**FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO, TECNOLOGIA**

**E EDUCAÇÃO**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**ÁTILEY DE CÁSSIA RIBEIRO SILVA**

**GABRIEL DA SILVA COELHO**

**JÔNATAS BALESTRA DE OLIVEIRA**

**SANARA MACIEL FELÍCIO**

**EXTRANET FÊNIX:**

**DESENVOLVIMENTO DE UMA EXTRANET PARA A EMPRESA FÊNIX INDÚSTRIA DE ELETRÔNICOS LTDA**

**SANTA RITA DO SAPUCAÍ - MG**

**2017**

**FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO, TECNOLOGIA**

**E EDUCAÇÃO**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**ÁTILEY DE CÁSSIA RIBEIRO SILVA**

**GABRIEL DA SILVA COELHO**

**JÔNATAS BALESTRA DE OLIVEIRA**

**SANARA MACIEL FELÍCIO**

**EXTRANET FÊNIX:**

**DESENVOLVIMENTO DE UMA EXTRANET PARA A EMPRESA FÊNIX INDÚSTRIA DE ELETRÔNICOS LTDA**

Projeto de final de curso apresentado à FAI – Centro de Ensino Superior em Gestão, Tecnologia e Educação, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação, sob a orientação da profa. Eunice Gomes de Siqueira.

**SANTA RITA DO SAPUCAÍ - MG**

**2017**

**FOLHA DE APROVAÇÃO**

**HISTÓRICO DE REVISÕES**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Autor | Descrição |
| 19/02/2017 | 0.01 | Jônatas | Primeiros esboços – formatação geral. |
| 05/03/2017 | 0.02 | Equipe | Elaboração do capítulo 3 (Objetivo). |
| 11/03/2017 | 0.03 | Equipe | Elaboração da Revisão Bibliográfica. |
| 18/03/2017 | 0.04 | Equipe | Elaboração do capitulo 4 e 5. |
| 31/03/2017 | 1.00 | Sanara e Átiley | Feita a revisão do documento antes da primeira entrega. |
| 08/04/2017 | 1.01 | Jônatas | Feita a correção do capítulo 2 Revisão Bibliográfica |
| 15/04/2017 | 1.02 | Jônatas, Átiley | Feita a correção do capítulo 3 Objetivos do Projeto |
| 22/04/2017 | 1.03 | Jônatas, Sanara | Feita a Correção do capítulo 4 e 5. |
| 27/05/2107 | 1.04 | Equipe | Elaboração da sessão 6.7 Projeto de Sistemas Distribuídos. |
| 30/06/2017 | 2.0 | Jônatas, Átiley | Feita a revisão do documento antes da primeira entrega. |

**AGRADECIMENTOS**

[A inclusão deste elemento é opcional. ]

**RESUMO**

[O resumo deve ser escrito em apenas um parágrafo. Não ultrapasse 250 palavras. Não utilize siglas. No final, apresente de 3 a 6 palavras-chave, separadas entre si por pontos. ]

**LISTA DE FIGURAS**

[FIGURA 1 – Revoluções industriais 19](#_3dy6vkm)

[FIGURA 2 - Níveis de Decisão e Grupos Funcionais Previstos 26](#_35nkun2)

[FIGURA 3 - Partes interessadas alocadas em uma matriz de poder x interesse 29](#_z337ya)

[FIGURA 4 - Modelo de Ciclo de Vida Incremental 30](#_1y810tw)

[FIGURA 5 – Repositório GitHub 37](#_3o7alnk)

[FIGURA 6 – Repositórios de arquivos Google Drive 38](#_23ckvvd)

[FIGURA 7 - Diagrama de caso uso Visão Geral 55](#_19c6y18)

[FIGURA 8 - Diagrama de caso uso Módulo Controle de Acesso 56](#_3tbugp1)

[FIGURA 9 - Diagrama de caso uso Módulo Administrador 56](#_28h4qwu)

[FIGURA 10 - Diagrama de caso uso Módulo Marketing 57](#_nmf14n)

[FIGURA 11 - Diagrama de caso uso Módulo Recursos Humanos 57](#_37m2jsg)

[FIGURA 12 - Diagrama Entidade-Relacionamento 58](#_111kx3o)

[FIGURA 13 - Diagrama de Pacotes 60](#_2zbgiuw)

[FIGURA 14 - Diagrama de Classe Entity 61](#_3ygebqi)

[FIGURA 15 - Diagrama de Classe BO 62](#_2dlolyb)

[FIGURA 16- Diagrama de Classe DAO 62](#_sqyw64)

[FIGURA 17- Diagrama de Objeto do cenário criar do Caso de Uso Manter Cadastro Grupo de Usuários 63](#_1rvwp1q)

[FIGURA 18 – Diagrama de sequência do cenário criar do Caso de Uso Manter Cadastro Grupo de Usuários. 65](#_1664s55)

[FIGURA 19- Diagrama de Visão geral de Interação do caso de uso Manter Cadastro Grupo Usuários 66](#_3q5sasy)

[FIGURA 20 – Diagrama de Atividades do caso de uso Manter Cadastro Grupo Usuários 67](#_kgcv8k)

[FIGURA 21 – Diagrama de Máquina de Estados do Caso de Uso Manter Cadastro Grupo de Usuários 68](#_1jlao46)

[FIGURA 22 – Modelo Operacional 69](#_xvir7l)

[FIGURA 23 – O Ambiente de Execução do CLR 73](#_2nusc19)

[FIGURA 24 – Visão de Distribuição 78](#_3mzq4wv)

[FIGURA 25 – O processo de execução de código CLR 80](#_319y80a)

[FIGURA 26 -Principais elementos do Framework .NET 82](#_1gf8i83)

**LISTA DE QUADROS**

[QUADRO 1 - Recursos Humanos 31](#_2xcytpi)

[QUADRO 2 - Recursos de Hardware 32](#_1ci93xb)

[QUADRO 3 - Recursos de Software 32](#_3whwml4)

[QUADRO 4 - Estimativas de esforço 33](#_qsh70q)

[QUADRO 5 - Esforços planejados e realizados 36](#_2p2csry)

[QUADRO 6 – Principais Mudanças 40](#_ihv636)

[QUADRO 7 - Entregas do projeto 51](#_1v1yuxt)

[QUADRO 8 - Apresentação dos design patterns utilizados 70](#_2afmg28)

[QUADRO 10 - Documentos relevantes para testes 86](#_3s49zyc)

[QUADRO 11 - Equipamentos a serem utilizados para a realização dos testes 86](#_meukdy)

[QUADRO 12 - Softwares a serem utilizados para a realização dos testes 86](#_36ei31r)

[QUADRO 13 - Identificação dos itens a serem testados 87](#_45jfvxd)

[QUADRO 14 - Requisitos e casos de testes 87](#_zu0gcz)

[QUADRO 15 - Descrição do caso de teste Xnnn 87](#_1yyy98l)

[QUADRO 16 - Situação do caso de teste Xn 88](#_1d96cc0)

[QUADRO 17 - Papéis e responsabilidades na implantação 90](#_4anzqyu)

[QUADRO 18 - Treinamentos previstos 90](#_14ykbeg)

[QUADRO 19 - Cronograma de atividades da implantação 90](#_243i4a2)

[QUADRO 20 - Documentos de apoio à implantação 91](#_338fx5o)

**LISTA DE ABREVIATURAS E SIGLAS**

ABNT -Associação Brasileira de Normas Técnicas

AD - *Active Directory*

BI - *Business Intelligence*

CLR - *Common Language Runtime*

CRM - *Customer Relationship* *Management*

CTS - *Common Type System*

DAO – *Data Access Object*

EAP - Estrutura Analítica do Projeto

EIS - *Executive* *Information System*

ERP - *Enterprise Resource Planning*

FAI - Centro de Ensino Superior em Gestão, Tecnologia e Educação

FAITEC - Feira de Tecnologia da FAI

GHz - *Giga Hertz*

HD - *Hard Disk*

HTML - *HyperText Markup Language*

HTTP - *HyperText Transfer Protocol*

IDE - *Integrated Development Environment*

IL *– Intermediate language*

J2EE - *Java For Entreprise Edition*

JVM - *Java Virtual Machine*,

LDAP -  *Lightweight Directory Access Protocol*

MER - Modelo Entidade Relacionamento

MSIL – *Microsoft Intemediate Language*

MVC *- Model-View-Controller*

PCU - Pontos de Caso de Uso

PF - Pontos de Função

PFC - Projeto Final de Curso

PMBOK - *Project Management Body of Knowledge*

PMI - *Project Management Institute*

RAM - *Random Access Memory*

SAD – Sistema de Apoio a Decisões

SD – Sistemas Distribuídos

SGDB - Sistema Gerenciador de banco de Dados

TI – Tecnologia da Informação

UML - *Unified Modeling Language*

WBS - *Work Breakdown Structure*

**SUMÁrio**

[**1 INTRODUÇÃO 14**](#_30j0zll)

[**2 REVISÃO BIBLIOGRÁFICA 15**](#_1fob9te)

[2.1 EXTRANET 15](#_3znysh7)

[2.2 ESTRATÉGIAS ORGANIZACIONAIS - INTERNET, INTRANET E EXTRANET 17](#_2et92p0)

[2.3 INDUSTRIA 4.0 19](#_tyjcwt)

[2.4 MICROSOFT ACTIVE DIRECTORY 22](#_1t3h5sf)

[**3 OBJETIVO DO PROJETO 24**](#_4d34og8)

[3.1 FORMULAÇÃO DO PROBLEMA 24](#_2s8eyo1)

[3.2 OBJETIVOS 24](#_17dp8vu)

[3.3 JUSTIFICATIVA 25](#_3rdcrjn)

[3.4 PÚBLICO-ALVO 25](#_26in1rg)

[3.5 NÍVEIS DE DECISÃO E GRUPOS FUNCIONAIS ATENDIDOS 26](#_lnxbz9)

[**4 GERÊNCIA DO PROJETO 27**](#_1ksv4uv)

[4.1 PLANO DE PROJETO 27](#_44sinio)

[**4.1.1 Partes Interessadas 27**](#_2jxsxqh)

[**4.1.2 Modelo de Ciclo de Vida 29**](#_3j2qqm3)

[**4.1.3 Recursos Necessários 30**](#_4i7ojhp)

[**4.1.4 Estimativas de Tempo 32**](#_2bn6wsx)

[4.2 ÁREAS DE CONHECIMENTO 33](#_3as4poj)

[**4.2.1 Gestão de Escopo 33**](#_1pxezwc)

[**4.2.2 Gestão do Tempo 34**](#_49x2ik5)

[**4.2.3 Gestão da Integração 36**](#_147n2zr)

[**4.2.4 Gestão da Qualidade 40**](#_32hioqz)

[**4.2.5 Gestão dos Riscos 42**](#_1hmsyys)

[**5 ESPECIFICAÇÃO E ANÁLISE DOS REQUISITOS 43**](#_41mghml)

[5.1 DESCRIÇÃO DE REQUISITOS 43](#_2grqrue)

[**5.1.1 Requisitos Funcionais 44**](#_vx1227)

[**5.1.2 Requisitos não Funcionais 49**](#_3fwokq0)

[5.2 VISÃO FUNCIONAL 54](#_4f1mdlm)

[**5.2.1 Diagramas de Casos de Uso 54**](#_2u6wntf)

[**5.2.2 Descrição dos Casos de Uso 58**](#_1mrcu09)

[5.3 VISÃO DE DADOS 58](#_46r0co2)

[**5.3.1 Projeto lógico 58**](#_2lwamvv)

[**6 ARQUITETURA E PROJETO DO SISTEMA 59**](#_3l18frh)

[6.1 VISÃO ESTRUTURAL 59](#_206ipza)

[**6.1.1 Diagrama de Pacotes 59**](#_4k668n3)

[**6.1.2 Diagramas de Classes 60**](#_1egqt2p)

[**6.1.3 Diagramas de Objetos 63**](#_3cqmetx)

[6.2 VISÃO COMPORTAMENTAL 64](#_4bvk7pj)

[**6.2.1 Projeto das Interações 64**](#_2r0uhxc)

[**6.2.2 Diagrama de Atividades 66**](#_25b2l0r)

[**6.2.3 Diagrama de Máquina de Estados 67**](#_34g0dwd)

[6.3 VISÃO DE DADOS 68](#_43ky6rz)

[**6.3.1 Modelo Operacional 68**](#_2iq8gzs)

[**6.3.2 Dicionário de Dados 69**](#_3hv69ve)

[**6.3.3 Processo de Elaboração do Projeto Físico 69**](#_1x0gk37)

[6.4 VISÃO FÍSICA 69](#_4h042r0)

[**6.4.1 Diagrama de Componentes 70**](#_2w5ecyt)

[6.5 PADRÕES DE PROJETO 70](#_1baon6m)

[**6.5.1 Design Patterns 70**](#_3vac5uf)

[**6.5.2 Convenções para Codificação 70**](#_pkwqa1)

[6.6 ANÁLISE DE COMPLEXIDADE 71](#_39kk8xu)

[6.7 PROJETO DE SISTEMAS DISTRIBUÍDOS 71](#_1opuj5n)

[**6.7.1 Procedimentos para Tratamento dos Desafios 71**](#_48pi1tg)

[**6.7.2 Tecnologias e Arquiteturas de Distribuição 77**](#_1302m92)

[6.8 PROJETO DA INTERAÇÃO HUMANO-COMPUTADOR 85](#_40ew0vw)

[**6.8.1 Perfil de Usuário 85**](#_2fk6b3p)

[**6.8.2 Aspecto Visual da Interface de Usuário 85**](#_upglbi)

[**6.8.3 Heurísticas de Usabilidade 85**](#_3ep43zb)

[**7 PLANO DE TESTES 86**](#_1tuee74)

[7.1 FINALIDADE 86](#_4du1wux)

[7.2 ESCOPO 86](#_2szc72q)

[**7.2.1 Referências a Documentos Relevantes 86**](#_184mhaj)

[**7.2.2 Ambiente para a Realização dos Testes 86**](#_279ka65)

[**7.3.1 Item a Testar 87**](#_1ljsd9k)

[**7.3.2 Rastreabilidade entre Requisitos e Casos de Teste 87**](#_2koq656)

[**7.3.3 Descrição dos Casos de Testes 87**](#_3jtnz0s)

[7.4 RESULTADOS DOS TESTES 88](#_4iylrwe)

[**7.4.1 Histórico de Realização 88**](#_2y3w247)

[**7.4.2 Resultados 88**](#_3x8tuzt)

[**8 PLANO DE IMPLANTAÇÃO 89**](#_2ce457m)

[8.1 METODOLOGIA 89](#_rjefff)

[**8.1.1 Descrição da Metodologia 89**](#_3bj1y38)

[**8.1.2 Matriz de Responsabilidade 89**](#_1qoc8b1)

[8.2 TREINAMENTO PREVISTO 90](#_2pta16n)

[8.3 CRONOGRAMA DE IMPLANTAÇÃO 90](#_3oy7u29)

[8.4 DOCUMENTOS DE APOIO À IMPLANTAÇÃO 90](#_j8sehv)

[8.5 VISÃO DE IMPLANTAÇÃO 91](#_1idq7dh)

[**9 CONCLUSÃO 92**](#_42ddq1a)

[**REFERÊNCIAS 93**](#_2hio093)

[**OBRAS CONSULTADAS 96**](#_4fsjm0b)

[**GLOSSÁRIO 97**](#_2uxtw84)

[**APÊNDICE A - ESTIMATIVAS POR PONTOS DE FUNÇÃO 100**](#_1a346fx)

[**APÊNDICE B – ESTIMATIVA POR PONTOS DE CASOS DE USO 101**](#_3u2rp3q)

[**APÊNDICE C – ESTRUTURA ANALÍTICA DO PROJETO 102**](#_2981zbj)

[**APÊNDICE D – DICIONÁRIO EAP DO PROJETO 103**](#_odc9jc)

[**APÊNDICE E – RELATÓRIO DE DESEMPENHO 104**](#_38czs75)

[**APÊNDICE F – PLANILHA GERÊNCIA DE RISCOS 105**](#_1nia2ey)

[**APÊNDICE G – DESCRIÇÃO DOS CASOS DE USO 106**](#_47hxl2r)

[**APÊNDICE G – DESCRIÇÃO DOS CASOS DE USO 107**](#_2mn7vak)

[**APÊNDICE H – Dicionário de Dados 108**](#_11si5id)

[**APÊNDICE I – Questionário de Perfil do Usuário 109**](#_3ls5o66)

# 1 INTRODUÇÃO

[Apresentação do projeto e do documento, onde constam a delimitação do assunto tratado e outros elementos necessários para situar o leitor quanto ao tema do trabalho. O último parágrafo deve apresentar os capítulos que este documento contém. ]

# 2 REVISÃO BIBLIOGRÁFICA

Este capítulo contém a revisão técnica que abrange todos os conceitos utilizados para o desenvolvimento deste projeto.

Segundo Trentini e Paim (1999, p.24):

A revisão bibliográfica é definida como uma fonte de informação para as pesquisas bibliográficas. Estas pesquisas incluem estudos que propõem a construção de teorias e marcos conceituais pelo método dedutivo, estudos conduzidos para traçar uma imagem do saber produzido ou os vazios em determinados fenômenos.

As seções contidas neste capítulo irão relatam os fundamentos dos seguintes conceitos: Extranet, Intranet, Industria 4.0 e *Active Directory*.

## 2.1 EXTRANET

Com o crescimento acelerado do mercado globalizado e da concorrência, cada vez mais as empresas se tornam dependentes dos benefícios e facilidades propostos pela tecnologia da informação. Para manterem-se competitivas empresas optam por incrementar formas de colaboração na execução de projetos de vários propósitos. As extranets funcionam como um canal de informações aberto para seus clientes e fornecedores, facilitando pedidos, pagamentos, contato com o cliente, entre outros.

Segundo Ruikar, Anumba e Carillo (2005, apud Andery; Arantes, 2015):

As extranets têm a função de monitorar, controlar, manipular e guardar as informações dos projetos, tanto no sentido de empreendimento como no sentido de concepção de um produto (design), tornando essa informação disponível para os participantes de uma organização temporária (como é o caso das equipes de projeto) ou para uma cadeia de suprimentos.

De acordo com Wikison (2005) destacam-se as principais funcionalidades das extranets: (a) módulos administrativos, módulos que permitem a configuração da estrutura do sistema, definindo os tipos de acesso e parâmetros de segurança, permitindo configurar o uso para múltiplos usuários; (b) funções de gerenciamento de documentos, envolvendo a carga e descarga destes, emissões de notificações, rastreabilidade dos documentos; (c) funções de gerenciamento do fluxo de tarefas e informações (*workflow* *and tasks management*), integrando disciplinas e tarefas, aplicando ferramentas de gerenciamento de projetos (custo, prazo, escopo) e permitindo a integração de fluxos de trabalho; (d) ferramentas de comunicação com os usuários, clientes e fornecedores..

Nitithamyong e Skibniewski (2011) conduziram um dos estudos mais completos sobre a utilização de extranets em projetos de arquitetura, engenharia e construção. Os autores destacam alguns aspectos como essenciais para garantir a eficiência na implementação e no funcionamento das extranets. Citam em primeiro lugar o comprometimento e apoio técnico e gerencial da alta administração das empresas envolvidas. Também consideram como aspecto crítico a presença de um líder de projeto comprometido com a utilização do ambiente colaborativo, que tenha a possibilidade de dar assistência aos membros da equipe no que diz respeito ao seu uso. Chamam também a atenção para o fato de que o sucesso da implementação depende de uma extranet amigável, intuitiva e fácil de ser utilizada, bem como do treinamento adequado do pessoal envolvido.

