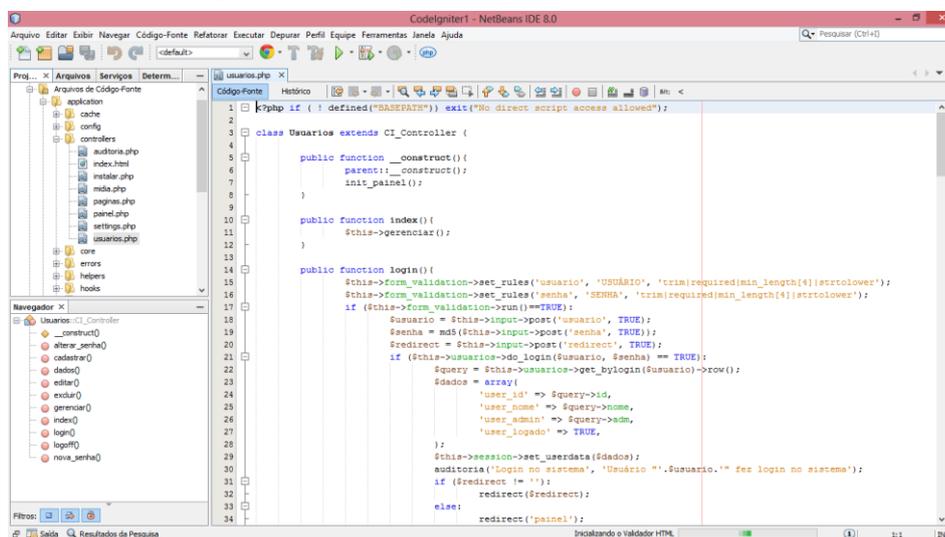
**Figura 55 – Sistema desenvolvido com Framework CodeIgniter
Fonte – Autoria própria**

A figura 56 demonstra o diretório config aberto com todos os seus arquivos de configuração em destaque



**Figura 56 – Diretório config em destaque
Fonte – Autoria própria**

A figura 57 demonstra o diretório controllers aberto com todos os seus arquivos de desenvolvimento em destaque e em particular o arquivo usuários aberto com os códigos expostos na página inicial do Netbeans.



```

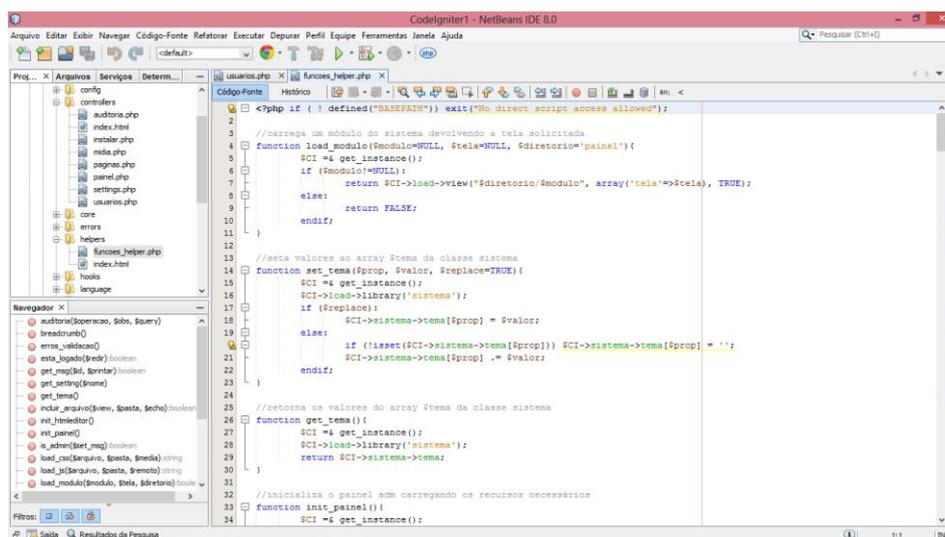
1 <?php if ( ! defined("BASEPATH")) exit("No direct script access allowed");
2
3 class Usuarios extends CI_Controller {
4
5     public function __construct(){
6         parent::__construct();
7         init_panel();
8     }
9
10    public function index(){
11        $this->gerenciar();
12    }
13
14    public function login(){
15        $this->form_validation->set_rules('usuario', 'USUARIO', 'trim|required|min_length[4]|strtolower');
16        $this->form_validation->set_rules('senha', 'SENHA', 'trim|required|min_length[4]|strtolower');
17        if ($this->form_validation->run()==TRUE){
18            $usuario = $this->input->post('usuario', TRUE);
19            $senha = md5($this->input->post('senha', TRUE));
20            $redirect = $this->input->post('redirect', TRUE);
21            if ($this->usuarios->do_login($usuario, $senha) == TRUE){
22                $query = $this->usuarios->get_bylogin($usuario->row());
23                $dados = array(
24                    'user_id' => $query->id,
25                    'user_nome' => $query->nome,
26                    'user_admin' => $query->adm,
27                    'user_logado' => TRUE,
28                );
29                $this->session->set_userdata($dados);
30                auditoria('Login no sistema', 'Usuário "' . $usuario . "' fez login no sistema');
31                if ($redirect != ''){
32                    redirect($redirect);
33                } else{
34                    redirect('panel');
35                }
36            }
37        }
38    }
39
40 }

