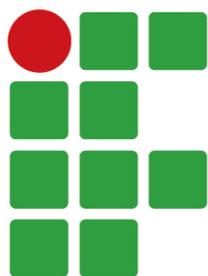


Instituto Federal de Educação, Ciência e Tecnologia de Goiás
Câmpus Formosa
Tecnologia em Análise e Desenvolvimento de Sistemas

**MyTask: Sistema Baseado na Web para
Geração Automatizada de Documentos Oficiais
Emitidos pelo Instituto Federal de Goiás.**



INSTITUTO FEDERAL
Goiás

Câmpus
Formosa

Autor: Juliana Alves Vieira
Orientador: (Me. Mário Teixeira Lemes)

Formosa, GO
2017

Juliana Alves Vieira

**MyTask: Sistema Baseado na Web para Geração
Automatizada de Documentos Oficiais Emitidos pelo
Instituto Federal de Goiás.**

Monografia submetida ao curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, câmpus Formosa, como requisito parcial para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Câmpus Formosa

Orientador: (Me. Mário Teixeira Lemes)

Formosa, GO

2017

Juliana Alves Vieira

MyTask: Sistema Baseado na Web para Geração Automatizada de Documentos
Oficiais Emitidos pelo Instituto Federal de Goiás./ Juliana Alves Vieira. – Formosa,
GO, 2017-

68 p. : il. (algumas color.) ; 30 cm.

Orientador: (Me. Mário Teixeira Lemes)

Trabalho de Conclusão de Curso – Instituto Federal de Educação, Ciência e Tec-
nologia de Goiás
Câmpus Formosa , 2017.

1. padronização. 2. documentos. I. (Me. Mário Teixeira Lemes). II. Instituto
Federal de Educação, Ciência e Tecnologia de Goiás. III. Câmpus Formosa. IV.
MyTask: Sistema Baseado na Web para Geração Automatizada de Documentos
Oficiais Emitidos pelo Instituto Federal de Goiás.

CDU 02:141:005.6

Juliana Alves Vieira

MyTask: Sistema Baseado na Web para Geração Automatizada de Documentos Oficiais Emitidos pelo Instituto Federal de Goiás.

Monografia submetida ao curso de graduação em Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, câmpus Formosa, como requisito parcial para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Trabalho aprovado. Formosa, GO, 17 de agosto de 2017:

(Me. Mário Teixeira Lemes)
Orientador

Esp. Alisson da Silveira Garcez
Convidado 1

Me. Leandro Rodrigues da Silva Souza
Convidado 2

Formosa, GO
2017

Dedico este trabalho primeiramente ao Senhor meu Deus, meu guia e autor da minha vida, auxiliador nas horas de angústia e solidão e amigo nas horas das dificuldades, a minha mãe Luzinete, ao meu pai Antonio e aos meus irmãos Gessica e Marcos Vinícius.

Agradecimentos

Agradeço a todos os amigos que me auxiliaram nessa longa jornada de aprendizagem. Em especial a minha família que acreditou no meu sonho, e não mediram esforços para me ajudar. Pois sei que a pessoa que me torno é reflexo das pessoas que convivo.

Agradeço aos meus professores que me incentivaram, me ajudaram e que de alguma forma não me deixaram desistir do meu sonho. Contiunuo agradecendo a toda a equipe administrativa do instituto que sempre me trataram com respeito e carinho, e que me acolheram como uma família. Em especial a psicóloga Fernanda, que com sua análise colaborou em minha orientação educacional. Agradeço ainda, minha orientadora de estágio Viviane Bueno, com seu grande desprendimento em ajudar e pela amizade sincera.

Ainda, gostaria de agradecer a todos os meus colegas de curso que durante esses anos contribuíram com seus conselhos e análises nessa jornada. Em especial agradeço às amigas Josilene Fernandes, Joyce Braga e Alice Lima pelo apoio e longas conversas juntas.

Finalmente, agradeço ao meu orientador Mário Teixeira Lemes, por acreditar na minha capacidade. E que nos momentos de dificuldades me aconselhou e não me deixou desistir. Agradeço por sua empatia e amizade, pois sem esse apoio não teria conseguido. Agradeço ainda, pelo incentivo e grande ajuda com o fornecimento de material para a realização deste trabalho. .

“Estamos na situação de uma criancinha que entra em uma imensa biblioteca, repleta de livros em muitas línguas. A criança sabe que alguém deve ter escrito aqueles livros, mas não sabe como. Não compreende as línguas em que foram escritos. Tem uma pálida suspeita de que a disposição dos livros obedece a uma ordem misteriosa, mas não sabe qual ela é”.

(Albert Einstein)

Resumo

Documentos de comunicação oficial são instrumentos presentes nas instituições humanas desde a antiguidade. Sua redação tem como principais características o uso culto da linguagem e a padronização na formatação. Este artigo tem o objetivo de apresentar o estado da arte relativo às formas de comunicação oficial, ferramentas de geração automatizada de documentos oficiais e apresentar uma proposta de aplicação, denominada *MyTask*, que tem como funcionalidades a geração, formatação e padronização de memorandos e ofícios. A geração automatizada destes documentos permite que os usuários do sistema concentrem esforços apenas no conteúdo dos mesmos, eliminando a necessidade de cuidados adicionais em sua padronização e formatação. Esta filosofia também é conhecida como a WYSIWYM (*What You See Is What You Mean*), utilizada em outras ferramentas, como o LaTeX. O *software* proposto viabilizará a produção padronizada de documentos oficiais de comunicação emitidos pelo Instituto Federal de Goiás. A pesquisa foi realizada com foco no diagnóstico dos procedimentos de padronização de documentos oficiais de uma instituição de ensino pública, a fim de compreender suas rotinas administrativas e garantir a geração de documentos claros, objetivos e padronizados.

Palavras-chaves: automatização. documentos oficiais. padronização.

Abstract

Official communication documents are instruments present in human institutions since antiquity. Its main features are the use of cultured language and standardization in formatting. This article aims to present the state of the art regarding official forms of communication, tools for automated generation of official documents and present an application proposal, called MyTask, whose functions are the generation, formatting and standardization of memos and trades. The automated generation of these documents allows system users to focus only on their content, eliminating the need for additional care in their standardization and formatting. This philosophy is also known as WYSIWYM, which is used in other tools such as LaTeX. The proposed software will enable the standardized production of official communication documents issued by the Federal Institute of Goiás. The research was carried out focusing on the diagnosis of the standardization procedures of official documents of a public educational institution, in order to understand its administrative routines and ensure the generation of clear, objective and standardized documents.

Key-words: automation. official documents;. standardization.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – <i>Processo de interação baseado no modelo cliente-servidor.</i> | 33 |
| Figura 2 – <i>Taxonomia dos diagramas UML.</i> | 35 |
| Figura 3 – <i>Diagrama de atividades.</i> | 40 |
| Figura 4 – <i>Diagrama de Máquina de Estado.</i> | 41 |
| Figura 5 – <i>Processo de levantamento e análise de requisitos.</i> | 42 |
| Figura 6 – <i>Diagrama de Caso de Uso.</i> | 45 |
| Figura 7 – <i>Diagrama de Classe.</i> | 46 |
| Figura 8 – <i>Página de autenticação do sistema MyTask.</i> | 52 |
| Figura 9 – <i>Página de login do sistema Mytask.</i> | 53 |
| Figura 10 – <i>Página de escolha de produção de documento oficial (Memorando ou Ofício).</i> | 54 |
| Figura 11 – <i>Página de inserção de conteúdo do documento oficial.</i> | 55 |
| Figura 12 – <i>Página de localização do documento oficial.</i> | 56 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 – Requisitos Funcionais do sistema <i>MyTask</i> | 43 |
| Tabela 2 – Requisitos Não-Funcionais do sistema <i>MyTask</i> | 43 |
| Tabela 3 – Especificação do Documento de Requisitos | 44 |

Lista de abreviaturas e siglas

| | |
|---------|--|
| ACID | Atomicidade, Consistência, Isolamento e Durabilidade |
| HTML | <i>HyperText Markup Language</i> |
| HTTP | <i>HyperText Transfer Protocol</i> |
| IDE | <i>Integrated Development Environment</i> |
| IIS | <i>Internet Information Services</i> |
| JSP | <i>JavaServer Pages</i> |
| JSS | <i>JasperSoft Studio</i> |
| LDAP | <i>Lightweight Directory Access Protocol</i> |
| PDF | <i>Portable Document Format</i> |
| SGBD | Sistema de Gerenciamento de Banco de Dados |
| SQL | <i>Structured Query Language</i> |
| UML | <i>Unified Modeling Language</i> |
| UP | <i>Unified Process</i> |
| WWW | <i>World Wide Web</i> |
| WYSIWYM | <i>What You See Is What You Mean</i> |

Sumário

| | | |
|------------|--|-----------|
| | Introdução | 23 |
| I | CONCEITOS PRELIMINARES | 29 |
| 1 | REFERENCIAL TEÓRICO | 31 |
| 1.1 | Introdução | 31 |
| 1.2 | Sistema e Modelo | 31 |
| 1.3 | Sistemas Computacionais | 32 |
| 1.3.1 | Sistemas <i>Stand-alone</i> | 32 |
| 1.3.2 | Sistemas <i>Web</i> | 32 |
| 1.3.3 | Linguagem de Programação | 33 |
| 1.4 | Engenharia de <i>Software</i> | 33 |
| 1.4.1 | Metodologias de Desenvolvimento de <i>Software</i> | 34 |
| 1.4.2 | Processo Unificado | 34 |
| 1.4.3 | UML | 35 |
| 1.4.4 | <i>Framework</i> | 35 |
| 1.5 | Banco de Dados | 36 |
| 1.5.1 | Modelo Relacional | 36 |
| 1.5.2 | SGBD | 36 |
| 1.6 | Considerações Finais | 37 |
| 2 | DESENVOLVIMENTO DO PROJETO | 39 |
| 2.1 | Introdução | 39 |
| 2.2 | Descrição do sistema | 39 |
| 2.3 | Diagrama de Atividade | 39 |
| 2.4 | Diagrama de Máquina de Estados | 40 |
| 2.5 | Levantamento e Análise de requisitos | 41 |
| 2.5.1 | Requisitos Funcionais | 42 |
| 2.5.2 | Requisitos Não-Funcionais | 43 |
| 2.5.3 | Documento de Requisitos | 43 |
| 2.6 | Diagrama de Caso de Uso | 44 |
| 2.7 | Diagrama de Classes | 46 |
| 2.8 | Codificação | 46 |
| 2.9 | Considerações Finais | 47 |

| | | |
|-----------|--|-----------|
| II | RESULTADOS | 49 |
| 3 | ESTUDO DE CASO | 51 |
| 3.1 | Introdução | 51 |
| 3.2 | Estrutura do <i>software</i> | 51 |
| 3.3 | Tecnologia | 51 |
| 3.4 | Funcionalidades | 52 |
| 3.4.1 | Autenticação no sistema | 52 |
| 3.4.2 | Tipo de documento | 53 |
| 3.4.3 | Formatação do documento | 53 |
| 3.4.4 | Localização do documento | 54 |
| 4 | CONCLUSÃO | 57 |
| 4.1 | Sugestões para Trabalhos Futuros | 57 |
| 4.2 | Produções científicas (Aguardando decisão editorial) | 57 |
| | REFERÊNCIAS | 59 |
| | APÊNDICES | 65 |
| | APÊNDICE A – DIAGRAMA DE CLASSES | 67 |

Introdução

A comunicação através de documentos oficiais é caracterizada pela predominância de formalidades. Comunicações oficiais devem seguir preceitos de impessoalidade, clareza, concisão e uso do padrão culto da linguagem. O uso destes preceitos permite que a comunicação seja transparente e livre de erros de interpretação. Neste contexto, despontam questões como a escolha adequada de pronomes de tratamento e a adoção de formatação de texto padronizada. A utilização de formalidade na escrita do documento oficial garante enfoque ao assunto do qual trata a comunicação (BRASIL, 2002).