Entre as pesquisas nessa direção, Ruikar, Anumba e Carillo, (2005, apud Andery; Arantes, 2015) ressaltam alguns benefícios da utilização das extranets:

A melhoria da comunicação entre os agentes envolvidos; a redução do tempo de resposta, em função da facilidade de acesso à informação; e a possibilidade de rastrear e classificar facilmente as informações. Ainda, a maior transparência na troca de informações pode conduzir a um clima de maior confiança e colaboração.

Desde que sejam implementadas rotinas para análise das informações, as extranets se tornam um instrumento útil para identificar durante os processos, os gargalos, retrabalhos e fluxos ineficientes de informação e/ou determinação de atividades não agregadoras de valor, dentro da ótica do pensamento enxuto (TRIBELSKY; SACKS, 2010). Nesse contexto, este projeto demonstra essas afirmações propondo uma extranet de qualidade, que atenda a organização Fênix de forma eficaz.

## 2.2 ESTRATÉGIAS ORGANIZACIONAIS - INTERNET, INTRANET E EXTRANET

No contexto atual as organizações estão utilizando, cada vez mais, os espaços eletrônicos/digitais (internet/intranet/extranet) como uma forma de gerenciar eficientemente informação e conhecimento no ambiente organizacional.

De acordo com Teixeira; Valentim (2012), os sistemas de informação propiciam aos gestores uma visão integrada das informações existentes (produzidas internamente e externamente), pois geralmente quando a organização não possui esse tipo de sistema, as informações ficam fragmentadas por setor e dispersas nos *websites*, dessa forma os sistemas de informação possibilitam a gestão coerente e articulada das informações, visando obter decisões e ações consistentes e direcionada às estratégias da organização.

Para a elaboração de um sistema de informação satisfatório é fundamental a arquitetura da informação para que viabilize a fácil localização e acesso de dados e informações, bem como ajuda no desenvolvimento de estrutura hierarquias distintas para a divulgação dos conteúdos, promovendo maior qualidade.

É difícil imaginar uma organização de médio e/ou grande porte sem um sistema de informação conectado à internet/intranet/extranet e para que isso seja feito de forma eficaz é importante uma equipe de profissionais competentes para elaborar a arquitetura do sistema de informação, bem como do *website*.

O desafio da arquitetura da informação no ambiente organizacional é integrar as informações geradas internamente e as produzidas externamente, dentro do ambiente eletrônico/digital vinculado a internet/intranet/extranet, em formato compreensível, facilitando assim o compartilhamento e a disseminação de informações. Finalmente, tem como desafio propiciar maior conectividade, interatividade e velocidade de resposta.

Segundo Carvalho et al. (1999, apud Teixeira; Valentim, 2012):

Através da arquitetura da informação foi possível criar sistemas de informações gerenciais que, por sua vez, assumiram um papel efetivo no que tange ao suporte à tomada de decisão, uma vez que são compostos por ferramentas de análise, representação, apresentação, relacionamento e integração, cuja interação com o usuário é dinâmica e flexível, propiciando o acesso, a recuperação e o armazenamento de produtos informacionais distintos e que podem ser alinhados ao objetivo do usuário que buscou a informação.

Sistemas de informação possuem potencialidades para proporcionar a aprendizagem e é possível encontrar no universo empresarial vários sistemas de informação com características e objetivos gerenciais como: *Executive* *Information System* (EIS), *Business Intelligence* (BI), *Customer Relationship* *Management* (CRM), *Enterprise Resource Planning* (ERP); Sistema de Apoio a Decisões (SAD). Dessa forma, funcionam como um importante instrumento de gestão que possibilitam ganhos em desempenho tanto operacional quanto de aprendizagem.

No que tange a usabilidade propicia criar *websites* e sistemas de informação direcionados a um público específico, respeitando as necessidades e demandas informacionais, cujas exigências são relacionadas às estratégias e objetivos do próprio usuário ou grupo de usuários, respeitando o tempo disponível do usuário para esse tipo de acesso e tomada de decisão, bem como respeitando o formato que mais se adequa a velocidade de resposta, assim o formato e a disposição das informações devem procurar ser bem organizadas.

As vantagens da encontrabilidade em tempo real ou *on-line* permitem à organização possuir um instrumento de gestão com todas as dimensões e agregações possíveis para análise e simulação de dados e informação. A arquitetura da informação aplica os fundamentos da usabilidade e da encontrabilidade na criação dos espaços eletrônicos/digitais, bem como na criação de sistemas de informação gerencial.

Observa-se na ambiente webque as organizações têm um grande número de *sites* de marcas/produtos com poucos aspectos em comum e nenhuma estrutura de organização ou navegação compartilhada, com *intranets* e interfaces múltiplas e extremamente diferentes para gerir seus empreendimentos. Internamente, quando um departamento precisa contactar outro departamento é preciso fazê-lo de forma *offline*.

Visando todo esse conceito o projeto consiste em desenvolver um *website* que integre todos os setores da empresa e funcione como extranet, levando também o desafio do ramo de Indústrias 4.0 ou indústrias inteligentes.

## 2.3 INDUSTRIA 4.0

Indústria 4.0 é um conceito de indústria que reúne os últimos avanços tecnológicos dos campos de automação, controle e tecnologia da informação, aplicadas aos processos de manufatura. Baseado em Sistemas Cyber-Físicos, Internet dos Serviços e Internet das Coisas, os processos de produção inclinam-se a se tornar cada vez mais eficientes, autônomos e customizáveis.

Segundo Ferreira e Stock (2016, p. 2) “Uma revolução industrial é caracterizada por mudanças abruptas e radicais, motivadas pela incorporação de tecnologias, tendo desdobramentos nos âmbitos econômico, social e político. ”

A Figura 1 apresenta as três primeiras revoluções industriais e suas inovações, e destaca aquilo que diferentes estudos técnicos consideram a quarta revolução industrial.

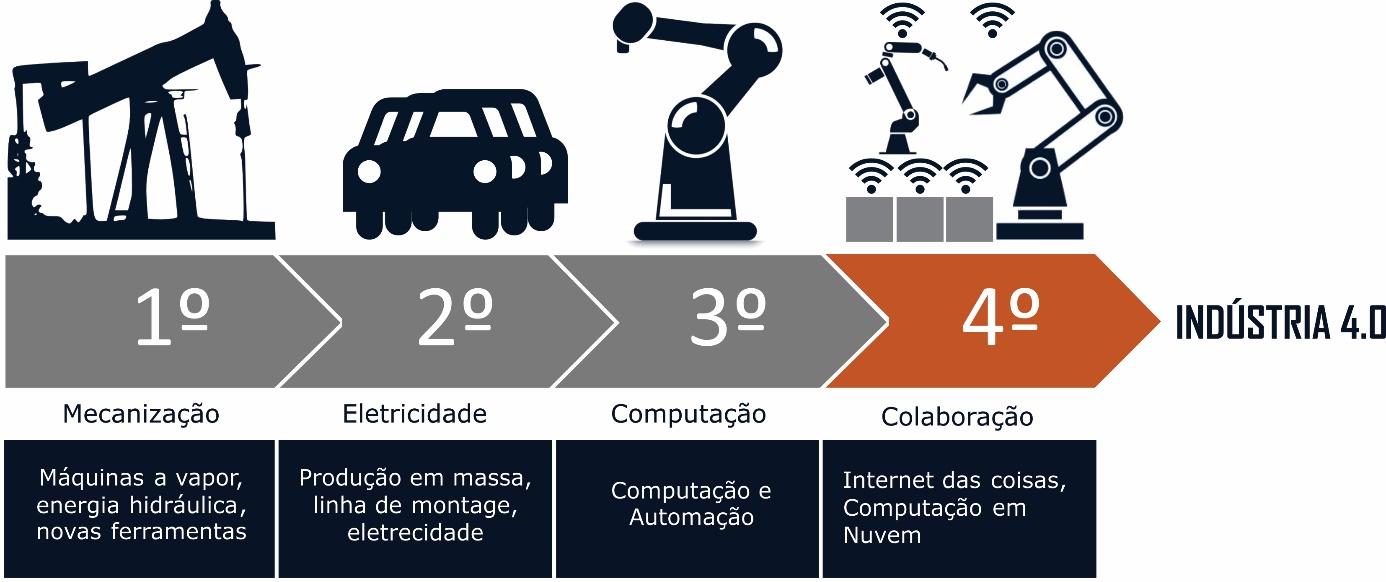


FIGURA 1 – Revoluções industriais

FONTE - www.group-promotion.com

A primeira revolução industrial iniciou-se na Inglaterra, no século XVIII (1780-1840), foi impulsionada por tecnologias como máquinas a vapor e linhas férreas. A segunda revolução industrial começou por volta de 1870. Mas a transparência de um novo ciclo só se deu nas primeiras décadas do século XX, tendo como inovações a metalurgia, a eletricidade, a eletromecânica, o petróleo, o motor a explosão, a petroquímica a linha de montagem e a difusão da produção em massa. A terceira revolução teve início na década de 1960, tendo por base a alta tecnologia, a tecnologia de ponta, rompeu com paradigmas por meio do desenvolvimento de semicondutores e tecnologias como computadores pessoais, mainframes e, mais tarde, nos anos 1990 com a popularização da internet.

No entanto, com um vasto progresso e propagação de algumas das tecnologias da terceira revolução industrial, [do mesmo modo que](https://www.sinonimos.com.br/do-mesmo-modo-que/) o chegada e integração de outras tecnologias, autores têm insinuado que, no início do século XXI, teríamos dado início a essa que seria a quarta revolução industrial. (FERREIRA, STOCK, 2016, p. 2)

De acordo com professor Klaus Schwab (apud FERREIRA, STOCK, 2016, p. 2), em um trabalho publicado em 2016 intitulado The Fourth Industrial Revolution, estamos vivenciando a quarta revolução industrial. Esta revolução seria uma revolução digital movida por meio de tecnologias como inteligência artificial, internet móvel, automação, “*machine learning*” (robôs e computadores com capacidade de autoprogramação), sensores aperfeiçoados (menores e mais potentes possibilitam o uso da internet das coisas). Avanços nos estudos de genética e nanotecnologia também, são apresentados por Schwab como causadores desta revolução em curso.

De acordo Ferreira e Stock (2016, p. 3) Apesar de algumas das tecnologias da quarta revolução industrial como *softwares, hardwares* e a internet tenham sido desenvolvidas na terceira revolução industrial, estas passaram e estão passando por um aperfeiçoamento visível. Tendo como exemplo, o barateamento e propalação do acesso da internet, junto do aumento da velocidade e capacidade de transferência de dados e o surgimento da internet móvel, que fizeram com que a internet se tornasse onipresente se confrontada à mesma internet dos anos 1990. As tecnologias como a internet das coisas e o Big Data que juntas, proporcionam maior envolvimento de consumidores por meio de produtos e serviços que antecipam necessidades dos usuários. O aperfeiçoamento destas tecnologias, conciliado a recentes inovações, fornecem oportunidades que definem a quarta revolução industrial.

Segundo relatório do Boston Consulting Group (apud FERREIRA, STOCK, 2016, p. 2), são nove as mais importantes tecnologias empregadas na indústria 4.0, sendo estas de grande importância para o crescimento e produtividade das indústrias sobre este novo conceito. São elas:

1. Manufatura aditiva: criação de peças em impressoras 3D, que forma o produto por meio de adição de matéria-prima, sem a necessidade do uso de moldes.
2. Integração horizontal e vertical de sistemas: sistemas de informação que incorporam uma cadeia de valor automatizada, por meio da digitalização de dados.
3. Internet das coisas industrial: máquinas são conectadas, por via de sensores e dispositivos, a uma rede de computadores, proporcionando a centralização e a automação do controle e da produção.
4. Robôs automatizados: além das atuais funções, no futuro aos robôs estarão aptos a comunicar-se r com outras máquinas e com os humanos, tornando-se mais flexíveis e cooperativos.
5. Big Data e Analytics: capaz de detectar falhas nos processos da empresa, auxilia a melhorar a qualidade da produção, poupa energia e torna mais eficiente a utilização de recursos na produção.
6. Nuvem: banco de dados criado pelo usuário, capaz de ser acessado de qualquer lugar do mundo, por meio de uma infinidade de dispositivos conectados à internet.
7. Simulação: permite operadores testarem e otimizarem processos e produtos ainda na fase de concepção, diminuindo os custos e o tempo de criação.
8. Segurança cibernética: meios de comunicação cada vez mais confiáveis e sofisticados.
9. Realidade aumentada (“Augmented Reality”): sistemas baseados nesta tecnologia executam uma variedade de serviços, como selecionar peças em um armazém e enviar instruções de reparação por meio de dispositivos móveis.

O emprego da indústria 4.0 traz ganhos de produtividade, substituição do trabalho por capital e retornos crescentes de escala.

## 2.4 MICROSOFT ACTIVE DIRECTORY

O *Active Directory* (AD)é uma implementação de serviço de diretório, armazena informações sobre objetos em rede de computadores e disponibiliza essas informações a usuários e administradores desta rede.

O AD surgiu junto com o lançamento do Windows 2000 Server, após o seu nascimento assumiu a liderança dos serviços de diretório, utilizando como base *o Lightweight Directory Access Protocol* **(**LDAP) e a comunicação através de replicação lançou vários atributos e principalmente ferramentas para facilitar o gerenciamento de informações nas empresas. O ADassumiu o mercado de serviços de diretório pelo seu desempenho, segurança e principalmente disponibilidade.

O desenvolvimento do *Active Directory* trouxe várias facilidades para os administradores de redes. Antes do seu surgimento, os usuários que faziam uso de sistemas nas empresas tinham um grande problema, o esquecimento de suas inúmeras senhas para diversas ampliações diferentes.

Com a implementação do *Active Directory,* tornou-se possível que um usuário tivesse apenas uma senha sincronizada com diversas aplicações utilizadas em uma empresa. O AD centraliza recursos fazendo uso de um banco de dados que possui as principais dados, como grupos, membro de grupos e senhas. Com isso facilitou o acesso e conferencia das informações, para manutenção dos recursos e administração do AD. (TORRES, 2001).

**2.5.1 Principais Funções do AD**

De acordo com RANDECK (2012), as principais funções do *Active Directory* no *Windows Server* 2008 são:

1. Serviços de Domínio no AD: em seu banco de dados mantém os dados dos usuários, computadores e outros serviços que são encontrados na rede. Essa facilidade auxilia o gerenciador da rede a verificar e manter a integridade de dados nele cadastradas. Para essa função funcionar corretamente, é necessária a sua instalação na rede para que todos os aplicativos possam sincronizar os seus recursos de dependência do *Active Directory*.
2. AD *Rights Managenebyte Services*: função responsável pela proteção dos dados. Garante que apenas usuários registrados possam ter acesso a determinado aplicativo, mantendo os dados seguros e persistentes.
3. AD *Lightweight Directory Services*: função que serve para aplicativos que estão disponíveis em um diretório. Fornece armazenamento e recuperação dos dados, que são otimizados para leitura.
4. Serviço de Federação do AD: faz com que o *Active Directory* possa ser utilizado em diversas plataformas e via web.
5. Serviço de Certificado no AD: gerencia cia a segurança baseando-se em certificados de chaves públicas, que podem ser vinculados a uma chave de privada aos aplicativos correspondentes.

# 3 OBJETIVO DO PROJETO

Neste capítulo descreve-se a problemática na qual o projeto se insere, seus objetivos gerais e específicos, a justificativa para o seu desenvolvimento, o público do futuro sistema e, por fim, seus níveis de decisão e grupos funcionais atendidos.

## 3.1 FORMULAÇÃO DO PROBLEMA

No atual contexto da empresa Fênix Indústria de Eletrônicos Ltda, o sistema interno (intranet) e o *website* (extranet) não são integrados, isso dificulta a manutenção e causa retrabalho por parte do setor de tecnologia da informação, e consequentemente a usabilidade dos usuários por conter vários links de diferentes ferramentas. Há várias aplicações em uso, porém, não existe um canal único que reúna todos os caminhos de acesso (*link’s*) dessas aplicações de forma organizada e especializada por usuários de cada setor e suas respectivas restrições ou permissões de acesso.

O desenvolvimento de um *website* que integre todos os setores da empresa e funcione como extranet e intranet, torna-se a solução adequada ao problema apresentado. Agregando o conceito de Indústria 4.0 que engloba as principais inovações tecnológicas, Internet das Coisas e Internet dos Serviços, os processos tendem a se tornar cada vez mais eficientes, autônomos e customizáveis.

## 3.2 OBJETIVOS

Este projeto tem como objetivo principal o desenvolvimento de um sistema capaz de melhorar a automação, o desempenho e a eficiência dos processos de negócios da empresa Fênix Indústria de Eletrônicos Ltda, e permitindo a evolução dos sistemas utilizados para uso em seus processos organizacionais.

Para atender o objetivo, será desenvolvido um sistema para plataforma *web*, chamado Extranet Fênix que consiste em atender as áreas de produção e marketing da empresa. Por meio dele, será gerenciado todo o ambiente produtivo de qualquer lugar a qualquer hora, adentrando no conceito de indústria 4.0 ou indústria inteligente, que engloba as principais inovações tecnológicas dos campos de automação, controle e tecnologia da informação.

## 3.3 JUSTIFICATIVA

O projeto apresentado apoia-se no fato de que, apesar de já existir sistemas computacionais que realizam os processos e procedimentos adotados pela empresa, surge um desafio de levar a organização para o ramo da Indústria 4.0 ou indústrias inteligentes este é um conceito de indústria proposto recentemente e que engloba as principais inovações tecnológicas dos campos de automação, controle e tecnologia da informação, aplicadas aos processos de manufatura, integrando todo o sistema computacional da empresa. Esta revolução industrial permitirá a gestão empresarial um controle das informações e dos recursos disponibilizados pela empresa.

## 3.4 PÚBLICO-ALVO

O sistema em questão é destinado à empresa Fênix Indústria de Eletrônicos Ltda fundada em 2010, pertencente do grupo Foxconn fundada em 1974 em Taiwan, mantendo sua sede em Santa Rita do Sapucaí, no estado de Minas Gerais. Atualmente, 150 empresas constituem o Arranjo Produtivo Eletroeletrônico (APL) de Santa Rita, e atuam no desenvolvimento e na produção de eletrônicos para atender mais de 41 países.

A Fênix possui como missão, “prover soluções de manufatura para produtos eletrônicos de forma a atender um mercado exigente e inovador”. (Disponível em: <http://www.fenixindustria.net.br>).

Possui como visão: “Expandir nossas operações de forma consistente com as expectativas dos clientes e nossos requisitos legais, acompanhando o rápido desenvolvimento tecnológico, através da aplicação eficaz dos nossos processos de melhoria contínua”. (Disponível em: <http://www.fenixindustria.net.br>).

Possui como valores a S.I.N.E.R.G.I.A : S- Sustentabilidade, I– Igualdade, N- Naturalidade, E- Eficiência, R- Respeito, G- Gestão, I- Inovação, A- Aprendizado.

## 3.5 NÍVEIS DE DECISÃO E GRUPOS FUNCIONAIS ATENDIDOS

Este projeto é voltado para atender os níveis operacional e tático da empresa Fênix Indústria de Eletrônicos Ltda.

Os grupos funcionais atingidos pelo sistema são desde um colaborador operacional até a gerência de marketing e de produção.

