

1 INTRODUÇÃO

O Sistema de Gerenciamento de Notificação (SNG), foi desenvolvido no segundo semestre do ano de 2016 , tem como publico alvo empresas que necessitam de notificar seus funcionários, através de e-mail, sobre sua escala, férias, feriados e dias de pagamento.

O SGN armazena informações dos funcionários e envia notificações, informando os funcionários suas escalas de serviço, férias, feriados e dias de pagamento, proporcionando comodidade para os usuários.

1.1 Objetivos

1.1.1 Geral

O objeto geral do Sistema de Gerenciamento de Notificação (SGN) e a criação de um sistema que cadastra os funcionários e informações a seu respeito, realize notificações por e-mail informando os dias de serviço, também conceda acesso aos mesmos para que possam consultar as devidas informações facilitando o serviço.

1.1.2 Específicos

a) Elaborar a visão do sistema, avaliando os possíveis riscos, estabelecendo as prioridades, levantando e documentando os requisitos e mantendo o acompanhamento das mudanças desses requisitos;

b) Modelar o sistema visualmente utilizando modelos os diversos diagramas da UML para apresentar as visões estáticas e dinâmicas do sistema;

c) Desenvolver de maneira incremental e iterativa as funcionalidades:

d) Testar, implantar e verificar a qualidade do sistema garantindo que atenda aos padrões de qualidade da organização.

1.2 Estrutura do Trabalho de Conclusão

No capítulo dois...

No capítulo três...

2. REFERENCIAL TEÓRICO

O referencial teórico permite verificar o estado do problema a ser pesquisado, sob o aspecto teórico e de outros estudos e pesquisas realizados (LAKATOS; MARCONI, 2003).

De forma que para a elaboração de um sistema existem vários fatores como metodologias e ferramentas para uma construção correta do sistema com ganho de tempo para finalizar o produto com qualidade.

2.1 Engenharia de *Software*

De acordo com Pressman (2011, p.39) engenharia de software é “o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais”.

Para Sommerville (2011, p.5) “a engenharia de software é uma disciplina de engenharia cujo foco está em todos os aspectos da produção de *software*, desde os estágios iniciais do sistema até sua manutenção”.

Segundo Sommerville (2011) Sobre técnicas e métodos:

Não existem técnicas e métodos universais na engenharia de software adequados a todos os sistemas e todas as empresas. Em vez disso, um conjunto diverso de métodos e ferramentas de engenharia de software tem evoluído nos últimos 50 anos. (SOMMERVILLE, 2011, p.7)

Segundo Pressman (2011) Cada atividade metodológica:

Composta por um conjunto de ações de engenharia de software. Cada ação é definida por um conjunto de tarefas, o qual identifica as tarefas de trabalho a ser completadas, os artefatos de software que serão produzidos, os fatores de garantia da qualidade que serão exigidos e os marcos utilizados para indicar progresso (PRESSMAN, 2011, p.53).

De acordo com Sommerville (2011, p.18) “Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software”.

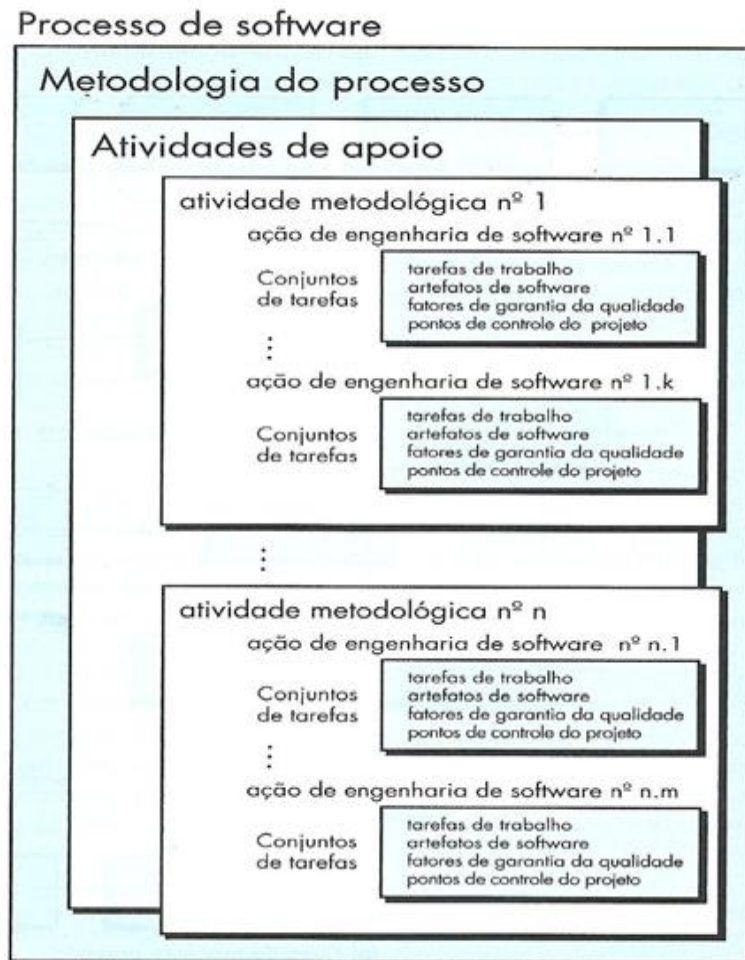
Segundo Pressman(2011, p.52) “Processo de software é definido como uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade”.

Para Sommerville (2011, p.18) Existem quatro atividades fundamentais para a engenharia de software:

- a) Especificação de software. A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
- b) Projeto e implementação. O software deve ser produzido para atender às especificações.
- c) Validação de software. O software deve ser validado para garantir que atenda às demandas do cliente.
- d) Evolução de software. O software deve evoluir para atender às necessidades de mudança do cliente.

Na figura 1, é representada esquematicamente a metodologia de processo de software.

Figura 1 Metodologia do processo de software



Fonte: Pressman, 2011

Segundo Pressman (2011, p.127): “a engenharia de requisitos constrói uma ponte para o projeto e para a construção”.

Para ter uma base do que já quer construir é necessário utilizar engenharia de requisitos para fazer um levantamento do que irá precisar para o sistema, do que o sistema irá precisar e de como os usuários irão lidar com o sistema, assim levantando os requisitos necessários e validando todos tem uma base para construir um sistema robusto (PRESSMAN, 2011).

Para Sommerville (2011) O processo de teste tem dois objetivos distintos:

- a) Demonstrar ao desenvolvedor e ao cliente que o software atende a seus requisitos. Para softwares customizados isso significa que deve haver pelo menos um teste para todas as características do sistema, além de suas combinações que serão incorporados ao release do produto.
- b) Descobrir situações em que o software se comporta de maneira incorreta, indesejável ou de forma indiferente das especificações. Essas são consequências de defeitos de software. O teste de defeito preocupa-se com a eliminação de comportamento indesejáveis do sistema, tais como panes, interações indesejáveis com outros sistemas, processamento incorretos e corrupção de dados.

O primeiro objetivo levar a testes de validação, nos quais você espera que o sistema execute corretamente usando determinado conjunto de casos de teste que refletem o uso esperado do sistema. O segundo objetivo levar a teste de defeitos, nos quais os casos de testes são projetados para expor defeitos.(SOMMERVILLE, 2011)

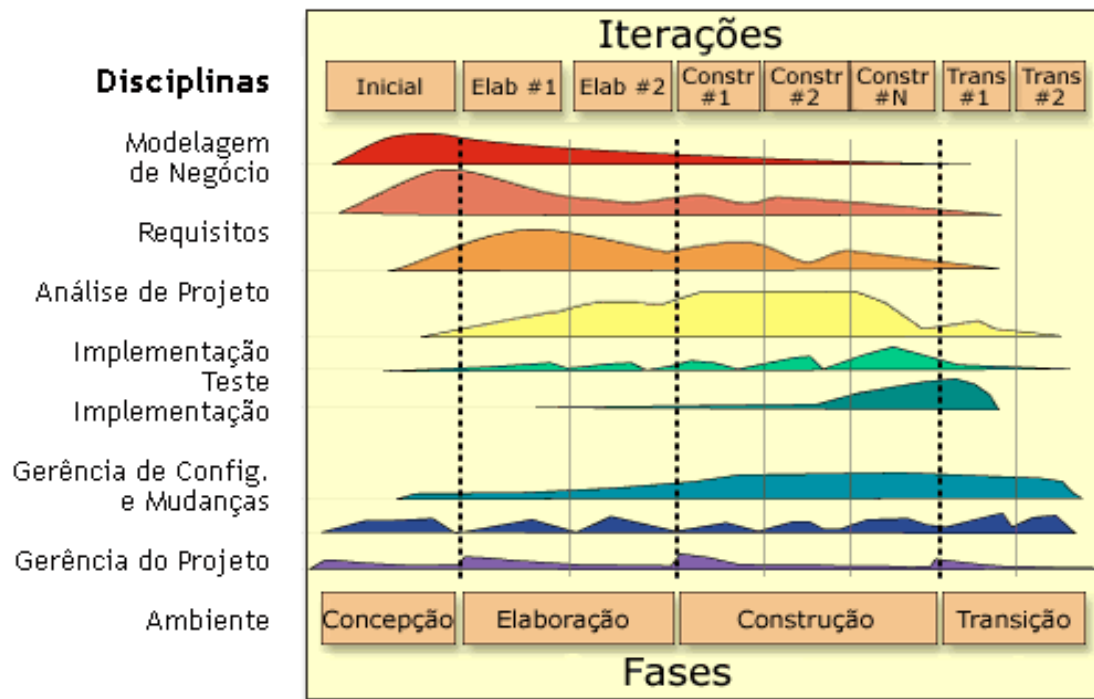
Segundo Sommerville (2011, p.144) “O teste é destinado a mostrar que um programa faz o que é proposto para descobrir os defeitos do programa antes do uso”.

2.2 IBM *Rational Unified Process* (IRUP)

Segundo Sommerville(2011, p.34) “O RUP é um modelo constituído de fases que identifica quatro fases distintas no processo de software. No entanto, ao contrário do modelo cascata, no qual as fases são equalizadas com as atividades do processo, as fases do RUP são estreitamente relacionadas a negócio, e não a assuntos técnicos”.

Na figura 2 mostra as disciplinas e as fases do modelo RUP.

Figura 2 Fases do RUP



Fonte: PTS-DATASUS, 2008

De acordo com Sommerville as fases do RUP são:

- a) **Concepção.** O objetivo da fase de concepção é estabelecer um business case para o sistema. Você deve identificar todas as entidades externas (pessoas e sistemas) que vão interagir com o sistema e definir as interações. Então, você deve usar essas informações para avaliar a contribuição do sistema para o negócio. Se essa contribuição for pequena, então o projeto poderá ser cancelado depois dessa fase.
- b) **Elaboração.** As metas da fase de elaboração são desenvolver uma compreensão do problema dominante, estabelecer um framework da arquitetura para sistema, desenvolver o plano do projeto e identificar os

maiores riscos do projeto. No fim dessa fase, você deve ter um modelo de requisitos para o sistema, que pode ser um conjunto de casos de uso da UML, uma descrição da arquitetura ou plano de desenvolvimento do software.

- c) Construção. A fase de construção envolve projeto, programação e testes do sistema. Durante essa fase, as partes do sistema são desenvolvidas em paralelo e integradas. Na conclusão dessa fase, você deve ter um sistema de software já funcionando, bem como a documentação associada pronta para ser entregue aos usuários.
- d) Transição. A fase final do RUP implica transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários e em seu funcionamento em um ambiente real. Isso é ignorado na maioria dos modelos de processo de software, mas é, de fato, uma atividade cara e, às vezes, problemática. Na conclusão dessa fase, você deve ter um sistema de software documentado e funcionando corretamente em seu ambiente operacional.

A visão estática do RUP prioriza as atividades que ocorrem durante o processo de desenvolvimento. Na descrição do RUP, essas são chamadas workflows. Existem seis workflows centrais, identificadas no processo, e três workflows de apoio. (SOMMERVILLE,2011).

De acordo com Sommerville os workflows do RUP são:

- a) Modelagem de negócios: Os processos de negócio são modelados por meio de casos de uso de negócios.
- b) Requisitos: atores que integram o sistema são identificados e casos de uso são desenvolvidos para modelar os requisitos dos sistemas.