Um dos primeiros registros de comunicação oficial remete ao ano de 490 *a.C.* De acordo com a lenda da *Batalha de Maratona*, soldados atenienses marcharam até a Planície de *Marathonas* para combaterem os persas. Os atenienses, apesar da previsão negativa, venceram a guerra. Então o general Milcíades escolheu Fidípides, um de seus mensageiros, para levar a notícia à cidade de Atenas. Fidípides correu quarenta e dois quilômetros do campo de batalha em *Marathonas* até Atenas, para transmitir a mensagem sobre a vitória. Após anunciar: “*Alegra-te, somos vitoriosos!*”, entrou em colapso e morreu de exaustão (LUCAS, 1976).

O primeiro documento oficial da História Brasileira, redigido pelo escrivão Pero Vaz de Caminha em 1500, foi a carta intitulada “*Carta a el-rei D. Manuel sobre o descobrimento da terra nova*”. Este documento possui considerável valor literário e também o relato mais confiável sobre a chegada dos portugueses ao Brasil. A carta descreve ao rei Dom Manuel as primeiras impressões de Caminha sobre a terra descoberta (MARTINS, 2005).

No século XVIII, as cartas redigidas pela rainha Carlota Joaquina, mulher de Dom João VI, revelam sua determinação em defender projetos políticos. A leitura de suas cartas oficiais, enviadas à aristocracia colonial, permite compreender o processo de desmoronamento do Antigo Regime na Espanha (AZEVEDO, 2007).

O Brasil saiu da condição de colônia de Portugal no século XIX, quando foi consolidado o modelo Imperial de governo. O Imperador Dom Pedro I enfrentou complicações políticas advindas tanto dos interesses dos brasileiros quanto dos portugueses, o que resultou em sua abdicação. O documento oficial de abdicação ao trono, escrito de próprio punho, findava o primeiro reinado no Brasil (ARMITAGE, 2011).

No final do século XIX instaura-se no Brasil a Primeira República. Este período constitui a primeira fase da organização republicana nacional, marcada pelo domínio político das elites agrárias mineiras, paulistas e cariocas. Marechal Deodoro da Fonseca foi o mentor do golpe armado pelos republicanos contra o governo imperial de Dom

Pedro II. O primeiro documento oficial assinado por Marechal Deodoro, já na condição de presidente do Brasil, comunicava ao imperador a sua destituição e a mudança do regime político (GOMES, 2013).

De 1964 a 1985 o Brasil teve seu poder político controlado pelo Comando das Forças Armadas, período denominado Ditadura Militar. A Constituição de 1988 suplantou os vestígios do Regime Militar e estabeleceu princípios democráticos. No entanto, diversos documentos oficiais da época, revelam atrocidades cometidas pelas autoridades militares durante a ditadura no Brasil. Esta documentação composta por relatórios, memorandos, dossiês, fotos e vídeos pode ser acessada em meios digitais e impressos (MEZAROBBA, 2010; STAMPA, 2011).

Marcada pelo caráter detalhista em que estabelece direitos sociais e regula a ordem econômica, após o Regime Militar, a nova República é estabelecida (KOUTZII et al., 1986). Em 1991, o presidente Fernando Collor assina decreto que autoriza a normatização da redação dos atos oficiais, resultando na primeira edição do Manual de Redação da Presidência da República (MENDES; JÚNIOR, 1991).

O uso de sistemas informatizados torna-se cada vez mais recorrente nas décadas de 1980 e 1990, alterando de maneira significativa os modos de comunicação acadêmicos, empresariais e pessoais (TANENBAUM; AUSTIN, 2012). No contexto da produção de documentos oficiais, isto implicou a adoção de meios digitais para sua produção e divulgação, face à clássica redação manual em papel.

Para Furuta, Scofield e Shaw (1982), pesquisadores da Universidade de Washington nos Estados Unidos, sistemas de formatação de documentos são viáveis para especificar, manipular e visualizar a aparência de documentos. Todavia, os primeiros formadores de texto puro surgiram na década de 1960 e endereçavam basicamente linhas e páginas do documento (FILHO, 2000).

Brown (1989), da Universidade de Kent na Inglaterra, aborda em trabalho publicado no periódico *The Computer Journal* as vantagens em manter a estrutura lógica do documento em todas as fases de edição, armazenamento e transmissão. Segundo o autor, isto permite desenvolver sistemas de processamento de texto ágeis e interoperáveis.

Na década seguinte, Appelt e Tetteh-Lartey (1993), pesquisadores europeus, propõem em trabalho apresentado também no periódico *The Computer Journal*, que estruturas de informação podem ser associadas à técnicas de descrição, para a criação de padrões de sistemas de processamento de documentos.

McKeown, Kukich e Shaw (1994) em trabalho aprovado na Quarta Conferência de Processamento de Linguagem Natural Aplicada em Stuttgart, Alemanha, abordam sistemas geradores de documentação automática de sistemas de Engenharia. A proposta apresentada sugere utilização sistemática de conjunções e combinações de frases previa-

mente fornecidas para geração de documentos concisos e fluentes. Por outro lado, [Belz \(2008\)](#) pesquisadora da Universidade de Brighton considera que trabalhar com regras de geração automática constitui tarefa complexa de combinação. Assim, para tornar a tarefa de produção automatizada de texto exequível é necessária eliminação da variação mediante aplicação de regras probabilísticas, o que nem sempre produz resultados com semântica correta.

De acordo com [Braga e Staa \(1995\)](#), a geração automatizada de documentos em forma de hipertexto constitui ferramenta de apoio ao desenvolvimento de *software*. Desta forma, a modelagem da estrutura lógica do *software* através de ferramentas auxilia a construção de diferentes formas de visualização do mesmo.

[Ferrari \(2006\)](#) em sua Tese de doutoramento afirma que sistemas de formatação de documentos devem prover vantagens como a eliminação de falhas de padronização, economia de tempo e o impedimento de erros humanos. Além disso, o autor propõe em seu trabalho que sistemas de *software* documentados através de técnicas de Engenharia de *Software* padronizadas, possuem manutenção simplificada.

[Hurst, Li e Marriott \(2009\)](#) da Universidade de Monash da Austrália, em trabalho aceito na 9º Simpósio de Engenharia de Documentos ocorrido em Munique, Alemanha, destacam o aumento do número de ferramentas de formatação automática disponíveis. Outra ideia apresentada neste trabalho é um processo de otimização e *layouts* automáticos de documentos eletrônicos.

A criação da *Web*, serviço de navegação em páginas de hipertexto da Internet, no início da década de 1980 modificou de maneira substancial a forma como documentos digitais são produzidos, armazenados, distribuídos e acessados. A Organização Europeia para a Investigação Nuclear foi a responsável pelo desenvolvimento do projeto *ENQUIRE* que deu origem às tecnologias empregadas na contemporânea *World Wide Web* (WWW) ([BERNERS-LEE et al., 2010](#)).

Para [Isotani et al. \(2009\)](#) do Instituto de Investigação Científica e Industrial, da Universidade de Osaka no Japão, a *Web* pode ser utilizada como ferramenta de suporte para desenvolvimento de ambientes informatizados. Uma das vantagens da utilização desta tecnologia é a possibilidade de criar mecanismos que permitam manipulação de informação em sistemas heterogêneos, propiciando a interoperabilidade entre diferentes sistemas operacionais. [Batista e Periotto \(2011\)](#) ressaltam que este novo paradigma tecnológico, contribui na construção de sistemas para a gestão de documentos oficiais, visando atender necessidades de produção e armazenamento deste tipo de documento.

Segundo [Pressman e Lowe \(2009\)](#), a *Web* tornou-se indispensável em diversas áreas do conhecimento humano. Desta maneira, a implantação de sistemas *Web* considera a necessidade da rápida disseminação de informação e extensa distribuição geográfica de

seus usuários, utilizando a Internet como ferramenta de suporte.

Santarosa, Conforto e Basso (2013), referem-se à *Web* como um conjunto de serviços que possibilitam formas de publicação, cooperação e organização da informação, constituindo novos espaços de interação humana.

De acordo com Gutiérrez (2014), do Departamento de Línguas e Sistemas de Informação, da Universidade de Sevilha, o desenvolvimento de sistemas *Web* é composto por um conjunto de componentes que constituem seu projeto. Para Seabra (2013), especialista em gerência de projetos, a adoção de metodologias de desenvolvimento de sistemas é a base para a consolidação de métodos para produção de um produto. Os conceitos tratados pela Engenharia de *Software* podem auxiliar na criação destes métodos, técnicas e ferramentas visando a eficiência do desenvolvimento do projeto.

Oliveira e Seabra (2015) defendem que estes métodos aliados à técnicas de desenvolvimento de sistemas *Web* constituem requisito fundamental para a otimização da utilização de recursos computacionais. Por outro lado, Sommerville (2015), afirma que não há processo ideal para o desenvolvimento de sistemas, e que para cada caso específico podem ser desenvolvidas diferentes metodologias, conforme a natureza do projeto.

O desenvolvimento tradicional pode não atender as singularidades envolvidas na construção de sistemas para *Web*. Neste contexto, surge a necessidade de uma nova abordagem para disciplinar e introduzir novos métodos e ferramentas para seu desenvolvimento, validação e implantação. Esta nova abordagem é definida como Engenharia *Web* (PRESSMAN; LOWE, 2009; PRESSMAN; MAXIM, 2014).

Diante do exposto, pode-se levantar a hipótese: se documentos de comunicação oficial possuem redação e apresentação normatizadas, então sua formatação pode ser realizada utilizando um sistema computacional. Logo, é possível desenvolver um sistema computacional para realizar a geração automatizada de documentos de comunicação oficial. Portanto, propõe-se desenvolver um sistema *Web* que possibilite simplificar a produção destes instrumentos de comunicação utilizando metodologias de análise e desenvolvimento de sistemas.

Sob o ponto de vista da produção científica nota-se que a quantidade de estudos sobre geração automatizada de documentos tem apresentado crescimento, sobretudo a partir da década de 2000. Com os dados obtidos em pesquisa realizada na *Web of Science* em novembro de 2016, utilizando os termos *automatic AND document AND generation* foi possível observar que as pesquisas nesta área são recentes e que o tema possui potencial para exploração, obtendo média de citações por item publicado de 4,17.

O objetivo geral deste trabalho é o desenvolvimento de um sistema *Web* que permita a seus usuários a criação de documentos de comunicação oficial padronizados conforme regras de formatação preestabelecidas. Ainda como objetivos tem-se: i) definir os

padrões de formatação de documentos oficiais a serem utilizados, ii) realizar projeto e implementação do sistema *Web* proposto e iii) implantar o sistema desenvolvido.

O trabalho encontra-se dividido em 5 capítulos. No Capítulo 1 é apresentada uma introdução do trabalho. No Capítulo 2 é apresentada revisão bibliográfica sobre conceitos, tecnologias e métodos empregados para o desenvolvimento deste trabalho de conclusão de curso. No Capítulo 3 é apresentada a metodologia de análise e desenvolvimento de sistemas adotada. No Capítulo 4 são apresentados os resultados obtidos acerca do desenvolvimento do sistema *MyTask*. Finalmente no Capítulo 5 é apresentado a conclusão e algumas recomendações de trabalhos futuros.

Parte I

Conceitos Preliminares

1 Referencial Teórico

1.1 Introdução

Neste capítulo são apresentadas definições gerais inerentes aos conceitos de sistema e modelo, além da caracterização específica de sistemas computacionais. São abordadas ainda as definições de Engenharia de *Software* e Banco de Dados.