A Extranet Fênix apoiará os níveis de decisão operacional, pois visa dar resposta imediata aos problemas do dia a dia, seguindo as orientações táticas ou gerenciais contando com processamento de transações e controle de dados, alocando recursos para atingir objetivos e metas produtivas.

A Figura 2 representa os níveis de decisão e os grupos funcionais atendidos pelo projeto.

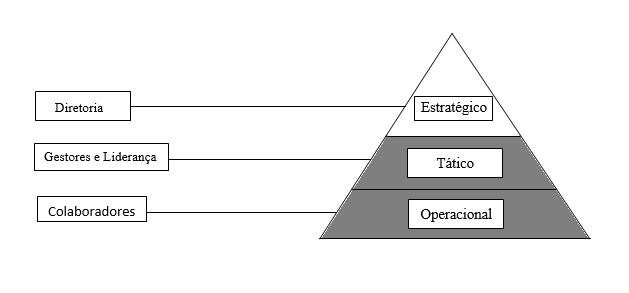


FIGURA 2 - Níveis de Decisão e Grupos Funcionais Previstos

FONTE: Elaborado pelos autores (2017).

# 4 GERÊNCIA DO PROJETO

Segundo Sommerville (2003, p.60), “necessitamos do gerenciamento de projetos de software porque a engenharia de software profissional está sempre sujeita a restrições de orçamentos e de prazo”.

A gestão de projetos é um conjunto de práticas que servem de guia para a um grupo trabalhar de maneira produtiva. Ela compreende métodos e ferramentas que organizam as tarefas, identificam sua sequência de execução e dependências existentes, apoia a alocação de recursos e tempo, além de permitir o rastreamento da execução das atividades e medição do progresso relativo ao que foi definido no plano de projeto. (SPÍNOLA, 2010, p.64).

No capítulo que segue, é apresentado o plano de projeto contendo o modelo de ciclo de vida adotado, as partes interessadas, recursos e esforços necessários, bem como as estimativas de esforço do projeto e a descrição da gestão de tempo, gestão de integração, gestão de escopo, gestão de risco e gestão de qualidade.

## 4.1 PLANO DE PROJETO

Nesta seção são apresentadas as partes interessadas no projeto com seus respectivos graus de poder e interesse, o modelo de ciclo de vida adotado para seu desenvolvimento, os recursos de *software*, *hardware* e humanos para sua implementação e a estima de tempo para a sua realização.

### 4.1.1 Partes Interessadas

Os interessados ou *Stakeholders são* todas as pessoas ou grupos que possuem participações e interesses em uma determinada empresa, negócio ou projeto. Esta seção trata destes interessados, baseado na matriz "poder x interesse".

a) Empresa Fênix: cliente, que possuirá a licença de uso do *WebSite* Extranet Fênix.

b) Supervisor de Tecnologia da Informação da Fênix Thiago Moraes: principal representante da cliente Fênix.

c) Orientadora deste projeto e professora da disciplina de Projeto Final de Curso, Profa. Ma. Eunice Gomes de Siqueira: tem o papel de sugerir, propor, avaliar o projeto realizado pela equipe.

d) Gerente de Projeto: papel do integrante Jônatas Balestra de Oliveira, que consiste em planejar e controlar a execução do projeto.

e) Equipe de desenvolvimento: composta por mais três integrantes Átiley de Cássia Ribeiro Silva, Gabriel da Silva Coelho e Sanara Maciel Felício, consiste no desenvolvimento do *WebSite,* produto deste projeto.

f) FAI – Centro de Ensino Superior em Gestão Tecnologia e Educação: papel na orientação do projeto de conclusão de curso em conjunto com seu corpo docente.

A Figura 4 representa a matriz de poder x interesse das partes em relação ao projeto.

FIGURA 3 - Partes interessadas alocadas em uma matriz de poder x interesse

### 4.1.2 Modelo de Ciclo de Vida

Segundo Pressman (2011), um modelo prescritivo de processo de software é um conjunto de elementos que inclui ações de engenharia de software, produtos de trabalho e mecanismos que possam garantir a qualidade e o controle de modificações em cada projeto necessárias para o desenvolvimento de um sistema de software. O modelo de ciclo de vida adotado é o modelo Incremental (Figura 4), com técnicas de prototipação.

O modelo de ciclo de vida Incremental é baseado na modelo Cascata ou Clássico, pelo qual o desenvolvimento do software é dividido em várias etapas denominadas “incrementos”, que produzirão incrementalmente o sistema, até a sua versão final estando de acordo com um conjunto de prioridades definidas nos requisitos do sistema.

Este modelo é recomendado quando se tem requisitos relativamente estáveis, pequenas equipes de projeto e riscos técnicos de projeto que necessitam ser gerenciados.

O Modelo Incremental apresenta diversas vantagens para o desenvolvimento de um software, como: pode-se adicionar novas funções ao software no final de cada incremento; a previsibilidade de riscos envolvidos no escopo do projeto, o cliente pode esclarecer melhor os seus requisitos e suas prioridades para os próximos incrementos. Em contrapartida, esse modelo como todos outros possuem algumas deficiências, dentre elas sua rigorosa formalidade, exigindo um grande cuidado na definição dos incrementos para que ele não se aproxime do modelo Clássico.

A Figura 4 representa o modelo de ciclo de vida adotado pelo projeto.

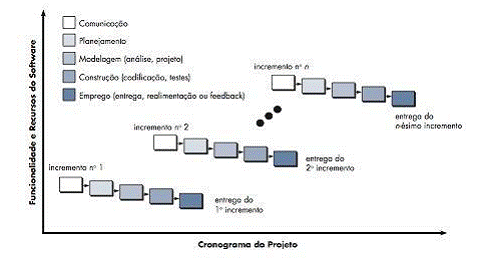


FIGURA 4 - Modelo de Ciclo de Vida Incremental

FONTE: - Pressman (2016)

### 4.1.3 Recursos Necessários

Nesta seção são apresentados os recursos necessários para a realização do projeto Extranet Fênix, sendo estes recursos humanos, de hardware, de software, entre outros.

1. Recursos Humanos: equipe formada por quatro integrantes de Sistemas de Informação.

|  |  |
| --- | --- |
| Gerente de Projeto | Jônatas Balestra de Oliveira |
| Desenvolvedores | Átiley De Cássia Ribeiro Silva, Gabriel da Silva Coelho e Sanara Maciel Felício |
| Elaboração de Relatórios | Átiley De Cássia Ribeiro Silva e Jônatas Balestra de Oliveira |

QUADRO 1 - Recursos Humanos

FONTE: Elaborado pelos autores (2017).

Os recursos humanos requerem ainda conhecimento das seguintes tecnologias:

1. Conhecimento em C# (CSharp);
2. Conhecimento de Engenharia de Software;
3. Conhecimento em *Unified Modeling Language* (U.M.L);
4. Conhecimento em Programação Orientada a Objetos;
5. Conhecimento em HTML,CSS e *Jquery*;
6. Conhecimento em SQL Server;
7. Conhecimento no Microsoft *Active Directory*.
8. Recursos de Hardware: são necessários os equipamentos a seguir.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tipo** | **Marca** | **Sistema Operacional** | **Processador** | **Memória RAM** | **Espaço de armazenamento permanente** |
| Desktop | Não possui | Windows 7 Pro x64 | Intel Core I5 - 4 GHz | 16 GBytes | 500 GB |
| Notebook | Dell | Windows 10 Pro X64 | Intel Core I3 - 2GHz | 4 Gbytes | 1 TB |
| Notebook | Lenovo | Windows 7 Pro x64 | Intel Core I3 2.4GHz | 4 Gbytes | 500 GB |
| Notebook | HP | Windows 7 Ultimate x64 | Intel Core I5 2.53GHz | 3 GBytes | 500 GB |

QUADRO 2 - Recursos de Hardware

FONTE: Elaborado pelos autores (2017).

1. Recursos de Software: no desenvolvimento do projeto, são utilizadas as seguintes ferramentas.

|  |  |
| --- | --- |
| **Ferramenta** | **Versão** |
| Visual Studio IDE | 2017 RC |
| Oracle SQL | 11G |
| BrModelo | 2.0 |
| Microsoft Project | 2016 |
| Visual Paradigm for UML | 13.0 |
| Google Docs | 2017 |
| Microsoft Visio | 2016 |
| Git Hub | 2017 |
| WBS Tool | 2017 |

QUADRO 3 - Recursos de Software

FONTE: Elaborado pelos autores (2017).

### 4.1.4 Estimativas de Tempo

Nesta seção é apresentado o processo de estimativas de projeto de software, utilizando a métrica de Pontos de Função (PF), para o tamanho funcional do projeto de software, e Pontos por Casos de uso (PCU).

As planilhas com o detalhamento das estimativas de pontos de caso de uso e de pontos de função do Extranet Fênix encontram-se nos Apêndices A e B respectivamente.

|  |  |  |  |
| --- | --- | --- | --- |
| Métrica | 2ª. Fase (HH) | 3ª. Fase  (reestimativa) | 4ª. Fase  (reestimativa) |
| Pontos por função | 2436 |  |  |
| PCU - Schneider e Winters | 721 |  |  |
| PCU - Karner | 721 |  |  |

QUADRO 4 - Estimativas de esforço

## 4.2 ÁREAS DE CONHECIMENTO

Segundo o Guia PMBoK, elaborado pelo *Project Management Institute* (PMI, 2004), uma área de conhecimento é definida por seus requisitos de conhecimentos e descrita em termos dos processos a que compõem, suas práticas, entradas, saídas, ferramentas e técnicas.

Esta seção aborda as áreas de conhecimento de gestão de: escopo, tempo, integração qualidade e riscos.

### 4.2.1 Gestão de Escopo

De acordo com o PMI, o gerenciamento do escopo do projeto é composto dos processos para garantir que o projeto inclua todo o trabalho exigido, e somente o trabalho exigido, para completar o projeto com o sucesso (PMI, 2004).

4.2.1.1 Estrutura analítica do projeto (EAP)

A EAP é uma poderosa ferramenta para facilitar o entendimento do projeto por parte da própria Equipe de Gerenciamento, bem como na comunicação com todos os demais participantes e interessados no projeto. Ela é definida pelo PMI como:

Uma decomposição hierárquica orientada às entregas do trabalho a ser executado pela equipe para atingir os objetivos do projeto e criar as entregas requisitadas, sendo que cada nível descendente da EAP representa uma definição gradualmente mais detalhada da definição do trabalho do Projeto (PMI, 2004, p. 116).

A EAP do Projeto Extranet Fênix pode ser encontrada no Apêndice C.

4.2.1.2 Dicionário EAP

O dicionário da EAP traz todo detalhe necessário para cada elemento da EAP de modo a orientar a equipe do projeto. Ele contém informações sobre como o trabalho será realizado, e outras questões técnicas.

Segundo o PMI (2004), ele pode servir como parte de um sistema de autorização de trabalho descrevendo para os integrantes da equipe cada componente da EAP. Pode ser usado também para controlar quando um trabalho específico é realizado, de modo a evitar aumento do escopo e aumentar a compreensão das partes interessadas sobre o esforço necessário para cada pacote de trabalho. O dicionário de dados da EAP define limites do que é incluído no pacote de trabalho.

O dicionário da EAP pode ser encontrado no Apêndice D.

### 4.2.2 Gestão do Tempo

O gerenciamento de tempo inclui processos necessários que permitem as entregas do projeto dentro dos prazos estabelecidos.

Define e realiza a sequência das atividades necessárias, estima os recursos e duração de atividades para desenvolver e controlar o cronograma planejado.

4.2.2.1 Lista de Atividades

É uma lista abrangente que inclui todas as atividades do cronograma planejadas para serem realizadas no projeto.

A lista de atividades inclui o identificador da atividade e uma descrição do escopo do trabalho para cada atividade do cronograma suficientemente detalhados para garantir que os membros da equipe do projeto compreendam que trabalho precisará ser terminado. Segundo o PMI (2012, p.133), “definir as atividades é um processo de identificação das ações específicas a serem realizadas para produzir as entregas”.

A lista de atividades do projeto está disponível no Apêndice D.

4.2.2.2 Diagrama de Rede

O diagrama de rede é o resultado de processos que sequenciam as atividades, no qual envolve identificar as atividades e marcos do projeto e colocá-los numa ordem lógica.

O diagrama de rede do projeto está disponível no Apêndice D.

4.2.2.4 Cronograma das Atividades

O cronograma é a composição de uma lista de atividades interligadas por relações de dependência, que aplicadas sobre um calendário e após a análise da disponibilidade de recursos humanos/ materiais, possibilita a identificação e controle da data de realização de atividades.Segundo Heldman (2006, p. 178), “o objetivo do processo Desenvolver o Cronograma é definir as datas de início e de término de cada atividade do projeto.”.

O cronograma do projeto está disponível no Apêndice D.

4.2.2.4 Quadro com Resumo de Esforço

O Quadro 5 a seguir apresenta a comparação de tempo de realização de cada etapa do projeto. Nela é possível visualizar as estimativas de tempo:

1. *ad-hoc*, sendo a quantidade de horas estimadas antes da criação das atividades e cronograma do projeto;
2. dicionário EAP, sendo o total horas das atividades previsto nos pacotes de trabalho;
3. efetivo realizado, sendo o total de horas gastas para a realização das atividades.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1ª. Fase | 2ª. Fase | 3ª. Fase | 4ª. Fase |
| Estimativa Ad-hoc | 165 horas | 161horas |  |  |
| Dicionário EAP | 121 horas  121 horas | 161 horas  282 horas |  |  |
| Efetivo realizado | 108 horas  108 horas | 159 horas  267 horas |  |  |

QUADRO 5 - Esforços planejados e realizados

### 4.2.3 Gestão da Integração

Gerenciamento de integração inclui características de unificação, consolidação, comunicação e ações integradoras que são essenciais para a execução controlada do projeto até a sua conclusão, a fim de gerenciar com sucesso as expectativas das partes interessadas e atender aos requisitos, segundo o Guia PMBOK (PMI, 2013).

4.2.3.1 Monitoramento

Os processos de monitoramento permitem verificar se os trabalhos do projeto estão ocorrendo conforme planejados, bem como medir seu desempenho.

O Relatório de Desempenho do projeto Extranet Fênix pode ser encontrado no arquivo eletrônico, citado no Apêndice E.

4.2.3.2 Controle de Configuração

Um controle de configuração é um conjunto de atividades de apoio que permite a absorção ordenada das mudanças inerentes ao desenvolvimento de software, mantendo a integridade e a estabilidade durante a evolução do projeto. Segundo Pressman (2006):

Tais atividades definem o mecanismo para o gerenciamento de diferentes versões deste produto, proporcionando um controle sobre as mudanças impostas, além de possibilitar o desenvolvimento paralelo pelos membros da equipe, impactando positivamente na produtividade e qualidade do projeto.

Como repositório do projeto é utilizado o serviço *GitHub* para de controle de versão de código fonte. Através do Git é possível trabalhar sempre em um mesmo diretório, podendo fazer as alterações do projeto, gravando documentação e ainda comentários. Sendo assim, ele é capaz de gravar tudo o que foi feito. O registro é muito útil para que se possa voltar à fase anterior quando surgir um possível problema. A equipe pode ainda desfazer as alterações que aparecem com problemas, voltando para a versão original quando desejar.

A Figura 5 apresenta o repositório para de controle de versão *GitHub* utilizado no projeto.

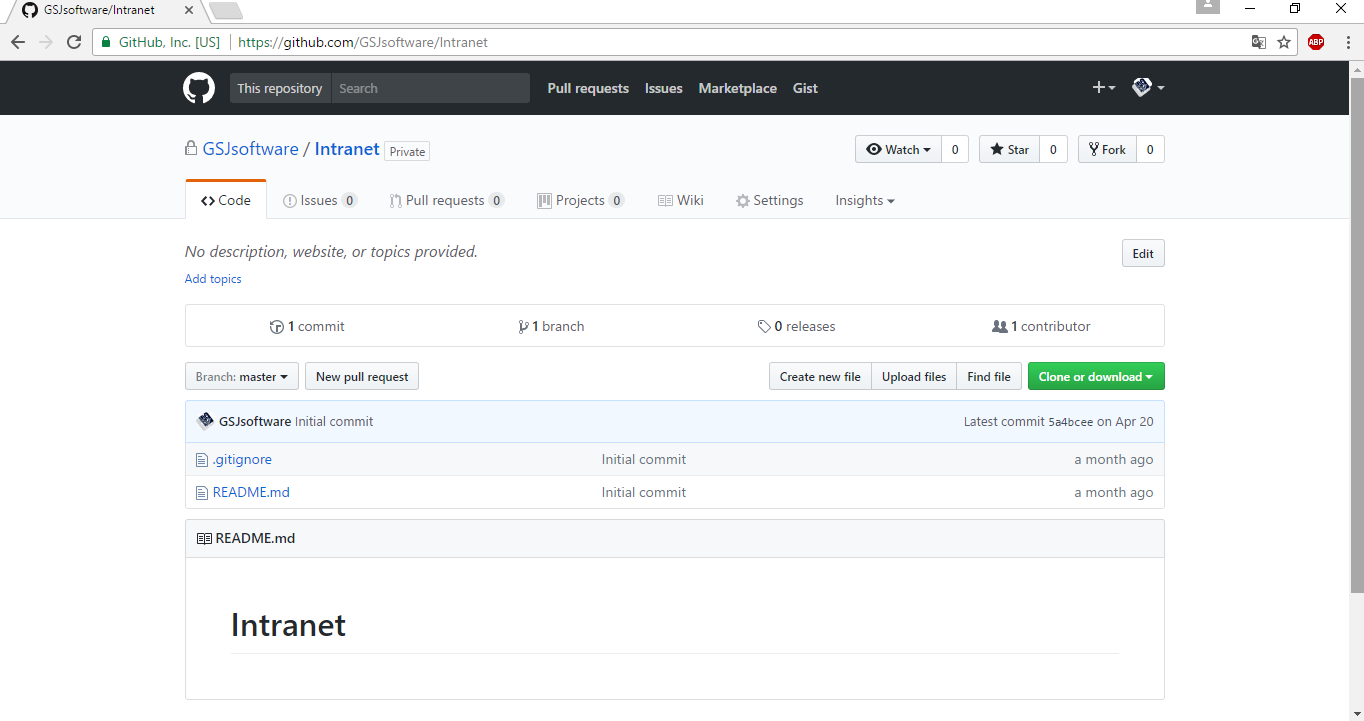


FIGURA 5 – Repositório GitHub

FONTE: Elaborado pelos autores (2017).

Outro serviço utilizado é o *Google Drive*, que disponibiliza gratuitamente 15GB de armazenamento *on-line* do Google. Com ele, é possível desenvolver e alterar o documento paralelamente pelos membros do grupo, sendo que pode disponibilizar qualquer arquivo, usando a conta do Google e compartilhá-los com diferentes usuários, além de gerenciar permissões.

A Figura 6 apresenta o repositório de arquivos Google Drive utilizado no projeto.