```

Figura 57 – Diretório controllers em destaque

Fonte – Autoria própria

A figura 58 demonstra o diretório helpers aberto com seu arquivo de desenvolvimento em destaque aberto com os códigos expostos na página inicial do Netbeans.



```

1 <?php if ( ! defined("BASEPATH")) exit("No direct script access allowed");
2
3 //carrega um módulo do sistema desenvolvendo a tela solicitada
4 function load_modulo($modulo=NULL, $tela=NULL, $diretorio='panel'){
5     $CI =& get_instance();
6     if ($modulo=NULL){
7         return $CI->load->view("#diretorio/$modulo", array('tela'=>$tela), TRUE);
8     } else{
9         return FALSE;
10    }
11 }
12
13 //seta valores ao array $tema da classe sistema
14 function set_tema($prop, $valor, $replace=TRUE){
15     $CI =& get_instance();
16     $CI->load->library('sistema');
17     if ($replace){
18         $CI->$sistema->tema[$prop] = $valor;
19     } else{
20         if (!isset($CI->$sistema->tema[$prop])) $CI->$sistema->tema[$prop] = '';
21         $CI->$sistema->tema[$prop] .= $valor;
22     }
23 }
24
25 //retorna os valores do array $tema da classe sistema
26 function get_tema(){
27     $CI =& get_instance();
28     $CI->load->library('sistema');
29     return $CI->$sistema->tema;
30 }
31
32 //inicializa o painel adm carregando os recursos necessários
33 function init_panel(){
34     $CI =& get_instance();
35 }

```

Figura 58 – Diretório helpers de funções em destaque

Fonte – Autoria própria

A figura 59 demonstra o diretório models aberto com seus arquivos de desenvolvimento em destaque e em particular o arquivo auditoria-models aberto com os códigos expostos na página inicial do Netbeans.

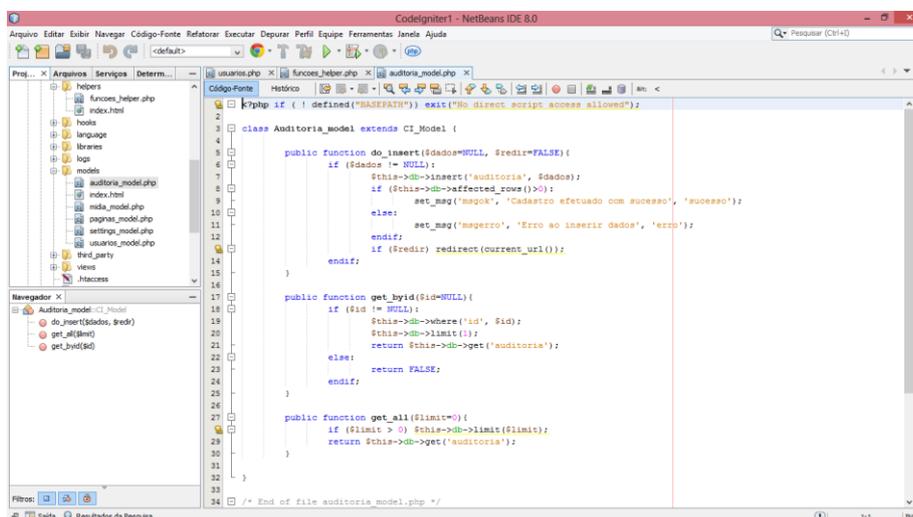


Figura 59 – Diretório models em destaque
Fonte – Autoria própria

A figura 60 demonstra o diretório views aberto com seus arquivos de desenvolvimento em destaque e em particular o arquivo painel-views aberto com os códigos expostos na página inicial do Netbeans.

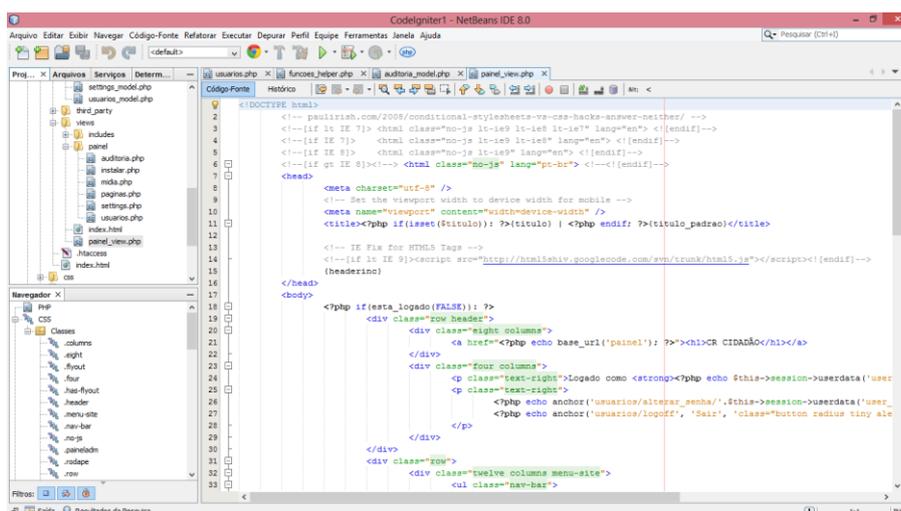


Figura 60 – Diretório views em destaque
Fonte – Autoria própria

APÊNDICE C - DETALHAMENTO DE TODOS OS CASOS DE USO DESENVOLVIDOS NA APLICAÇÃO

- ACESSAR PÁGINA CR-PONTA GROSSA (A1)

Caso de uso primário

Atores

Cliente

Pré-condição:

O cliente deve ter o endereço do site SISCRPF-CIDADÃO, <http://www.crpacidade.com.br> (SISTEMA DE CERTIFICADO DE REGISTRO – CIDADÃO).