- c) **Análise de projeto:** Um modelo de projeto é criado e documentado com modelos de arquitetura, modelos de componentes, modelos de objetos e modelo de sequência.
- d) **Implementação:** Os componentes do sistema são implementados e estruturados em subsistemas de implementação. A geração automática de código a partir de modelos de projeto ajuda a acelerar esse processo.
- e) **Teste:** O teste é um processo iterativo que é feito em conjunto com a implementação. O teste segue a conclusão da implementação.
- f) **Implantação:** Um release do produto é criado, distribuído aos usuários e instalado em seu local de trabalho.
- g) **Gerenciamento de configuração e mudanças:** Esse workflow de apoio gerencia as mudanças do sistema.
- h) **Gerenciamento de projeto:** Esse workflow de apoio gerencia o desenvolvimento do sistema.
- i) **Meio ambiente:** Esse workflow está relacionado com a disponibilização de ferramentas apropriadas para a equipe de desenvolvimento de software.

Segundo Sommerville (2011, p.34) “no RUP, a iteração é apoiada de duas maneiras. Cada fase pode ser executada de forma iterativa com os resultados desenvolvidos de forma incremental”.

O Sistema de Gerenciamento de Notificação (SGN) baseou-se no IRUP, por ser um processo bem definido, com características que permite uma adaptação a diversos tipos de projetos e possui as melhores práticas de desenvolvimento de software. Utiliza-

se as disciplinas de requisitos, análise, design, implementação e teste juntamente com as fases de concepção, elaboração e construção.

2.3 Unified Modeling Language (UML)

De acordo com Guedes (2011, p.19) “UML é uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem adotada internacionalmente pela indústria de engenharia de software”.

Segundo Fowler (2006, p.25) “UML é uma família de notações gráficas, apoiada por um mega modelo único, que ajuda na descrição e no projeto de software particularmente daqueles construídos utilizando o orientado a objetos”.

Para Guedes (2011) sobre modelar um software:

Por mais simples que seja, todo e qualquer sistema deve ser modelado antes de se iniciar sua implementação, entre outras coisas, porque os sistemas de informação frequentemente costumam ter a propriedade de “crescer”, isto é, aumentar em tamanho, complexidade e abrangência. Muitos profissionais costumam afirmar que sistemas de informação são “vivos”, porque nunca estão completamente finalizados. Na verdade, o termo correto seria “dinâmicos”, pois normalmente os sistemas de informação estão em constante mudança. (GUEDES, 2011, p.20).

Segundo Guedes (2011) sobre diagramas da UML:

Cada diagrama da UML analisa o sistema, ou parte dele, sob uma determinada óptica. É como se o sistema fosse modelado em camadas, sendo que alguns diagramas enfocam o sistema de forma mais geral, apresentando uma visão externa do sistema, como é o objetivo do Diagrama de Casos de Uso, enquanto outros oferecem uma visão de uma camada mais profunda do software, apresentando um enfoque mais técnico ou ainda visualizando apenas uma característica específica do sistema ou um

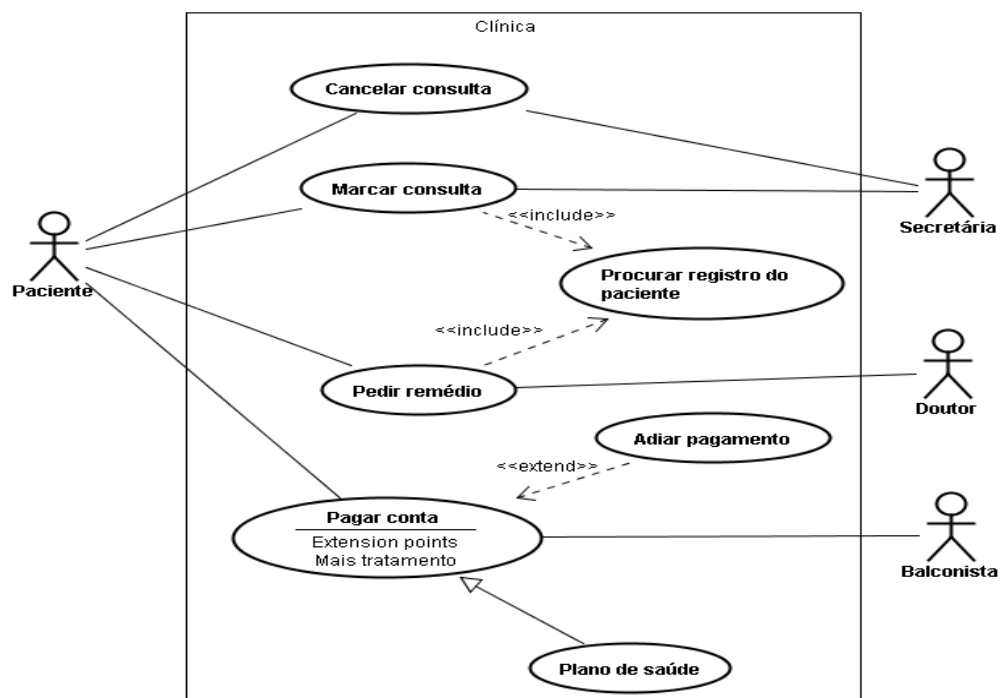
determinado processo. A utilização de diversos diagramas permite que falhas sejam descobertas, diminuindo a possibilidade da ocorrência de erros futuros.

2.3.1 Diagrama de Caso de Uso

O diagrama de casos de uso é o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas (Guedes, 2011, p.30).

A figura 3 é um exemplo de diagrama de caso de uso:

Figura 3 Diagrama de caso de uso



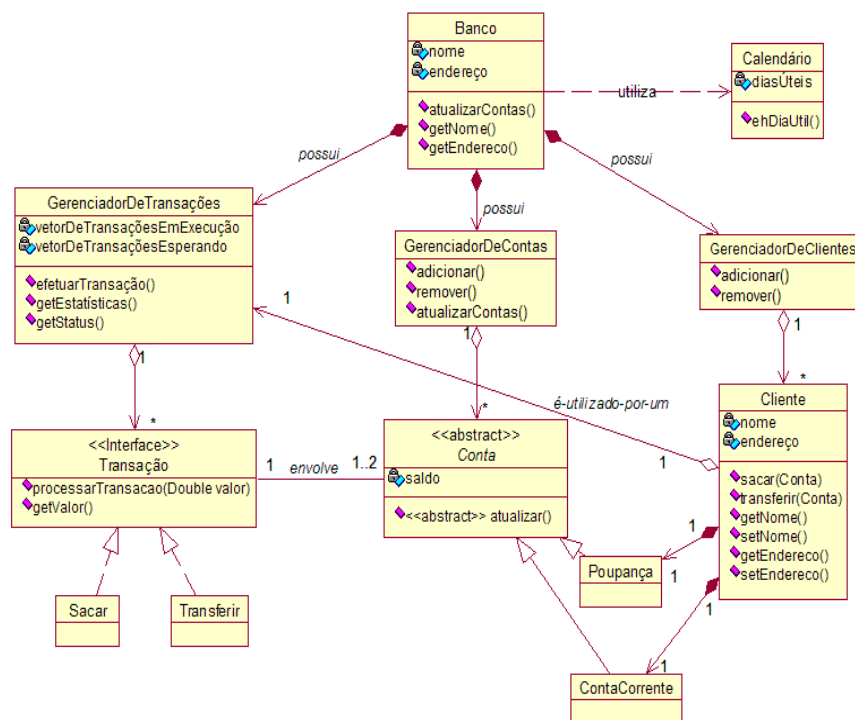
Fonte: DSC.UFCG,2015

2.3.2 Diagrama de Classes

De acordo com Guedes (2011, p.31) o diagrama de classes é provavelmente o mais utilizado e é um dos mais importantes da UML. Serve de apoio para a maioria dos demais diagramas. Como o próprio nome diz, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si.

A figura 4 apresenta um exemplo do diagrama de classes:

Figura 4 Diagrama de Classes



Fonte: DSC.UFCG, 2015

2.3.3 Diagrama de Objetos

Segundo Guedes (2011, p.32) o diagrama de objetos está amplamente associado ao diagrama de classes. Na verdade, o diagrama de objetos é praticamente um complemento do diagrama de classes e bastante dependente deste. O diagrama fornece

uma visão dos valores armazenados pelos objetos de um diagrama de classes em um determinado momento da execução de um processo do software.

2.3.4 Diagrama de Pacotes

De acordo com Guedes (2011, p.33) o diagrama de pacotes é um diagrama estrutural que tem por objetivo representar os subsistemas ou submódulos englobados por um sistema de forma a determinar as partes que o compõem. Pode ser utilizado de maneira independente ou associado com outros diagramas. Esse diagrama pode ser utilizado também para auxiliar a demonstrar a arquitetura de uma linguagem, como ocorre com a própria UML ou ainda para definir as camadas de um software ou de um processo de desenvolvimento.

2.3.5 Diagrama de Sequência

Para Guedes (2011, p.33) o diagrama de sequência é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso definido pelo diagrama de mesmo nome e apoia-se no diagrama de classes para determinar os objetos das classes envolvidas em um processo. Um diagrama de sequência costuma identificar o evento gerador do processo modelado, bem como o ator responsável por esse evento, e determina como o processo deve se desenrolar e ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos.

2.3.6 Diagrama de Comunicação

Segundo Guedes (2011, p.35) o diagrama de comunicação está amplamente associado ao diagrama de sequência: na verdade, um complementa o outro. As informações mostradas no diagrama de comunicação com frequência são praticamente as mesmas apresentadas no de sequência, porém com um enfoque distinto, visto que esse diagrama não se preocupa com a temporalidade do processo, concentrando-

se em como os elementos do diagrama estão vinculados e quais mensagens trocam entre si durante o processo.

2.3.7 Diagrama de Máquina de Estado

De acordo com Guedes (2011, p.35) diagrama de máquina de estados demonstra o comportamento de um elemento por meio de um conjunto finito de transições de estado, ou seja, uma máquina de estados. Além de poder ser utilizado para expressar o comportamento de uma parte do sistema, quando é chamado de máquina de estado comportamental, também pode ser usado para expressar o protocolo de uso de parte de um sistema, quando identifica uma máquina de estado de protocolo.

2.3.8 Diagrama de Atividade

Segundo Guedes (2001, p.36) o diagrama de atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. O diagrama de atividade concentra-se na representação do fluxo de controle de uma atividade.

2.3.9 Diagrama de Visão Geral de Interação

Para Guedes (2011, p.37) diagrama de visão geral de interação é uma variação do diagrama de atividade que fornece uma visão geral dentro de um sistema ou processo de negócio.

2.3.10 Diagrama de Componentes

De acordo com Guedes (2011, p.38) o diagrama de componentes está amplamente associado à linguagem de programação que será utilizada para desenvolver o sistema modelado. Esse diagrama representa os componentes do sistema quando o mesmo for ser implementado em termos de módulos de código-fonte, bibliotecas,

formulários, arquivos de ajuda, módulos executáveis etc. e determina como tais componentes estão estruturados e irão interagir para que o sistema funcione de maneira adequada.

2.3.11 Diagrama de Implementação

Segundo Guedes (2011, p.39) o diagrama de implantação determina as necessidades de hardware do sistema, as características físicas como servidores, estações, topologias e protocolos de comunicação, ou seja, todo o aparato físico sobre o qual o sistema deverá ser executado. Esse diagrama permite demonstrar também como se dará a distribuição dos módulos do sistema, em situações em que estes forem ser executados em mais de um servidor.

O Sistema de Gerenciamento de Notificação (SGN) optou por usar o modelo UML por ele facilitar a utilização das informações para o desenvolvedor tornando assim claro o que o sistema irá utilizar para a implementação, criando diagramas para especificar cada função do sistema.

2.4 Padrões de Projeto

Segundo Stephen (2008) “padrão de projeto é uma solução para um problema de projeto genérico na forma de um conjunto de classes que interagem e devem ser personalizadas para criar o projeto específico.”

2.4.1 Model View Control (MVC)

De acordo com Gamma (2000, p.20) MVC é usada para construir interfaces para usuários *Smalltalk-80*. O modelo é objeto de aplicação, a vista é apresentação na tela e controlador define a maneira como a interface do usuário reage a entrada do mesmo. Antes do MVC os projetos de interface para usuário tendiam a agrupar esses objetos. MVC separa esses objetos para aumentar a flexibilidade e a reutilização. O MVC

também pode mudar a maneira como uma vista responde às entradas do usuário sem mudar sua apresentação visual.

Outra característica do MVC é que as vistas podem ser encaixadas. MVC suporta vista encaixadas com a classe *CompositeView*, uma subclasse de *View*.(GAMMA, 2000, p.21)

MVC também permite mudar a maneira como uma vista responde às entradas do usuário sem mudar sua apresentação visual. MVC encapsula o mecanismo de resposta em um objeto Controlador.(GAMMA, 2000, p.22)

A utilização do MVC no Sistema de Gerenciamento de Notificação (SGN) justifica-se pelo fato de ser um padrão de projeto, que torna fácil a manutenção de sua aplicação, com pacotes modulares de rápido desenvolvimento.