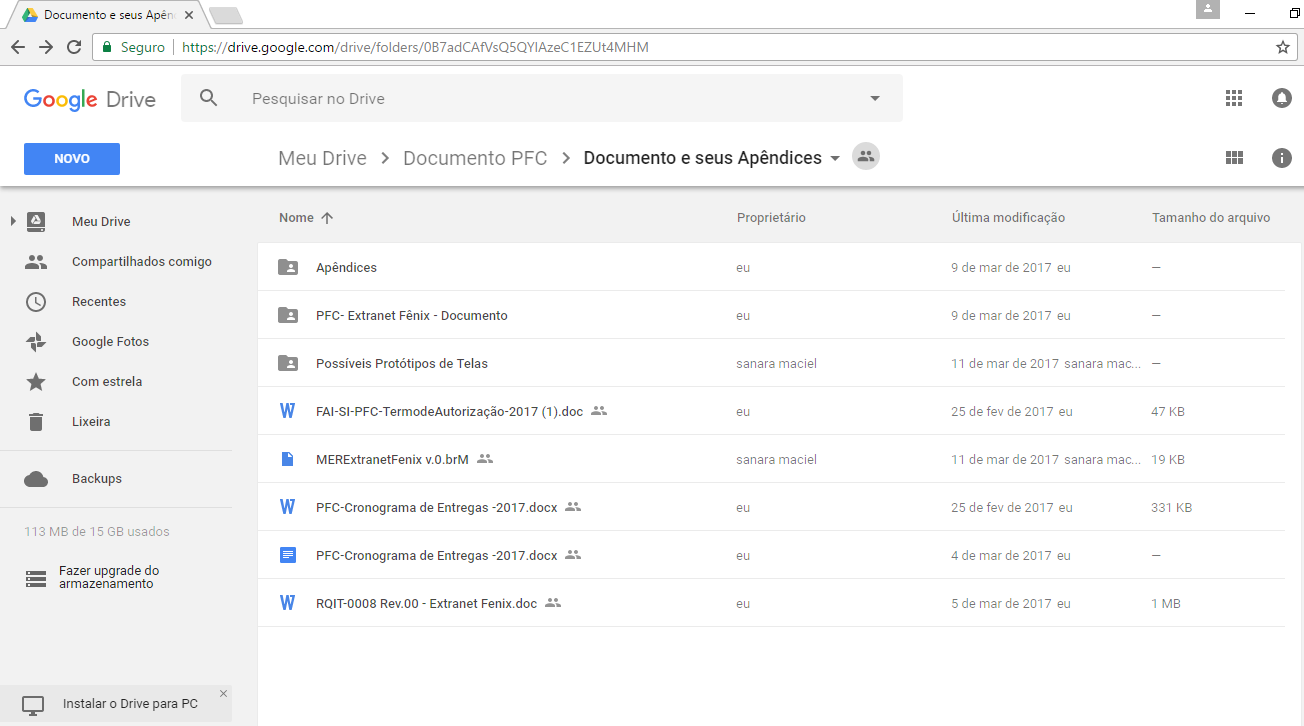


FIGURA 6 – Repositórios de arquivos Google Drive

FONTE: Elaborado pelos autores (2017).

4.2.3.3 Controle de Mudanças

Controle de mudança é uma ferramenta de apoio para as atividades da gerência de configuração. Mudanças aparecem durante todo o desenvolvimento e devem ser registradas, avaliadas e agrupadas de acordo com sua prioridade, sendo assim, é possível planejar melhor o escopo, prazo e o custo de cada iteração. Em seguida, à medida que o desenvolvimento acontece, pode-se acompanhar o estado da solicitação da mudança até sua implementação e até o lançamento de uma versão em produção.

No projeto Extranet Fênix, as solicitações de mudanças são realizadas de forma verbal ou via *e-mail*. Primeiramente são levantados os possíveis impactos dos custos e benefícios que a mudança pode gerar, portanto antes de qualquer decisão, a mudança é analisada e revisada pela equipe e o gerente de projeto, sob a orientação da Profª. Ma. Eunice Gomes de Siqueira para que juntos decidam se a mudança será viável ou não.

No caso de aprovação de mudança, é feita uma nova análise no escopo do projeto para implantação da mesma. Deve-se ressaltar que é importante manter a integridade do projeto como também incorporar as modificações que foram aprovadas ao plano de gerenciamento do projeto e na documentação.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mudança** | **Responsável** | **Data** | **Situação** |
| Alteração do Capítulo revisão bibliográfica e inclusão da seção Indústria 4.0. | Jônatas | 08/04/2017 | Realizado |
| Revisão dos objetivos e da formulação do problema. | Jônatas e Átiley | 17/04/2017 | Realizado |
| Avaliação e melhor definição dos requisitos funcionais e não funcionais | Jônatas | 12/05/2017 | Realizado |
| Verificação de autores e obras presentes nas referências bibliográficas | Jônatas Átiley | 30/06/2017 | Realizado |

QUADRO 6 – Principais Mudanças

### 4.2.4 Gestão da Qualidade

Segundo Valeriano (2005), a gestão da qualidade visa garantir que todas as necessidades que originaram o projeto serão satisfeitas. Inclui também todas as atividades pertencentes a qualquer nível de gerência, que determinem tanto a política, como os objetivos e as reponsabilidades.

As atividades que fazem parte do projeto foram distribuídas entre os componentes da equipe. Após a realização de cada atividade, esta é revisada pelo gerente de projeto, Jônatas Balestra de Oliveira

Nesta seção são apresentadas as medidas tomadas para o controle da qualidade do início ao termino do projeto, também foi determinado o responsável por monitorar o atendimento desses critérios.

A gestão de qualidade do projeto foi dividida em duas etapas: controle da qualidade da documentação e controle da qualidade do *software*.

As medidas tomadas para assegurar a qualidade da documentação são listadas a seguir:

1. Usar como base trabalhos semelhantes e fontes de referência confiáveis;
2. Ajustar o documento de acordo com as Diretrizes para Elaboração de trabalhos científicos da FAI;
3. Averiguar se todas as citações contidas no documento foram corretamente referenciadas no capítulo Referências, e se se todas as fontes referenciadas no capítulo referências constam no decorrer do documento;
4. Assegurar o entendimento e adequação de ilustrações (tabelas, gráficos, figuras) de acordo com o conteúdo abordado;
5. Garantir a elaboração do conteúdo de cada capítulo dentro do prazo especificado no cronograma de atividades;
6. Revisar e corrigir o documento antes de cada entrega (Fase I, Fase II, Fase III e Fase IV);
7. Corrigir o documento após observações feitas pela orientadora;
8. Garantir as referências dos apêndices de acordo com o conteúdo exposto, tanto no arquivo de mídia, como no documento escrito;
9. Manter o sumário atualizado, assim como, a lista de figuras, lista de quadros e de tabelas;
10. Criar uma lista de abreviaturas e de siglas quando estas são utilizadas mais de uma vez no documento e manter a lista em ordem alfabética;
11. Definir um código para controle de versão e mudanças realizadas no projeto;
12. Definir um critério de back-up dos arquivos projeto;
13. Buscar a conformidade da documentação do projeto conforme o Modelo de Documento da FAI para o Projeto final de Curso.

Para garantir a qualidade do software são seguidos os seguintes critérios:

1. Realizar a documentação necessária de análise antes do desenvolvimento do sistema.
2. Adotar e respeitar o conjunto de padrões definidos pelo *framework* ASP.Net MVC.
3. Definir uma data de entrega das partes do software possível de ser cumpridas.
4. Realizar e monitorar a testes de unidade, integração, sistema e validação.
5. Acrescentar comentários no código para melhor entendimento dos blocos de códigos.
6. Garantir a implementação dos requisitos funcionais e não funcionais estabelecidos na fase de levantamento de dados.
7. Implementar o *software* de modo que facilite a sua manutenção e reaproveitamento de código.
8. Revisar e integrar periodicamente os artefatos produzidos pelo processo de desenvolvimento.
9. Utilizar mecanismos adequados de armazenamento e controle de versão dos artefatos produzidos.
10. Definir os recursos e ferramentas adequados para o desenvolvimento do *software.*
11. Buscar a melhoria constante do processo de desenvolvimento do *software.*

### 4.2.5 Gestão dos Riscos

Risco do projeto é um evento que poderá ou não ocorrer. Caso ocorra, poderá ter um efeito positivo ou negativo nos objetivos do projeto. O gerenciamento de riscos, conforme PMBOK, é o processo de identificação, análise, desenvolvimento de respostas e monitoramento dos riscos em projeto, com o objetivo de diminuir impactos negativos e de aumentar eventos de impactos positivos.

Conforme o citado no guia PMBOK, o processo irá se preocupar com a atribuição de responsabilidades, quais os limites e tolerâncias a riscos das partes interessadas, quais as definições para probabilidade e impacto, quais serão os recursos, tempo e orçamento serão alocados para o gerenciamento dos riscos, assim como documentar tudo isto.

O plano de gerência de riscos deste projeto está contido no Apêndice F.

# 5 ESPECIFICAÇÃO E ANÁLISE DOS REQUISITOS

# 

“A engenharia de requisitos estabelece uma base sólida para o projeto e para a construção. Sem ela, o software resultante tem grande probabilidade de não atender as necessidades do cliente”. (PRESSMAN, 2011, p. 127).

Este capítulo descreve todos os requisitos funcionais e não funcionais necessários para o correto funcionamento do sistema, a visão dos casos de uso e a visão dos dados.

## 5.1 DESCRIÇÃO DE REQUISITOS

Sommerville (2007) classifica os requisitos de sistema de software em funcionais, não funcionais e requisitos de domínio:

* Requisitos funcionais: definem as funcionalidades do sistema como deve reagir em condições específicas e como se comportar em determinadas situações. Podem ainda declarar o que o sistema não deve fazer.
* Requisitos não funcionais: são restrições sobre serviços ou funções oferecidas pelo sistema. Dentre elas destacam-se restrições de tempo, sobre o processo de desenvolvimento e de padrões.
* Requisitos de domínio: são restrições originárias do domínio da aplicação do sistema e refletem características do mesmo. Podem ser requisitos funcionais ou não funcionais.

Ainda de acordo com Sommerville (2007) A diferenciação entre esses tipos de requisitos não é tão clara como sugere as definições. Um requisito pode parecer-se inicialmente não funcional, mas quando é desenvolvido com mais detalhes pode dar origem a uma série de novos requisitos funcionais. Na elaboração dos requisitos devemos considerar que na realidade a distinção entre eles é artificial.

### 5.1.1 Requisitos Funcionais

Nesta seção são descritos os requisitos funcionais do Projeto Extranet Fênix separados em módulos, os quais são:

* 1. Módulo Controle de Acesso: este módulo agrupa os requisitos funcionais associados ao controle de acesso dos usuários.
  2. Módulo Administrador: este módulo trata de operações especificas para o administrador do sistema.
  3. Módulo Marketing: este módulo trata de operações especificas para a administração das páginas web.
  4. Módulo Recursos Humanos: este módulo trata de operações específicas para o usuário do setor de recursos humanos.
  5. Módulo Estatísticas : este módulo trata da visualização de gráficos e estáticas e quem pode visualizar esses gráficos.

Estes módulos por sua vez possuem seus requisitos divididos em prioridades, podendo ser “essencial” “importante” ou “desejável”.

1. **Essencial:** requisito de crucial importância para o sistema, sendo imprescindível para seu correto funcionamento;
2. **Importante:** este requisito é importante para o funcionamento satisfatório do sistema, mas que não será impedido de funcionar, caso não seja implantado;
3. **Desejável:** este tipo de requisito pode ser implantado por último sem comprometer o funcionamento do sistema de forma satisfatória

**Módulo Controle de Acesso**

**[RF01] – Controlar Acesso**

Este requisito refere-se à facilidade dos usuários já cadastrados poderem realizar acesso à aplicação. Estes usuários possuem autenticação que será recuperada do *Active Directory* (AD) da Microsoft *Windows*.

Não será permitida a alteração de dados cadastrais através do sistema. As alterações cadastrais serão realizadas por um funcionário autorizado, onde o *Active Directory* (AD) da Microsoft *Windows*, apenas irá resgatar essas alterações realizadas no cadastro.

**Prioridade:** Essencial.

**[RF02] – Finalizar Sessão**

Este requisito refere-se à facilidade de um usuário que esteja com acesso [RF 01] possa sair da aplicação.

**Prioridade:** Essencial.

**[RF03] Manter Histórico**

Este requisito refere-se à função de armazenamento de dados de acesso, o sistema armazenará a data e hora de acesso de cada usuário.

**Prioridade:** Essencial.

**Módulo Administrador**

**[RF04] - Manter Cadastro de Grupo de Usuários**

Este requisito refere-se à facilidade que o usuário administrador terá para cadastrar grupos de usuários. Um grupo de usuários determina privilégios para acesso às funções especificas do sistema. É restrita a exclusão de um grupo de usuários se este ainda possuir um usuário associado.

**Prioridade:** Essencial.

**[RF05] – Mante Cadastro Menu**

O sistema deverá possibilitar o cadastro de um menu para cada grupo de usuários. Um menu é composto por um conjunto de links externos. Este menu ficará disponível na tela principal da intranet.

Ao selecionar um grupo de usuários será possível adicionar *links* aos quais os usuários pertencentes a este grupo poderão acessar.

Não existe um número limitado de cadastro de *links* para um grupo de usuários.

Será possível a exclusão desses menus/*links*.

**Prioridade:** Essencial.

**[RF06] - Renderizar Menu**

Após cadastrado ou exclusão o menu deve ser renderizado na página inicial da intranet de forma dinâmica.

**Prioridade:** Essencial.

**[RF07] - Associar Usuário ao Grupo de Usuários**

Este requisito refere-se à facilidade do administrador associar um perfil já cadastrado com os usuários recuperados do AD ao grupo de usuário que o mesmo pertence.

Somente o administrador do sistema poderá realizar essa associação.

Não será possível excluir um grupo de usuário que tenha usuários associados, somente desativá-lo.

**Prioridade:** Essencial.

**[RF08] Cadasstrar Dashboard**

Este requisito refere-se à função de visualização de painéis (*dashboard*).

Todos os gráficos deverão ser desenvolvidos com o *google* *charts*.

A visualização de um determinado gráfico será limitada por grupo de usuários.

**Prioridade**: Importante.

**Módulo Marketing**

**[RF09] - Manter Layout do *Website***

Este requisito refere-se à facilidade que o usuário do tipo Marketing possui para configurar a página inicial do sistema.

Deve haver a possibilidade de edição direta do texto, imagens e que fazem parte da página inicial assim como seu plano de fundo.

**Prioridade:** Importante.

**[RF10] Renderizar Página Inicial**

Consiste em renderizar a página inicial de acordo com as modificações realizadas.

**Prioridade:** Importante.

**Módulo Recursos Humanos**

**[RF11] - Manter Postagem de Informação**

Este requisito refere-se à facilidade que o usuário de perfil relacionado à área de RH possuirá para realizar as postagens de texto e/ou imagens no sistema.

Essas informações poderão ser vistas por todos os usuários do sistema que estiverem autenticados.

As postagens poderão ser alteradas, criadas e removidas pelo usuário com privilégios para tal.

**Prioridade:** Importante.

**[RF 12] – Renderizar Postagem**

Após realizada o cadastro da postagem deve ser renderizada na página inicial da intranet de forma dinâmica.

**Prioridade:** Importante.

**[RF13] - Manter Cadastro de Currículo**

Este requisito refere-se à facilidade que o usuário de perfil relacionado ao RH possuirá para utilizar

O banco de currículos disponibilizado pelo sistema.

O sistema deverá possuir um ambiente para cadastro de currículo pelo website, sendo esse cadastro realizado por meio da extranet.

Este cadastro de currículo poderá conter arquivos anexados.

Somente o perfil relacionado ao RH possui permissão para acesso ao banco de currículo.

O responsável pela área de currículo, poderá ler e excluir currículo.

**Prioridade**: Importante.

### 5.1.2 Requisitos não Funcionais

Os requisitos não funcionais definem as propriedades do sistema, bem como suas restrições. A classificação adotada está baseada na obra Engenharia de Software (SOMMERVILLE, 2007).

Requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema, são restrições sobre os serviços ou as funções oferecidas pelo sistema e geralmente são mais importantes do que os requisitos funcionais.

Esta seção agrupa todos os requisitos não funcionais, aplicando-se ao sistema como um todo.

5.1.2.1 Requisitos de Produto

Esta seção apresenta todos os RNF de produtos, que especificam o comportamento do produto de software tais como: usabilidade, eficiência (desempenho e espaço), confiabilidade e portabilidade.

1. Requisitos de eficiência

**[RNF01] - Eficiência no uso do sistema**

As requisições do sistema devem ser atendidas com máxima eficiência para que os usuários possam obter respostas corretas de modo ágil.

Este requisito será medido por meio do *feedback* da empresa Fênix, como também os usuários do sistema.

**Prioridade:** Essencial.

1. Requisitos de confiabilidade

**[RNF02] - Acesso Limitado**

O administrador da intranet deverá realizar o cadastro de acesso somente para funcionários da empresa F ênix ou do grupo Foxconn.

**Prioridade:** Essencial.

**[RNF03] - Disponibilidade**

O sistema *web* deve ficar disponível vinte e quatro horas por dia, todos os dias da semana. Para atender esse requisito a empresa Fênix possui um servidor próprio no qual realizam *backup’s* diariamente.

**Prioridade**: Essencial.

**[RNF04] - Segurança**

O sistema *web* oferecerá autenticação a partir do Active Directory via código para que o administrador e os usuários acessem a intranet com segurança.

**Prioridade:** Importante.

1. Requisitos de portabilidade

**[RNF05] - Acesso por diversos dispositivos**

O acesso ao sistema*web* poderá ser feito por meio de dispositivos como: desktops, notebooks, tablets e smartphones.

**Prioridade:** Importante.

**[RNF06] – Navegadores**

O sistema *web* irá funcionar nos principais navegadores como Mozzila Firefox e Google Chrome.

**Prioridade:** Essencial.

1. Requisitos de usabilidade

**[RNF07] - Interface de usuário**

As interfaces para os usuários devem ser intuitivas e interativas, devido a isso o *design* da página inicial do sistema *web* édefinido pelo setor de Marketing da empresa Fênix.

Os usuários devem ser capazes de compreender a linguagem visual e utilizar suas funções de maneira fácil, intuitiva e com elevado nível de satisfação.

**Prioridade:** Essencial.

**[RNF08] - Suporte à internacionalização**

A configuração de idioma deverá ser Português ou Inglês, para que usuários estrangeiros do grupo Foxconn também acessem o sistema web de forma satisfatória.

**Prioridade:** Importante.

5.1.2.2 Requisitos Organizacionais

Esta seção apresenta todos os RNF organizacionais, que são derivados de políticas e procedimentos da organização do cliente e do desenvolvedor*.* Dos quais se abordam entrega, implementação e padrões.

1. Requisitos de entrega

**[RNF09] - Data marco do projeto**

Durante o desenvolvimento do projeto, haverá quatro entregas, conforme o Quadro 6:

|  |  |
| --- | --- |
| **Data** | **Entrega** |
| 01/04/2017 | 1ª entrega |
| 01/07/2017 | 2ª entrega |
| 09/09/2017 | 3ª entrega |
| 04/11/2017 | 4ª entrega |

QUADRO 7 - Entregas do projeto

**Prioridade:** Essencial.

1. Requisitos de codificação

**[RNF10] - Linguagem para codificação**

A linguagem utilizada para a codificação será C#.

**Prioridade:** Essencial.

**[RNF11] - Sistema Gerenciador de Banco de Dados (SGDB)**

O sistema *web* operará juntamente a dois SGDB’s, sendo eles: SQL Server 8.0 e Oracle 11G\*.

\*O Oracle será utilizado, porém será implementado pela própria empresa Fênix.

**Prioridade:** Essencial.

**[RNF12] - Plataforma de desenvolvimento**

O desenvolvimento do sistema *web* deve-se utilizar a plataforma de desenvolvimento Visual Studio 2017.

**Prioridade:** Essencial.

c) Requisitos de padrões

**[RNF13] - Modelagem arquitetural de projeto**

Deve ser seguido o padrão arquitetural *Model-View-Controller* (MVC) para a criação do sistema *web,* utilizando o *Framework* Asp Net.