1.2 Sistema e Modelo

Sistemas são conjuntos de unidades básicas (denominadas elementos, componentes ou partes) ordenadas que realizam interação mútua, podendo também interagir com outros sistemas (BERTALANFFY, 1969; RECHTIN; MAIER, 2010). Para Lemes (2012), este conceito está diretamente relacionado ao cotidiano humano, podendo ser utilizado em contextos como saúde, política, educação, engenharia ou esportes. A definição de sistema incorpora tanto sistemas artificiais quanto sistemas reais. Sistemas artificiais são baseados em definições e símbolos pertencentes ao pensamento. Sistemas reais são compostos por componentes materiais tangíveis (CHWIF; MEDINA, 2014).

A estrutura do sistema é determinada pelas conexões existentes entre suas unidades. Desta maneira, o comportamento que o sistema manifesta é consequência da interação entre seus elementos constituintes. O conceito de emergência é intrínseco à sistemas, pois diferentes comportamentos podem emergir de forma inesperada durante sua dinâmica de funcionamento. (CURY, 2001; CASSANDRAS; LAFORTUNE, 2009).

Um sistema pode ser representado através de modelos, que se distinguem pela ênfase em determinados aspectos ou pelo grau de abstração adotado em relação ao sistema real. Portanto, o modelo não deve ser uma representação simples a ponto de não fornecer detalhes suficientes sobre o sistema e nem complexa a ponto de possuir representação mais detalhada do que o próprio sistema real (COSTA; LIMA, 2005; PERROW, 2011; MONTGOMERY, 2004).

A construção de modelos permite a observação de características e a realização de experimentos que seriam inviáveis em termos de custo ou risco no sistema real. Assim, o modelo é capaz de representar o sistema real quando dado mesmo conjunto de entradas, fornece mesmo conjunto de saídas ou respostas do sistema que modela (MONTGOMERY, 2004; CHWIF; MEDINA, 2014).

1.3 Sistemas Computacionais

Sistemas constituídos por componentes de entrada, processamento, armazenamento e saída de informação digital são denominados sistemas computacionais. Estes sistemas funcionam através da execução de processos lógicos e de maneira automatizada, utilizando um computador eletrônico com plataforma (TANENBAUM; AUSTIN, 2012; STALLINGS, 2015).

Para realizar a representação computacional de sistemas reais é necessário realizar adaptações, visto que o computador não é capaz de representar estruturas da mesma maneira que elas se apresentam em um sistema real (CORMEN, 2009). A utilização do computador como principal meio de comunicação implicou a reconfiguração dos espaços de interação humana. Isto alterou de forma significativa relações econômicas, políticas e sociais (LOGAN, 2000).

1.3.1 Sistemas *Stand-alone*

No século XIX, Charles Babbage da Universidade de Cambridge apresentou o projeto da máquina analítica, precursora da definição de computador programável. Um programa é composto por uma sequência de instruções passível de ser executada por um computador. Os primeiros computadores executavam apenas uma tarefa por vez, limitação superada com a criação dos sistemas operacionais, que permitiram aumento na quantidade de aplicações processadas simultaneamente (MANZANO; OLIVEIRA, 2008; TANENBAUM; BOS, 2014).

Um programa que não depende de outras aplicações para operar é denominado *stand-alone* (TANENBAUM; BOS, 2014). Este tipo de programa não necessita da utilização de recursos de rede e geralmente proporciona operações locais mais rápidas. Porém, para Deitel e Deitel (2010), no contexto de computação colaborativa e distribuída, aplicações isoladas são praticamente inviáveis.

1.3.2 Sistemas *Web*

Sistemas *Web* são sistemas distribuídos que podem ser desenvolvidos e acessados através de tecnologias *Web* (W3C, 2016). Aplicações *Web* possuem características dirigidas ao fornecimento de conteúdo dinâmico e com prazo restrito de elaboração. Sistemas *Web* utilizam arquitetura cliente-servidor. Neste tipo de arquitetura, um cliente solicita tarefas específicas à um servidor que fornece respostas processadas ao cliente requisitante. Em um sistema *Web* convencional, o cliente é um *software* navegador *Web* que realiza requisições utilizando protocolo de transferência de hipertexto (*HTTP - HyperText Transfer Protocol*) à um computador servidor, que possui os objetos solicitados. A Figura 1 ilustra

estes conceitos, com ênfase no fluxo do processo (TANENBAUM; WETHERALL, 2010; KUROSE; ROSS, 2017).



Figura 1 – Processo de interação baseado no modelo cliente-servidor.

Os principais objetos de sistemas *Web* são páginas escritas em linguagem de marcação de hipertexto (HTML - *HyperText Markup Language*) que possuem ligações para outras páginas de mesmo tipo. O acesso a estas páginas é gerenciado por servidores *Web*, como o Apache (APACHE, 2016), Glassfish (GLASSFISH, 2016) ou IIS (IIS, 2016).

1.3.3 Linguagem de Programação

O desenvolvimento de sistemas computacionais, tanto do tipo *stand-alone* quanto do tipo distribuído, como os sistemas *Web*, passa pela etapa de codificação. Nesta fase, o sistema modelado é representado através de algoritmos em linguagem de programação. Usualmente são utilizadas linguagens de programação de alto nível de abstração, que após processo de compilação fornecem programas que podem ser executados diretamente em máquina (MANZANO; OLIVEIRA, 2008; SEBESTA, 2016).

Linguagens de programação apresentam diferentes abordagens para representação de sistemas reais, denominadas paradigmas de programação. Sebesta (2016) destaca como principais paradigmas: procedural, funcional, lógico, declarativo e orientado à objetos.

1.4 Engenharia de Software

A Engenharia de *Software* atua em aspectos relacionados ao estabelecimento de métodos, técnicas e ferramentas para tornar o processo de desenvolvimento de *software* mais eficiente. Este conjunto de conhecimentos é aplicado por equipes de desenvolvimento, que dividem a produção do *software* em etapas, desde o levantamento inicial até o produto final (PRESSMAN; MAXIM, 2014; SEABRA, 2013; REZENDE, 2005).

Para o desenvolvimento eficiente de *software* é necessária descrição detalhada dos requisitos que o sistema deve atender. Existem dois tipos de requisitos de *softwares*. Requisitos funcionais referem-se às funcionalidades que o sistema deve desempenhar e os não funcionais referem-se à restrições de desempenho, usabilidade, segurança e tecnologia para o desenvolvimento do sistema (SOMMERVILLE, 2015; GUEDES, 2008).

Os requisitos levantados para o desenvolvimento de um sistema de informação podem ser organizados em casos de uso, que descrevem fluxos para a execução das funcionalidades do sistema. Os casos de uso podem ser sintetizados em diagramas de caso de uso, utilizados em diversas metodologias de desenvolvimento de *software* (GUEDES, 2008).

1.4.1 Metodologias de Desenvolvimento de *Software*

As fases do desenvolvimento de *software* incluem métodos e metodologias, motivo de discussão científica, por serem amplamente utilizados como sinônimos. Métodos têm relação direta com a forma como as etapas de um processo são desenvolvidas. A metodologia é a área de estudo de diferentes métodos. Portanto, o método caracteriza o processo de desenvolvimento de *software* como uma série de atividades, e a metodologia incorpora um conjunto de boas práticas de desenvolvimento. Neste sentido, um método de Engenharia de *Software* pode ser considerado parte de uma metodologia. (PRESSMAN; MAXIM, 2014; WAZLAWICK, 2014).

A metodologia orientada a objetos considera dados e objetos do sistema como elementos logicamente relacionados, baseando-se em conceitos de herança e classe. A metodologia estruturada utiliza métodos orientados a processos para especificação e separação de funcionalidades do sistema, utilizando diagramas de fluxo de dados. A metodologia ágil fundamenta-se na característica de reformulação contínua dos sistemas, inclusive durante seu processo de desenvolvimento. Nesta metodologia de desenvolvimento é adotada abordagem de sistematização, interação e resposta rápida à mudanças nas regras de negócio do sistema. (PRESSMAN; MAXIM, 2014; SOMMERVILLE, 2015).

1.4.2 Processo Unificado

O Processo Unificado (*Unified Process* - UP) de desenvolvimento de *software* é um método utilizado pela metodologia orientada a objetos, que objetiva transformar requisitos de usuários em funcionalidades de sistema. O Processo Unificado possui o seguinte conjunto de características: i) iterativo e incremental, pois cada iteração resulta em incremento do sistema com funcionalidades melhoradas; ii) dirigido a casos de uso, pois utiliza casos de uso para captar requisitos funcionais; iii) centrado na arquitetura, uma vez que a arquitetura do sistema determina a base de todo desenvolvimento e iv) focado no risco, visto que os maiores risco do projeto devem ser tratados no início do ciclo de vida (PRESSMAN; MAXIM, 2014; WAZLAWICK, 2014).

As iterações do Processo Unificado são organizadas em fases. Na fase de concepção é realizada justificativa acerca da execução do projeto de desenvolvimento de *software*. Na fase de elaboração, os requisitos do produto são detalhados suficientemente para o

planejamento da próxima fase. A implementação (codificação) dos elementos e preparação para instalação do software são realizadas na fase de construção. Na transição ocorrem os testes finais e a implantação do sistema (PRESSMAN; MAXIM, 2014; JACOBSON et al., 1999).

1.4.3 UML

A UML (*Unified Modeling Language*) é uma linguagem visual para a especificação, visualização, construção e documentação orientada a objetos utilizada no processo de desenvolvimento de *software*. Utilizando UML é possível modelar sistemas de *software* por intermédio de diagramas padronizados, o que permite visualização parcial do produto. Estas representações são divididas em estruturais e comportamentais (GUEDES, 2008; BOOCH; RUMBAUGH; JACOBSON, 2006; BEZERRA, 2006).

Os diagramas criados a partir de elementos UML são capazes de retratar objetos do mundo real. A Figura 2 ilustra a taxonomia dos diagramas UML. (BOOCH; RUMBAUGH; JACOBSON, 2006; BEZERRA, 2006).

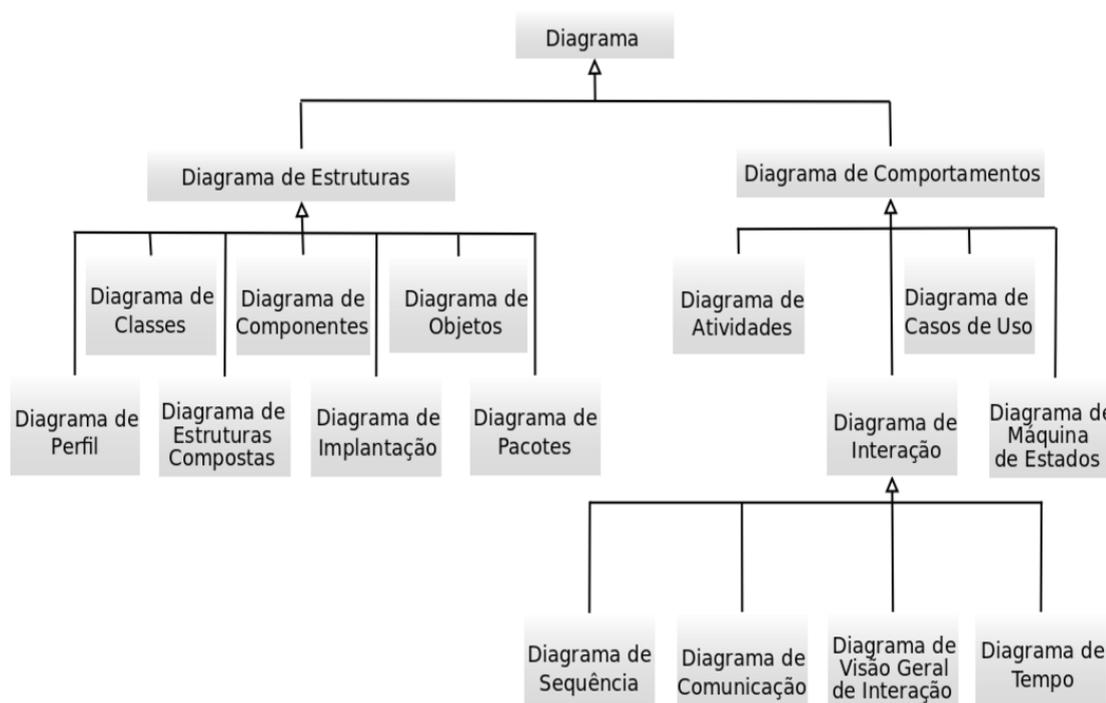


Figura 2 – Taxonomia dos diagramas UML.