Fluxo de Eventos primários:

1. Cliente escolhe o navegador web;
2. Cliente digita o endereço <http://www.crpacidade.com.br>;
3. O sistema carrega e fornece a página inicial;
4. Cliente visualiza informações.

Fluxo de eventos secundários:

1. O sistema apresenta página de erro;
2. O Cliente reinicia a ação (A1);

Pós-condição:

O sistema fornece a visualização da página.

- REALIZAR CADASTRO (A2)

Caso de uso primário

Atores

Cliente

Pré-condição:

O Cliente deve ter entrado no site <http://www.crpacidade.com.br> (SISTEMA DE CERTIFICADO DE REGISTRO – CIDADÃO).

Fluxo de Eventos primários: (A2.1)

1. O Cliente clica em “REALIZAR CADASTRO”;
2. O Cliente digita o “CPF”, o “E-MAIL”, a “SENHA” e os “CARACTERES DE AUTENTICAÇÃO” para o cadastro;
3. O Cliente clica no botão “CADASTRAR”;
4. O Sistema reconhece as informações e envia para o e-mail do Cliente um link para ativação da página;
5. O Cliente acessa o e-mail e ativa o link;
6. O Sistema reconhece as informações e habilita o cadastro para o acesso à página;

Fluxo de eventos secundários:

1. O Cliente digita o “CPF” incorreto ou deixa de digitar;
2. O Cliente digita o “E-MAIL” incorreto ou deixa de digitar;
3. O Cliente clica no botão “CADASTRAR” antecipadamente;

4. O sistema informa que o CPF não existe, ou que o e-mail está incorreto, ou que o CPF e o e-mail deverão ser informados, ou ainda que a tecla CAPSLOCK pode estar acionada;
5. O Cliente retorna para a ação **(A2.1)**.

Pós-condição:

O sistema fornece acesso à área restrita.

- FAZER LOGIN (A3)

Caso de uso primário

Atores

Cliente

Pré-condição:

O Cliente já deve estar com o cadastro realizado e ter entrado no site <http://www.crp'gcidadao.com.br> (SISTEMA DE CERTIFICADO DE REGISTRO – CIDADÃO).

Fluxo de Eventos primários: (A3.1)

1. O Cliente digita o “nome” de login;
2. O Cliente digita a “senha” para o login;
3. O Cliente clica no botão “acessar”;
4. O Sistema reconhece as informações e habilita a página;

Fluxo de eventos secundários:

1. O Cliente digita o “nome” de login incorreto ou deixa de digitar;
2. O Cliente clica no botão “acessar” antecipadamente;
3. O sistema informa que o Cliente e a senha deverão ser informados;
4. O Cliente digita a senha para o login incorreta ou deixa de digitar;
5. O sistema informa que o Cliente não existe, ou que o login está incorreto, ou que a senha está incorreta, ou ainda que a tecla CAPSLOCK pode estar acionada;
6. O Cliente retorna para a ação **(A3.1)**.

Pós-condição:

O sistema fornece acesso à área restrita.

- REQUERER NOVA SENHA (A4)

Caso de uso primário

Atores

Cliente

Pré-condição:

Ações (A1 e A2).

Fluxo de Eventos primários: (A4.1)

1. O Cliente clica em “Esqueci minha senha”;
2. O Sistema pede para informar o CPF e o e-mail;
3. O Cliente informa o CPF e o e-mail;
4. O Sistema informa que a senha foi enviada para o email;
5. O Cliente reinicia a ação **(A3.1)**.

Fluxo de eventos secundários:

1. O Cliente digita um e-mail incorreto;
2. O sistema informa que o e-mail não existe ou está incorreto;

3. O sistema pede para informar e-mail válido;
4. O Cliente retorna a ação **(A4.1)**.

Pós-condição:

O sistema fornece nova senha via e-mail ao Cliente.

- **ACESSAR PROCESSOS EXÉRCITO CR – CONCESSÃO DE CR (A5)**

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A5.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS EXÉRCITO”;
3. O Sistema habilita a página com opção para “PROCESSOS CR” ou “PROCESSOS DE ARMAS”;
4. O Cliente pode escolher sua opção: . “PROCESSOS CR” ou “PROCESSOS ARMAS”;
5. O Cliente clica em “PROCESSOS CR”;
6. O sistema habilita página com botões de opções para: “CONCESSÃO DE CR”, “REVALIDAÇÃO DE CR”, “APOSTILAMENTO DE CR” e “CANCELAMENTO DE CR”;
7. O Cliente escolhe sua opção clicando no botão “CONCESSÃO DE CR”;
8. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado;

Fluxo de eventos secundários (A5.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;

4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO CR – REVALIDAÇÃO DE CR (A6)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A6.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS EXÉRCITO”;
3. O Sistema habilita a página com opção para “PROCESSOS CR” ou “PROCESSOS DE ARMAS”;
4. O Cliente pode escolher sua opção: . “PROCESSOS CR” ou “PROCESSOS ARMAS”;
5. O Cliente clica em “PROCESSOS CR”;
6. O sistema habilita página com botões de opções para: “CONCESSÃO DE CR”, “REVALIDAÇÃO DE CR”, “APOSTILAMENTO DE CR” e “CANCELAMENTO DE CR”;
7. O Cliente escolhe sua opção clicando no botão “REVALIDAÇÃO DE CR”;
8. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado;

Fluxo de eventos secundários (A6.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;