2.5 World Wide Web Consortium (W3C)

No SNG foi utilizado o W3C que são padrões da Web as quais são destinadas a orientar os desenvolvedores para o uso de boas práticas que torna web acessível para todos. Por se tratar de um sistema voltado para *internet* os padrões do WC3 são essenciais para o bom funcionamento do sistema, para tal é utilizado à Linguagem de Marcação de Hipertexto – HTML5 e Sistema de Estilo em Cascata – CSS3.

O Consórcio *World Wide Web* (W3C) é um consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a Web. (Disponível em: <<http://www.w3c.br/Sobre/>>. Acesso em 24 de Novembro de 2016).

Liderado pelo inventor da Web Tim Berners-Lee e pelo CEO Jeffrey Jaffe, o W3C tem como missão conduzir a *World Wide Web* a atingir todo seu potencial, desenvolvendo protocolos e diretrizes que garantam seu crescimento de longo prazo.

(Disponível em: <<http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>>. Acesso em 24 de Novembro de 2016).

O W3C desenvolve especificações técnicas e orientações através de um processo projetado para maximizar a consenso sobre as recomendações, garantindo qualidades técnicas e editoriais, sendo assim disponibilizando diversos modos de visualização de suas recomendações:

- a) Por tecnologia (como todas as especificações relacionadas a HTML, CSS, JAVASCRIPT).
- b) Por status (ordenada inicialmente pelas mais recentes e, em seguida, agrupadas por status, das que já são Recomendação até as que ainda estão no status de Rascunho)
- c) Por data (a partir das mais recentes)
- d) Por título
- e) Por grupo (os grupos que trabalham nas especificações)

Disponível em: <<http://www.w3c.br/Padroes/>>. Acesso em 24 de Novembro de 2016.

No ambiente de *software*, os sistemas são as ferramentas utilizadas para a execução de tarefas pelo usuário e encontramos na usabilidade o momento do diálogo entre o usuário e a interface do software.

De acordo com Walter Cybis(2010, p.16) Usabilidade é a qualidade que caracteriza o uso dos programas e aplicações. Assim, ela não é qualidade intrínseca de um sistema, mas depende de um acordo entre as características de sua interface e as características de seus usuários ao buscarem determinados objetos em determinadas

situações de seus usuários ao buscarem determinados objetos em determinadas situações de uso.

A usabilidade é capacidade do sistema em fazer com que o usuário tenha sucesso na execução de suas tarefas. Fácil aprendizagem, utilização eficiente e gestão de erros são pontos fundamentais para que o usuário perceba a boa usabilidade.

Segundo a Cartilha Acessibilidade na Web W3C Brasil, existem muitas definições de acessibilidade.

O Decreto Federal nº 5.296/2004 [1], em seu artigo 8º, I, estabelece:

“I – acessibilidade: condição para utilização, com segurança e autonomia, total ou assistida, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação, por pessoa portadora de deficiência ou com mobilidade reduzida;”

A Convenção Internacional Sobre os Direitos das Pessoas com Deficiência, adotada pela Organização das Nações Unidas (ONU) em 30 de março de 2007, em nova York, e ratificada pelo Decreto Federal nº 6.949 de 25 de agosto de 2009 [2], estabelece em seu artigo 9º, item 1:

“A fim de possibilitar às pessoas com deficiência viver com autonomia e participar plenamente de todos os aspectos da vida, os Estados Partes deverão tomar as medidas apropriadas para assegurar-lhes o acesso, em igualdade de oportunidades com as demais pessoas, ao meio físico, ao transporte, à informação e comunicação, inclusive aos sistemas e tecnologias da informação e comunicação, bem como a outros serviços e instalações abertos ou propiciados ao público, tanto na zona urbana como na rural”.

A norma Brasileira ABNT NBR 9050:2004 [3] define em seu item 3.1:

“Acessibilidade: Possibilidade e condição de alcance, percepção e entendimento para a utilização com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e elementos.”

De acordo com a Cartilha Acessibilidade na Web W3C Brasil.

”Acessibilidade na *web* significa que pessoas com deficiência podem usar a *web*. Mais especificamente, a acessibilidade na *web* significa que pessoas com deficiência podem perceber, entender, navegar, interagir e contribuir para a *web*. E mais. Ela também beneficia outras pessoas, incluindo pessoas idosas com capacidades em mudança devido ao envelhecimento”

Se for aplicada a definição geral de acessibilidade ao ambiente específico da *web*, pode-se dizer que se trata da possibilidade e da condição de alcance, percepção e entendimento para a utilização, em igualdade de oportunidades, com segurança e autonomia, dos sítios e serviços disponíveis na *web*. Porém, para abarcar a complexidade do conceito de acessibilidade na *web*, existem alguns aspectos específicos que precisam ser considerados, segundo a Cartilha Acessibilidade na Web W3C Brasil.

- A. A importância, a abrangência e a universalidade da Web.
- B. Reciprocidade.
- C. A multiplicidade e a diversidade de Fatores envolvidos, a W3C identifica sete componentes conteúdo, navegadores, tecnologia assistiva, o conhecimento do usuário, desenvolvedores, ferramentas de autoria e ferramenta de avaliação.

2.6 Banco de Dados

Para Date (2003, p.10) “Um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa”.

Um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros; em outras palavras, é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar.(DATE, 2003, p.6).

De acordo com Date(2003) O sistema de gerenciamento de banco de dados (SGBD) é o software que trata de todo o acesso ao banco de dados.

As principais funções do SGBD são:

- a) Definição de dados: o SGBD deve ser capaz de aceitar definições de dados em formato fonte e convertê-los para o formato objeto apropriado.
- b) Manipulação de dados: o SGBD deve ser capaz de lidar com requisições do usuário para buscar, atualizar ou excluir dados existentes no banco de dados, ou para acrescentar novos dados ao banco de dados.
- c) Otimização e execução: as requisições de DML, planejadas e não planejadas, devem ser processadas pelo componente otimizado, cuja finalidade é determinar um modo eficiente de implementar a requisição.
- d) Segurança e integridade de dados: o SGBD, ou algum subsistema chamado pelo SGBD, deve monitorar requisições de usuários e rejeitar






toda tentativa de violar as restrições de segurança e integridade definidas pelo DBA.

- e) Recuperação de dados e concorrência: o SGBD ou ,mais provavelmente, algum outro componente de software relacionado, em geral chamado gerenciador de transações ou monitor de TP(transaction processing) deve impor certos controles de recuperação e concorrência.
- f) Dicionário de dados: o SGBD deve fornecer uma função de dicionário de dados. O dicionário de dados pode ser considerado um banco de dados isolado.
- g) Desempenho: é necessário com o SGBD realize todas as funções identificadas anteriormente de forma tão eficiente quanto possível.

No quadro 1 é possível ver o recorde dos sistemas de gerenciamento de banco de dados..

Quadro 1 Recorde de Sistemas de Gerenciamento de Banco de Dados.

121 systems in ranking, November 2016

Rank			DBMS	Database Model	Score		
Nov 2016	Oct 2016	Nov 2015			Nov 2016	Oct 2016	Nov 2015
1.	1.	1.	Oracle 	Relational DBMS	1413.01	-4.09	-67.94
2.	2.	2.	MySQL 	Relational DBMS	1373.56	+10.91	+86.71
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1213.80	-0.38	+91.48
4.	4.	4.	PostgreSQL	Relational DBMS	325.82	+7.12	+40.13
5.	5.	5.	DB2	Relational DBMS	181.46	+0.90	-21.07
6.	6.	6.	Microsoft Access	Relational DBMS	125.97	+1.30	-14.99
7.	7.	7.	SQLite	Relational DBMS	112.00	+3.43	+8.55
8.	8.	 9.	Teradata	Relational DBMS	75.16	-1.07	-1.92
9.	9.	 8.	SAP Adaptive Server	Relational DBMS	70.16	+0.68	-13.55
10.	10.	 11.	FileMaker	Relational DBMS	53.92	-1.03	+2.19

Fonte:DB-Engines(2016).

Para Date (2004) “As instruções SQL podem incluir referências a variáveis hospedeiras; essas referências devem incluir um prefixo de dois-pontos para distingui-las dos nomes de colunas da SQL”.

A SQL é uma linguagem para banco de dados, suas sintaxes utilizam formas de manipular os dados tais como buscar, alterar e excluir dados de determinado banco de dados da forma que for necessário para o sistema ou usuário que estiver utilizando (DATE, 2004).

2.7 Linguagem de Programação

Segundo Prates (2015, p.88) A orientação a objetos é um paradigma que representa uma filosofia para construção de sistemas. Em vez de construir um sistema formado por um conjunto de procedimentos e variáveis nem sempre agrupadas de acordo com o contexto, como se fazia em linguagens estruturadas (Cobol, Clipper, Pascal), na orientação a objetos utilizamos uma ótica mais próxima do mundo real. Lidamos com objetos: estruturas que carregam dados e comportamento próprio, além de trocarem mensagens entre si com o objetivo de formar algo maior, um sistema.

De acordo com Soares (2013, p.28). PHP é acrônimo de *Hypertext Preprocessor* (pré-processador de Hipertexto), uma poderosa linguagem de programação *open source*, mundialmente utilizada, principalmente no ambiente web (apesar de existir a versão PHP-GTK para ambiente desktop).

Segundo Soares (2013, p.28) Uma das características mais marcantes no PHP é sua capacidade de se misturar ao HTML, tornando mais fácil a geração de páginas web dinâmicas.

O índice *TIOBE Programming Community* é um indicador da popularidade de linguagens de programação. O índice é atualizado uma vez por mês. As classificações são baseadas no número de engenheiros qualificados a nível mundial, cursos e fornecedores de terceiros. Motores de busca populares, como *Google*, *Bing*, *Yahoo* !,

Wikipedia, Amazon, YouTube e Baidu são usados para calcular as classificações. É importante notar que o índice TIOBE não é sobre a melhor linguagem de programação ou a língua em que a maioria das linhas de código foram escritas. Disponível em: <<http://www.tiobe.com/tiobe-index/>> Acesso em 26 de Novembro de 2016.

O índice pode ser utilizado para verificar se suas habilidades de programação ainda estão atualizados ou para tomar uma decisão estratégica sobre o linguagem de programação deve ser adotada quando começar a construir um novo sistema de software.

Quadro 2, ranking completo do Índice Tiobe

Jan 2016	Jan 2015	Mudança	Linguagem	%	Alteração
1	2	▲	Java	21.465%	+5.94%
2	1	▼	C	16.036%	-0.67%
3	4	▲	C++	6.914%	+0.21%
4	5	▲	C#	4.707%	-0.34%
5	8	▲	Python	3.854%	+1.24%
6	6		PHP	2.706%	-1.08%
7	16	▲▲	Visual Basic .NET	2.582%	+1.51%
8	7	▼	JavaScript	2.565%	-0.71%
9	14	▲▲	Assembly language	2.095%	+0.92%
10	15	▲▲	Ruby	2.047%	+0.92%
11	9	▼	Perl	1.841%	-0.42%
12	20	▲▲	Delphi/Object Pascal	1.786%	+0.95%
13	17	▲▲	Visual Basic	1.684%	+0.61%
14	25	▲▲	Swift	1.363%	+0.62%
15	11	▼▼	MATLAB	1.228%	-0.16%
16	30	▲▲	Pascal	1.194%	+0.52%
17	82	▲▲	Groovy	1.182%	+1.07%
18	3	▼▼	Objective-C	1.074%	-5.88%
19	18	▼	R	1.054%	+0.01%
20	10	▼▼	PL/SQL	1.016%	-1.00%

Fonte: www.tiobe.com

2.8 Servidor Web

Servidor Web é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que os sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

(Disponível em: <<http://www.soawebservices.com.br/como-funciona.aspx>>. Acesso em 25 de Novembro de 2016).

Os Serviços Web são componentes que permitem às aplicações enviar e receber dados em formato *Extensible Markup Language* (XML). Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, o formato XML. (Disponível em: <<http://www.soawebservices.com.br/como-funciona.aspx>>. Acesso em 25 de Novembro de 2016).

2.9 Requisitos de Segurança de Software

Segundo Sommerville (2011, p.57) Os requisitos de um sistema são descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para o sistema que serve a finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações.

O termo 'requisito' não é usado de forma consistente pela indústria de software. Davis (1993) explica por quê.