**Prioridade:** Essencial.

**[RNF14] – Modelagem de dados**

A modelagem de dados será feita utilizando-se o modelo Entidade-Relacionamento, elaborado por meio das ferramentas *Visual Paradigm* e brModelo.

**Prioridade:** Essencial.

**[RNF15] - Linguagem de modelagem unificada**

A modelagem de análise e de projeto seguirá a Linguagem de Modelagem Unificada (UML), utilizando a ferramenta *Visual Paradigm*.

**Prioridade:** Essencial.

**[RNF16] - Documentação**

Os documentos entregues seguirão as diretrizes da FAI – Centro de Ensino Superior em Gestão, Tecnologia e Educação.

**Prioridade:** Essencial.

5.1.2.3 Requisitos Externos

Esta seção apresenta todos os RNF externos, que são derivados de fatores externos ao sistema e ao processo de desenvolvimento do software tais como: Interoperabilidade, legal (privacidade e segurança) e ético.

1. Requisitos de interoperabilidade

**[RNF17] - Sistema Operacional**

Deve ser utilizado o sistema operacional *Windows Server* 2008 ou superior.

**Prioridade**: Importante

**[RNF18] - Servidor *Web***

A empresa Fênix possui um servidor *web* interno IIS 7.5, no qual irá atender ao requisito de disponibilidade do sistema.

**Prioridade:** Importante.

1. Requisitos éticos

**[RNF19]** **Termo de autorização**

O termo de autorização deve ser assinado pelo representante e Supervisor de Tecnologia da Informação da Fênix Thiago Moraes, no qual assume o compromisso de autorizar e fornecer as informações sobre a rotina da empresa Fênix e que são necessárias para realização do projeto final de curso (PFC).

**Prioridade:** Essencial.

**[RNF20] Privacidade de dados**

O sistema *web* garantirá a privacidade e segurança dos dados armazenados nos bancos de dados, de modo que não permitirá aos usuários terem acessos aos dados sigilosos sem autorização.

O controle de privacidade ocorrerá via atribuição de permissões que serão configuradas de acordo com o perfil do usuário.

**Prioridade:** Essencial.

## 5.2 VISÃO FUNCIONAL

A visão funcional é elaborada de acordo com o levantamento dos requisitos funcionais do projeto. Portanto, sua apresentação é indispensável, pois permite organizar o relacionamento entre os papéis (atores) do sistema e os casos de uso.

O caso de uso descreve um conjunto de sequências de ações que o sistema desempenhará para produzir um resultado esperado pelo usuário. Cada sequência representa a interação de entidades externas e o sistema. Estas entidades são chamadas de atoresque podem ser usuários ou outros sistemas.

### 5.2.1 Diagramas de Casos de Uso

À medida que os requisitos são levantados, uma visão geral das funções e características começa a se materializar. Entretanto, é difícil progredir para atividades de engenharia de software mais técnicas até que entendamos como tais funções e características serão usadas por diferentes classes de usuários. Para tanto, os desenvolvedores e usuários podem criar um conjunto de cenários que identifique um roteiro de uso para o sistema a ser construído. Os cenários normalmente chamados de casos de uso, fornecem uma descrição de como o sistema será utilizado. (PRESSMAN, 2011, p. 136).

Nesta seção, estão definidos os casos de uso do projeto Extranet Fênix – Desenvolvimento de uma extranet para a empresa Fênix.

Um diagrama de Caso de Uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário com o objetivo de auxiliar a comunicação entre os desenvolvedores e o cliente, de forma que este veja no diagrama de Casos de Uso as principais funcionalidades do sistema adquirido.

A Figura 7 apresenta o diagrama de casos uso Visão Geral (ou de contexto) do Extranet Fênix.

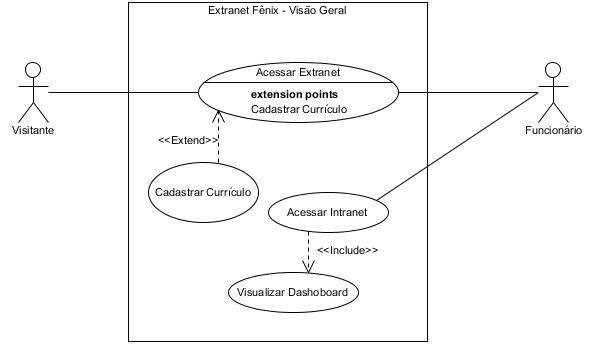


FIGURA 7 - Diagrama de caso uso Visão Geral

A Figura 8 apresenta o diagrama de caso uso Módulo Controle de Acesso.

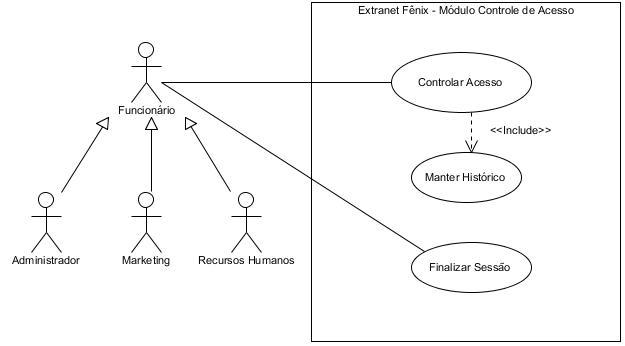


FIGURA 8 - Diagrama de caso uso Módulo Controle de Acesso

A Figura 9 apresenta o diagrama de caso uso Módulo Administrador.

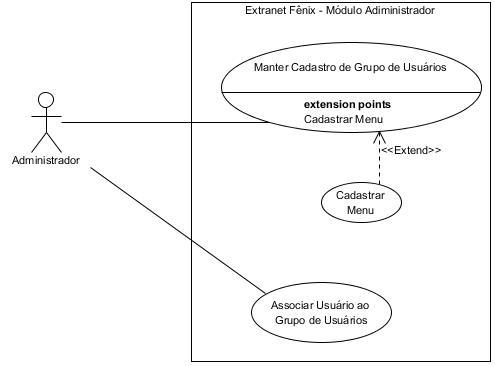


FIGURA 9 **-** Diagrama de caso uso Módulo Administrador

A Figura 10 apresenta o diagrama de caso uso Módulo Marketing.

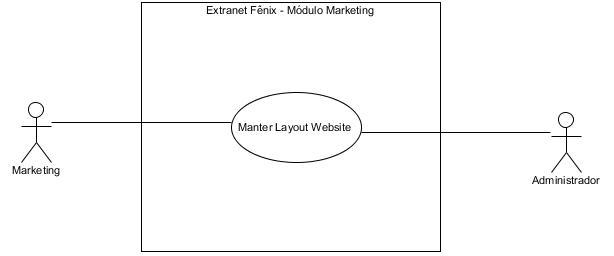


FIGURA 10 -Diagrama de caso uso Módulo Marketing

A Figura 11 apresenta o diagrama de caso uso Módulo Recursos Humanos.

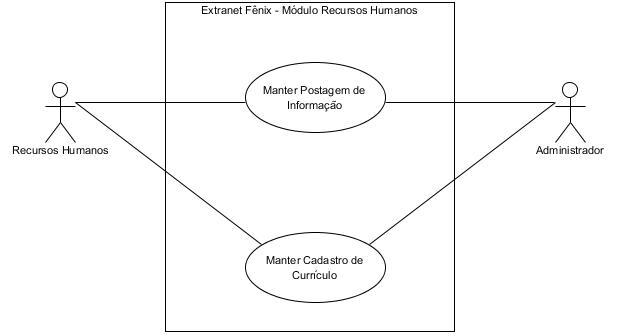


FIGURA 11 - Diagrama de caso uso Módulo Recursos Humanos

### 5.2.2 Descrição dos Casos de Uso

A descrição de um caso de uso apresenta informações sobre formas possíveis de utilização deste caso de uso, chamadas cenários. Cada cenário de um caso de uso apresenta informações como pré-condição para que o caso de uso seja executado, pós-condição após a finalização do cenário, atores envolvidos, fluxo de eventos entre os atores e o sistema, etc.

Os cenários de caso de uso deste projeto podem ser consultados no Apêndice G.

## 5.3 VISÃO DE DADOS

Nesta seção é apresentado o projeto lógico do sistema.

### 5.3.1 Projeto lógico

O Modelo Entidade e Relacionamento (MER) é uma abstração esquemática que apresenta as entidades do mundo real sobre as quais o sistema deverá armazenar dados, bem como o relacionamento entre estas entidades.

A Figura 12 apresenta o Diagrama Entidade e Relacionamento deste sistema.

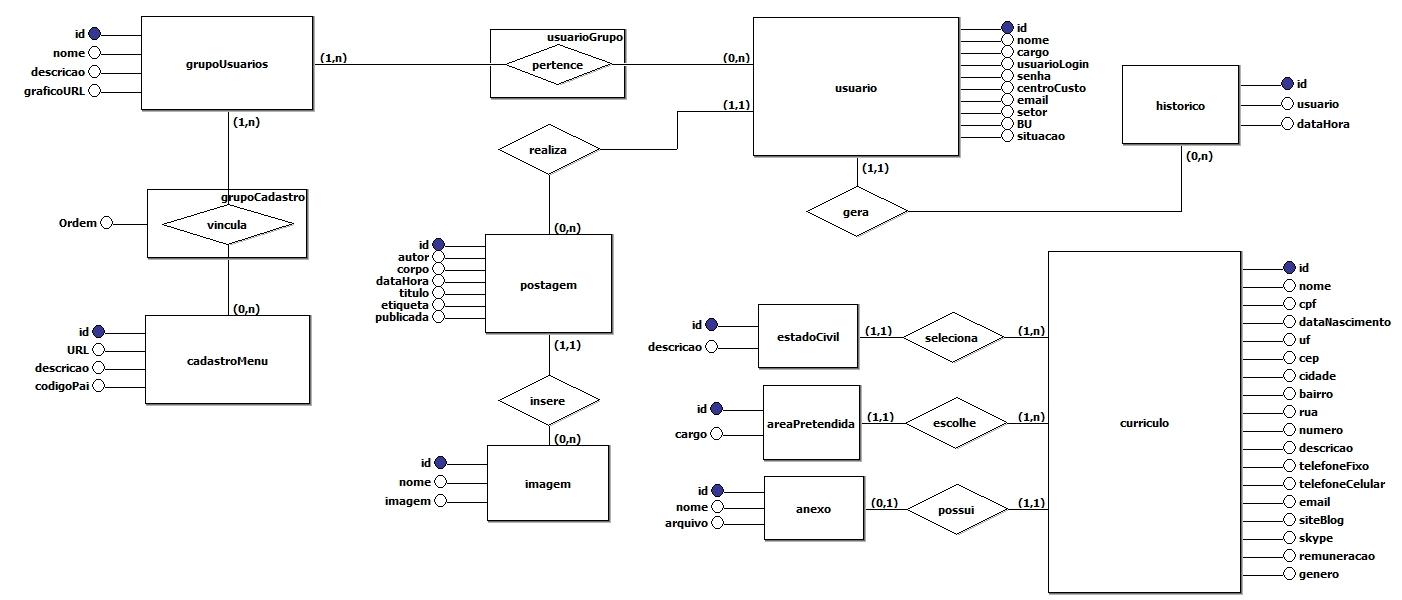


FIGURA 12 - Diagrama Entidade-Relacionamento

# 6 ARQUITETURA E PROJETO DO SISTEMA

A arquitetura do software determina a sua estrutura, expõe os componentes com suas propriedades e os relacionamentos entre eles. A elaboração de uma documentação da arquitetura facilita a comunicação entre os *stakeholders* e o reuso de componentes em futuros projetos.

Projeto Arquitetural representa a estrutura dos componentes e programas que são necessários para construir um sistema baseado em computador. Ele considera o estilo arquitetural que o sistema vai adotar, a estrutura e as propriedades dos componentes que constituem o sistema e os inter-relacionamentos que ocorrem entre todos os componentes arquiteturais de um sistema. (PRESSMAN, p.357, 2002).

Neste capítulo são apresentadas a arquitetura e projeto do sistema Extranet Fênix, por meio das visões estrutural, comportamental, de dados e física. Neste capítulo também se encontram os padrões de projeto, uma análise de complexidade, projeto de sistemas distribuídos e o projeto da interação-humano computador.

## 6.1 VISÃO ESTRUTURAL

Para modelar as estruturas que irão compor o sistema, amparando-se nos conceitos de Orientação a Objetos, utilizam-se os diferentes diagramas que integram *a Unified Modeling Language* (UML). Os diagramas estruturais da UML servem para visualizar, especificar, construir e documentar os sistemas de *software*, permitindo uma melhor abstração da arquitetura e seus relacionamentos.

Os diagramas estruturais comtemplados neste documento são: diagrama de pacotes, diagrama de classes e diagrama de objetos.

### 6.1.1 Diagrama de Pacotes

O diagrama de pacotes é um modelo que descreve os pacotes idealizados para o sistema.

Agrupados logicamente, estes mostram como os elementos são organizados dentro dos pacotes e as dependências entre eles, conforme exibido na Figura 13.

O padrão de arquitetura *Model-View-Controller* (MVC) é uma forma de dividir a aplicação, criando separações do código em três camadas, descritas como *Controller* (Controle) que identifica as entradas do mouse e teclado e mapeia as solicitações do usuário. O *Model* (Modelo) que gerencia os elementos de dados e responde as solicitações do *Controller.* E por fim, o *View* (Visão) que organiza a área de visualização e apresentação das informações para o usuário.

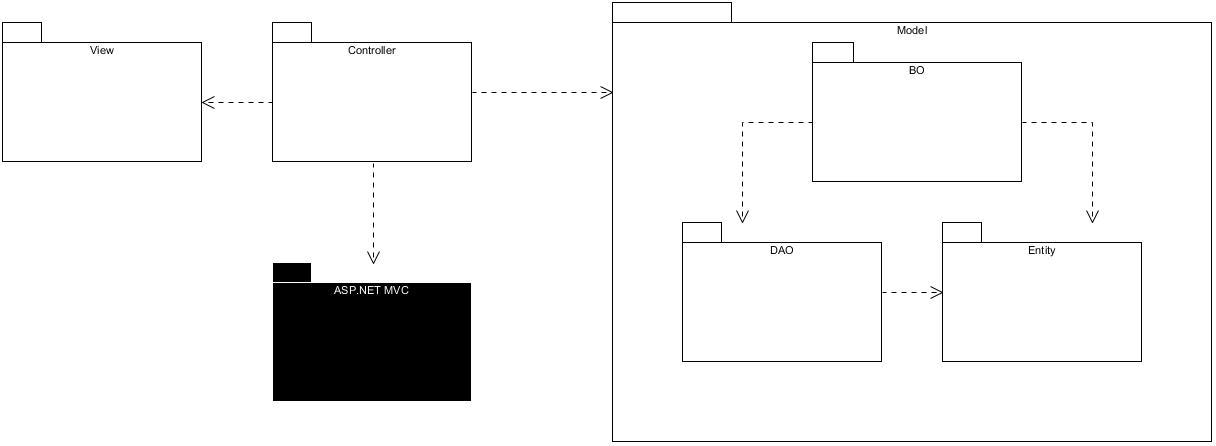


FIGURA 13 - Diagrama de Pacotes

### 6.1.2 Diagramas de Classes

O diagrama de classes representa os objetos que o sistema irá manipular, os métodos e operações que serão aplicados aos objetos para efetuar a manipulação, os relacionamentos entre os objetos e as colaborações que ocorrem entre as classes definidas (PRESSMAN,2011).

As figuras 14, 15 e 16 apresentam os diagramas de classe construídos de acordo com os pacotes identificados na Seção 6.1.1

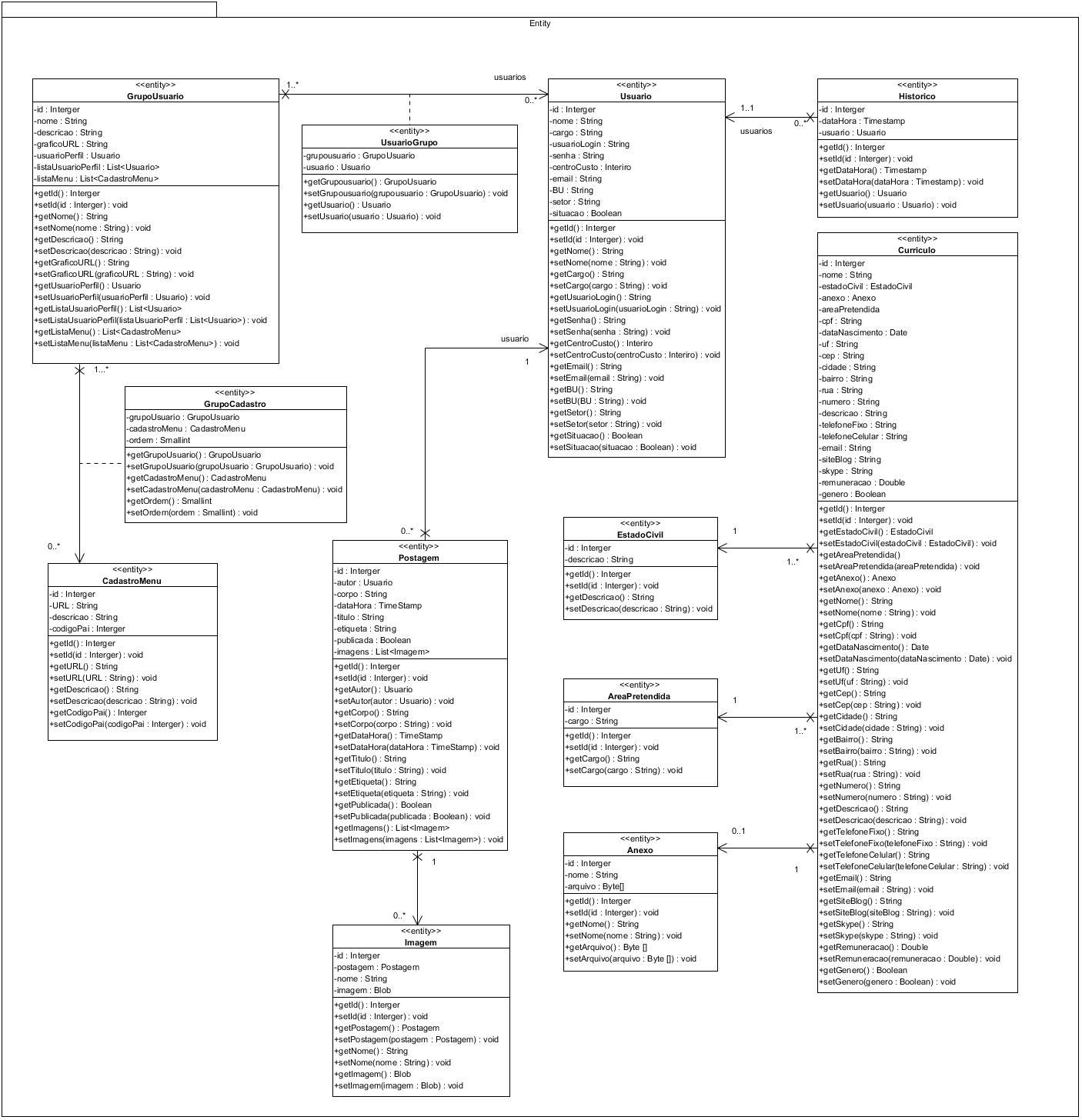


FIGURA 14 - Diagrama de Classe Entity

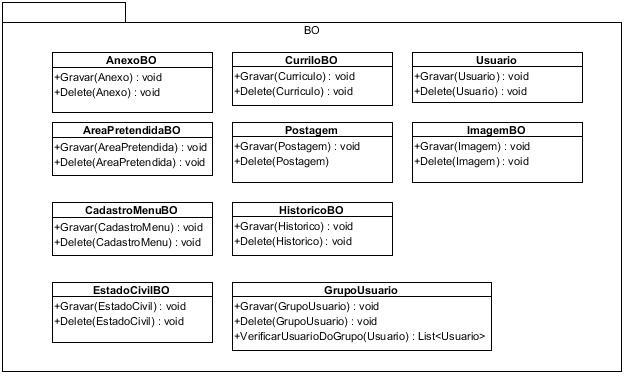


FIGURA 15 - Diagrama de Classe BO

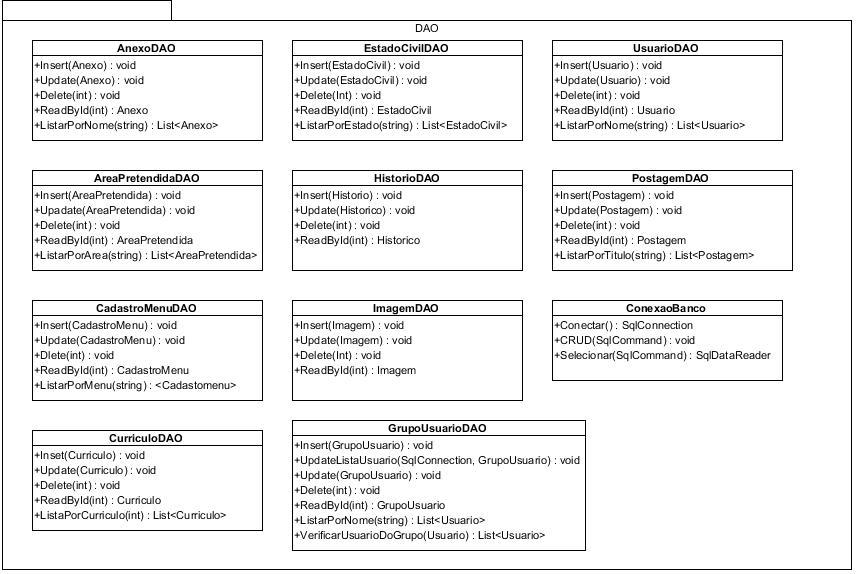


FIGURA 16- Diagrama de Classe DAO

### 6.1.3 Diagrama de Objetos

De acordo com Medeiros (2004, p.212), “O diagrama de objetos é uma instância do diagrama de classes. Cada classe mostra seu objeto em um determinado ponto de tempo”.