3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO CR – APOSTILAMENTO DE CR (A7)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A7.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS EXÉRCITO”;
3. O Sistema habilita a página com opção para “PROCESSOS CR” ou “PROCESSOS DE ARMAS”;
4. O Cliente pode escolher sua opção: . “PROCESSOS CR” ou “PROCESSOS ARMAS”;
5. O Cliente clica em “PROCESSOS CR”;
6. O sistema habilita página com botões de opções para: “CONCESSÃO DE CR”, “REVALIDAÇÃO DE CR”, “APOSTILAMENTO DE CR” e “CANCELAMENTO DE CR”;
7. O Cliente escolhe sua opção clicando no botão “APOSTILAMENTO DE CR”;
8. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo utilizando as informações dos documentos salvos no banco de dados;

16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado;

Fluxo de eventos secundários (A7.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO CR – CANCELAMENTO DE CR (A8)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A8.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS EXÉRCITO”;
3. O Sistema habilita a página com opção para “PROCESSOS CR” ou “PROCESSOS DE ARMAS”;
4. O Cliente pode escolher sua opção: . “PROCESSOS CR” ou “PROCESSOS ARMAS”;
5. O Cliente clica em “PROCESSOS CR”;
6. O sistema habilita página com botões de opções para: “CONCESSÃO DE CR”, “REVALIDAÇÃO DE CR”, “APOSTILAMENTO DE CR” e “CANCELAMENTO DE CR”;
7. O Cliente escolhe sua opção clicando no botão “CANCELAMENTO DE CR”;
8. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);

13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado;

Fluxo de eventos secundários (A8.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO ARMAS - AQUISIÇÃO NA INDÚSTRIA (A9)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A9.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em "PROCESSOS EXÉRCITO";
3. O Sistema habilita a página com opção para "PROCESSOS CR" ou "PROCESSOS DE ARMAS";
4. O Cliente pode escolher sua opção: . "PROCESSOS CR" ou "PROCESSOS ARMAS";
5. O Cliente clica em "PROCESSOS ARMAS";
6. O sistema habilita página com botões de opções para: "AQUISIÇÃO NA INDÚSTRIA", "AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO", "TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL", "TRANSFERÊNCIA ENTRE ATIRADORES", "TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR" e "IMPORTAÇÃO";
7. O Cliente escolhe sua opção clicando no botão "AQUISIÇÃO NA INDÚSTRIA";
8. O sistema gera "procuração" para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de

documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;

10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.

Fluxo de eventos secundários (A9.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO ARMAS - AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO (A10)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A10.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS EXÉRCITO”;
3. O Sistema habilita a página com opção para “PROCESSOS CR” ou “PROCESSOS DE ARMAS”;
4. O Cliente pode escolher sua opção: . “PROCESSOS CR” ou “PROCESSOS ARMAS”;
5. O Cliente clica em “PROCESSOS ARMAS”;
6. O sistema habilita página com botões de opções para: “AQUISIÇÃO NA INDÚSTRIA”, “AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL”,

“TRANSFERÊNCIA ENTRE ATIRADORES”, “TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR” e “IMPORTAÇÃO”;

7. O Cliente escolhe sua opção clicando no botão “AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO”;
8. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.

Fluxo de eventos secundários (A10.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO ARMAS - TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL (A11)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A11.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS EXÉRCITO”;

3. O Sistema habilita a página com opção para “PROCESSOS CR” ou “PROCESSOS DE ARMAS”;
4. O Cliente pode escolher sua opção: . “PROCESSOS CR” ou “PROCESSOS ARMAS”;
5. O Cliente clica em “PROCESSOS ARMAS”;
6. O sistema habilita página com botões de opções para: “AQUISIÇÃO NA INDÚSTRIA”, “AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL”, “TRANSFERÊNCIA ENTRE ATIRADORES”, “TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR” e “IMPORTAÇÃO”;
7. O Cliente escolhe sua opção clicando no botão “TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL”;
8. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.

Fluxo de eventos secundários (A11.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO ARMAS - TRANSFERÊNCIA ENTRE ATIRADORES (A12)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A12.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS EXÉRCITO”;
3. O Sistema habilita a página com opção para “PROCESSOS CR” ou “PROCESSOS DE ARMAS”;
4. O Cliente pode escolher sua opção: . “PROCESSOS CR” ou “PROCESSOS ARMAS”;
5. O Cliente clica em “PROCESSOS ARMAS”;
6. O sistema habilita página com botões de opções para: “AQUISIÇÃO NA INDÚSTRIA”, “AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL”, “TRANSFERÊNCIA ENTRE ATIRADORES”, “TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR” e “IMPORTAÇÃO”;
7. O Cliente escolhe sua opção clicando no botão “TRANSFERÊNCIA ENTRE ATIRADORES”;
8. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
10. O Cliente clica em “continuar”;
11. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.

Fluxo de eventos secundários (A12.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO ARMAS - TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR (A13)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A13.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em "PROCESSOS EXÉRCITO";
3. O Sistema habilita a página com opção para "PROCESSOS CR" ou "PROCESSOS DE ARMAS";
4. O Cliente pode escolher sua opção: . "PROCESSOS CR" ou "PROCESSOS ARMAS";
5. O Cliente clica em "PROCESSOS ARMAS";
6. O sistema habilita página com botões de opções para: "AQUISIÇÃO NA INDÚSTRIA", "AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO", "TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL", "TRANSFERÊNCIA ENTRE ATIRADORES", "TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR" e "IMPORTAÇÃO";
7. O Cliente escolhe sua opção clicando no botão "TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR";
8. O sistema gera "procuração" para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de "continuar";
10. O Cliente clica em "continuar";
11. O sistema apresenta tela para pagamento idealizada pelo "PAGUE SEGURO", com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;
15. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.