Se uma empresa pretende fechar um contrato para um projeto de desenvolvimento de software de grande porte, deve definir as necessidades de forma abstrata o suficiente para que a solução para essas necessidades não seja predefinidas. Os requisitos precisam ser escritos de modo que vários contratantes possam concorrer pelo contrato e oferecer diferentes maneiras de atender às necessidades da organização do cliente. Uma vez que o contrato tenha sido adjudicado, o contratante deve escrever para o cliente uma definição mais detalhada do sistema, para que este entenda e

possa validar o que o software fará. Ambos os documentos podem ser chamados documentos de requisitos para o sistema.

De acordo com Sommerville (2011, p.59) Os requisitos de software são frequentemente classificados como requisitos funcionais e requisitos não funcionais:

- a) Requisitos funcionais. São declarações de serviço que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar que o sistema não deve fazer.
- b) Requisitos não funcionais. São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como todo.

De acordo com Pádua (2000, p. 14) Os requisitos podem ser dos seguintes tipos:

- a) Os requisitos explícitos são aqueles descritos em um documento que arrola os requisitos de um produto, ou seja, um documento de especificação de requisitos.
- b) Os requisitos normativos são aqueles que decorrem de leis, regulamentos, padrões e outros tipos de normas a que o tipo de produto deve obedecer.
- c) Os requisitos implícitos são expectativas dos clientes e usuários, que são cobradas por estes, embora não documentadas.

2.10 Ferramentas *Case*

< conceito, versão, justificativa de utilização, print da tela principal: Ferramentas de Modelagem de dados, Modelagem UML, IDE Desenvolvimento, Teste de Software>

2.10.1 IDE Desenvolvimento

2.10.2 Modelagem de dados

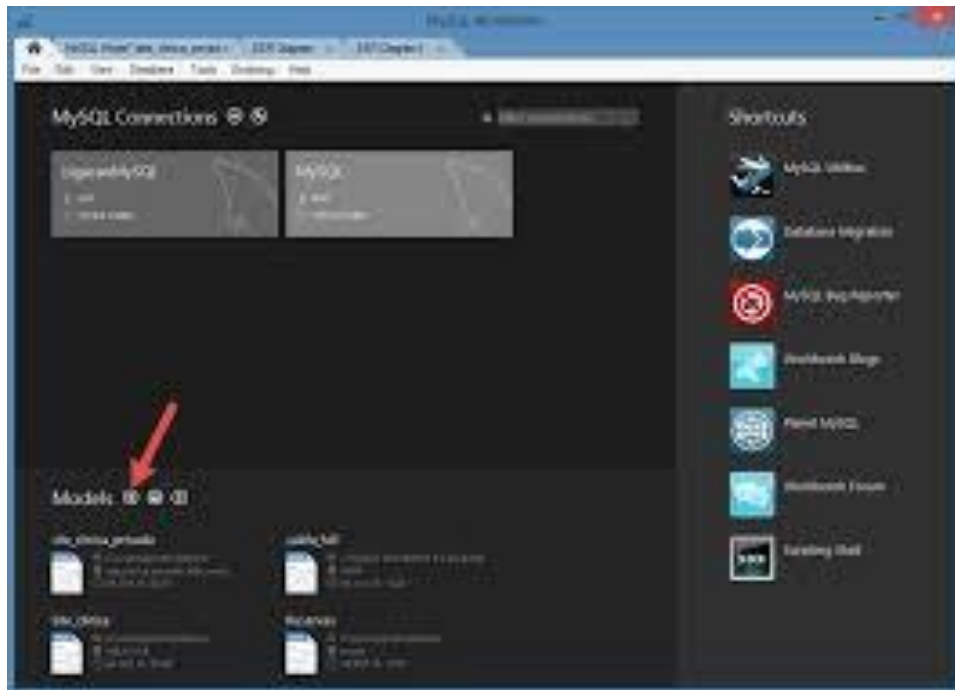
Modelagem de dados é o ato de explorar estruturas orientadas a dados, é conceitualmente similar à modelagem de classes. Com a modelagem de dados identificamos tipos de entidades da mesma forma que na modelagem de classes identificamos classes.

No SGN está sendo utilizado o app MySQL Workbench versão 6.3, porque trata-se de uma ferramenta de base de dados aberto que permite criar diagrama de relação entidade para as bases de dados MySQL, assim produzindo uma representação visual da tabela de dados.

MySQL Workbench é uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e DBAs. MySQL Workbench fornece modelagem de dados, desenvolvimento de SQL, e ferramentas de administração de abrangente para configuração de servidor, administração de usuários, backup e muito mais. MySQL Workbench está disponível em Windows, Linux e Mac OS X.

(Disponível em: <<http://www.mysql.com/products/workbench/>>. Acesso em 26 de Novembro de 2016).

Na figura 5 mostra a tela principal do MySQL Workbench .



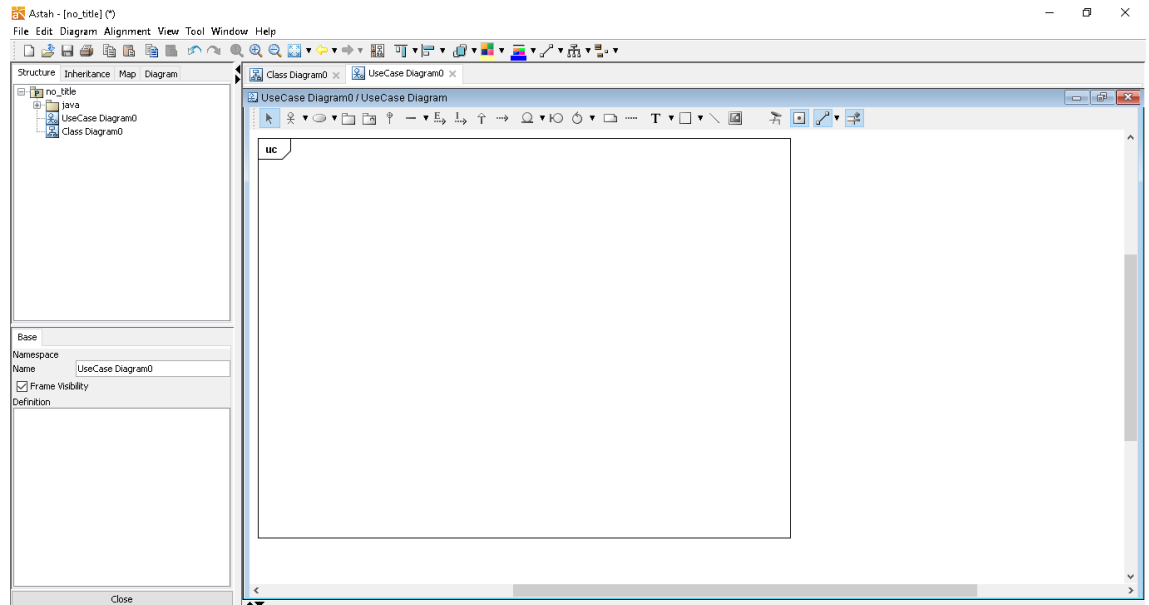
Fonte: <http://www.mysql.com>

2.10.3 Modelagem UML

UML é uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem adotada internacionalmente pela indústria de engenharia de software.

No SGN utiliza-se o *Astah Community 6.6.4* por se tratar de uma ferramenta gratuita, de fácil usabilidade e com todas as funções que este projeto necessita.

Na figura 6, mostra a tela principal do *Astah Community*



2.10.3 Gerenciamento de Projeto

De acordo com a norma ISO 10006 (Diretrizes para Qualidade de Gerenciamento de Projetos).

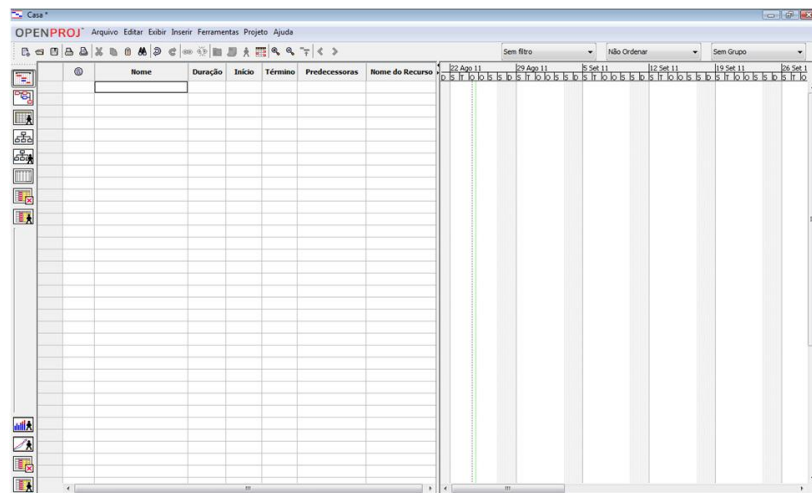
Projeto é um processo único, consistindo de um grupo de atividades coordenadas e controladas com data para início e término, que é a chave para se determinar se realmente estamos em um projeto. Se você estiver empenhando forças para realizar ou desenvolver um produto ou serviço e não possui data de início e fim, é provável que você não esteja em um projeto, sendo assim seu gerenciamento pelo *Project Management Body of Knowledge* (PMBOK) fica comprometido.

No SGN foi utilizado o OpenProj versão 1.4, porque analisar as atividades necessárias e criar um plano detalhado que mostra duração das tarefas numa linha de tempo e designar prioridades as entregas definidas no escopo do projeto.

O OpenProj é uma aplicação gratuita e open source (código-fonte aberto) que o pode ajudar a gerir os projetos entre outras coisas, que é absolutamente conhecido pela sua boa usabilidade, e acima de tudo o seu preço, totalmente gratuito.

(Disponível em: <<https://openproj.br.uptodown.com/windows>>. Acesso em 26 de Novembro de 2016).

Na figura 7, demonstra a tela principal do OpenProj.



Fonte: <https://openproj.br>

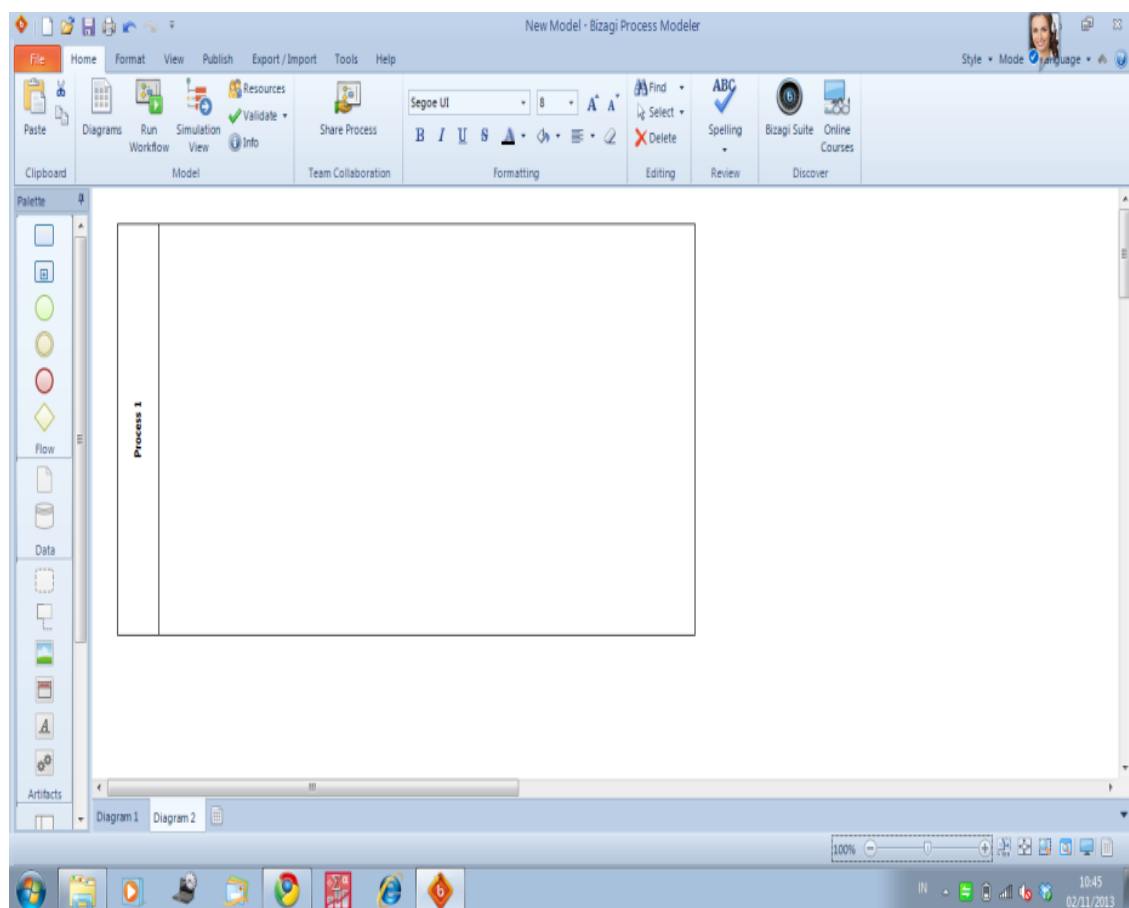
2.10.4 Modelagens de Processos

Para que haja uma melhor compreensão daquilo que se quer desenvolver, uma das medidas adotadas no processo de desenvolvimento de software é a criação de modelos (BOOCH; RUMBAUGH; JACOBSON, 2005). Os modelos de sistemas, buscam representar relação aos requisitos do mesmo. Em outras palavras, representam a estrutura do software a ser desenvolvido e suas funcionalidades. Os modelos de sistemas são criados com base nos requisitos de um sistema e podem ter diferentes níveis de abstração, dependendo da necessidade. (Casati, 2016 p.16).