1.4.4 Framework

A definição de *framework* de *softwares* está relacionada a *design* reutilizável de objetos representados através de elementos abstratos e concretos, isto permite a construção de modelos extensíveis e adaptáveis (JOHNSON, 1997; HOWARD; SHIP, 2004; FAYAD;

[SCHMIDT, 1997](#)). Desta maneira, *frameworks* permitem introduzir, alterar ou reconectar componentes do sistema, simplesmente analisando suas propriedades.

Frameworks se diferenciam de bibliotecas, pois estas concentram-se apenas em oferecer implementação de funcionalidades, enquanto *frameworks* definem a reutilização de toda uma solução de arquitetura. Os principais exemplos de *frameworks* para desenvolvimento de sistemas *Web* são: Cordova ([CORDOVA, 2016](#)), JQuery ([JQUERY, 2016](#)), AngularJS ([ANGULARJS, 2016](#)), Bootstrap ([BOOTSTRAP, 2016](#)), Meteor ([METEOR, 2016](#)) e Vaadin ([VAADIN, 2016](#)).

1.5 Banco de Dados

De acordo com [Silberschatz, Korth e Sudarshan \(2010\)](#), banco de dados é uma coleção de dados inter-relacionados, representando informações sobre domínio específico de maneira estruturada e com menor redundância possível. Como bancos de dados são ao mesmo tempo acessados e atualizados, é necessária precaução com a coerência dos objetos persistentes ([DATE, 2004](#); [ELMASRI; NAVATHE, 2011](#)).

A coerência de objetos é obtida através de sequência de operações executadas em uma unidade lógica de trabalho. Esta unidade lógica deve apresentar quatro propriedades, representadas pelas iniciais ACID (atomicidade, consistência, isolamento e durabilidade), para ser qualificada como uma transação ([SILBERSCHATZ; KORTH; SUDARSHAN, 2010](#); [ELMASRI; NAVATHE, 2011](#)).

1.5.1 Modelo Relacional

O modelo relacional, baseado na teoria dos conjuntos matemáticos (álgebra relacional), emprega tabelas (relações) para representar características de entidades do mundo real. As colunas da tabela representam atributos e cada linha (tupla) um registro ou entrada de dados ([CODD, 1970](#)).

O modelo relacional pode ser utilizado para manipular dados e estruturas de bancos de dados. Os dados são armazenados e acessados com base em restrições, que evidenciam condições obrigatórias impostas pelo modelo ([SILBERSCHATZ; KORTH; SUDARSHAN, 2010](#); [CHAMBERLIN et al., 1981](#)).

1.5.2 SGBD

Sistemas de Gerenciamento de Banco de Dados (SGBDs) são formados por coleções de *softwares* utilizados para gerir bancos de dados. Utilizando SGBDs é possível inserir, atualizar, recuperar e excluir dados em tabelas. Esta manipulação de dados em bancos de dados relacionais é realizada através de comandos em linguagem SQL (*Struc-*

tered Query Language) (SILBERSCHATZ; KORTH; SUDARSHAN, 2010; DATE, 2004). Alguns exemplos de SGBDs são: SQL Server (SERVER, 2016), Firebird (FIREBIRD, 2016), PostgreSQL (POSTGRESQL, 2016) e MySQL (MYSQL, 2016).

1.6 Considerações Finais

Características de interesse de um sistema podem ser observadas através da elaboração de modelos, frequentemente desenvolvidos utilizando o computador como ferramenta. Sistemas computacionais surgiram para automatizar tarefas que cotidianamente eram realizadas de forma manual. Estes sistemas são desenvolvidos utilizando técnicas e métodos específicos, estabelecidos pela Engenharia de *Software*. O sistema apresentado neste trabalho foi desenvolvido utilizando conceitos e tecnologias apresentados neste capítulo.

2 Desenvolvimento do projeto

2.1 Introdução

Neste capítulo é exposto a motivação associada ao projeto, e uma breve descrição do sistema. Posteriormente, são apresentadas as modelagens do sistemas através da UML (*Unified Modeling Language*). O método de desenvolvimento deste sistema foi fundamentado no Processo Unificado (*Unified Process - UP*), portanto, são utilizados alguns dos diferentes tipos de diagramas que compõem a UML. São também apresentadas as tecnologias utilizadas para a construção do sistema.

2.2 Descrição do sistema

O projeto *MyTask* constitui uma aplicação *Web* para geração automatizada de documentos oficiais e objetiva a promoção de simplificação na criação, gerenciamento e padronização deste tipo de documento. O sistema caracteriza-se como uma ferramenta de suporte para funcionários de instituições de diversas áreas, inicialmente voltado para o Instituto Federal de Goiás, permitindo que o usuário crie e gerencie seu documento de forma automatizada.

A geração automatizada de documentos oficiais permite que os usuários do sistema concentrem esforços apenas no conteúdo dos mesmos, eliminando a necessidade de esforços adicionais em sua formatação.

Após fornecer ao sistema as informações inerentes ao documento, o sistema realizará sua geração automatizada e o usuário poderá realizar o *download* do projeto, podendo armazená-lo na extensão pdf (*portable document format*). O sistema será capaz de gerir dois tipos diferentes de documentos oficiais: memorando e ofício.

2.3 Diagrama de Atividade

Na Figura 3 é apresentado o diagrama que descreve as atividades que compõem o processo e o fluxo de trabalho do sistema *MyTask*.

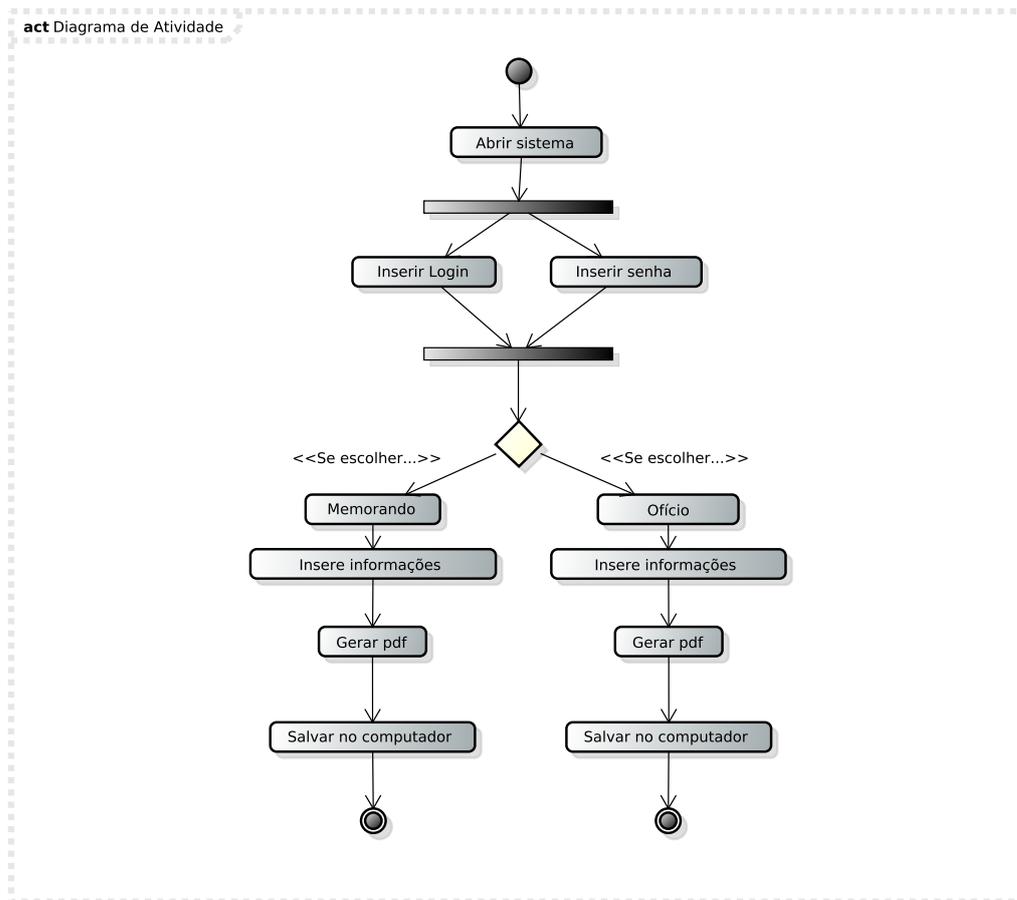


Figura 3 – Diagrama de atividades.

O diagrama de atividades ilustrado na figura 3, tem como objetivo especificar o comportamento do sistema *Mytask* e suas funcionalidades. O nó inicial do diagrama é seguido de bifurcações que indicam atividades simultâneas e que são ligadas por seus conectores, podendo ser observada ainda a possibilidade de escolha de atividade, indicando onde o sistema termina sua ação.

2.4 Diagrama de Máquina de Estados

A Figura 4 apresenta o diagrama de estados do sistema *MyTask*. Neste diagrama, os estados dos objetos que representam o sistema são apresentados assim como os eventos que levam a transição de um estado para outro.

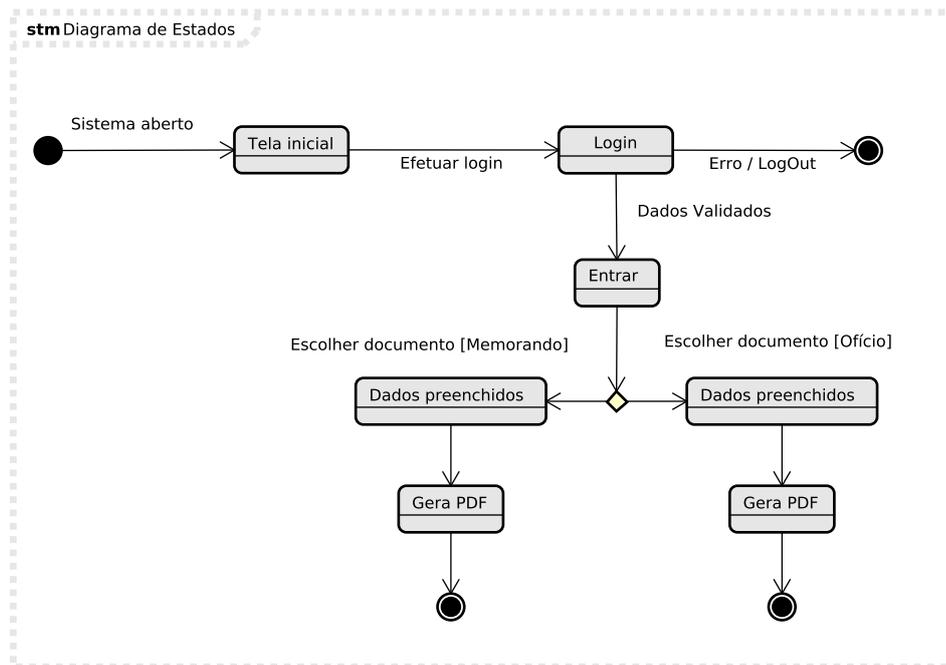


Figura 4 – Diagrama de Máquina de Estado.