Fluxo de eventos secundários (A13.2):

1. O Cliente não fornece a documentação solicitada;

2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS EXÉRCITO ARMAS - IMPORTAÇÃO (A14)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A14.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em "PROCESSOS EXÉRCITO";
3. O Sistema habilita a página com opção para "PROCESSOS CR" ou "PROCESSOS DE ARMAS";
4. O Cliente pode escolher sua opção: . "PROCESSOS CR" ou "PROCESSOS ARMAS";
5. O Cliente clica em "PROCESSOS ARMAS";
6. O sistema habilita página com botões de opções para: "AQUISIÇÃO NA INDÚSTRIA", "AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO", "TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL", "TRANSFERÊNCIA ENTRE ATIRADORES", "TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR" e "IMPORTAÇÃO";
7. O Cliente escolhe sua opção clicando no botão "IMPORTAÇÃO";
8. O sistema gera "procuração" para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
9. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de "continuar";
10. O Cliente clica em "continuar";
11. O sistema apresenta tela para pagamento idealizada pelo "PAGUE SEGURO", com as opções padrão de pagamento;
12. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
13. O Cliente apresenta a documentação necessária;
14. O Gerente do Sistema anexa a documentação ao banco de dados;

15. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
16. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.

Fluxo de eventos secundários (A14.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS POLÍCIA FEDERAL - AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO (A15)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A15.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS POLÍCIA FEDERAL”;
3. O sistema habilita página com botões de opções para: “AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DE ARMA DE FOGO DO SIGMA – EXÉRCITO”, “TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO”, “RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO” e “PORTE DE ARMA DE FOGO”;
4. O Cliente escolhe sua opção clicando no botão “AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO”;
5. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
6. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
7. O Cliente clica em “continuar”;
8. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;

9. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
10. O Cliente apresenta a documentação necessária;
11. O Gerente do Sistema anexa a documentação ao banco de dados;
12. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
13. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.
14. O Gerente do Sistema apresenta a documentação ao Cliente para assinatura.

Fluxo de eventos secundários (A15.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS POLÍCIA FEDERAL - TRANSFERÊNCIA DE ARMA DE FOGO DO SIGMA – EXÉRCITO (A16)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A16.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS POLÍCIA FEDERAL”;
3. O sistema habilita página com botões de opções para: “AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DE ARMA DE FOGO DO SIGMA – EXÉRCITO”, “TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO”, “RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO” e “PORTE DE ARMA DE FOGO”;
4. O Cliente escolhe sua opção clicando no botão “TRANSFERÊNCIA DE ARMA DE FOGO DO SIGMA – EXÉRCITO”;
5. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
6. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de

documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;

7. O Cliente clica em “continuar”;
8. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
9. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
10. O Cliente apresenta a documentação necessária;
11. O Gerente do Sistema anexa a documentação ao banco de dados;
12. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
13. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.
14. O Gerente do Sistema apresenta a documentação ao Cliente para assinatura.

Fluxo de eventos secundários (A16.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS POLÍCIA FEDERAL - TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO (A17)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A17.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS POLÍCIA FEDERAL”;
3. O sistema habilita página com botões de opções para: “AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DE ARMA DE FOGO DO SIGMA – EXÉRCITO”, “TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO”, “RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO” e “PORTE DE ARMA DE FOGO”;
4. O Cliente escolhe sua opção clicando no botão “TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO”;

5. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
6. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
7. O Cliente clica em “continuar”;
8. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
9. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
10. O Cliente apresenta a documentação necessária;
11. O Gerente do Sistema anexa a documentação ao banco de dados;
12. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
13. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.
14. O Gerente do Sistema apresenta a documentação ao Cliente para assinatura.

Fluxo de eventos secundários (A17.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS POLÍCIA FEDERAL - RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO (A18)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A18.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS POLÍCIA FEDERAL”;
3. O sistema habilita página com botões de opções para: “AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA

DE ARMA DE FOGO DO SIGMA – EXÉRCITO”, “TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO”, “RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO” e “PORTE DE ARMA DE FOGO”;

4. O Cliente escolhe sua opção clicando no botão “RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO”;
5. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
6. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
7. O Cliente clica em “continuar”;
8. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
9. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
10. O Cliente apresenta a documentação necessária;
11. O Gerente do Sistema anexa a documentação ao banco de dados;
12. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
13. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.
14. O Gerente do Sistema apresenta a documentação ao Cliente para assinatura.

Fluxo de eventos secundários (A18.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- ACESSAR PROCESSOS POLÍCIA FEDERAL - PORTE DE ARMA DE FOGO (A19)

Caso de uso primário

Atores

Cliente

Gerente do Sistema

Pré-condição:

Ações (A1, A2 e A3).