O digrama de objeto é usado para determinar os relacionamentos entre as classes e colaborar para a identificação de possíveis problemas que poderão acontecer com o sistema em funcionamento. A Figura 17 apresenta o diagrama de objeto do cenário criar do caso de uso Manter Cadastro grupo de Usuário.

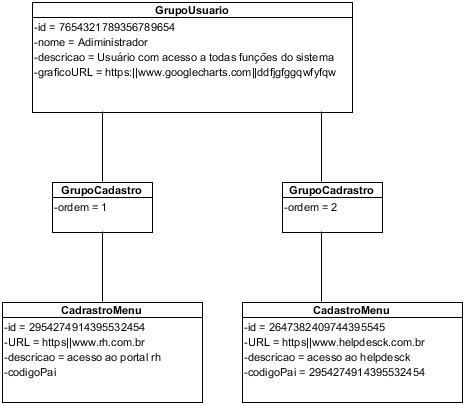


FIGURA 17- Diagrama de Objeto do cenário criar do Caso de Uso Manter Cadastro Grupo de Usuários

## 6.2 VISÃO COMPORTAMENTAL

Segundo Pressman (2011, p.177), a visão comportamental é a representação do comportamento do sistema em função de tempos e eventos específicos. O modelo comportamental indica como o software responderá a eventos ou estímulos externos.

Nesta seção são abordados os seguintes diagramas comportamentais: diagrama de sequência e diagrama de visão geral de interação (sendo esses três primeiros os que compõem o projeto das interações), diagrama de atividades e por fim o diagrama de máquina de estados.

### 6.2.1 Projeto das Interações

O projeto das interações é considerado um subgrupo dos diagramas comportamentais, sendo utilizado na representação

6.2.1.1 Diagramas de Sequência

Segundo Pressman (2011, p.179), o diagrama de sequência representa o comportamento pela descrição de como os objetos se movem de um estado para o outro. É um tipo de representação de como os eventos causam fluxo de um objeto para outro como função do tempo. É uma versão abreviada do caso de uso, representa classes-chave e os eventos que fazem o comportamento fluir de classe para classe.

A Figura 18 apresenta a comunicação real entre os objetos do sistema durante sua execução.

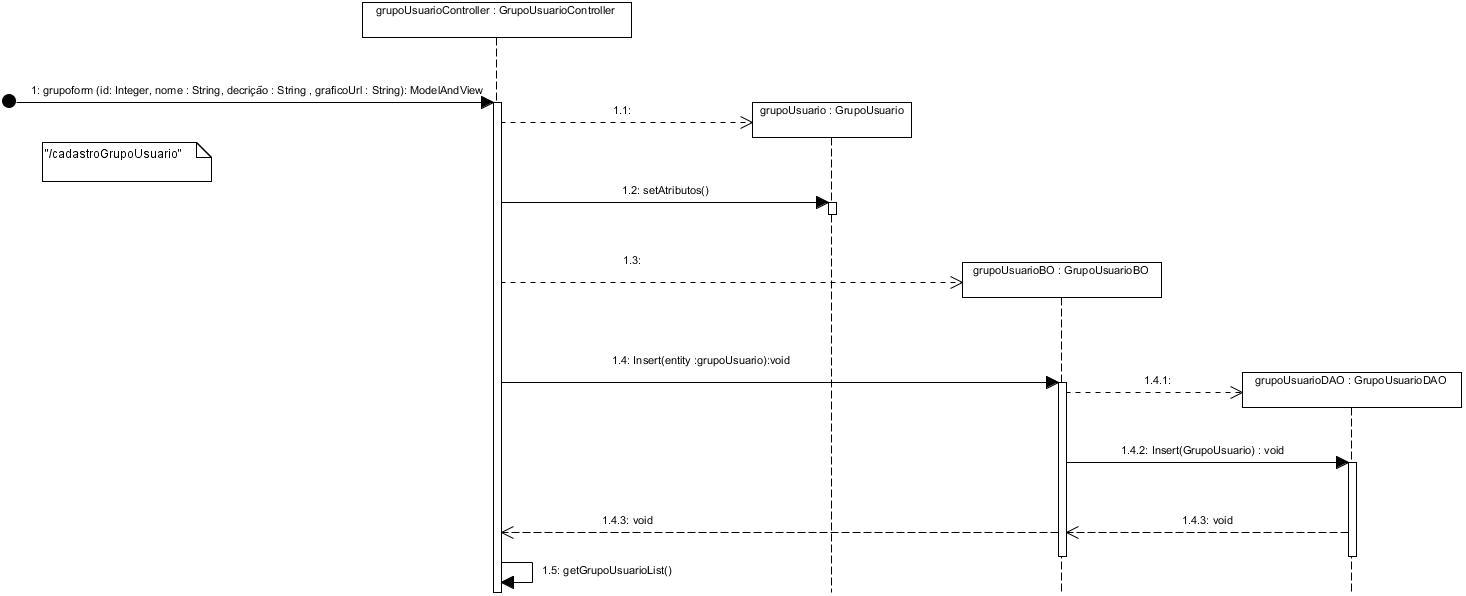
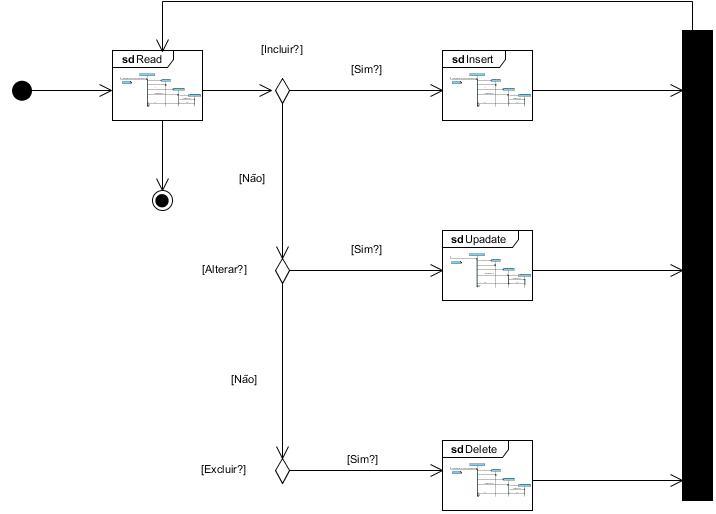


FIGURA 18 – Diagrama de sequência do cenário criar do Caso de Uso Manter Cadastro Grupo de Usuários.

6.2.1.2 Diagrama de Visão Geral de Interação

É uma variação do diagrama de atividades que mostra de uma forma geral o fluxo de controle dentro do sistema ou processo de negócios. Cada nó ou atividade dentro do diagrama pode representar outro diagrama de interação entre objetos. A Figura 19 apresenta o Diagrama de Visão Geral de Interação do sistema do Caso de Uso Mante Cadastro Grupo de Usuários.



**FIGURA 19- Diagrama de Visão geral de Interação do caso de uso Manter Cadastro Grupo Usuários**

### 6.2.2 Diagrama de Atividades

Os diagramas de atividades representam o fluxo de controle de uma atividade para outra em etapas sequenciais. A Figura 20 apresenta o diagrama de atividades do sistema do caso de uso manter cadastro grupo de usuários.

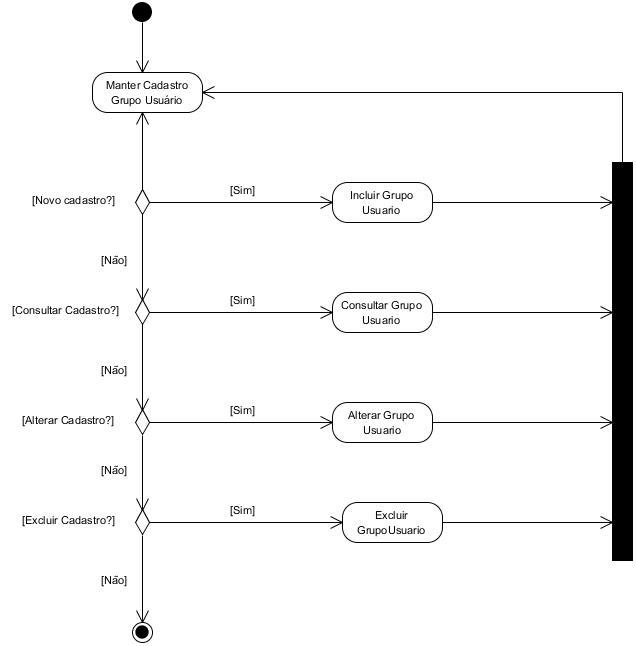


FIGURA 20 – Diagrama de Atividades do caso de uso Manter Cadastro Grupo Usuários

### 6.2.3 Diagrama de Máquina de Estados

Os diagramas de máquina de estados são utilizados para demonstrar os estados pelo qual passa a instância de uma classe durante a execução de um caso de uso. A Figura 21 apresenta o diagrama de máquina de estados do caso de uso manter cadastro de grupo de usuários.

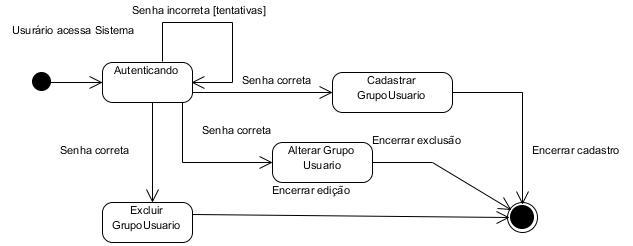


FIGURA 21 – Diagrama de Máquina de Estados do Caso de Uso Manter Cadastro Grupo de Usuários

## 6.3 VISÃO DE DADOS

Nesta seção é apresentado o modelo operacional do projeto Extranet Fênix, seu dicionário de dados e o processo utilizado para a elaboração do projeto físico.

### 6.3.1 Modelo Operacional

Segundo Martins (2007), o modelo lógico é criado através do modelo conceitual. Este modelo é um modelo intermediário entre o nível conceitual e o nível físico. Ao contrário dos modelos conceituais, os modelos lógicos são os modelos em que os objetos, suas características e relacionamentos têm sua representação de acordo com as regras de implementação e limitantes impostos por alguma tecnologia, modelo esse utilizado já na fase de projeto, mais independente de dispositivo físico, implementado conceitos como chave primaria, normalização, integridade referencial, cave composta e outros.

A Figura 22 apresenta o modelo operacional do projeto Extranet Fênix.

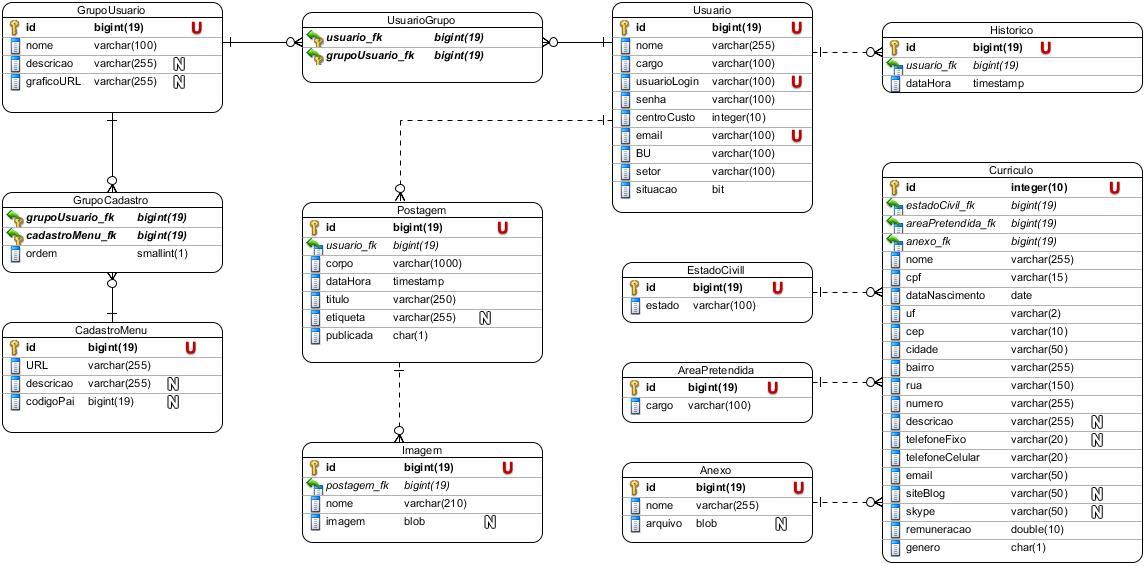


FIGURA 22 – Modelo Operacional

### 6.3.2 Dicionário de Dados

Dicionário de dados uma lista organizada que contém todos os elementos de dados pertinentes ás relações apresentadas no modelo lógico.

O dicionário de dados do projeto Extranet Fênix encontra-se disponível no apêndice H.

### 6.3.3 Processo de Elaboração do Projeto Físico

A elaboração do projeto físico consiste primeiramente no projeto conceitual, no qual é denominado modelo entidade relacionamento (MER), ou seja, o [modelo de dados](https://pt.wikipedia.org/wiki/Modelo_de_dados) e sua descrição, no qual gerou o projeto lógico da aplicação. Após essa fase, inicia-se a construção do banco de dados no qual foi utilizado o SGDB SQL Server, onde são definidos os detalhes técnicos, como: permissões de acesso, scripts de criação e armazenamento de dados.

## 6.4 VISÃO FÍSICA

[Insira uma apresentação sobre o assunto tratado nesta seção.]

### 6.4.1 Diagrama de Componentes

[Todos os componentes do sistema de software deverão ser identificados. Recomenda-se construir um diagrama para componentes logicamente agrupáveis, como componentes fontes, componentes de execução, banco de dados e XML].

## 6.5 PADRÕES DE PROJETO

[Insira uma apresentação sobre o assunto tratado nesta seção.]

### 6.5.1 Design Patterns

[Cite, descreva e exemplifique os design patterns utilizados na codificação, podendo ser eles de propósitos criacionais, estruturais, comportamentais, de concorrência ou arquiteturais].

|  |  |  |
| --- | --- | --- |
| Propósito | Padrões utilizados | Classes e/ou métodos aplicados |
| Arquitetural |  |  |
| Criacional |  |  |
| Comportamental |  |  |
| Concorrência |  |  |
| Estrutural |  |  |

QUADRO 8 - Apresentação dos design patterns utilizados

### 6.5.2 Convenções para Codificação

[Inclua as convenções adotadas para codificação (nomes de classes, objetos, métodos, comentários, entre outros).]

## 6.6 ANÁLISE DE COMPLEXIDADE

[ A análise da complexidade deve ser feita para pelo menos três dos principais métodos de classes do projeto e mostrar qual é a sua complexidade final. Especificar o que significa o parâmetro "n" que aparece nos estudos de complexidade. Exemplo:

Seja “n” o número de bimestres. As instruções dentro do laço "for" no método GeraMediaFinal da classe “Disciplina” são executadas 4 vezes. Este loop só contém comandos de leitura e atribuição que são O(1). Logo, a complexidade deste método é O(1), constante, pois as instruções dentro do loop são executadas um número fixo de vezes.]

## 6.7 PROJETO DE SISTEMAS DISTRIBUÍDOS

Nesta seção são apresentados os desafios de Sistemas Distribuídos encontrados durante o desenvolvimento do projeto Extranet Fênix.

“Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente. ” (TANENBAUM, 2007, p. 1)

### 6.7.1 Procedimentos para Tratamento dos Desafios

Nesta seção são apresentados os desafios de sistemas distribuídos encontrados pelo projeto Extranet Fênix. É importante ressaltar que o desenvolvimento de um sistema distribuído é fundamental para uma arquitetura de qualidade do projeto. Sendo assim, ao projetar esta arquitetura são encontrados alguns desafios: heterogeneidade, escalabilidade, abertura, segurança, manuseio de falhas, concorrência e transparência. Também são descritos os procedimentos a serem adotados para tratar cada desafio.

6.7.1.1 Heterogeneidade

A heterogeneidade (isto é, variedade e diferença) é a característica que determina que um sistema distribuído pode possuir diferentes recursos em diferentes níveis que não são uniformes entre si.

Entre esses recursos podem estar incluídos diferentes tipos de:

1. redes
2. hardware (diferentes representações de dados, diferente código máquina)
3. sistemas operacionais (diferentes interfaces para os protocolos de comunicação)
4. linguagens de programação (diferentes representações de estruturas de dados como *arrays* ou registos)
5. implementações de diversos programadores

O desafio de heterogeneidade da aplicação Extranet Fênix é tratado parcialmente por meio do CLR (Common Language Runtime), o sistema faz uso do framework .NET que permite o desenvolvimento em qualquer linguagem de programação que for suportada pelas ferramentas da Microsoft, dentre as que mais se destacam são: C# e Visual Basic, ASP.NET, Jscript .NET, além de linguagens como Managed Extensions for C++, F#, IronRuby, VB.NET , J# , Cobol e IronPhyton.