O diagrama de estado consiste na transição e relacionamento entre estados. O sistema *Mytask*, como ilustrado na figura 4, contém condições e situações que satisfazem o evento que devem ser reconhecidas e gerar uma reação pelo sistema.

2.5 Levantamento e Análise de requisitos

O levantamento e análise dos requisitos é a representação do que o *software* deve fazer, a partir do processo de observação e pesquisa dos elementos do sistema. Este processo descreve de forma sistemática as propriedades funcionais e não-funcionais do *software*, auxiliando na comunicação entre analistas e projetistas do sistema.

O levantamento e análise de requisitos, conforme mostrado na figura 5, é um processo iterativo com validação contínua de uma atividade para a outra.

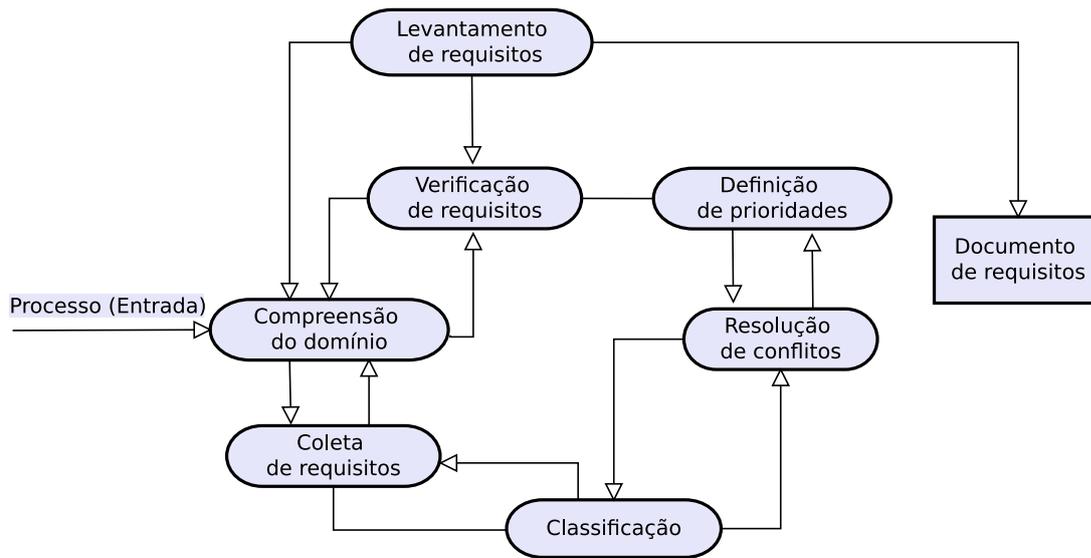


Figura 5 – Processo de levantamento e análise de requisitos.

O processo de levantamento e análise dos requisitos do *software MyTask* será inicialmente realizado através de pesquisas com foco no diagnóstico dos procedimentos de padronização de documentos oficiais de uma instituição de ensino pública. Desta maneira será possível obter conhecimento acerca de suas rotinas administrativas e inseri-las como regras de negócio e casos de uso na aplicação a ser desenvolvida, garantindo a geração de documentos claros e objetivos no âmbito da instituição.

2.5.1 Requisitos Funcionais

Os requisitos funcionais definem as funcionalidades e serviços que o sistema será capaz de realizar. Sendo assim, envolve cálculos, manipulação e processamento de dados de eventos internos ou externos ao sistema. Normalmente são expresso em linguagem natural ou notações fáceis para o entendimento.

Desta forma, os requisitos funcionais encontrados para o primeiro ciclo do projeto *MyTask* são apresentados na tabela (1):

| ID | Descrição |
|--------|---|
| [RF01] | O sistema deve permitir autenticação dos usuários através do uso de <i>login</i> e senha. |
| [RF02] | O sistema deve ser capaz de armazenar todos os dados referentes aos documentos. |
| [RF03] | O sistema deve permitir o cadastro, exclusão e alteração de usuários. |
| [RF04] | O sistema deverá gerar o documento oficial em formato pdf. |
| [RF05] | O sistema deverá listar todos os documentos, através de uma pesquisa simplificada. |
| [RF06] | O sistema deve ser capaz de armazenar todos os dados referentes aos documentos. |

Tabela 1 – Requisitos Funcionais do sistema *MyTask*.

2.5.2 Requisitos Não-Funcionais

Requisitos não-funcionais são relacionados as especificações técnicas em termos de desempenho, usabilidade, confiabilidade, disponibilidade, segurança e tecnologias envolvidas no desenvolvimento do sistema. Resumindo, os requisitos não-funcionais descrevem atributos do sistema ao invés de suas funcionalidades.

Desta forma, os requisitos não-funcionais encontrados para o primeiro ciclo do projeto *MyTask* são apresentados na tabela (2):

| ID | Descrição |
|---------|--|
| [RNF01] | O sistema de armazenamento será estruturado com banco de dados Entidade-relacionamento. |
| [RNF04] | O sistema funcionará em todos os browsers: Mozilla Firefox, Google Chrome, Opera, etc. |
| [RNF06] | Cada processo realizado no sistema, deverá ser feito com o mínimo de cliques possíveis. |
| [RNF07] | O sistema deverá ser <i>Web</i> . |
| [RNF08] | O sistema deverá armazenar todos os dados em banco de dados persistente. |
| [RNF09] | O banco de dados que o sistema utilizará será o postgresSQL e o gestor gráfico será o pgAdmin. |

Tabela 2 – Requisitos Não-Funcionais do sistema *MyTask*.

2.5.3 Documento de Requisitos

O documento de requisitos determina o escopo das funcionalidades que o sistema deve desempenhar. Este documento deve ser completo e escrito de forma coesa para a compreensão das partes interessadas.

| Itens | Conteúdo |
|------------------------------|--|
| 1. Introdução | Escopo e propósito do projeto. |
| 2. Requisitos Funcionais | Principais funcionalidades do software. |
| 3. Requisitos Não-Funcionais | Descrição dos requisitos relacionados ao uso da aplicação. |
| 4. Escopo Não-Contemplado | Descrição das funcionalidades não contempladas no escopo do sistema. |
| 5. Documentação de apoio | Descrição dos diagramas do sistema. |

Tabela 3 – Especificação do Documento de Requisitos

A Tabela (3) apresenta uma relação dos itens considerados na construção do documento de requisitos do *software MyTask*. A relação de itens destacados na Tabela (3) não pressupõe a intenção de ser completo.

2.6 Diagrama de Caso de Uso

Na Figura 6 é apresentado o diagrama que descreve as atividades que compõem o processo e o fluxo de trabalho do sistema *MyTask*.

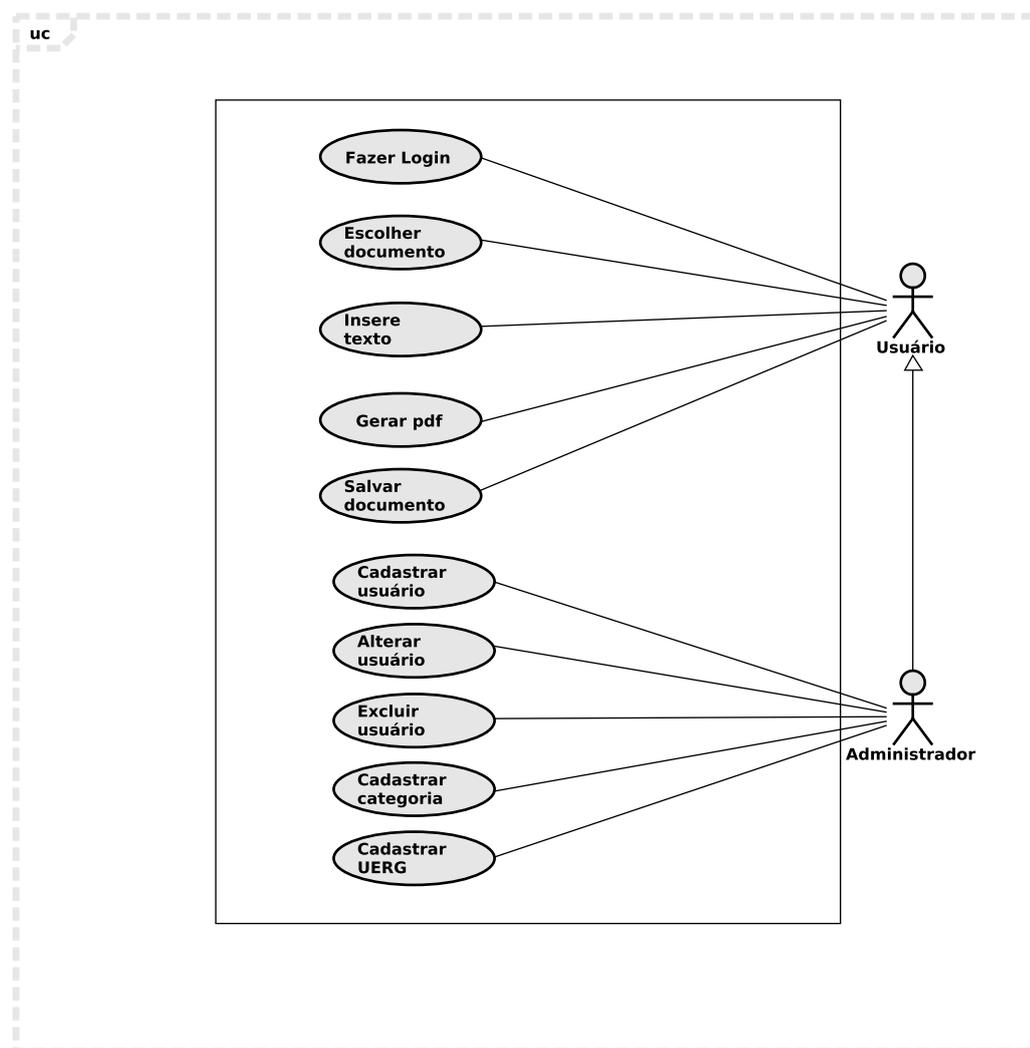


Figura 6 – Diagrama de Caso de Uso.

Como podemos observar na 6 esse diagrama composto por desenhos descrevem de maneira objetiva o fluxo do sistema. Nele vemos as funcionalidades e as interações dos usuários que fará a execução do caso de uso.

A figura 6 ilustra a interação do ator usuário com sistema a partir dos casos de usos: Fazer login, Escolher documento, Inserir texto e Salvar Documento. O caso de uso fazer login autentica o usuário no sistema. Após a autenticação poderá escolher o tipo de documento, sendo eles memorando ou ofício. Logo, poderá inserir o conteúdo do documento e salvá-lo. Já o ator administrador herdará os casos de uso do ator usuário. Além disso, poderá interagir com o sistema a partir dos casos de uso: Cadastrar usuário, Alterar usuário, Excluir usuário, Cadastrar categoria e Cadastrar UERG.