Fluxo de Eventos primários: (A19.1)

1. O Sistema habilita a página da área restrita;
2. O Cliente clica em “PROCESSOS POLÍCIA FEDERAL”;
3. O sistema habilita página com botões de opções para: “AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DE ARMA DE FOGO DO SIGMA – EXÉRCITO”, “TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO”, “RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO” e “PORTE DE ARMA DE FOGO”;
4. O Cliente escolhe sua opção clicando no botão “PORTE DE ARMA DE FOGO”;
5. O sistema gera “procuração” para o Cliente imprimir, assinar e reconhecer firma e enviar junto com a documentação;
6. O sistema informa a documentação necessária ao processo que o Cliente deverá providenciar, com endereço para envio e telefone para contato e agendamento de visita se assim o desejar (se o Cliente já tiver cadastro de documentos, o sistema informa somente os documentos já vencidos) e a opção de “continuar”;
7. O Cliente clica em “continuar”;
8. O sistema apresenta tela para pagamento idealizada pelo “PAGUE SEGURO”, com as opções padrão de pagamento;
9. O Cliente faz sua opção de pagamento e realiza o mesmo (cartão de crédito ou débito ou boleto bancário);
10. O Cliente apresenta a documentação necessária;
11. O Gerente do Sistema anexa a documentação ao banco de dados;
12. O Gerente do Sistema gera o processo através de relatórios pré-determinados com modelos padrão, utilizando as informações dos documentos salvos no banco de dados;
13. O sistema confirma o pagamento e envia para o e-mail do Cliente a confirmação e o nº do protocolo do processo iniciado.
14. O Gerente do Sistema apresenta a documentação ao Cliente para assinatura.

Fluxo de eventos secundários (A19.2):

1. O Cliente não fornece a documentação solicitada;
2. O Cliente não realiza o pagamento requerido para o processo;
3. O sistema dispara e-mail a cada 03 (três) dias para o Cliente informando que ainda não consta o recebimento da documentação e a confirmação do pagamento;
4. Após 30 (trinta) dias sem a apresentação da documentação ou realização do pagamento (Gerente consulta relatórios do sistema), o Gerente do Sistema exclui o processo, habilitando o disparo de e-mail para o Cliente informando tal fato.

Pós-condição:

O sistema fornece e-mail ao Cliente, informando nº do protocolo do processo requerido efetivado com sucesso.

- FAZER LOGIN GERENTE (A20)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

O Gerente já deve estar com o cadastro realizado e ter entrado no site <http://www.crp'gcidadao.com.br> (SISTEMA DE CERTIFICADO DE REGISTRO – CIDADÃO).

Fluxo de Eventos primários: (A20.1)

1. O Gerente digita o “nome” de login;
2. O Gerente digita a “senha” para o login;
3. O Gerente clica no botão “acessar”;
4. O Sistema reconhece as informações e habilita a página inicial do Gerente;

Fluxo de eventos secundários:

1. O Gerente digita o “nome” de login incorreto ou deixa de digitar;
2. O Gerente clica no botão “acessar” antecipadamente;
3. O sistema informa que o Gerente e a senha deverão ser informados;
4. O Gerente digita a senha para o login incorreta ou deixa de digitar;
5. O sistema informa que o Gerente não existe, ou que o login está incorreto, ou que a senha está incorreta, ou ainda que a tecla CAPSLOCK pode estar acionada;
6. O Gerente retorna para a ação (A20.1).

Pós-condição:

O sistema fornece acesso à área restrita.

- GERENCIAR CLIENTES – CADASTRAR (A21)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A21.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica em na aba “GERENCIAR CLIENTES”;
3. O sistema habilita sub-abas a partir da aba “GERENCIAR CLIENTES”, com os botões de opções para: “CADASTRAR CLIENTES”, “EDITAR CLIENTES”, e “EXCLUIR CLIENTES”;
4. O Gerente do Sistema escolhe sua opção clicando no botão “CADASTRAR CLIENTES”;
5. O sistema abre página com a lista dos primeiros clientes já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o Gerente do Sistema queira procurar por um cliente específico, ainda, a página possui o botão “INCLUIR NOVO CLIENTE”, quando o Gerente do Sistema quer cadastrar um novo cliente. O Gerente do Sistema clica no botão “INCLUIR NOVO CLIENTE”;
6. O sistema abre página com campos abertos para digitação, referente aos dados particulares do cliente para cadastro. O Gerente do Sistema preenche todos os campos abertos e clica no botão “SALVAR”;
7. O Sistema salva os dados digitados no Banco de dados e mostra a mensagem “CLIENTE CADASTRADO COM SUCESSO”;

Fluxo de eventos secundários (A21.2):

5. O Gerente do Sistema realiza o procedimento (A20 e A21.1);
6. O Sistema não salva os dados digitados no Banco de dados;
7. O Sistema mostra mensagem de ERRO;

Pós-condição:

O novo cliente é cadastrado no sistema com sucesso.

- GERENCIAR CLIENTES – EDITAR (A22)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A22.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica em na aba “GERENCIAR CLIENTES”;
3. O sistema habilita sub-abas a partir da aba “GERENCIAR CLIENTES”, com os botões de opções para: “CADASTRAR CLIENTES”, “EDITAR CLIENTES”, e “EXCLUIR CLIENTES”;
4. O Gerente do Sistema escolhe sua opção clicando no botão “EDITAR CLIENTES”;
5. O sistema abre página com a lista dos primeiros clientes já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o Gerente do Sistema queira procurar por um cliente específico. O Gerente do Sistema digita o primeiro nome do cliente que deseja editar;
6. O sistema faz procura automática pelo nome digitado, a medida que o Gerente do Sistema digita os caracteres do nome desejado. O Gerente clica no nome desejado, assim que ele aparecer como opção;
7. O Sistema abre página com opção de editar os campos relativos aos dados do cliente;
8. O Gerente do Sistema edita os campos que deseja e clica em “SALVAR”;
9. O Sistema salva os dados digitados no Banco de dados e mostra a mensagem “DADOS EDITADOS COM SUCESSO”;

Fluxo de eventos secundários (A22.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A22.1);
2. O Sistema não salva os dados digitados no Banco de dados;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O dados são editados e salvos no sistema com sucesso.