O Bizagi Modeler foi utilizado no SGN, porque é específico para modelar processos departamentais.

Bizagi Modeler é uma ferramenta de modelagem e documentação de processos de negócios. O Modeler permite que você visualmente diagrama, modelo e empresariais documento processos em padrão da indústria *Business Process Notation Model* (BPMN). BPMN é um formato em todo o mundo aceite para modelagem de processos. Disponível em: <<http://help.bizagi.com/process-modeler/en/>>. Acesso em 26 de Novembro de 2016.

Na figura 8, mostra a tela principal do Bizagi.



2.10.5 Testes Automatizados

3 METODOLOGIA

Do ponto de vista da sua natureza o presente trabalho é uma pesquisa aplicada que objetiva gerar conhecimentos para aplicação prática e dirigida à solução de problemas específicos.

Do ponto de vista da forma de abordagem do problema trata-se de uma pesquisa quantitativa e do ponto de vista de seus objetivos é uma pesquisa exploratória.

Do ponto de vista dos procedimentos técnicos trata-se de um estudo de caso, pois, envolve o estudo que se permita o seu amplo e detalhado conhecimento.

Segundo Ponte (2006) o estudo de caso é:

É uma investigação que se assume como particularística, isto é, que se debruça deliberadamente sobre uma situação específica que se supõe ser única ou especial, pelo menos em certos aspectos, procurando descobrir a que há nela de mais essencial e característico e, desse modo, contribuir para a compreensão global de um certo fenômeno de interesse. (PONTE, 2006)

Sendo assim, constam na metodologia deste projeto a concepção do sistema, a elaboração do sistema, a construção do sistema e a transição do sistema. <texto complementar>

3.1 Concepção

Nesta fase foi estabelecido o escopo do projeto de *software* e as condições limite, incluindo uma visão operacional, critérios de aceitação e o que deve ou não estar no Sistema XYZ.

3.1.1 Modelagem de Negócio

Para a modelagem de negócio foi elaborado o Documento de Visão. Este documento é importante, pois tem a finalidade de coletar, analisar e definir as necessidades de alto nível e os recursos do SGN. No documento de visão é feito a descrição do problema e uma solução para o mesmo.

Neste documento também é estabelecido os envolvidos no projeto do SGN. É apresentado visão geral do produto e os requisitos funcionais e não funcionais. Através deste artefato é possível ter uma visão clara sobre o negócio que o sistema está envolvido. Todos os detalhes do Documento de Visão encontram-se no Apêndice XXX.

3.1.2 Requisitos

Durante esta etapa o analista de sistema entrevista o usuário para entender e documentar seus requisitos. O resultado dessa etapa é uma entrevista com um conjunto de perguntas e respostas escritas de forma concisa, especificado da forma mais detalhada e completa possível.

Para facilitar a comunicação entre o analista e o cliente e feito o diagrama de caso de uso, que descreve um cenário que mostra as funcionalidade do SGN do ponto de vista do usuário.

Como requisito de referência ao desenvolvedor do SGN, a especificação de caso de uso e um documento de software que é claro, consistente e completo.

3.1.3 Gerenciamento do Projeto

<análise de riscos e plano de contingência, plano de cronograma do projeto e estimativa de custo>

3.2 Elaboração

3.2.1 Análise e Projeto

<diagrama de atividades, diagrama de sequência, diagrama de classes, projeto de banco de dados (MER e dicionário de dados)>

3.3 Construção

3.3.1 Implementação

<programação orientada a objetos, diagrama de implantação>

3.3.2 Teste

<teste funcional, teste de carga, teste de segurança>

3.4 Transição

3.4.1 Implantação

<diagrama de implantação, plano de implantação, manual de implantação>

4. CONCLUSÃO

[A conclusão, última etapa do TCC, deve retomar o que foi feito ao longo do trabalho de forma resumida e mostrar que objetivo geral foi atingido. Se ao longo do trabalho foi exposta uma situação-problema, esse é o momento de apresentá-la novamente e respondê-la. Após discorrer sobre os pontos mais relevantes, exponha as limitações do trabalho, as dificuldades encontradas para realizá-lo, e dê sugestões de pesquisas futuras que venham a contribuir para aquele mercado.]

Como a conclusão deve ser escrita com as palavras do aluno, não é comum usar citações de autores. Além disso, lembre-se que, assim como em todo o trabalho, não se deve escrever em primeira pessoa – concluí, acredito, etc. –, e sim de forma impessoal – conclui-se, acredita-se, etc. A conclusão é, portanto, um texto breve, objetivo e claro, contendo no máximo duas páginas, e que não apresentará novas descobertas, apenas fará o fechamento das ideias.]

REFERÊNCIAS BIBLIOGRÁFICAS

PÁDUA, Wilson. **Engenharia de Software: fundamentos, métodos e padrões**, Editora LTC 2000.

SOARES, Wallace. **PHP 5**, Editora Érica, Sétima Edição, 2013.

CASATI, João Paulo. **Modelagem de Sistemas**. Editora Estácio. Primeira Edição SESES 2016.

PONTE, João Pedro (2006). **Estudos de caso em educação matemática**. Bolema, 25, 105-132. Este artigo é uma versão revista e atualizada de um artigo anterior: Ponte, J. P. (1994). O estudo de caso na investigação em educação matemática. Quadrante, 3(1), pp3-18. (re-publicado com autorização).

PRATES, Rubens. **PHP Programando com Orientação a Objeto**, Editora Novatec, Terceira Edição 2015.

NBR ISO 10006, *Quality management - Guidelines to quality in project management* (Gestão da qualidade - Diretrizes para a qualidade no gerenciamento de Projetos) Descriptors: Quality assurance. Quality assurance systems. Quality management. Project Esta Norma é equivalente à ISO 10006:1997(E) Válida a partir de 29.01.2001.

CYBIS, Walter, BETIOL, Adriana Holtz, FAUST, Richard. **Ergonomia e Usabilidade**, Editora Novatec, Segunda Edição 2010;

POWER, Martin, Cris. KOBRYN. UML Essencial: **Um Breve guia para a linguagem padrão de modelagem de objetos**. Editora Bookman, Terceira Edição 2006;

SCHACH, Stephen R, **Engenharia de Software Os Paradigmas Clássico Orientado a Objeto**, Editora Mc Graw Hill, Sétima Edição, 2008;

DATE, C J., **Introdução a sistemas de bancos de dados**; Tradução de Daniel Vieira. Elsevier, 2003 - 19ª reimpressão.

GAMMA Erich, HELM Richard, JOHNSON Ralph, VLISSIDES John. Padrões de Projeto: **Soluções Reutilizáveis de Software Orientado a Objeto**, Editora Bookman Primeira Edição 2000;

GUEDES, Gilleanes T.A. **UML 2 Uma Abordagem Prática**, Novatec Editora, Segunda Edição, 2011;

PRESSMAN, Roger S. Engenharia de Software - **Uma Abordagem Profissional**. Editora Artmed, Sétima Edição, 2011;

SOMMERVILLE, Ian. **Engenharia de Software**. Editora Pearson Education do Brasil, Nona Edição 2011;

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

Portal uptodwn – Conceito de OpenProj. Disponível em:

<<https://openproj.br.uptodown.com/windows>>. Acesso em 26 de Novembro de 2016.

Portal W3C Brasil – Cartilha de Acessibilidade na web W3C Brasil. Disponível em:

<<http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>>. Acesso em 24 de Novembro de 2016.

Portal W3C Brasil – Padrões do W3C. Disponível em:

<<http://www.w3c.br/Padroes/>>. Acesso em 24 de Novembro de 2016.

Portal MySQL – MySQL Workbench. Disponível em:

<<http://www.mysql.com/products/workbench>>. Acesso em 26 de Novembro de 2016.

Portal TIOBE - Tiobe. Disponível em:

<<http://www.tiobe.com/tiobe-index/>>. Acesso em 26 de Novembro de 2016.

Portal W3C Brasil – Conceito de W3C. Disponível em:

< <http://www.w3c.br/Sobre/> >. Acesso em 24 de Novembro de 2016.

Portal Bizagi – Conceito de Bizagi. Disponível em:

<<http://help.bizagi.com/process-modeler/en/>>. Acesso em 26 de Novembro de 2016.

APÊNDICES A - ENTREVISTA

Data: 01 de Setembro de 2016

Entrevistado: Gestor de Notificações

Entrevistador: Analista de Requisito

1)Qual o objetivo do desenvolvimento deste software?

O objetivo deste software é que os funcionários possam ser notificados com antecedência sobre suas atividades importantes.

2)Quem pode acessar o Sistema de Gerenciamento de Notificações?

O gestor, administrador e funcionários.

3)Quais os principais pontos deste projeto de software?

Gerenciar as notificações e a comunicação com o funcionário.

4)Qual o nível de conhecimento dos funcionários em relação a computação?

Mexemos com o trivial então, quanto mais fácil melhor.

5)Quem pode modificar/acrescentar/apagar dados?

O gestor, administrador e funcionário.

6)Como o Funcionário cadastrado terá acesso?

Através de usuário e senha.

7)Dados obrigatórios ao cadastrar o funcionário?

Nome completo, CPF, data de nascimento, endereço, telefone, função.

8)Dados obrigatórios ao cadastrar o Gestor e Administrador?

Nome completo, CPF, data de nascimento, endereço, telefone, função.

9)Existe algo específico que você deseje que o sistema faça?

Sim, que gere um relatório das notificações enviadas aos funcionários.

10)Com que frequência você gostaria de atualizações?

Atualizar no mínimo de seis em seis meses.

12)Deve ser fácil para o usuário descobrir como desativar ou excluir o Sistema de Gerenciamento de Notificação?

Não, Somente o Gestor pode desativar ou excluir o software.

13) Como você deseja que a interface seja?

Na primeira tela abra uma janela pedindo login e senha, na segunda se for Funcionário apareça atualização de dados e logo depois uma tela de notificações.

14)Depois da interface de login, se o usuário for o Gestor ou o Administrador como deseja que seja as demais interface?

Abra uma janela com as opções de manter funcionário, excluir funcionários, adicionar notificações, atualização de software, tire suas dúvidas.

15)O que se refere ao acrescentar este item (Tire suas dúvidas)?

Neste item vai conter explicações de como o software pode ser usado, seria como um mini manual.

APÊNDICES B - DOCUMENTO DE VISÃO

1. Introdução

A finalidade deste documento é coletar, analisar e definir necessidades e recursos de nível superior do Sistema de Gerenciamento de Notificação. Ele se concentra nos recursos necessários aos envolvidos e aos usuários-alvo e nas razões que levam a essas necessidades. Os detalhes de como o SGN Sistema de Gerenciamento de Notificação (SGN) satisfaz essas necessidades são descritos no caso de uso e nas especificações suplementares.

1.1 Referências

Não se aplica

2. Posicionamento

2.1 Descrição do Problema

O problema de	Dificuldade do funcionário em saber sua escala, férias, feriados, dias de pagamento, comunicação com a chefia, com facilidade e comodidade.
Afeta	Funcionários.
Cujo impacto é	Perda de tempo dos funcionários na realização de tarefas relativamente fáceis, de informações de funcionários sobre

	suas férias, escalas, dias de plantão e demais informações pessoais.
Uma boa solução seria	Criação de um sistema que cadastra os funcionários e suas informações e informe aos mesmos por e-mail os dias de serviço e afins e também conceda acesso aos mesmos para que eles possam consultar as devidas informações facilitando o serviço e poupando tempo.