2.7 Diagrama de Classes

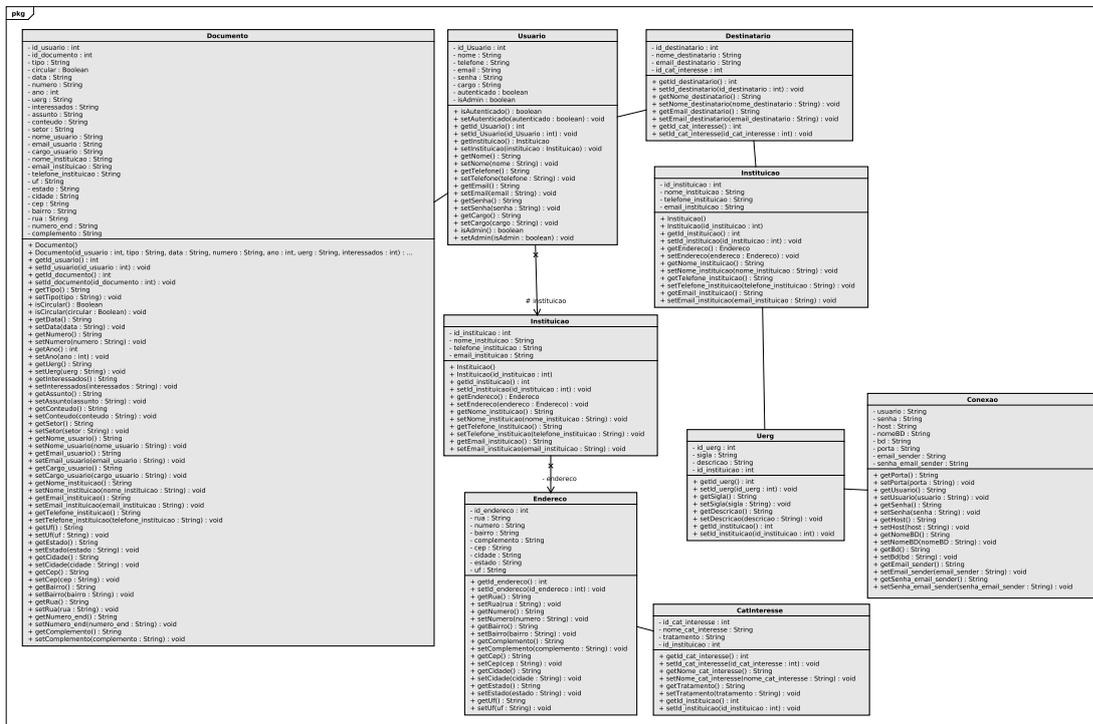


Figura 7 – Diagrama de Classe.

Na figura A é ilustrado o diagrama de classe do sistema *MyTask*. O diagrama mostra as estruturas e relações entre as classes. As classes representadas neste diagrama pertencem ao pacote “model” do projeto. Portanto, essas classes são algumas do projeto.

2.8 Codificação

Para a codificação do sistema a nível de *back-end* será utilizada a linguagem de programação orientada à objetos JAVA (JAVA, 2017), que possui especificação simplificada tanto de sintaxe quanto arquitetural. A metodologia de desenvolvimento adotada será o Processo Unificado (*Unified Process*) que proporciona ferramenta para o desenvolvimento do processo de transformação de requisitos de usuário em um sistema informatizado.

A construção das interfaces (*front-end*) será realizada utilizando o *framework* Bootstrap (BOOTSTRAP, 2016) e as bases de dados do sistema serão modeladas conforme o modelo relacional, que utiliza conceitos de entidade e relação. Para o gerenciamento das bases de dados será utilizado o Sistema Gerenciador de Banco de Dados *postgreSQL* (POSTGRESQL, 2016), em conjunto com o gestor gráfico de dados *pgAdmin* (PGADMIN, 2017).

2.9 Considerações Finais

O sistema proporcionará elaboração e gerenciamento de documentos oficiais de forma automatizada. Isto permite que os usuários concentrem esforços apenas no conteúdo do documento. A utilização de diagramas padronizadas nos permite visualização parcial do *software*, facilitando o processo de desenvolvimento.

Parte II

Resultados

3 Estudo de caso

3.1 Introdução

Neste capítulo apresenta-se a metodologia proposta para o desenvolvimento do sistema. Apresenta-se ainda a descrição do sistema e do processo utilizado referentes as etapas do projeto e os conhecimentos utilizados para projetar o mesmo.

3.2 Estrutura do *software*

O *software* desenvolvido, estruturado sobre plataforma *web* será utilizado através de aplicativos comuns de navegação na *Internet*. A interação é possível por meio do uso de “páginas dinâmicas” produzidas no servidor, de acordo com os dados informados ou requisitados pelo usuário. O sistema caracteriza-se como uma ferramenta de suporte para funcionários de instituições de diversas áreas, tais como o Instituto Federal de Goiás, permitindo que o usuário crie e gerencie seu documento de forma automatizada e padronizada.

Para o desenvolvimento do *software* realizou-se pesquisa com foco no diagnóstico dos procedimentos de padronização de documentos oficiais do Instituto Federal de Goiás. Desta maneira, após entrevistas com os gestores que lidam com a produção de documentos oficiais de comunicação, foi possível obter conhecimento acerca de suas rotinas administrativas e inseri-las como regras de negócio e casos de uso no desenvolvimento do sistema, garantindo a geração de documentos claros e objetivos no âmbito da instituição.

O sistema permite a geração de documentos de comunicação oficial utilizando todas as regras do gênero textual. Esta escolha pode ser realizada através de uma interface simples. Após o fornecimento das informações relativas às preferências de conteúdo do documento, o sistema disponibiliza ao usuário um documento oficial (memorando ou ofício) preenchido e padronizado.

3.3 Tecnologia

O sistema utilizará o conceito de arquitetura cliente-servidor (TANENBAUM; WETHERALL, 2010; KUROSE; ROSS, 2017). Desta forma, o servidor conserva o banco de dados e as informações são compartilhadas por usuários através de aplicações nos clientes.

Para a criação de páginas *web* dinâmicas foi utilizada a tecnologia *JavaServer Pages* (*JSP*), que usa a linguagem de programação Java (HALL, 2001). Para a utilização da tecnologia foi necessária a instalação do Eclipse uma *Integrated Development Environment* (*IDE*), versão *Neon.3 Release* (4.6.3) para desenvolvimento JAVA que funciona como uma ferramenta de apoio ao desenvolvedor.

Nesta mesma máquina deverá estar rodando um aplicativo servidor *Web*, que irá proporcionar a interação com o navegador do usuário. O servidor *Web* escolhido foi o Apache devido sua popularidade e gratuidade (APACHE, 2016). O banco de dados utilizado foi o PostgreSQL, em sua versão 9.6.3.

O *layout* do documento oficial é elaborado pelo *JasperSoft Studio* (*JSS*) baseado em eclipse, sendo uma reescrita completa do *iReport*, disponível como *plugin*. O *iReport* permite a criação de *layouts* sofisticados contendo, gráficos, imagens, relatórios, entre outros (JASPERSOFT, 2017).

3.4 Funcionalidades

3.4.1 Autenticação no sistema

A página inicial do Sistema *MyTask* é exibido na Figura 8. O usuário deve realizar *login* no sistema para ter permissão de uso, como pode ser visto na Figura 9.

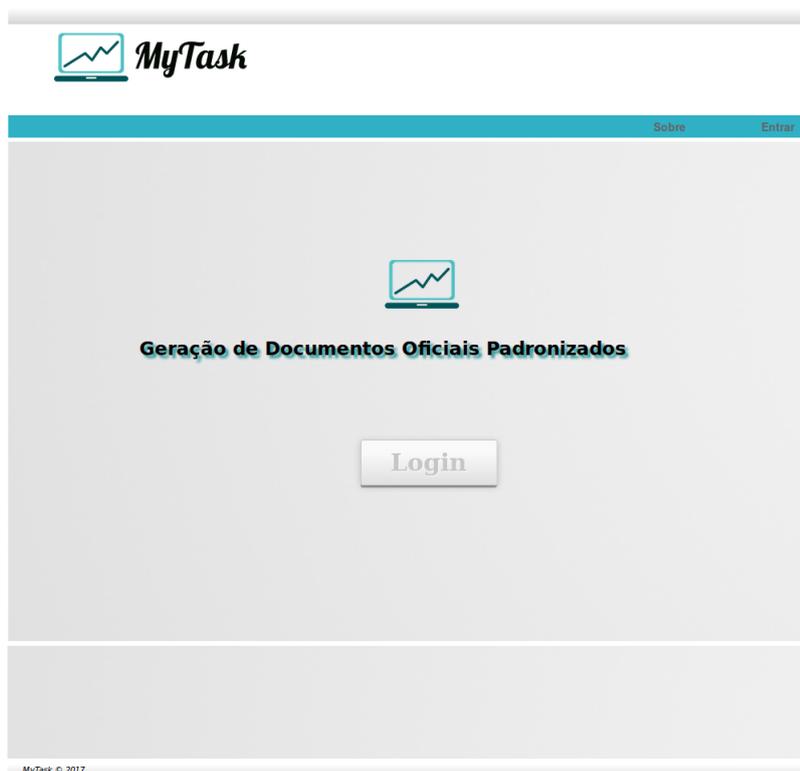


Figura 8 – Página de autenticação do sistema *MyTask*.

A vantagem na utilização de *login* para validação de usuários é que desta maneira pode-se implementar políticas de controle de acesso ao sistema de geração de documentos oficiais, garantido que somente usuários autorizados possam redigir e enviar este tipo de documento.

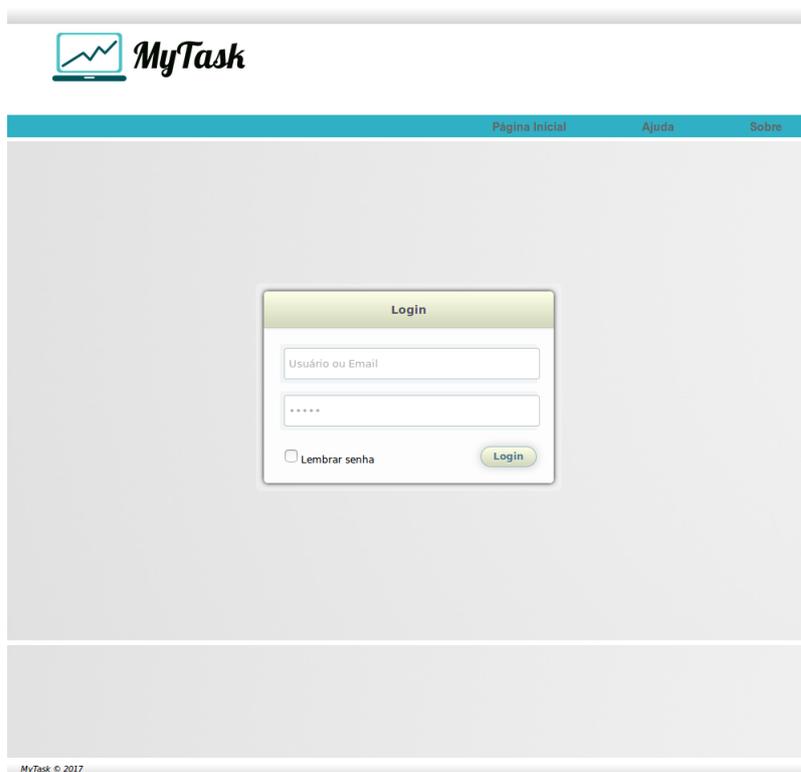


Figura 9 – Página de login do sistema Mytask.