- GERENCIAR CLIENTES – EXCLUIR (A23)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A23.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica em na aba “GERENCIAR CLIENTES”;
3. O sistema habilita sub-abas a partir da aba “GERENCIAR CLIENTES”, com os botões de opções para: “CADASTRAR CLIENTES”, “EDITAR CLIENTES”, e “EXCLUIR CLIENTES”;
4. O Gerente do Sistema escolhe sua opção clicando no botão “EXCLUIR CLIENTES”;
5. O sistema abre página com a lista dos primeiros clientes já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o Gerente do Sistema queira procurar por um cliente específico. O Gerente do Sistema digita o primeiro nome do cliente que deseja excluir;
6. O sistema faz procura automática pelo nome digitado, a medida que o Gerente do Sistema digita os caracteres do nome desejado. O Gerente clica no nome desejado, assim que ele aparecer como opção;
7. O Sistema abre página com opção de excluir o cliente desejado;
8. O Gerente do Sistema clica em “EXCLUIR”;
9. O Sistema salva os dados no Banco de dados e mostra a mensagem “CLIENTE EXCLUÍDO COM SUCESSO”;

Fluxo de eventos secundários (A23.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A23.1);
2. O Sistema não salva os dados digitados no Banco de dados;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O cliente é excluído e os dados são salvos e atualizados no sistema com sucesso.

- GERENCIAR PROCESSOS – CADASTRAR (A24)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A24.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica em na aba “GERENCIAR PROCESSOS”;
3. O sistema habilita sub-abas a partir da aba “GERENCIAR PROCESSOS”, com os botões de opções para: “CADASTRAR PROCESSOS”, “EDITAR PROCESSOS”, e “EXCLUIR PROCESSOS”;
4. O Gerente do Sistema escolhe sua opção clicando no botão “CADASTRAR PROCESSOS”;
5. O sistema abre página com a lista dos primeiros processos já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o Gerente do Sistema queira procurar por um processo específico, ainda, a página possui o botão “INCLUIR NOVO PROCESSO”, quando o Gerente do Sistema quer cadastrar um novo processo. O Gerente do Sistema clica no botão “INCLUIR NOVO PROCESSO”;

6. O sistema abre página com campos abertos para digitação, referente aos dados particulares do processo para cadastro. O Gerente do Sistema preenche todos os campos abertos e clica no botão “SALVAR”;
7. O Sistema salva os dados digitados no Banco de dados e mostra a mensagem “PROCESSO CADASTRADO COM SUCESSO”;

Fluxo de eventos secundários (A24.2):

1. O Gerente do Sistema realiza o procedimento (A20 e A24.1);
2. O Sistema não salva os dados digitados no Banco de dados;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O novo processo é cadastrado no sistema com sucesso.

- GERENCIAR PROCESSOS – EDITAR (A25)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A25.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica em na aba “GERENCIAR PROCESSOS”;
3. O sistema habilita sub-abas a partir da aba “GERENCIAR PROCESSOS”, com os botões de opções para: “CADASTRAR PROCESSOS”, “EDITAR PROCESSOS”, e “EXCLUIR PROCESSOS”;
4. O Gerente do Sistema escolhe sua opção clicando no botão “EDITAR PROCESSOS”;
5. O sistema abre página com a lista dos primeiros processos já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o Gerente do Sistema queira procurar por um processo específico. O Gerente do Sistema começa a digitar o nome desejado;
6. O sistema faz procura automática pelo nome digitado, a medida que o Gerente do Sistema digita os caracteres do nome desejado. O Gerente clica no nome desejado, assim que ele aparecer como opção;
7. O Sistema abre página com opção de editar os campos relativos aos dados do processo;
8. O Gerente do Sistema edita os campos que deseja e clica em “SALVAR”;
9. O Sistema salva os dados digitados no Banco de dados e mostra a mensagem “DADOS EDITADOS COM SUCESSO”;
10. O Sistema salva os dados digitados no Banco de dados e mostra a mensagem “PROCESSO EDITADO COM SUCESSO”;

Fluxo de eventos secundários (A25.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A25.1);
2. O Sistema não salva os dados digitados no Banco de dados;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O dados são editados e salvos no sistema com sucesso.

- GERENCIAR PROCESSOS – EXCLUIR (A26)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A26.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica em na aba “GERENCIAR PROCESSOS”;
3. O sistema habilita sub-abas a partir da aba “GERENCIAR PROCESSOS”, com os botões de opções para: “CADASTRAR PROCESSOS”, “EDITAR PROCESSOS”, e “EXCLUIR PROCESSOS”;
4. O Gerente do Sistema escolhe sua opção clicando no botão “EXCLUIR PROCESSOS”;
5. O sistema abre página com a lista dos primeiros processos já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o Gerente do Sistema queira procurar por um processo específico. O Gerente do Sistema digita o primeiro nome do processo que deseja excluir;
6. O sistema faz procura automática pelo nome digitado, a medida que o Gerente do Sistema digita os caracteres do nome desejado. O Gerente clica no nome desejado, assim que ele aparecer como opção;
7. O Sistema abre página com opção de excluir o processo desejado;
8. O Gerente do Sistema clica em “EXCLUIR”;
9. O Sistema salva os dados no Banco de dados e mostra a mensagem “PROCESSO EXCLUÍDO COM SUCESSO”;

Fluxo de eventos secundários (A26.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A26.1);
2. O Sistema não salva os dados digitados no Banco de dados;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O processo é excluído e os dados são salvos e atualizados no sistema com sucesso.