2.2 Sentença de Posição do Produto

Para	Empresas que necessitam de notificar seus funcionários
Que	Necessita de um produto de software que possa facilitar a informação dos funcionários sobre seus plantões, férias, folgas, comunicação da chefia sem maiores dificuldades
O	Sistema de Gerenciamento de Notificação
Que	Irà guardar as informações de funcionários e facilitar a notificação e a consulta às suas informações como férias, escalas dentre outros com facilidade e comodidade.
Ao contrário de	Sistemas desenvolvidos voltado a esse tipo de usuário, que não provem as necessidades

	exigidas.
Nosso produto	Comodidade para os usuários, fácil de usar, sistema seguro.

3. Descrições dos Envolvidos e dos Usuários

3.1 Resumo dos Envolvidos

Nome	Descrição	Responsabilidades
Cliente	Usuário Final	Manter o Sistema.
Analistas de Sistemas	Soluções de problemas	Sistematizar informações.
Gerente de Projetos	Fornece o sistema para o alcance de futuros clientes	Gerência, planeja e acompanha o progresso das rotinas.
Administrador de dados-DBA	Administra o Banco de Dados	Desenvolvimento e manutenção do Banco de Dados.
Desenvolvedor	Codificar o Sistema	Desenvolvimento e manutenção do sistema.

Testador	Realiza os testes no sistema	Testar as funcionalidades do sistema.
----------	------------------------------	---------------------------------------

3.2 Resumo dos Usuários

Nome	Descrição	Responsabilidades
Administrador	Acesso total às funcionalidades do sistema	Manter Gestor, Gerar Relatórios.
Gestor	Acesso parcial às funcionalidades do sistema	Manter Funcionário, Gerenciar Notificações, Enviar Notificação
Funcionário	Acesso parcial às funcionalidades do sistema	Consulta Notificação, Receber Notificação.

3.3 Ambiente do Usuário

3.4 Principais Necessidades dos Usuários ou dos Envolvidos

Necessidade	Prioridade	Preocupações	Solução Atual	Soluções Proposta
-------------	------------	--------------	---------------	-------------------

Facilitar a consulta sobre dia de plantão, escalas, férias, feriados, dias de pagamento, comunicação com a chefia.	Média	Facilidade na utilização do programa e agilidade para realizar as tarefas nele.	Consulta feita pessoalmente	Fazer com que cada funcionário possa receber notificações em seu e-mail
--	-------	---	-----------------------------	---

3.5 Alternativas e Concorrência

Não se aplica

4. Visão Geral do Produto

4.1 Perspectiva do Produto

O objetivo do produto é suprir a necessidade do usuário que irá utilizá-lo, facilitando o recebimento de notificações, agilizando a comunicação com a chefia com comodidade.

4.2 Suposições e Dependências

Esperamos que o software tenha uma interface de fácil usabilidade para o cliente, que seja de grande ajuda e que supra as necessidades dos potenciais usuários, futuramente pretendemos incrementar o software com a tecnologia de envio de notificações por SMS sendo assim mais fácil para o cliente, já que esta versão só envia e-mail.

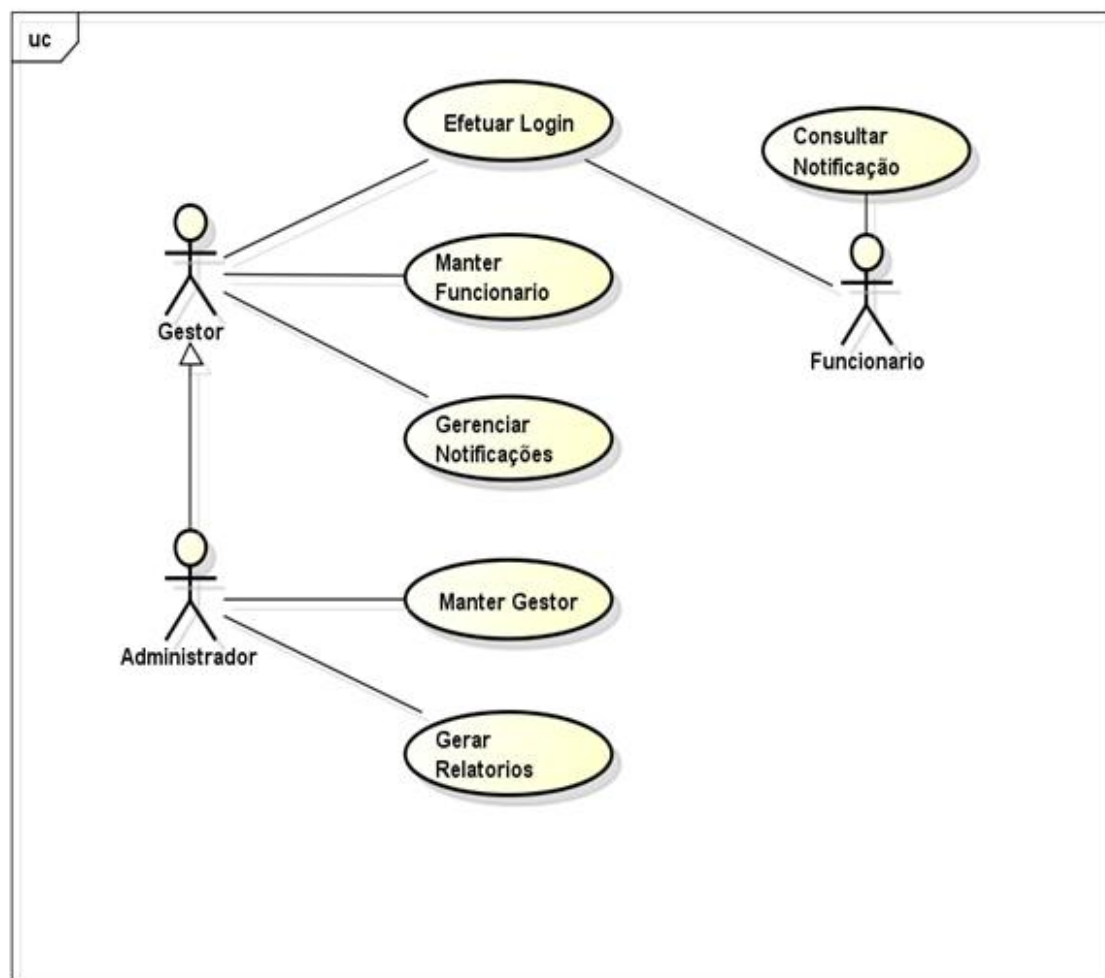
5. Recursos do Produto

- Efetuar Login – Usuário deverá efetuar o Login usando usuário e senha;
- Consultar Notificação - Esse caso de uso permite o usuário realizar consultas;
- Gerenciar Notificações – Esse caso de uso permite realizar consultas, adição, alteração, exclusão e envio de notificações;
- Manter Gestor - Esse caso de uso permite realizar consultas, adição, alteração e exclusão de Gestor;
- Gerar Relatórios – Esse caso de uso permite o administrador realizar consultas e gerar relatórios sobre a notificação feita ao funcionário.

6. Outros Requisitos do Produto

- O produto deverá oferecer resposta que não ultrapasse 10 segundos ao usuário;
- O processo de desenvolvimento de sistema e os documentos a serem entregues deverão estar de acordo com o processo e os produtos a serem entregues ao administrador do SGN;
- O produto fornece telas apropriadas para o administrador/gestor possa adicionar e visualizar dados sobre o funcionário;
- O Sistema deverá revelar ao administrador/gestor informações pessoais sobre os funcionários;
- Deve haver uma interface padrão com senha e usuário.

APÊNDICES C – DIAGRAMA E ESPECIFICAÇÃO DE CASO DE USO



Efetuar Login

1. Descrição

Este caso de uso tem por objetivo permitir acesso ao sistema e verificar se o ator já existe no Sistema de Gerenciamento de Notificação - SGN.

2. Referências

Caso de uso, Doc. Visão

3. Atores

3.1.Administrador

Responsável por administrar o sistema.

4. Fluxo de eventos

4.1.Fluxo principal

- P1 O caso de uso inicia quando o ator acessa o sistema;
- P2 O sistema apresenta a tela de login; (RA1), (RA2);
- P3 O ator informa o usuário e a senha e clicar em “Acessar”. (A1), (E1), (E2);
- P4 O sistema valida os dados;
- P5 O sistema apresenta a tela inicial;
- P6 O caso de uso se encerra.

4.2.Fluxos alternativos

A1 Recuperar senha

A1.1 O caso de uso inicia quando o ator acessa o sistema;

A1.2 O ator aciona a opção recuperar senha;

A1.3 O sistema pede o CPF, Usuário e Data de Nascimento. (RA1), (RA3), (RA4), (RN1), (E2);

A1.4 O ator informa os dados e clica em “Salvar”;

A1.5 O sistema valida os dados e cria uma nova senha;

A1.6 O sistema pede um e-mail para o envio da nova senha;

A1.7 O sistema apresenta a tela inicial;

A1.8 O caso de uso se encerra.

4.3.Fluxos de Exceção

E1 Usuário não encontrado

E1.1 O sistema apresenta uma mensagem. “Usuário ou senha incorretos”.

E2 Campos obrigatórios

E2.1 O sistema apresenta uma mensagem. “Campos obrigatórios não informado”.

5. Regras

5.1. Regras de Negócio

RN1 – Os dados só poderão ser alterados se o ator estiver cadastrado no sistema

5.2. Regras de Apresentação

RA1 – O campo usuário é obrigatório e deverá ser composto de no máximo 15 (quinze) caracteres alfas numéricos.

RA2 – O campo senha é composto de 8 (oito) dígitos alfas numéricos.

RA3 – O campo CPF deverá ser composto por 11 (onze) dígitos numéricos.

RA4 – O campo Data de Nascimento deverá ser composto por 8 (oito) dígitos numéricos e 3 (três) alfanuméricos.

6. Requisitos Especiais

Não se aplica.

7. Pré-condições

O ator deverá estar devidamente cadastrado no sistema.

8. Pós-condições

Cadastro do ator mantido no sistema.

9. Pontos de extensão

Não se aplica.

Manter Gestor

1. Descrição

Este caso de uso tem por objetivo cadastrar, alterar e excluir dados do Gestor no sistema no Sistema de Gerenciamento de Notificação - SGN.

2. Referências

Caso de uso, Doc. Visão.

3. Atores

3.1. Administrador

Acesso Total às Funcionalidades; Manter Gestor.

4. Fluxo de eventos

4.1. Fluxo principal

- P1 O caso de uso inicia quando o ator seleciona a opção Gestor;
- P2 O sistema apresenta a tela de gerenciamento de gestor;(RA1)
- P3 O ator seleciona a opção consultar gestor;(A1), (A2), (A3)
- P4 O ator informa o CPF do gestor e clica em Consultar; (E1), (E2), (E3), (RA1)
- P5 O sistema apresenta uma tabela com os dados do gestor: Nome completo, CPF, data de nascimento, endereço, telefone, função.
- P6 O caso de uso se encerra.

4.2. Fluxos alternativos

A1 Inserir

A1.1 O caso de uso se inicia quando o ator seleciona a opção gestor;

A1.2 O ator seleciona a opção inserir gestor;

A1.3 O sistema apresenta a tela para uma nova inclusão; (RN1), (RA1), (RA2), (RA3), (RA4), (RA5), (RA6)

A1.4 O ator informa os dados: nome completo, CPF, data de nascimento, endereço, telefone, função;

A1.5 O ator seleciona a opção adicionar; (E1), (E2), (E3)

A1.6 O sistema apresenta a mensagem “Salvo com Sucesso”;

A1.7 O caso de uso é finalizado.

A2 Alterar

A2.1 O caso de uso é iniciado quando o ator seleciona a opção gestor;

A2.2 O ator seleciona a opção alterar gestor;

A2.3 O sistema apresenta uma lista com os gestores cadastrados;

A2.4 O ator seleciona o gestor a ser alterado;

A2.5 O ator informa os parâmetros necessários; (RN2), (RA1), (RA2), (RA3), (RA4), (RA5), (RA6)

A2.6 O ator confirma a alteração; (E1), (E2), (E3)

A2.7 O sistema apresenta a mensagem “Alterado com sucesso”;

A2.8 O caso de uso é finalizado.

A3 Excluir

A3.1 O caso de uso é iniciado quando o ator seleciona a opção gestor;

A3.2 O ator seleciona a opção excluir gestor;

A3.3 O sistema apresenta uma lista com os gestores cadastrados;

A3.4 O ator seleciona o funcionário a ser excluído;(RN3)

A3.5 O ator confirma a exclusão;

A3.6 O sistema apresenta a mensagem “Excluído com sucesso”;

A3.7 O caso de uso é finalizado.

4.3.Fluxos de Exceção

E1 CPF inválido

E1.1 O sistema apresenta a mensagem “CPF inválido”.

E2 Campos obrigatórios

E2.1 O sistema apresenta uma mensagem. “Campos obrigatórios não informado”.

