Sistema Amnésia

Sistema de captura e gerenciamento de anotações pessoais

Estácio

Lucas Gonçalves Bento

Rio de Janeiro
2012

LUCAS GONÇALVES BENTO

SISTEMA AMNÉSIA
SISTEMA DE CAPTURA E GERENCIAMENTO DE ANOTAÇÕES PESSOAIS

TRABALHO MONOGRÁFICO SUBMETIDO AO CORPO DOCENTE DA COORDENAÇÃO DE SISTEMAS DE INFORMAÇÃO DA UNIVERSIDADE ESTÁCIO DE SÁ, COMO PROJETO FINAL DE CURSO PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

RIO DE JANEIRO
2012

FICHA CATALOGRÁFICA

BENTO, LUCAS GONÇALVES

Sistema Amnésia – Sistema de captura e gerenciamento de anotações pessoais.

65pp

Orientadora: Patrícia Fiúza de Castro

Monografia (Bacharelado) - Universidade Estácio de Sá, Curso de Sistemas de Informação, 2011.

Palavras chave: Anotações, Versionamento, Sincronização

I. Bento, Lucas Gonçalves. II. Universidade Estácio de Sá, Curso de Sistemas de Informação.

AGRADECIMENTOS

A Deus, por tudo.
A minha namorada Morgana, pelo auxilio, motivação e ajuda na revisão.
A meus pais e irmãos pelo silêncio.
Ao meu cachorro Eddie que sempre se deita perto de mim quando eu estou trabalhando.

RESUMO

Este trabalho desenvolve uma aplicação para uso em computadores desktops com o intuito servir como gerenciador de anotações pessoais, com funcionalidades de versionamento e sincronização remota dos dados do usuário. O projeto inclui análise, planejamento e implementação do software, realizados seguindo a metodologia orientada a objetos. O desenvolvimento deverá ser feito utilizando a linguagem de programação Groovy e o banco de dados orientado a documentos OrientDB.

ABSTRACT

This paper develops an application for use on desktop computers in order to serve as personal notes manager with features versioning and remote synchronization of user data. The project includes analysis, planning and implementation of software, made following the object-oriented methodology. The development must be done using the Groovy programming language and the document-oriented database OrientDB.

SUMÁRIO

INTRODUÇÃO

OBJETIVOS, JUSTIFICATIVA E ANÁLISE DE MERCADO

OBJETIVOS

Este trabalho tem como objetivo criar um sistema para gerenciar dados pessoais de usuários de computadores, em forma de anotações, ou textos, que serão persistidas em uma base de dados, versionadas e indexadas para maior facilidade de acesso e conveniência do usuário.

O projeto não tem a pretensão de criar ou fingir criar um sistema comercial, sendo assumidamente um trabalho de conclusão de curso e tendo toda sua documentação pautada nisto. O projeto tem no entanto, a pretensão de criar uma ferramenta de software livre que pode ser utilizada integralmente com as funcionalidades propostas.

O software resultante deste projeto se chamará Amnésia. De acordo com o Wikipedia: ***Amnésia*** *é a perda de memória (total ou parcial), na maioria dos casos é temporária e pode ser causada por diversas razões*. É opinião do Desenvolvedor deste projeto que o ser humano moderno não é capaz de reter a quantidade de informação disponível atualmente, principalmente quando o seu trabalho ou estilo de vida envolve o contato constante com esta massa infinita de dados. Por esta razão estou propondo o sistema Amnésia, para ser uma ferramenta que ajudará os seus usuários quando por alguma razão eles sofrerem uma falta temporária de memória, e não se lembrarem de alguma informação importante.

Outro ponto importante do projeto, é que o software resultante será totalmente freesource, e terá tanto seu código fonte como executáveis binários disponíveis na internet para qualquer pessoa que se interessar por ele.