A princípio a ideia é bem semelhante ao Java onde temos a JVM – Java Virtual Machine, a qual executa byte code, que é o código gerado pelas aplicações Java.

Conforme demonstrado na Figura 23 o código C# é compilado para CLR e para uma linguagem intermediária MSIL (IL), que é semelhante ao *bytecode* do Java, esse código intermediário é passado para o compilador JIT (Just in Time), o compilador JIT compila o IL para código de máquina totalmente nativa, esse código nativo é guardado na memória para a execução do programa.

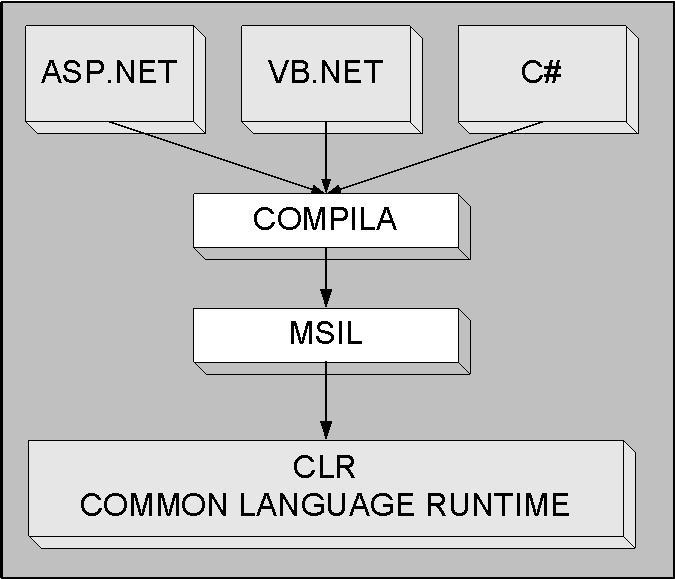


FIGURA 23 – O Ambiente de Execução do CLR

FONTE: Battisti (2016)

Este funcionamento possui uma enorme diferença se compararmos com uma máquina virtual, pois o código totalmente compilado é executado pela CPU diretamente, em vez de interpretado ou traduzido por uma camada de abstração de software adicional, como exemplo *Middleware*.

6.7.1.2 Escalabilidade

A escalabilidade é uma das mais importantes metas de projeto para desenvolvedores de sistemas distribuídos (SD). Um SD é um sistema aberto. Isto significa que está sujeito a modificações ao longo do tempo. Desta forma, a qualquer momento novas recursos podem ser incorporadas ao sistema, assim como outros recursos podem deixar de existir. O mesmo vale para usuários do sistema e suas requisições. Toda essa dinamicidade deve ser considerada ao implementar um SD. Um sistema é considerado escalável se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários.

No projeto Extranet Fênix, não é previsto um número alto de usuários, sendo que não é um sistema de trabalho e sim um sistema de apoio, ou seja, não é utilizado a todo tempo. E com isso não haverá sobrecarga de processamento no sistema.

“Um sistema é descrito como escalável se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários. ” (COULOURIS; DOLLIMORE; KINDBERG, 2001, p.31).

Feita uma análise de complexidade foi identificado nível baixo. Sendo assim, este desafio não representa um grande problema, devido haver um servidor (externo) da própria empresa Fênix, que disponibiliza os recursos de escalabilidade, visto que os números de requisições são suportáveis e de forma satisfatória.

6.7.1.3 Abertura

Abertura é a característica de um sistema distribuído que determina a capacidade de o sistema ser extensível. Podendo acontecer por meio do acréscimo de novos componentes de *hardware* ou *software*. Novos componentes devem poder ser adicionados sem prejudicar o funcionamento dos já existentes.

O desafio de abertura não deve ser contemplado pelo Projeto Extranet Fênix, pois não será desenvolvida nenhuma API *Application Programming Interface*, para acesso a funcionalidades do sistema web.

6.7.1.4 Segurança

Esta certamente é uma das propriedades que mais causa preocupações aos usuários de sistemas distribuídos. Em sistemas distribuídos a importância das informações se mostram evidentes por ser justamente o centro e foco da computação. Nesse contexto, ao modelar e implementar um sistema distribuído, a existência de questões relacionadas à segurança tornam-se mais nítidas e consideravelmente indispensáveis para o bom funcionamento e sucesso do sistema.

Manter recursos computacionais seguros significa:

1. manter o nível de confidencialidade exigido pelos utilizadores (proteção contra acessos não autorizados)
2. garantir a integridade dos dados (proteção contra alteração ou corrupção de dados ou programas)
3. manter a disponibilidade do sistema (proteção contra interferências com os meios de acesso aos recursos)

O sistema ao manter dados no banco de dados deva garantir a confidencialidade e proteção de tais dados protegendo-os de ações maliciosas ou acesso indevido.

O mecanismo adotado para a segurança autenticidade são: controle de acesso com autenticação de usuário e senha.

6.7.1.5 Manuseio de falhas

Tolerar uma falha significa conter os seus efeitos de forma a que o sistema continue a funcionar. Os sistemas distribuídos estão sujeitos a falhas (como qualquer outra entidade computacional), elas podem ocorrer em qualquer um dos vários componentes do sistema, como na rede, no servidor, no cliente, no *software*, no *hardware*, entre outros. Geralmente as falhas em um sistema distribuído são parciais (alguns componentes falham, enquanto outros continuam a funcionar). Quando uma falha for identificada, ela deve ser reportada ao usuário e tratada, a fim de manter o sistema coerente e confiável.

“Às vezes, os sistemas de computador falham. Quando ocorrem falhas no *hardware* ou no software, os programas podem produzir resultados incorretos ou podem parar antes de terem concluído a computação pretendida. ” (COULOURIS; DOLLIMORE; KINDBERG, 2001, p.32).

No projeto Extranet Fênix o manejo de falhas será executado através da técnica de isolamento de transação oferecida pelo banco de dados SQL Server. Ao obter uma mensagem de erro o sistema irá interromper a transação corrente e efetuar outra vez toda a transação do início, com início nesse ponto a transação passa a enxergar a alteração efetivada anteriormente como parte da sua visão inicial do banco de dados e, portanto, não existirá conflito lógico em usar a nova versão da linha como ponto de partida para atualização na nova transação.

6.7.1.6 Concorrência

Concorrência em sistemas distribuídos é a característica que determina que um componente do sistema possa ter seus recursos compartilhados, onde outros componentes ou processos concorrem entre si para a utilização desses recursos. Por existirem diversos componentes em um sistema distribuído, é possível que múltiplas requisições ou múltiplos acessos sejam realizados ao mesmo recurso, no mesmo instante de tempo. A implementação do sistema deve garantir que todas os acessos/requisições sejam respondidos (mesmo que a resposta seja negativa).

“Qualquer objeto que represente um recurso compartilhado em sistema distribuído deve ser responsável por garantir que ele opere corretamente em um ambiente concorrente. ” (COULOURIS; DOLLIMORE; KINDBERG, 2007, p.34).

Neste projeto, possui o tratamento de conexões com o SGDB para evitar problemas relacionados à concorrência de seus recursos. O sistema deva permitir o acesso de seus usuários e processos aos seus recursos compartilhados simultaneamente, sem que haja inconsistência de dados. Para que isso aconteça será utilizado o padrão *Singleton*, onde para cada requisição de acesso ao banco de dados é instanciada uma conexão dentre o pool de conexões disponíveis.

6.7.1.7 Transparência

Principal objetivo de um sistema distribuído é acessar recursos ao longo do sistema, como se fossem locais. Para que isto ocorra, o sistema deve executar processamentos externos, utilizar recursos compartilhados, efetuar trocas de dados entre componentes ou concorrer aos processamentos com diversas estações sem que o usuário perceba que isto esteja sendo feito. A concorrência tornou os sistemas distribuídos acessíveis a todos os tipos de usuários (dos avançados aos leigos) e deve ser preservada e almejada.

A transparência é definida como senso a ocultação, para um usuário final ou para um programador de aplicativos, da separação dos componentes em um sistema distribuído de modo que o sistema seja percebido como um todo, em vez de uma coleção de componentes independentes. A implicações da transparência têm grande influência sobre o projeto do software de sistema (COULOURIS; DOLLIMORE; KINDBERG, 2001, p.34).

O projeto deve aderir aos conceitos de transparência em todos os módulos, visto que a aplicação será utilizada por vários usuários ao mesmo tempo. O sistema deve efetuar essas operações sem intervir na utilização da aplicação pelo usuário.

O método mais empregue no projeto para implementação da transparência na aplicação é a execução de tarefas em segundo plano, que evitam espera por processamentos.

### 6.7.2 Tecnologias e Arquiteturas de Distribuição

O desenvolvimento de um sistema distribuído leva em conta a forma e a disposição com que os componentes se comunicam, bem como as tecnologias que são utilizadas para dar suporte ao processo de distribuição. A Figura 24 apresenta a visão de distribuição do sistema Extranet Fênix.

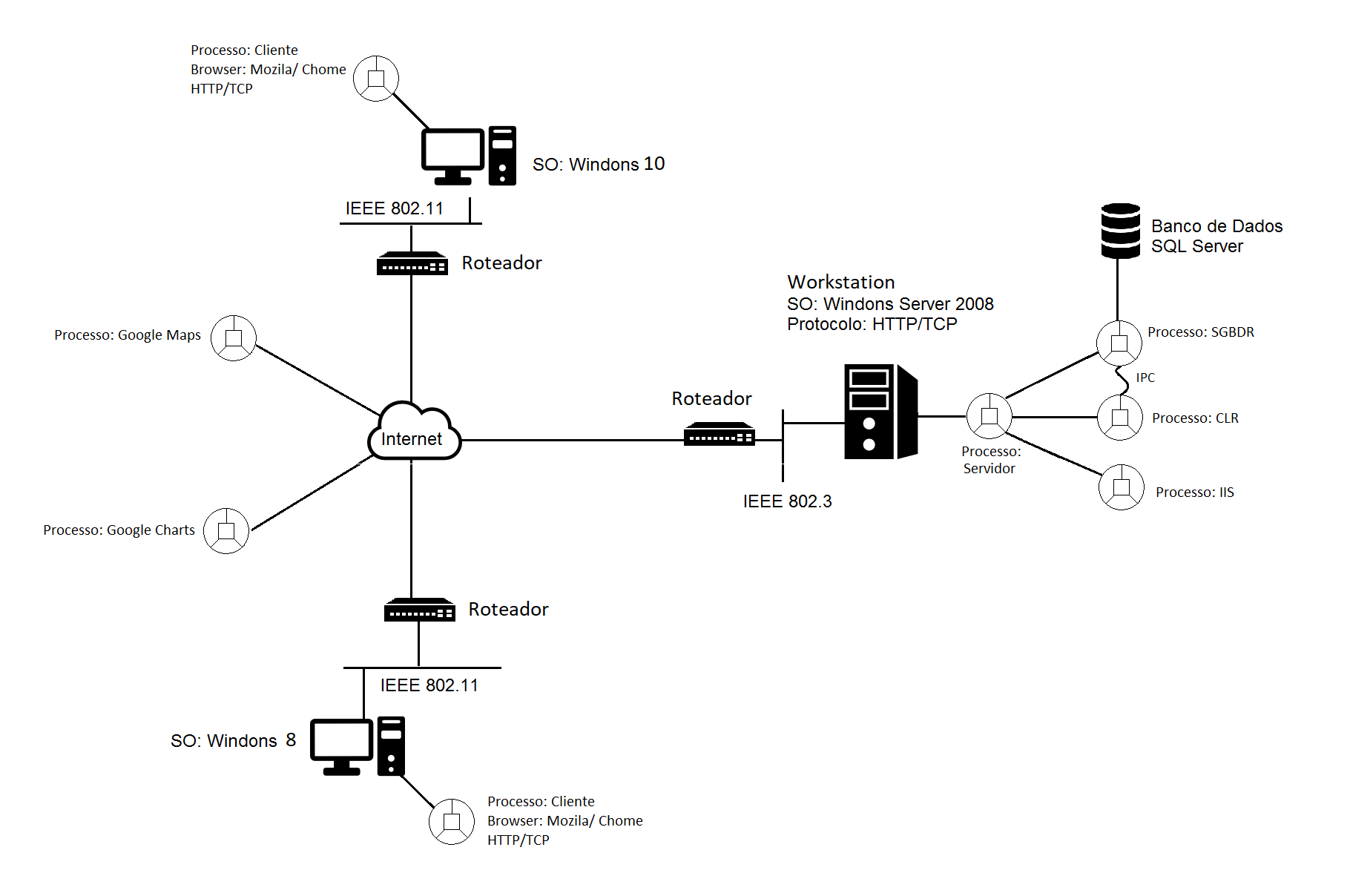


FIGURA 24 – Visão de Distribuição

FONTE: Elaborado pelos autores (2017)

As tecnologias de distribuição usadas no projeto Extranet Fênix são do Framework .NET. O ASP.NET MVC é um padrão de arquitetura leve e completo que provê a criação de aplicações Web baseadas no padrão [MVC](https://pt.wikipedia.org/wiki/MVC) (*Model, View, Controller*), de forma que haja uma maior independência entre as camadas. Dentro do padrão MVC, há três elementos principais: o *model*, responsável por representar as entidades da lógica de negócios da aplicação; a *view*, responsável por apresentar uma interface para o usuário; e o *controller*, que realiza o controle dos outros elementos, fornecendo uma ligação entre eles.

6.7.2.1 CLR

O CLR é um ambiente de execução, e poderíamos até dizer o coração do .NET, o qual dá suporte a todas as linguagens de programação habilitadas para o .NET . Ao instalarmos o Microsoft .NET SDK, temos disponíveis as linguagens: VB.NET (Visual basic .NET), C#, ASP.NET, Jscript .NET, entre outras.

As aplicações criadas em uma das linguagens para o .NET, ao serem compiladas, geram um código intermediário conhecido como MSIL – *Microsoft Intemediate* Language também conhecido como “*Managed Code*” ou código gerenciado, o qual é abreviado simplesmente como IL *– Intermediate language*, Este código que é executado pelo CLR.

A princípio a ideia é bem semelhante ao Java onde temos a JVM – *Java Virtual Machine*, a qual executa “*byte code”*, que é o código gerado pelas aplicações Java.

Conforme ilustrado na Figura 25, quando compilamos o código fonte criado por uma linguagem habilitada ao .NET estamos gerando MSIL, a qual é um conjunto de instruções para carregar, armazenar, inicializar e chamar métodos, instruções para operações aritméticas e lógicas, controle de fluxo do programa, acesso direto à memória, tratamento de exceções e demais operações necessárias para a execução do programa isto independente de CPU. Antes que o MSIL possa ser executado o mesmo precisa ser convertido para o código específico da CPU – código nativo, esta conversão é feita pelo JIT *– Just in Time compiler*. O CLR disponibiliza compiladores JIT para cada arquitetura suportada pelo CLR; desta maneira o mesmo conjunto de instruções MSIL pode ser compilado e executado em qualquer arquitetura para a qual exista um compilador JIT. Esta é a estratégia da Microsoft para fazer com que o código MSIL possa ser executado em diferentes plataformas, desde servidores Windows ou não-Windows, até dispositivos móveis para os quais esteja disponível um compilador JIT.



FIGURA 25 – O processo de execução de código CLR

Processo de execução do CLR

1. Cria a aplicação ou componente utilizando uma linguagem habilitada ao .NET
2. Compila o código da aplicação para gerar o código MSIL. O compilador da linguagem que você está utilizando compila o seu código-fonte para MSIL e acrescenta o metadados necessários.
3. Em tempo de execução, o compilador JIT (*Just in Time*) do CLR transforma o código MSIL em código nativo compilado.
4. O código nativo é executado, utilizando toda a infraestrutura disponibilizado pelo CLR.

Com isso o CLR oferece várias vantagens aos desenvolvedores das quais:

1. Fácil integração e interoperabilidade entre programas criados em diferentes linguagens;
2. Fácil implementação e controle da segurança da aplicação;
3. Utilizar a biblioteca de classes do framework .NET;
4. Melhor gerenciamento das diferentes versões de um mesmo componente, inclusive a possibilidade de execução simultânea de diferentes versões do mesmo componente, o que é conhecido por *syde-by-syde execution*;
5. Gerenciamento automático de alocação de objetos, bem como referência aos objetos e liberação de memória quando não há nenhuma referência ao objeto o que indica que este não está sendo mais usado o próprio CLR remove o objeto.

6.7.2.2 Common Type System

Tipo de dados comuns que podem ser utilizados por todas as linguagens habilitadas ao .NET é o facilitador da integração entre programas e serviços criados, utilizando-se de diferentes linguagens do .NET. Uma facilidade do CTS (*Common Type System*) é que não existe a necessidade do programador mapear os tipos de uma linguagem para os tipos correspondentes a outra linguagem, o que o torna mais simples uma vez que todas as linguagens têm acesso a um conjunto de tipos comuns a todas elas.

Principais funções do CTS:

1. Fornece uma estrutura que possibilita a integração entre diferentes linguagens habilitadas ao .NET, com uma execução mais rápida, uma vez que a sobrecarga para a conversão entre os diferentes tipos de diferentes linguagens deixa de existir.
2. Fornece uma estrutura de tipos com base em um modelo orientado a objetos, o que facilita a criação de novas linguagens habilitadas ao .NET, favorecendo a utilização de boas práticas de programação como exemplo a herança.
3. O CTS define algumas regras que toda linguagem deve seguir, para ser habilitada ao .NET, por seguirem um conjunto de regras comum, a interação entre programas escritos em diferentes linguagens fica bem mais fácil.

6.7.2.3 .NET Framework Class Library

Biblioteca de classes do framework .NET é uma coleção de classes ou tipos completamente integrados com o ambiente de execução CLR. São muitas as funções disponibilizadas pela biblioteca de classes do .Net conforme é apresentado na Figura 26.

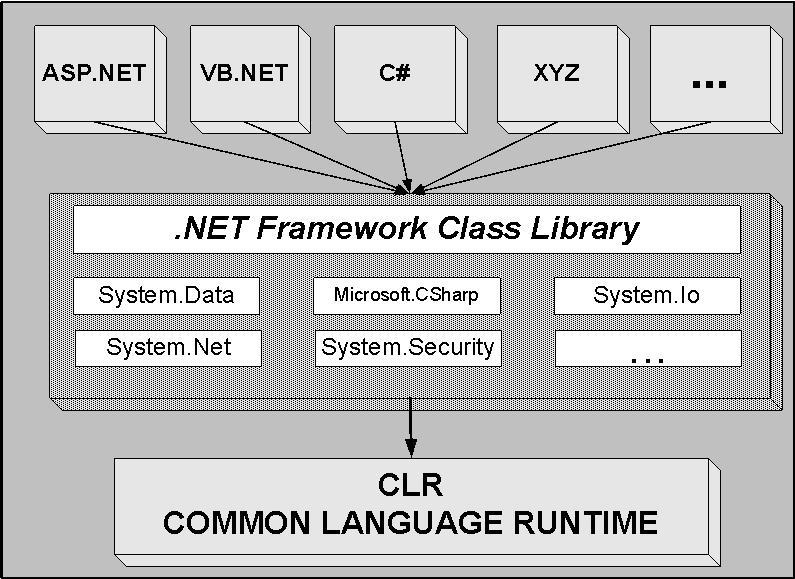


FIGURA 26 -Principais elementos do Framework .NET

6.7.2.4 WEB Services

Serviços que podem ser acessados via internet ou através de qualquer intranet, são chamados de Web Services.