E3 Dados inválidos

E3.1 O sistema apresenta uma mensagem de “Dados inválidos”.

5. Regras

5.1. Regras de Negócio

RN1 – Os dados só poderão ser criados se o ator não estiver cadastrado no sistema.

RN2 – Os dados só poderão ser alterados se o ator estiver cadastrado no sistema.

RN3 – Os dados só poderão ser excluídos se o ator estiver cadastrado no sistema

5.2. Regras de Apresentação

RA1 – O campo CPF deverá ser composto por 11 (onze) dígitos numéricos.

RA2 – O campo Nome não faz distinção de maiúsculas e minúsculas com tamanho máximo de 50 caracteres;

RA3 – O campo Data de Nascimento deverá ser composto por 8 (oito) dígitos numéricos e 3 (três) alfanuméricos.

RA4 – O campo Endereço não faz distinção de maiúsculas e minúsculas com tamanho máximo de 50 caracteres;

RA5 – O campo Telefone do tamanho máximo de 14 caracteres;

RA6 – O campo Função não faz distinção de maiúsculas e minúsculas com tamanho máximo de 50 caracteres;

6. Requisitos Especiais

Não se aplica.

7. Pré-condições

O ator deverá estar devidamente cadastrado no sistema.

8. Pós-condições

Cadastro do gestor mantido no sistema.

9. Pontos de extensão

Não se aplica.

Manter Funcionário

1. Descrição

Este caso de uso tem por objetivo cadastrar, alterar e excluir dados do funcionário no sistema no Sistema de Gerenciamento de Notificação - SGN.

2. Referências

Caso de uso, Doc. Visão.

3. Atores

3.1. Gestor

Acesso parcial às funcionalidades.

4. Fluxo de eventos

4.1. Fluxos principais

P1 O caso de uso inicia quando o ator seleciona a opção funcionário;

P2 O sistema apresenta a tela funcionário; (RA1), (RA2), (RA3), (RA4), (RA5), (RA6);

P3 O ator seleciona a opção consultar funcionário; (A1), (A2), (A3)

P4 O ator informa o CPF do funcionário e clica em Consultar; (E1), (E2), (E3);

P5 O sistema apresenta uma tabela com os dados do funcionário: Nome completo, CPF, data de nascimento, endereço, telefone, função;

P6 O caso de uso se encerra.

4.2. Fluxos alternativos

A1 Inserir

A1.1 O caso de uso se inicia quando o ator seleciona a opção funcionário;

A1.2 O ator seleciona a opção inserir funcionário;

A1.3 O sistema apresenta a tela para uma nova inclusão;
(RN1),(RA1),(RA2),(RA3),(RA4),(RA5),(RA6)

A1.4 O ator informa os dados: nome completo, CPF, data de nascimento, endereço, telefone, função;

A1.5 O ator seleciona a opção adicionar;

A1.6 O sistema apresenta a mensagem “ Salvo com Sucesso ”;

A1.7 O caso de uso é finalizado.

A2 Alterar

A2.1 O caso de uso é iniciado quando o ator seleciona a opção funcionário;

A2.2 O ator seleciona a opção alterar funcionário;

A2.3 O sistema apresenta uma lista com os funcionários cadastrados;

A2.4 O ator seleciona o funcionário a ser alterado;

A2.5 O ator informa os parâmetros necessários; (RN2), (RA1), (RA2),
(RA3), (RA4), (RA5), (RA6);

A2.6 O ator confirma a alteração;

A2.7 O sistema apresenta a mensagem “ Alterado com sucesso ”;

A2.8 O caso de uso é finalizado.

A3 Excluir

A3.1 O caso de uso é iniciado quando o ator seleciona a opção funcionário;

A3.2 O ator seleciona a opção excluir funcionário;

A3.3 O sistema apresenta uma lista com os funcionários cadastrados;

A3.4 O ator seleciona o funcionário a ser excluído; (RN3);

A3.5 O ator confirma a exclusão;

A3.6 O sistema apresenta a mensagem “ Excluído com sucesso ”;

A3.7 O caso de uso é finalizado.

4.3. Fluxos de Exceção

E1 CPF inválido

E1.1 O sistema apresenta a mensagem “ CPF inválido ”.

E2 Campos obrigatórios

E2.1 O sistema apresenta uma mensagem “ Campos obrigatórios não informado ”.

E3 Dados inválidos

E3.1 O sistema apresenta uma mensagem de “ Dados inválidos ”.

5. Regras

5.1. Regras de Negócio

RN1 – Os dados só poderão ser criados se o ator não estiver cadastrado no sistema.

RN2 – Os dados só poderão ser alterados se o ator estiver cadastrado no sistema.

RN3 – Os dados só poderão ser excluídos se o ator estiver cadastrado no sistema

5.2. Regras de Apresentação

RA1 – O campo CPF deverá ser composto por 11 (onze) dígitos numéricos.

RA2 – O campo Nome não faz distinção de maiúsculas e minúsculas com tamanho máximo de 50 caracteres;

RA3 – O campo Data de Nascimento deverá ser composto por 8 (oito) dígitos numéricos e 3 (três) alfanuméricos.

RA4 – O campo Endereço não faz distinção de maiúsculas e minúsculas com tamanho máximo de 50 caracteres;

RA5 – O campo Telefone deverá ser composto por 11 (onze) dígitos numéricos

RA6 – O campo Função não faz distinção de maiúsculas e minúsculas com tamanho máximo de 50 caracteres.

6. Requisitos Especiais

Não se aplica.

7. Pré-condições

O ator deverá estar devidamente cadastrado no sistema.

8. Pós-condições

Cadastro do funcionário mantido no sistema.

9. Pontos de extensão

Não se aplica.

Gerenciar Notificações

1. Descrição

Este caso de uso tem por objetivo consultar, enviar, inserir, alterar e excluir notificações no Sistema de Gerenciamento de Notificação - SGN.

2. Referências

Caso de Uso, Doc. Visão.

3. Atores

3.1.Gestor

Acesso parcial às funcionalidades; Gerenciar Notificações.

4. Fluxo de eventos

4.1.Fluxo principal

P1 O caso de uso inicia quando o ator seleciona a opção Gerenciar Notificações;

P2 O sistema apresenta a tela de gerenciamento de notificações;

P3 O ator seleciona a opção consultar notificações; (A1), (A2), (A3), (A4)

P4 O sistema apresenta uma tabela com as notificações cadastradas;

P5 O caso de uso se encerra.

4.2.Fluxos alternativos

A1 Enviar

A1.1 O caso de uso se inicia quando o ator seleciona a opção gerenciar notificações;

A1.2 O sistema apresenta a tela para envio de notificação; (RA2)

A1.3 O ator informa o nome do funcionário a ser notificado; (E1), (E2), (E3);

A1.4 O sistema valida os dados e apresenta a tela de envio;

A1.5 O ator informa a notificação a ser enviada;(RA1)

A1.6 O sistema apresenta a mensagem “Enviado com Sucesso”;

A1.7 O caso de uso se encerra.

A2 Inserir

A2.1 O caso de uso se inicia quando o ator seleciona a opção gerenciar notificações;

A2.2 O ator seleciona a opção inserir notificação;

A2.3 O sistema apresenta a tela para uma nova inclusão de notificação; (RN1), (RA1)

A2.4 O ator informa a notificação a ser adicionada;(E2)

A2.5 O ator seleciona a opção adicionar;

A2.6 O sistema apresenta a mensagem notificação “Salva com Sucesso”;

A2.7 O caso de uso é finalizado.

A3 Alterar

A3.1 O caso de uso é iniciado quando o ator seleciona a opção gerenciar notificações;

A3.2 O ator seleciona a opção alterar notificação;

A3.3 O sistema apresenta uma lista com as notificações cadastradas;

A3.4 O ator seleciona a notificação a ser alterada; (RN2), (RA1)

A3.5 O ator informa as alterações; (E2)

A3.6 O ator confirma a alteração;

A3.7 O sistema apresenta a mensagem “Alterado com sucesso”;

A3.8 O caso de uso é finalizado.

A4 Excluir

A4.1 O caso de uso é iniciado quando o ator seleciona a opção gerenciar notificações;

A4.2 O ator seleciona a opção excluir notificação;

A4.3 O sistema apresenta uma lista com as notificações cadastradas;

A4.4 O ator seleciona a notificação a ser excluída;(RN3)

A4.5 O ator confirma a exclusão;

A4.6 O sistema apresenta a mensagem “Excluído com sucesso”;

A4.7 O caso de uso é finalizado.

4.3.Fluxos de Exceção

E1 CPF inválido

E1.1 O sistema apresenta a mensagem “CPF inválido”.

E2 Campos obrigatórios

E2.1 O sistema apresenta uma mensagem. “Campos obrigatórios não informados”.

E3 Dados inválidos

E3.1 O sistema apresenta uma mensagem de “Dados inválidos”

5. Regras

5.1. Regras de Negócio

RN1 – Os dados só poderão ser criados se a notificação não estiver cadastrada no sistema.

RN2 – Os dados só poderão ser alterados se a notificação estiver cadastrada no sistema.

RN3 – Os dados só poderão ser excluídos se a notificação estiver cadastrada no sistema

5.2. Regras de Apresentação

RA1 – O campo inserir notificação deverá ter o tamanho máximo de 500 caracteres.

RA2 – O campo CPF deverá ser composto por 11 (onze) caracteres numéricos.

6. Requisitos Especiais

Não se aplica.

7. Pré-condições

O ator deverá estar devidamente cadastrado no sistema.

8. Pós-condições

Notificações mantidas no sistema.

9. Pontos de extensão

Não se aplica.

Gerar Relatórios de Notificações

1. Descrição

Este caso de uso tem por objetivo gerar relatórios das notificações enviadas aos funcionários no Sistema de Gerenciamento de Notificação - SGN.

2. Referências

Caso de Uso, Doc. Visão.

3. Atores

3.1.Gestor

Responsável por administrar o sistema.

4. Fluxo de eventos

4.1.Fluxo principal

- P1 O caso de uso inicia quando o ator seleciona a opção gerar relatório;
- P2 O sistema apresenta a tela de consulta de relatório de notificações;
- P3 O ator informa o CPF do funcionário a ser gerado o relatório (RA1), (E1), (E2), (E3);
- P4 O sistema apresenta o resultado da consulta;
- P5 O caso de uso se encerra.

4.2.Fluxos de Exceção

E1 Campos obrigatórios

E1.1 O sistema apresenta uma mensagem. “Campos obrigatórios não informados”.

E2 CPF inválido

E2.1 O sistema apresenta a mensagem “CPF inválido”.

E3 Dados inválidos

E3.1 O sistema apresenta uma mensagem de “Dados inválidos”

5. Regras de Negócio

5.1. Regras de Negócio

Não se aplica.

5.2. Regras de Apresentação

RA1 – O campo CPF deverá ser composto por 11 (onze) dígitos numéricos.

6. Requisitos Especiais

Não se aplica.

7. Pré-condições

O ator deverá estar devidamente cadastrado no sistema.