- ANEXAR DOCUMENTAÇÃO (A27)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A27.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica na aba “ANEXAR DOCUMENTAÇÃO”;
3. O sistema abre página com a lista dos primeiros clientes já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o

- Gerente do Sistema queira procurar por um cliente específico. O Gerente do Sistema digita o primeiro nome do cliente que deseja encontrar;
4. O sistema faz procura automática pelo nome digitado, a medida que o Gerente do Sistema digita os caracteres do nome desejado. O Gerente clica no nome desejado, assim que ele aparecer como opção;
 5. O Sistema abre página específica do cliente, com as abas de opções disponíveis “ARMAMENTO”, “MÁQUINA DE RECARGA”, “GERAR PROCESSOS” e “ANEXAR DOCUMENTAÇÃO”;
 6. O Gerente do Sistema clica em “ANEXAR DOCUMENTAÇÃO”;
 7. O Sistema abre página para upload de arquivos;
 8. O Gerente clica em “PROCURAR” para fazer a busca do documento que deseja anexar;
 9. O Sistema realiza a busca e quando encontrado, o Gerente anexa aquele documento, que será salvo numa pasta pré-definida;
 10. O Sistema mostra a mensagem “DOCUMENTO ANEXADO COM SUCESSO”;

Fluxo de eventos secundários (A27.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A27.1);
2. O Sistema não salva os documentos no Banco de dados;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O documento é incluído e os dados são salvos e atualizados no sistema com sucesso.

- GERAR PROCESSOS (A28)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A28.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica na aba “GERAR PROCESSOS”;
3. O sistema abre página com a lista dos primeiros clientes já cadastrados em ordem alfabética, com a opção de “PROCURAR” para digitação, caso o Gerente do Sistema queira procurar por um cliente específico. O Gerente do Sistema digita o primeiro nome do cliente que deseja encontrar;
4. O sistema faz procura automática pelo nome digitado, a medida que o Gerente do Sistema digita os caracteres do nome desejado. O Gerente clica no nome desejado, assim que ele aparecer como opção;
5. O Sistema abre página específica do cliente, com as abas de opções disponíveis “ARMAMENTO”, “MÁQUINA DE RECARGA”, “GERAR PROCESSOS” e “ANEXAR DOCUMENTAÇÃO”;
6. O Gerente do Sistema clica em “GERAR PROCESSOS”;
7. O Sistema abre página com lista de processos para escolha: “CONCESSÃO DE CR”, “REVALIDAÇÃO DE CR”, “APOSTILAMENTO DE CR”, “CANCELAMENTO DE CR”, “AQUISIÇÃO NA INDÚSTRIA”,

“AQUISIÇÃO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DO SINARM – POLÍCIA FEDERAL”, “TRANSFERÊNCIA ENTRE ATIRADORES”, “TRANSFERÊNCIA ENTRE MILITAR E ATIRADOR”, “IMPORTAÇÃO”, “AQUISIÇÃO DE ARMA DE FOGO NO COMÉRCIO ESPECIALIZADO”, “TRANSFERÊNCIA DE ARMA DE FOGO DO SIGMA – EXÉRCITO”, “TRANSFERÊNCIA DE ARMA DE FOGO ENTRE CIDADÃO”, “RENOVAÇÃO DE REGISTRO DE ARMA DE FOGO” e “PORTE DE ARMA DE FOGO”;

8. O Gerente clica no processo que deseja gerar;
9. O Sistema gera o processo no formato .PDF, com disponibilidade de impressão;
10. O Gerente do Sistema clica no botão “IMPRIMIR” e realiza a impressão do processo;

Fluxo de eventos secundários (A28.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A28.1);
2. O Sistema não gera o processo pretendido;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O processo é gerado e realizado sua impressão pelo Gerente do Sistema.

- EMITIR RELATÓRIOS (A29)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A29.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica na aba “EMITIR RELATÓRIOS”;
3. O sistema abre página com a lista dos relatórios disponíveis para gerar: “CRs VENCIDOS”, “RELAÇÃO DE CAC CADASTRADOS”, “RELAÇÃO DE CIDADÃOS CADASTRADOS”, “ARMAS CADASTRADAS SIGMA”, “ARMAS CADASTRADAS SINARM”, e “MÁQUINAS DE RECARGA CADASTRADAS”;
4. O Gerente clica no relatório que deseja gerar;
5. O Sistema gera o relatório no formato .PDF, com disponibilidade de impressão;
6. O Gerente do Sistema clica no botão “IMPRIMIR” e realiza a impressão do relatório;

Fluxo de eventos secundários (A29.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A29.1);
2. O Sistema não gera o relatório pretendido;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O relatório é gerado e realizado sua impressão pelo Gerente do Sistema.

- VERIFICAR AUDITORIA (A30)

Caso de uso primário

Atores

Gerente do Sistema

Pré-condição:

Ações (A20).

Fluxo de Eventos primários: (A30.1)

1. O Sistema habilita a página da área restrita;
2. O Gerente do Sistema clica na aba “VERIFICAR AUDITORIA”;
3. O sistema abre página de auditoria com as opções de visualizar os últimos dados auditados ou listar os últimos 50 (cinquenta) registros realizados pelo Sistema;
4. O Gerente clica no botão para listar os últimos 50 (cinquenta) registros realizados pelo Sistema;
5. O Sistema mostra os últimos 50 (cinquenta) registros realizados;

Fluxo de eventos secundários (A30.2):

1. O Gerente do Sistema realiza o procedimento (A20 E A30.1);
2. O Sistema não gera os dados auditados;
3. O Sistema mostra mensagem de ERRO;

Pós-condição:

O dados auditados são mostrados pelo Sistema.