3.4.2 Tipo de documento

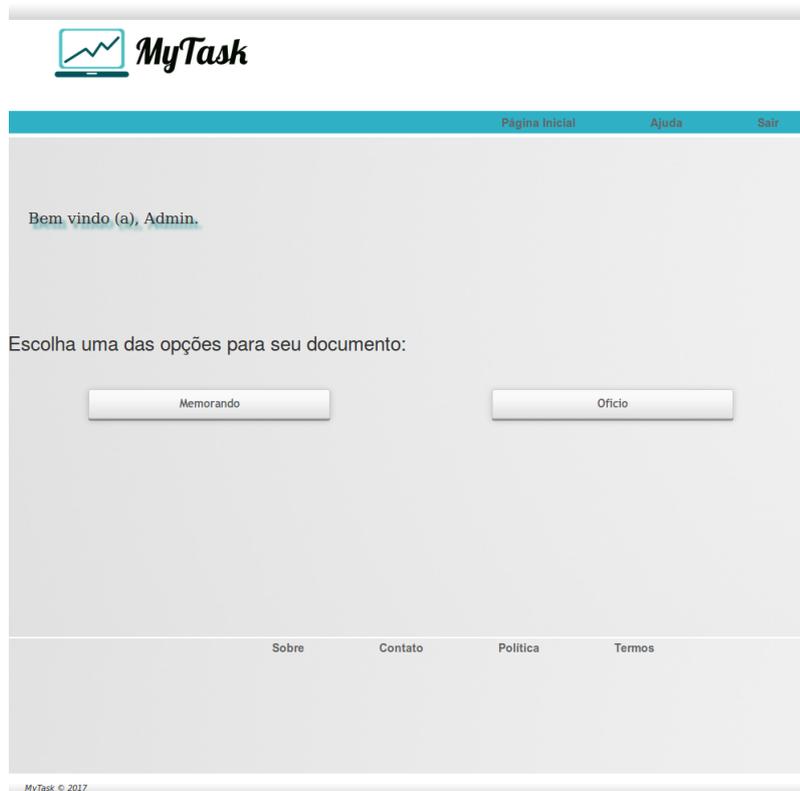


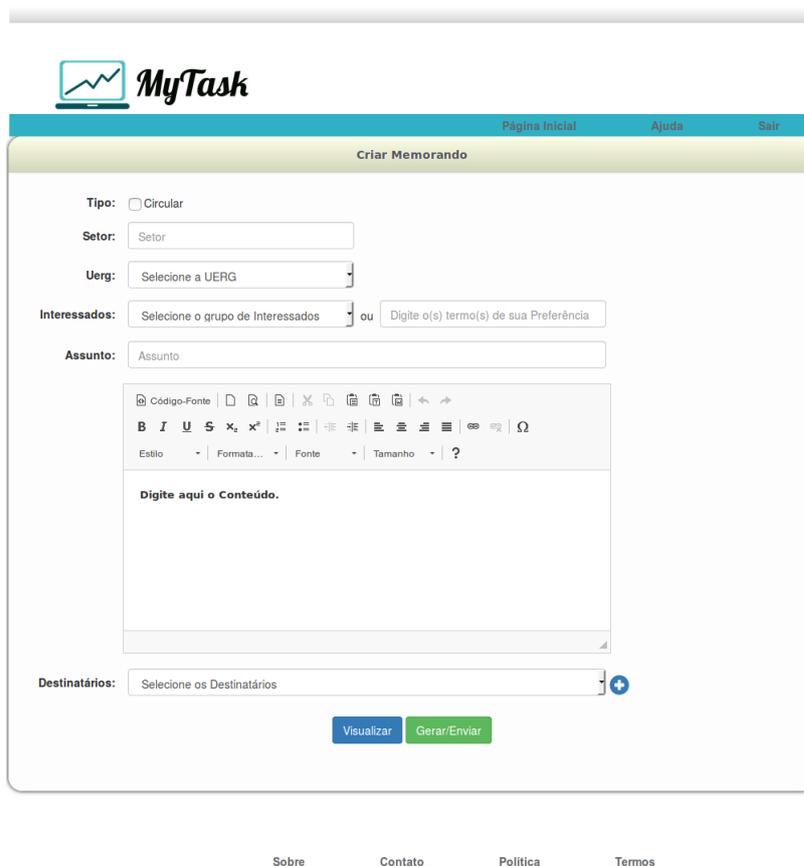
Figura 10 – Página de escolha de produção de documento oficial (*Memorando ou Ofício*).

Caso seja selecionado ofício, aplica-se na formatação do documento regras de comunicação externa. Caso seja selecionado memorando, aplica-se regras de comunicação interna. Na figura 10 pode ser observada a etapa de seleção do tipo de documento oficial.

3.4.3 Formatação do documento

A Figura 11 ilustra a página de definição das regras de formatação, no qual é necessária a inserção do conteúdo do documento. O conteúdo diz respeito ao assunto do qual trata a comunicação. Com as informações relativas ao conteúdo, o sistema é capaz de gerar o documento padronizado e fornecer via interface *Web* opções para seu download.

A página de inserção de conteúdo é a mesma para memorando e ofício. No caso do memorando, ele poderá ter a opção de ser circular e ainda terá a opção de colocar o setor que trabalha e a UERG sigla representando o setor. Poderá ainda adicionar a categoria de interessados e o assunto do documento. No campo de conteúdo, o usuário terá várias opções de formatação do texto. O usuário poderá então adicionar os *emails* que deseja enviar o *pdf* gerado.



The screenshot shows the 'Criar Memorando' (Create Memo) form in the MyTask system. The form is titled 'Criar Memorando' and has a teal header with navigation links for 'Página Inicial', 'Ajuda', and 'Sair'. The form fields include:

- Tipo:** A radio button for 'Circular'.
- Setor:** A text input field with the placeholder 'Setor'.
- Uerg:** A dropdown menu with the placeholder 'Selecione a UERG'.
- Interessados:** A dropdown menu with the placeholder 'Selecione o grupo de Interessados' and an 'ou' (or) option followed by a text input field for 'Digite o(s) termo(s) de sua Preferência'.
- Assunto:** A text input field with the placeholder 'Assunto'.
- Editor:** A rich text editor with a toolbar containing icons for bold, italic, underline, strikethrough, text color, background color, bulleted list, numbered list, indent, outdent, link, unlink, and help. Below the toolbar are dropdown menus for 'Estilo', 'Formata...', 'Fonte', and 'Tamanho', along with a help icon.
- Destinatários:** A dropdown menu with the placeholder 'Selecione os Destinatários' and a plus icon to add more.

At the bottom of the form are two buttons: 'Visualizar' (blue) and 'Gerar/Enviar' (green). Below the form, there are links for 'Sobre', 'Contato', 'Política', and 'Termos'. The footer of the page reads 'MyTask © 2017'.

Figura 11 – *Página de inserção de conteúdo do documento oficial.*

3.4.4 Localização do documento

Na localização dos arquivos importados no sistema, entramos na tela de localização dos documentos. Ela contém os campos de filtros, que possibilita a busca por todos, data, assunto, número, ofícios, memorandos e último criado. A figura 12 ilustra esse processo.

The screenshot displays the MyTask web application interface. At the top, there is a logo for "MyTask" and a navigation bar with links for "Urgs", "Categorias de Interesse", "Usuários", and "Salir". Below the navigation bar, a welcome message "Bem vindo (a), Admin." is shown next to a "Novo Documento" button. A search bar with the placeholder "Buscar por..." and a "Buscar" button is present. Below the search bar, it indicates "5 Resultados". The results are listed in a table-like format with six entries, each containing a document icon, title, date, subject, and action buttons (Download, Editar, Apagar).

| Document Title | Date | Subject | Download | Edit | Delete |
|--------------------------|------------|---|----------|------|--------|
| MEMORANDO. Nº 2 | 12/08/2017 | Assunto: Reunião | Download | Edit | Delete |
| OFICIO. Nº 2 | 12/08/2017 | Assunto: Reuniao | Download | Edit | Delete |
| MEMORANDO-CIRCULAR. Nº 3 | 12/08/2017 | Assunto: Reunião | Download | Edit | Delete |
| MEMORANDO-CIRCULAR. Nº 4 | 12/08/2017 | Assunto: Reunião | Download | Edit | Delete |
| MEMORANDO-CIRCULAR. Nº 5 | 12/08/2017 | Assunto: Reunião | Download | Edit | Delete |
| MEMORANDO-CIRCULAR. Nº 6 | 12/08/2017 | Assunto: aquisição de material de expedientes | Download | Edit | Delete |

Página 1 de 1

[Sobre](#) [Contato](#) [Política](#) [Termos](#)

MyTask © 2017

Figura 12 – Página de localização do documento oficial.

4 Conclusão

Documentos oficiais são as ferramentas utilizadas para comunicação oficial no campo administrativo. Esta comunicação torna-se efetiva por meio de atos administrativos, praticados pelos agentes da administração pública investidos de poderes. Neste contexto, a utilização de um sistema automatizado de geração de documentos oficiais pode descomplexificar, no que diz respeito ao formato e modalidades, a produção destas ferramentas de comunicação.

O software proposto neste artigo visa viabilizar esta produção padronizada de documentos oficiais de comunicação emitidos por Instituições. A utilização de *templates* (modelos de *layout*) permite que o usuário do sistema consiga elaborar documentos oficiais sem a necessidade do conhecimento prévio de normas de formatação. Como trabalhos futuros, pretende-se implantar o sistema *MyTask* no Instituto Federal de Goiás para que gestores produzam documentos de comunicação oficiais (memorandos e ofícios) de forma ágil e padronizada, o que proverá celeridade na resolução de problemas relativos a comunicação no âmbito da Instituição.

4.1 Sugestões para Trabalhos Futuros

- Desenvolver método de credenciamento de funcionários já utilizado na instituição com integração LDAP (*Lightweight Directory Access Protocol*) (SUNGAILA et al., 2008);
- Desenvolver método capaz de agregar características de segurança através da assinatura digital.

4.2 Produções científicas (Aguardando decisão editorial)

- VIEIRA, J. A. LEMES, M. T. MyTask: Um Sistema Baseado na Web para Geração Automatizada de Documentos Oficiais Emitidos Pelo Instituto Federal de Goiás. Revista de Educação, Ciência e Tecnologia do IFG. ISSN: 2526-2122
- VIEIRA, J. A. LEMES, M. T. MyTask: Sistema Baseado na Web para Geração Automatizada de Documentos Utilizados em Comunicações Oficiais. Revista de Sistemas e Computação - RSC. ISSN: 2237-2903.

Referências

- ANGULARJS. *AngularJs*. 2016. Disponível em: <<https://angularjs.org/>>. Citado na página 36.
- APACHE. *Apache HTTP server project*. 2016. Disponível em: <<https://httpd.apache.org/>>. Citado 2 vezes nas páginas 33 e 52.
- APPELT, W.; TETTEH-LARTEY, N. The formal specification of the iso open document architecture (oda) standard. *The Computer Journal*, Br Computer Soc, v. 36, n. 3, p. 269–279, 1993. Citado na página 24.
- ARMITAGE, J. *Historia do brasil*. Conselho Editorial do Senado Federal, 2011. Citado na página 23.
- AZEVEDO, F. N. *Carlota Joaquina: cartas inéditas*. [S.l.]: Casa da Palavra, 2007. Citado na página 23.
- BATISTA, D. J.; PERIOTTO, Á. J. Gestão de informações e documentos oficiais: especificação de uma solução de uma unidade administrativa da universidade estadual de maringá. *Caderno de Administração*, v. 17, n. 2, p. 70–83, 2011. Citado na página 25.
- BELZ, A. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, Cambridge University Press, v. 14, n. 04, p. 431–455, 2008. Citado na página 25.
- BERNERS-LEE, T. et al. World-wide web: the information universe. *Internet Research*, Emerald Group Publishing Limited, v. 20, n. 4, p. 461–471, 2010. Citado na página 25.
- BERTALANFFY, L. V. *General system theory: foundations, development, applications (revised edition)*. George Braziller Inc., 1969. Citado na página 31.
- BEZERRA, E. *Princípios De Análise E Projeto De Sistemas Com Uml-3ª Edição*. [S.l.]: Elsevier Brasil, 2006. v. 3. Citado na página 35.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. [S.l.]: Elsevier Brasil, 2006. Citado na página 35.
- BOOTSTRAP. *Bootstrap*. 2016. Disponível em: <<http://getbootstrap.com/>>. Citado 2 vezes nas páginas 36 e 46.
- BRAGA, C. O.; STAA, A. V. *Ferramentas para a geração automática de documentação*. 1995. Citado na página 25.
- BRASIL. *Manual de Redação da Presidência da República*. [S.l.]: Brasília: Departamento de Imprensa Nacional, 2002. Citado na página 23.
- BROWN, H. Standards for structured documents. *The Computer Journal*, v. 32, n. 6, p. 505–514, 1989. Citado na página 24.
- CASSANDRAS, C. G.; LAFORTUNE, S. *Introduction to discrete event systems*. [S.l.]: Springer Science & Business Media, 2009. Citado na página 31.