A iniciativa .NET é a visão da Microsoft de um mundo onde o software se transforma em serviços, na verdade pequenos componentes que podem ser utilizados por qualquer aplicação. A comunicação é feita porque todo programa criado na plataforma .NET utiliza um padrão para troca de informações. Existem dois padrões:

1. Para o formato de dados, o framework .NET utiliza XML(*Extensive Markup Language).*
2. Como protocolo de transporte, o framework .NET utiliza o SOAP (*Single Object Access Protocol)*.Com SOAP podemos fazer com que toda comunicação entre diferentes Web Services e demais componentes de um programa seja feita através do protocolo padrão da internet que é o HTTP . O SOAP é um padrão da indústria e não criado pela Microsoft.

A grande vantagem dos Web Services, no framework .NET, é que os mesmos podem ser acessados, facilmente através da utilização de um protocolo padrão: SOAP e trocando informações em um formato padrão: XML. Esta abordagem torna a criação e utilização dos *web services* uma tarefa muito mais simples do que, por exemplo, a criação de componentes COM+ ou CORBA, os quais utilizam formatos de dados e protocolos de comunicação proprietários.

6.7.3.4 Metadados

São dados sobre dados, seriam as informações que o componente .NET possui a respeito de si mesmo, e que podem ser utilizadas por outros componentes e serviços, para acessar o componente em questão. Além de fazer com que não seja necessário o registro do componente, as informações de Metadata facilitam a interoperabilidade entre diferentes componentes, mesmo estes componentes escritos em diferentes linguagens. Estas informações são geradas, automaticamente no momento da compilação do componente e são gravadas no arquivo .DLL ou .EXE do componente.

Quando o componente é acessado, o CLR carrega os metadados do componente na memória e faz referência a estes metadados, para obter informações sobre as classes, membros, herança e dependências do componente.

O CLR utiliza metadados para localizar e carregar programas ou componentes, organizar as várias instâncias de um mesmo componente na memória, resolver a chamada de métodos, gerar código nativo, garantir a segurança de acesso e definir os limites para o contexto de execução de um componente ou programa.

6.7.2.5 Assemblies

É uma coleção de tipos e recursos que foram construídos para trabalharem juntos, formando, uma unidade com funcionalidade e escopos bem definidos. *Assembly* é um mecanismo utilizado pelo .NET para “empacotar” todos os elementos e informações necessárias ao funcionamento de uma aplicação ou componente, simplificando a distribuição de aplicações e resolvendo o problema de “versões” existentes em uma aplicação baseada em componentes. Composição 2 elementos básicos: manifesto e um conjunto de módulos.

Um *assembly* fornece ao CLR importantes informações sobre a implementação de tipos da aplicação. Para o CLR, um tipo somente existe no contexto de um *assembly*.

6.7.2.6 ADO.NET

*Activex Data Objects* ou ADO.NET é uma biblioteca de acesso a dados da Microsoft,

Que fornece as funcionalidades de acesso a dados, ele é quem fornece "o código" de acesso a dados, para isto ele fornece uma implementação dás interfaces de acesso a dados para cada um dos *providers* existentes, utilizando as seguintes bibliotecas:

1. *System.Data.SqlClient* - Fornece as funcionalidades de acesso a dados utilizando o *Provider SqlClient*
2. *System.Data.OleDb* - Fornece as funcionalidades de acesso a dados utilizando o *Provider OleDB*;
3. *System.Data.Odbc* - Fornece as funcionalidades de acesso a dados utilizando o *Provider Odbc*

ODBC é um *provider* de acesso a dados, que permite acesso a diversos banco de dados distintos. Fazendo uma simples analogia o ADO.NET determina o que será feito já o *provider*, como o ODBC, determinada como será feito, Sendo assim podemos dizer que é possível utilizar ODBC e ADO.NET ao mesmo tempo, Cada p*rovider* tem suas características especificas, o projeto Extranet Fênix utiliza o *provider* SqlClient que só pode ser usado para acesso a bancos de dados SQLServer, banco de dados utilizado pela aplicação que é mais eficiente quando comparado ao ODBC e o OLEDB que podem acessar diversos bancos de dados diferentes, diminuindo o desempenho da aplicação.

## 6.8 PROJETO DA INTERAÇÃO HUMANO-COMPUTADOR

[Apresente o estudo dos usuários, mostrar os padrões ergonômicos e as heurísticas de usabilidade para a implementação da interface de usuário.]

### 6.8.1 Perfil de Usuário

Os usuários que irão utilizar a extranet serão os próprios colaboradores da empresa Fênix. O primeiro perfil de usuário será o Administrador, ou seja, pessoa autorizada para utilização diária e com permissão de uso de todo sistema, o administrador restringe-se principalmente ao setor de Tecnologia da Informação. Já o perfil de Recursos Humanos (RH), realizará alteração de postagens da intranet e manutenção de currículos. O perfil do Marketing, será responsável pela definição de *design* da extranet. Esses três perfis são os principais na utilização do *website*, uma vez que, possuem permissões de definição ou mudança. Os demais perfis irão ter acesso a extranet e intranet, onde cada setor ou usuário terá suas permissões para acessos aos *links.*

No Apêndice I estão disponíveis os questionários de perfil do usuário da empresa Fênix.

### 6.8.2 Aspecto Visual da Interface de Usuário

[Apresente a padronização de cores, fontes, plano de fundo e ícones e demais elementos da interface de usuário.]

### 6.8.3 Heurísticas de Usabilidade

[Projeto da interface de usuário visando atender às heurísticas propostas por pesquisadores da área.]

# 7 PLANO DE TESTES

[Insira uma apresentação sobre o assunto tratado neste capítulo.]

## 7.1 FINALIDADE

[Defina qual é o tipo de plano de teste realizado, caso seja mais de um, repetir a mesma estrutura. Os testes podem ser unitários, de integração, de validação e/ou de sistema.]

## 7.2 ESCOPO

[Insira uma apresentação sobre o assunto tratado nesta seção.]

### 7.2.1 Referências a Documentos Relevantes

|  |  |
| --- | --- |
| Tipo do Material | Referência |
|  |  |

QUADRO 10 - Documentos relevantes para testes

### 7.2.2 Ambiente para a Realização dos Testes

|  |  |  |
| --- | --- | --- |
| Equipamento | Marca/Modelo/Configuração | Finalidade |
|  |  |  |

QUADRO 11 - Equipamentos a serem utilizados para a realização dos testes

|  |  |  |
| --- | --- | --- |
| Software/Versão | Fabricante | Finalidade |
|  |  |  |

QUADRO 12 - Softwares a serem utilizados para a realização dos testes

7.3 ESPECIFICAÇÃO DOS CASOS DE TESTES

[Insira uma apresentação sobre o assunto tratado nesta seção.]

### 7.3.1 Item a Testar

|  |  |
| --- | --- |
| Identificação do Item | Descrição |
|  |  |

QUADRO 13 - Identificação dos itens a serem testados

### 7.3.2 Rastreabilidade entre Requisitos e Casos de Teste

|  |  |
| --- | --- |
| Identificação do Requisito | Caso(s) de teste(s) aplicável (eis) |
|  |  |

QUADRO 14 - Requisitos e casos de testes

### 7.3.3 Descrição dos Casos de Testes

|  |  |  |
| --- | --- | --- |
| Item: | No. teste: | Nome do Teste: |
| Fluxo de Execução: | | |
| Entradas: | | |
| Saídas esperadas: | | |
| Dependências: [Condições prévias para a execução do caso de teste, inclusive outros casos de teste que devam ser executados obrigatoriamente antes deste.] | | |

QUADRO 15 - Descrição do caso de teste Xnnn

## 7.4 RESULTADOS DOS TESTES

[Insira uma apresentação sobre o assunto tratado nesta seção.]

### 7.4.1 Histórico de Realização

[Para cada caso de teste, preencha o quadro a seguir. Deve constar como Apêndice].

|  |  |  |
| --- | --- | --- |
| No. do teste: | Responsável pelo teste: | Data de realização: |
| Versão do artefato testado: | |  |
| Resultado observável: | |  |
| Situação: | |  |
| Impacto: [Evento indesejável que tenha ocorrido] | |  |

QUADRO 16 - Situação do caso de teste Xn

### 7.4.2 Resultados

[Nesta seção faça uma avaliação da eficácia do plano de testes. Indique o nível de qualidade do sistema, se há necessidade de testes adicionais ou há deficiência de alguns itens do Plano de testes. Indique a situação final do sistema, em relação aos testes previstos.]

# 8 PLANO DE IMPLANTAÇÃO

[Insira uma apresentação sobre o assunto tratado neste capítulo.]

## 8.1 METODOLOGIA

[Insira uma apresentação sobre o assunto tratado nesta seção.]

### 8.1.1 Descrição da Metodologia

[Determine como será a implantação e a sua sequência.]

### 8.1.2 Matriz de Responsabilidade

[O Quadro 11 apresenta as atividades preparatórias para a implantação e as responsabilidades

do cliente e dos fornecedores do sistema de software.]

|  |  |
| --- | --- |
| Atividades | Responsáveis |
| Planejamento |  |
| Definição da equipe de implantação |  |
| Levantamento de recursos necessários de hardware |  |
| Levantamento de recursos necessários de software |  |
| Definição de estratégias para conversão e migração de dados |  |
| Programação dos treinamentos |  |
| Preparação dos testes de aceitação |  |
| Execução |  |
| Configuração de infraestrutura de hardware |  |
| Configuração de infraestrutura de software |  |
| Configuração de infraestrutura de telecomunicações |  |
| Instalação do produto |  |
| Conversão e migração de base de dados |  |
| Treinamentos |  |
| Realização de testes de aceitação |  |
| Avaliação |  |
| Acompanhamento pós-implantação |  |
| Reunião final da implantação |  |

QUADRO 17 - Papéis e responsabilidades na implantação

## 8.2 TREINAMENTO PREVISTO

[O Quadro 15 detalha os treinamentos a serem ministrados para a capacitação dos usuários durante a fase de implantação do produto.]

|  |  |  |
| --- | --- | --- |
| Treinamento | Conteúdo | Grupo de Usuários |
|  |  |  |

QUADRO 18 - Treinamentos previstos

## 8.3 CRONOGRAMA DE IMPLANTAÇÃO

[No Quadro 16 são apresentadas as tarefas previstas durante a implantação, a duração em horas e o período de realização.]

|  |  |  |
| --- | --- | --- |
| Tarefas | Duração | Período |
|  |  |  |
| Tempo estimado total: |  |  |

QUADRO 19 - Cronograma de atividades da implantação

## 8.4 DOCUMENTOS DE APOIO À IMPLANTAÇÃO

[Os documentos oferecidos para apoiar o processo de implantação e posterior uso do produto são listados no Quadro 17.]

|  |  |
| --- | --- |
| Documento | Referência |
| Manual de Instalação e Configuração | Vide Apêndice xx |
| Manuais do usuário | Vide Apêndice xx |
| Outros | Vide Apêndice xx |

QUADRO 20 - Documentos de apoio à implantação

## 8.5 VISÃO DE IMPLANTAÇÃO

[Mostre os componentes que serão implantados numa instância do ambiente real. Utilize o diagrama de distribuição/implantação da UML.]

# 9 CONCLUSÃO

[Faça um relato das principais atividades realizadas ao longo das fases. Inclua os aspectos mais relevantes observados em cada fase. Apresente quais objetivos do projeto foram alcançados e quais requisitos foram atendidos. Apresente dados que comprovam o alcance desses. Discuta os possíveis pontos de melhoria do projeto e o que poderá ser feito no futuro para continuar o trabalho. Apresente onde o trabalho foi divulgado. Finalize com as lições aprendidas.]

# REFERÊNCIAS

ANDERY, P. R. P.; ARANTES, E. M. Revisitando as extranets de projeto: uma análise dos processos de projeto. **Revista Ambiente Construído**, Porto Alegre, v.15, n.4, p. 149 - 164, out. /dez. 2015.

BASTTISTI, Júlio; ASP.NET: Uma Revolução na Construção de Sites e Aplicações Web.

Disponível: < http://www.juliobattisti.com.br/artigos/aspnet/principal.asp >. Acesso em: 30 jun. 2017.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Distributed systems: concepts and desing**. 3. ed. Harlow: Addison-Wesley, 2001.

\_\_\_\_\_\_\_\_. Sistemas Distribuídos: Conceitos e projetos, 4. ed. Porto Alegre: Bookman Companhia Ed, 2007.

FERREIRA, Hugo Braga Tadeu; STOCK, Eduardo dos Santos; **O que seria a Indústria 4.0?.** Fundação Dom Cabral. 2016.

HELDMAN, K. **Gerência de projetos**: guia para o exame oficial do PMI. Rio de Janeiro: Elsevier, 2006.

MARTINS, José Carlos Cordeiro. Gerenciando Projetos de Desenvolvimento de Software. 4 ed. 2007.

MESEIROS, Ernani Sales de. **Desenvolvimento software com UML 2.0:** definitivo. São Paulo:Pearson Markron Books, 2004.

NITITHAMYONG, P.; SKIBNIEWSKI, M. Success Factors For the Implementation of Web Based Construction Project Management Systems. **Construction Innovation**, v. 11, n.1, p. 13-42, 2011.

PROJECT MANAGEMENT INSTITUTE. **PMBOK Guide**: a guide to the Project management body of knowledge. 3 ed. Project Management Institute, 2004.

PRESSMAN, Roger S. **Engenharia de Software:** uma abordagem profissional. 7 ed. Porto Alegre: AMGH, 2011.

RADECK, Fernando**. Configuração de políticas de segurança no Windows Server 2008: Active Directory**. Curitiba: Universidade Tecnológica Federal do Paraná, 2012. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/1497/1/CT\_GESER\_II\_2012\_07.pdf>. Acesso em: 10 fev. 2017.

RUIKAR, K.; ANUMBA, C.; CARILLO, P. End User Perspectives on Use of Project Extranets in construction organizations. **Engineering, Construction and Architectural Management**, v. 12, n. 3, p. 225-235, 2005.

SOMMERVILLE, Ian. **Engenharia de software**. 8. ed. São Paulo: Pearson Addison – Wesley. 2007.

\_\_\_\_\_\_\_\_. **Engenharia de software**. 6. ed. São Paulo: Addison Wesley, 2003.

SPÍNOLA, Rodrigo Oliveira. Introdução a Gerência de Projetos. **SQL magazine**, Rio de Janeiro, ano 7, 75ª ed. 2010.

TANENBAUM, Andrew S. **Distributed operating systems**. Englewood Cliffs: Prentice Hall, 1995.

TEIXEIRA, Thiciane Mary Carvalho; VALENTIM, Marta Lígia Pomim. **Estratégias para disseminação do conhecimento organizacional: O papel da arquitetura da informação**. Inf. Inf., Londrina, v. 17, n. 3, p. 165-180, set./dez. 2012.

TORRES, A. et al. **Diseño de una plataforma interoperable para un observatorio de hábitos y estilos de vida saludable**. Escuela de Ingeneria de Antioquia-Universidad CES, Colombia, Volume 9, Número 17.45-55, 2015.

TORRES, Gabriel. **Redes de computadores**: curso completo. Rio de Janeiro: Axel Books do Brasil Editora Ltda, 2001.

TRIBELSKY, G.; SACKS, R. Measuring Information Flow in Detailed Design of Construction Projects. **Research in Engineering Design**, v. 21, n. 3, p. 189-206, 2010.

VALERIANO, Dalton. **Moderno gerenciamento de projeto**. São Paulo: Prentice Hall, 2005.

WILKINSON, P. Features and Functionality of Construction Collaboration Technology. In: WILKINSON, P. Construction Collaboration Technologies. New York: Taylor & Francis, 2005.

# OBRAS CONSULTADAS

[Coloque as obras pesquisadas, mas que não foram referenciadas no documento. Também em ordem alfabética por autor.]

# GLOSSÁRIO

A

Artefato: é um dos vários tipos de subprodutos concretos produzidos durante o desenvolvimento de *software*.

Ator: representa um papel de que um ser humano, um dispositivo de *hardware* ou até outro sistema desempenha com o sistema.

B

BrModelo: ferramenta de modelagem de banco de dados.

C

Caso de Uso: técnica de especificação que descreve uma sequência de ações que o sistema deve realizar para produzir uma resposta para um ator.

D

Diagrama: representação visual estruturada e gráfica de informações.

E

Escopo: definição do trabalho necessário para concluir o projeto, serve como ponto de referência para determinar que o trabalho não está incluso no projeto.

F

G

H

I

Iteração: processo chamado na programação de repetição de uma ou mais ações.

Internet das coisas: Consiste na conexão em rede de objetos físicos, ambientes, veículos e máquinas por meio de dispositivos eletrônicos embarcados que permitem a coleta e troca de dados.

J

K

L

M

Microsoft Project: *software* de gestão de projetos.

N

O

P

Q

R

Requisito: consiste da definição documentada de uma propriedade ou comportamento que um produto ou serviço particular deve atender.

S

T

U

V

Visual Paradigma for UML: ferramenta para desenvolver aplicações utilizando modelagem UML.

W

X

Y

Z

# APÊNDICE A - ESTIMATIVAS POR PONTOS DE FUNÇÃO

O arquivo “Estimativa\_por\_Ponto\_de\_Funcao.xlsx” pode ser encontrado na pasta “APÊNDICE A - Estimativa por Ponto de Função”.

# APÊNDICE B – ESTIMATIVA POR PONTOS DE CASOS DE USO

O arquivo “Estimativa\_por\_Ponto\_de\_Casos\_de\_Uso.xlsx” pode ser encontrado na pasta “APÊNDICE B - Estimativa por Pontos de Casos de Uso”.

# APÊNDICE C – ESTRUTURA ANALÍTICA DO PROJETO

Os arquivos “PFC\_Estrutura\_Analitica\_do\_Projeto.xlm” e “PFC\_Estrutura\_Analitica\_do\_Projeto.jpg” podem ser encontrados na pasta “APÊNDICE C – Estrutura Analítica do Projeto”.

# APÊNDICE D – DICIONÁRIO EAP DO PROJETO

O arquivo “PFC\_DicionárioEAP\_2017\_ExtranetFenix.mpp” pode ser encontrado na pasta “APÊNDICE D – Dicionário EAP do Projeto”.

# APÊNDICE E – RELATÓRIO DE DESEMPENHO

O arquivo “Relatorio\_de\_Desempenho.xls” pode ser encontrado na pasta “APÊNDICE E - Relatório de Desempenho”.

# APÊNDICE F – PLANILHA GERÊNCIA DE RISCOS

O arquivo “PFC\_2017\_GerenciaRiscos\_ExtranetFenix.xlsm” pode ser encontrado na pasta “APÊNDICE F - Planilha de Gerência de Riscos”.

# APÊNDICE G – DESCRIÇÃO DOS CASOS DE USO

O arquivo “Descricao\_dos\_Casos\_de\_Uso.docx” pode ser encontrado na pasta “APÊNDICE G - Descrição dos Casos de Uso”.

# APÊNDICE G – DESCRIÇÃO DOS CASOS DE USO

O arquivo “Descricao\_dos\_Casos\_de\_Uso.docx” pode ser encontrado na pasta “APÊNDICE G - Descrição dos Casos de Uso”.

# APÊNDICE H – Dicionário de Dados

O arquivo “Dicionário\_de\_Dados.docx” pode ser encontrado na pasta “APÊNDICE H - Dicionário de Dados”.

# APÊNDICE I – Questionário de Perfil do Usuário

O arquivo “Questionário\_de\_Perfil\_do\_Usuário.docx” pode ser encontrado na pasta “APÊNDICE I - Questionário de Perfil do Usuário”.