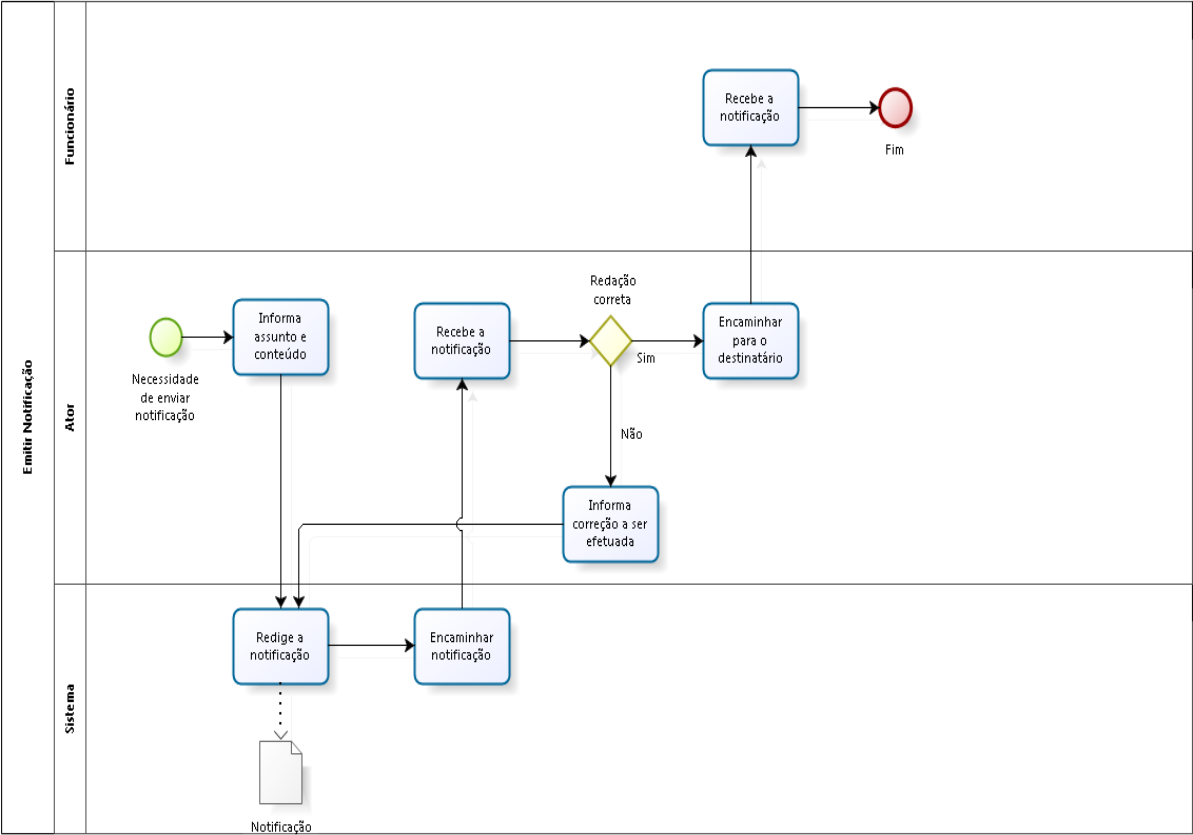
8. Pós-condições

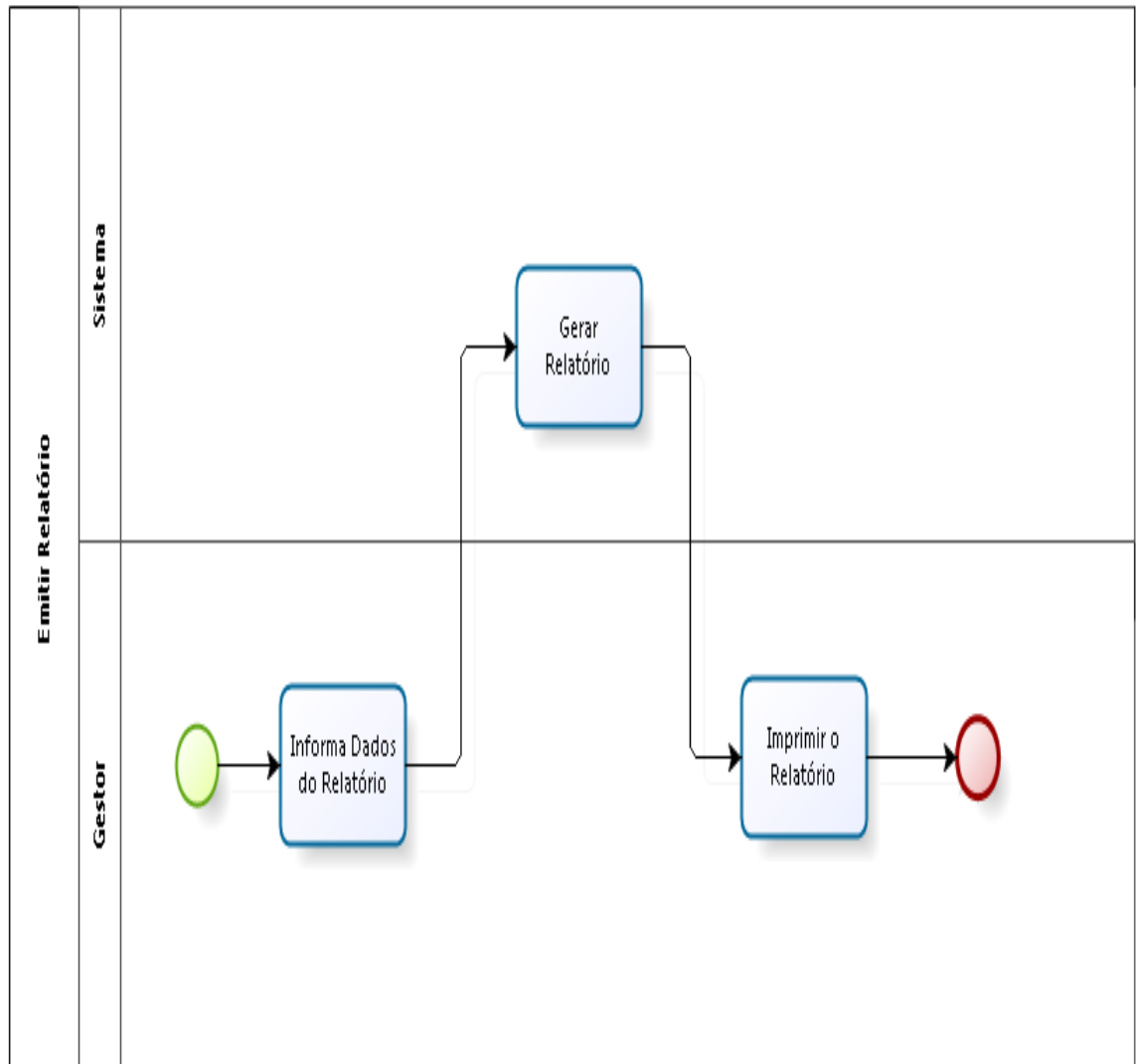
O sistema gera o relatório solicitado.

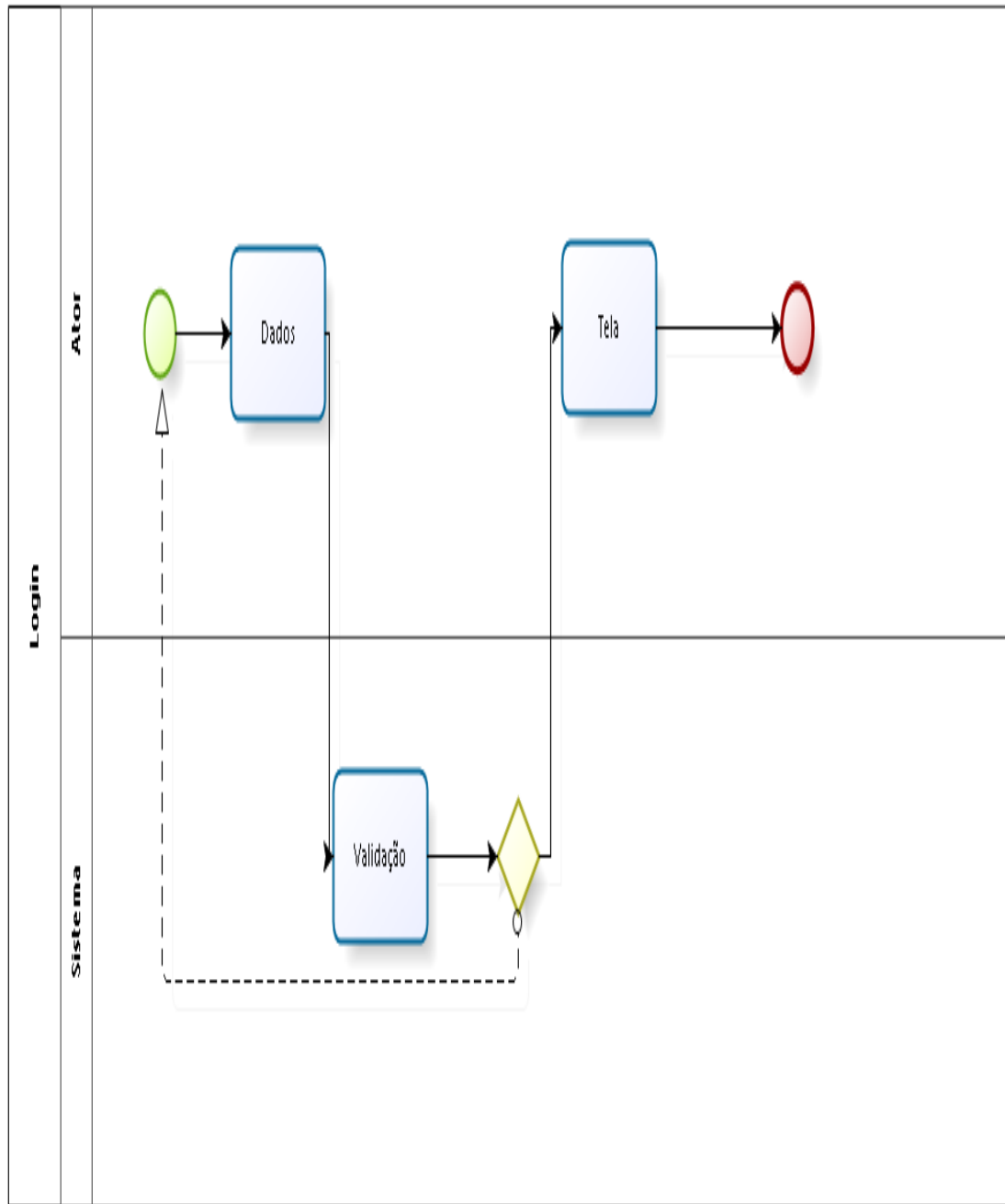
9. Pontos de extensão

Não se aplica.

APÊNDICES D – MODELAGEM DE PROCESSO







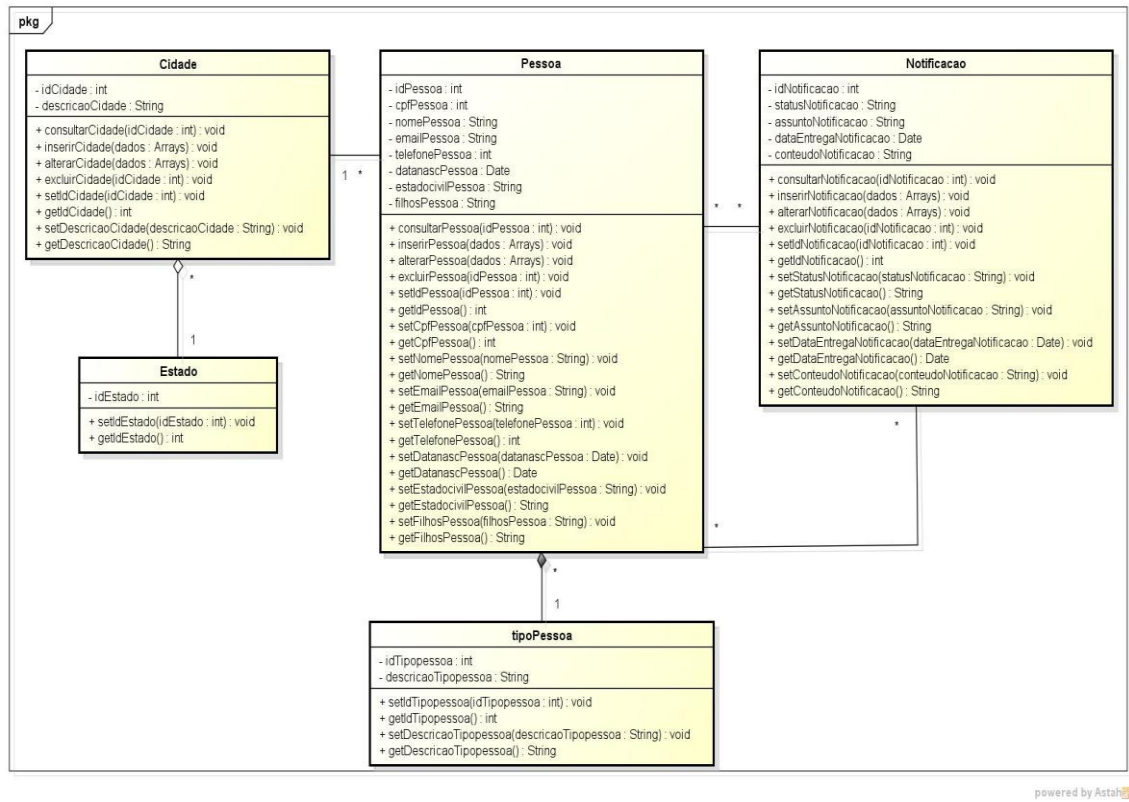
APÊNDICES E – ESTIMATIVA DE PONTO DE FUNÇÃO

APÊNDICES F - DIAGRAMAS DE SEQUÊNCIA

APÊNDICES G - DIAGRAMAS DE ATIVIDADES

APÊNDICES H - DIAGRAMA DE IMPLANTAÇÃO

APÊNDICES I - DIAGRAMA DE CLASSES



APÊNDICES J - MER FÍSICO E DICIONÁRIO DE DADOS