- CHAMBERLIN, D. D. et al. A history and evaluation of system r. *Communications of the ACM*, ACM, v. 24, n. 10, p. 632–646, 1981. Citado na página 36.
- CHWIF, L.; MEDINA, A. *Modelagem e Simulação de Eventos Discretos, 4ª Edição: Teoria e Aplicações*. [S.l.]: Elsevier Brasil, 2014. v. 4. Citado na página 31.
- CODD, E. F. A relational model of data for large shared data banks. *Communications of the ACM*, ACM, v. 13, n. 6, p. 377–387, 1970. Citado na página 36.
- CORDOVA. *Apache Cordova*. 2016. Disponível em: <<https://cordova.apache.org/>>. Citado na página 36.
- CORMEN, T. H. *Introduction to algorithms*. [S.l.]: MIT press, 2009. Citado na página 32.
- COSTA, E. M. M.; LIMA, A. M. N. *Sistemas Dinâmicos a Eventos Discretos: Fundamentos Básicos para a Moderna Automação Industrial, Ed. Salvador, BA: EDUFBA*, 2005. Citado na página 31.
- CURY, J. E. R. Teoria de controle supervisorio de sistemas a eventos discretos. V *Simpósio Brasileiro de Automação Inteligente (Minicurso)*, 2001. Citado na página 31.
- DATE, C. J. *Introduction to database systems*. [S.l.]: Addison Wesley, 2004. Citado 2 vezes nas páginas 36 e 37.
- DEITEL, H. M.; DEITEL, P. J. *Java: How to Program, 10a. Edição*. [S.l.]: Prentice Hall, 2010. Citado na página 32.
- ELMASRI, R.; NAVATHE, S. B. *Database systems*. Pearson Addison Wesley, 2011. Citado na página 36.
- FAYAD, M.; SCHMIDT, D. C. Object-oriented application frameworks. *Communications of the ACM*, ACM, v. 40, n. 10, p. 32–38, 1997. Citado 2 vezes nas páginas 35 e 36.
- FERRARI, T. M. *Departamento de Engenharia Elétrica. Tese (Doutorado) — Universidade Federal de Itajubá*, 2006. Citado na página 25.
- FILHO, W. d. P. P. *Multimídia: Conceitos e Aplicações*. [S.l.]: Livros Técnicos e Científicos, 2000. Citado na página 24.
- FIREBIRD. *Firebird*. 2016. Disponível em: <<https://firebirdsql.org/>>. Citado na página 37.
- FURUTA, R.; SCOFIELD, J.; SHAW, A. Document formatting systems: Survey, concepts, and issues. 1982. Citado na página 24.
- GLASSFISH. *GlassFish*. 2016. Disponível em: <<https://glassfish.java.net/>>. Citado na página 33.
- GOMES, L. Como um imperador cansado, um marechal vaidoso e um professor injustiçado contribuíram para o fim da monarquia e a proclamação da república no brasil. *São Paulo: Globo*, p. 231, 2013. Citado na página 24.
- GUEDES, G. T. *UML 2-Uma Abordagem Prática-1ª Edição*. [S.l.: s.n.], 2008. Citado 3 vezes nas páginas 33, 34 e 35.

- GUTIÉRREZ, J. J. ¿ qué es un framework web? 2014. Citado na página 26.
- HALL, M. *More servlets and JavaServer pages*. [S.l.]: Prentice Hall PTR, 2001. Citado na página 52.
- HOWARD, M.; SHIP, L. *Tapestry in Action*. [S.l.]: Softbound Press, 2004. Citado 2 vezes nas páginas 35 e 36.
- HURST, N.; LI, W.; MARRIOTT, K. Review of automatic document formatting. In: ACM. *Proceedings of the 9th ACM symposium on Document engineering*. [S.l.], 2009. p. 99–108. Citado na página 25.
- IIS. *Internet Information Services*. 2016. Disponível em: <<https://www.iis.net/>>. Citado na página 33.
- ISOTANI, S. et al. Estado da arte em web semântica e web 2.0: potencialidades e tendências da nova geração de ambientes de ensino na internet. *Revista Brasileira de Informática na Educação*, v. 17, n. 01, p. 30, 2009. Citado na página 25.
- JACOBSON, I. et al. *The unified software development process*. [S.l.]: Addison-wesley Reading, 1999. v. 1. Citado na página 35.
- JASPERSOFT. *JasperSoft*. 2017. Disponível em: <<https://www.jaspersoft.com>>. Citado na página 52.
- JAVA. *JAVA*. 2017. Disponível em: <<https://www.java.com>>. Citado na página 46.
- JOHNSON, R. E. Components, frameworks, patterns. In: ACM. *ACM SIGSOFT Software Engineering Notes*. [S.l.], 1997. v. 22, n. 3, p. 10–17. Citado 2 vezes nas páginas 35 e 36.
- JQUERY. *jQuery*. 2016. Disponível em: <<https://jquery.com/>>. Citado na página 36.
- KOUTZII, F. et al. Nova república: um balanço. *Porto Alegre: L&PM*, 1986. Citado na página 24.
- KUROSE, J. F.; ROSS, K. W. *Computer Networks and the Internet*. [S.l.: s.n.], 2017. Citado 2 vezes nas páginas 33 e 51.
- LEMES, M. J. R. *Complexidade, acoplamento e criticalidade (C^2A) como indicadores de risco em projetos de sistemas*. Tese (Doutorado) — Universidade de São Paulo, 2012. Citado na página 31.
- LOGAN, R. K. The extended mind: understanding language and thought in terms of complexity and chaos theory. *2000) Communication and Speech Annual*, v. 14, 2000. Citado na página 32.
- LUCAS, J. A. A history of the marathon race. *Journal of Sport History*, JSTOR, v. 3, n. 2, p. 120–138, 1976. Citado na página 23.
- MANZANO, J. A. N.; OLIVEIRA, J. F. de. *Estudo dirigido de algoritmos*. [S.l.]: Érica, 2008. Citado 2 vezes nas páginas 32 e 33.
- MARTINS, A. V. A carta de pero vaz de caminha: breve estudo das palavras gramaticais. 2005. Citado na página 23.

- MCKEOWN, K.; KUKICH, K.; SHAW, J. Practical issues in automatic documentation generation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the fourth conference on Applied natural language processing*. [S.l.], 1994. p. 7–14. Citado na página 24.
- MENDES, G. F.; JÚNIOR, N. J. F. Manual de redação da presidência da república. *Brasília: Presidência da República*, 1991. Citado na página 24.
- METEOR. *Meteor*. 2016. Disponível em: <<https://www.meteor.com/>>. Citado na página 36.
- MEZAROBBA, G. Entre reparações, meias verdades e impunidade: o difícil rompimento com o legado da ditadura no brasil. *Revista Internacional de Direitos Humanos*, p. 7, 2010. Citado na página 24.
- MONTGOMERY, E. Introdução aos sistemas a eventos discretos e à teoria de controle supervisorio. *Alta Books*, v. 122, 2004. Citado na página 31.
- MYSQL. *MySQL*. 2016. Disponível em: <<https://www.mysql.com>>. Citado na página 37.
- OLIVEIRA, F. G.; SEABRA, J. M. P. Metodologias de desenvolvimento de software: uma análise no desenvolvimento de sistemas na web. *Tecnologias em projeção*, v. 6, n. 1, p. 20–34, 2015. Citado na página 26.
- PERROW, C. *Normal accidents: Living with high risk technologies*. [S.l.]: Princeton University Press, 2011. Citado na página 31.
- PGADMIN. *pgAdmin*. 2017. Disponível em: <<https://www.pgadmin.org>>. Citado na página 46.
- POSTGRESQL. *PostgreSQL*. 2016. Disponível em: <<https://www.postgresql.org/>>. Citado 2 vezes nas páginas 37 e 46.
- PRESSMAN, R.; LOWE, D. *Web engineering: a practitioner's approach*. [S.l.]: Pearson, 2009. Citado 2 vezes nas páginas 25 e 26.
- PRESSMAN, R.; MAXIM, B. *Software Engineering: A Practitioner's Approach*. [S.l.]: McGraw-Hill Science, 2014. Citado 4 vezes nas páginas 26, 33, 34 e 35.
- RECHTIN, E.; MAIER, M. W. *The art of systems architecting*. [S.l.]: CRC Press, 2010. Citado na página 31.
- REZENDE, D. A. *Engenharia de software e sistemas de informação*. [S.l.]: Brasport, 2005. Citado na página 33.
- SANTAROSA, L. M. C.; CONFORTO, D.; BASSO, L. de O. Ferramentas de autoria e de colaboração: discutindo a acessibilidade e a usabilidade na perspectiva da web 2.0. *Revista Brasileira de Informática na Educação*, v. 21, n. 01, p. 121, 2013. Citado na página 26.
- SEABRA, J. *UML: Uma ferramenta para o design de software*. [S.l.]: Ciência Moderna, 2013. Citado 2 vezes nas páginas 26 e 33.

- SEBESTA, R. W. *Programming language concepts*. [S.l.]: Pearson, 2016. Citado na página 33.
- SERVER, S. *SQL Server*. 2016. Disponível em: <<https://www.microsoft.com/en-us/sql-server/sql-server-2016>>. Citado na página 37.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Database System Concepts*. [S.l.]: McGraw-Hill Science, 2010. Citado 2 vezes nas páginas 36 e 37.
- SOMMERVILLE, I. *Software Engineering*. [S.l.]: Pearson, 2015. Citado 3 vezes nas páginas 26, 33 e 34.
- STALLINGS, W. *Computer Architecture and Organization*. [S.l.]: Pearson, 2015. Citado na página 32.
- STAMPA, I. Memórias reveladas e os arquivos do período da ditadura militar. *ComCiência*, SciELO Brasil, n. 127, 2011. Citado na página 24.
- SUNGAILA, M. et al. *Autenticação centralizada com OpenLDAP*. [S.l.]: Novatec, 2008. 978-8575221426 p. Citado na página 57.
- TANENBAUM, A.; WETHERALL, D. J. *Computer Networks*. [S.l.]: Prentice Hall, 2010. Citado 2 vezes nas páginas 33 e 51.
- TANENBAUM, A. S.; AUSTIN, T. *Structured Computer Organization*. [S.l.]: Pearson, 2012. Citado 2 vezes nas páginas 24 e 32.
- TANENBAUM, A. S.; BOS, H. *Modern operating systems*. [S.l.]: Prentice Hall Press, 2014. Citado na página 32.
- VAADIN. *Vaadin*. 2016. Disponível em: <<https://vaadin.com/>>. Citado na página 36.
- W3C. *W3C Consortium*. 2016. Disponível em: <<https://www.w3.org/>>. Citado na página 32.
- WAZLAWICK, R. *Análise e Design Orientados a Objetos para Sistemas de Informação*. [S.l.]: Elsevier Brasil, 2014. v. 3. Citado na página 34.

Apêndices

APÊNDICE A – Diagrama de Classes