JUSTIFICATIVA

A escolha do tema se deve a observações realizadas no fluxo de trabalho de usuários de computador. Uma pessoa que tenha seu trabalho realizado através do computador tem a necessidade de persistir uma série de dados de uso pessoal que lhes são fornecidos por várias fontes, como por exemplo:

- Endereços de email.
- Comandos de terminal.
- Host-names ou endereços IP de máquinas ou sistemas remotos.
- Nomes de usuário e senhas, ou outros dados para conexão a algum sistema.
- Trechos de código.
- Tarefas a concluir (TODOs).
- Observações pertinentes a alguma tarefa.

Estes dados, que podem conter informações importantes para a realização de algum trabalho,

geralmente ficam relegados a anotações físicas em cadernos ou bilhetes, ou na melhor das hipóteses algum tipo de coleção de arquivos de texto mantida de forma manual pelo usuário. Estes métodos não são eficientes pelos seguintes motivos:

- Não permitem a fácil recuperação dos dados armazenados:
  Em que página do caderno, ou em qual arquivo de texto será que está aquela informação?
- Não garantem a persistência dos dados em caso de erro humano:
  O que acontece se o usuário esquecer de salvar o arquivo antes de desligar a máquina?
  Ou se decidir fazer uma "limpeza" e acabar apagando algo que será necessário mais tarde?
- Não garantem a segurança dos dados em caso de falha da máquina.
  E se o HD morrer, ou ou algum vírus deletar os arquivos?
- Geralmente não possuem disponibilidade fora da mesa de trabalho do usuário.

Se precisar lembrar daquele comando quando estiver ajudando um colega a instalar um programa vai ter que ir até sua maquina para conseguir a informação.

O software a ser construído ao londo deste projeto se propõe a atender as necessidades dos usuários de computador de possuir um sistema que gerencie estas informações e não sofrer com as limitações dos métodos ad hoc existentes. Desta maneira o programa desenvolvido deve:

- Estruturar e indexar os dados inseridos para facilitar a recuperação das informações quando necessário.
- Ser fácil e prático de usar. Caso contrario os usuário irão simplesmente continuar usando suas soluções atuais.
- Ser tolerante a falhas humanas para evitar perda acidental de dados.
- Fornecer funcionalidades de backup. Para permitir a recuperação em caso de falha humana ou da máquina.
- Permitir o acesso aos dados em mais de um local.

CONCORRENTES

Existem no mercado diversas opções de aplicativos com a proposta ser uma ferramenta pessoal de captura de anotações. Muitos dos concorrentes mais bem posicionados permitem realizar a captura não apenas de texto simples, mas também texto formatado, imagens, sons, e até mesmo vídeos.

Apesar de parecer interessante fazer um comparativo completo entre todos os concorrentes, isto acabaria ocupando muito espaço, logo foi decidido realizar uma comparação com dois dos maiores representantes do mercado de gerenciadores de anotações: Microsoft One Note e Evernote.

ONE NOTE

De acordo com o site oficial do produto:

> *O Microsoft OneNote 2010 é um bloco de anotações digital que fornece um único local onde você pode coletar todas as suas anotações e informações, com os benefícios adicionais de recursos avançados de pesquisa, para localizar o que quiser rapidamente, e de blocos de anotações compartilhados e fáceis de usar, para que você possa gerenciar a sobrecarga de informações e trabalhar em equipe com mais eficiência.*
>
> *Diferentemente dos sistemas baseados em papel, programas de processamento de texto, sistemas de email ou outros programas de produtividade, o OneNote proporciona a flexibilidade de coletar e organizar textos, imagens, manuscritos digitais, gravações de áudio e vídeo etc. — tudo em um único bloco de anotações digital, no seu computador. O OneNote pode ajudar você a se tornar mais produtivo, pois mantém as informações necessárias à mão e reduz o tempo gasto com pesquisas de informações em mensagens de email, blocos de anotações em papel, pastas de arquivos e impressos.*
>
> *O OneNote 2010 é parte integrante do Microsoft Office 2010, que facilita a coleta, a organização, a localização e o compartilhamento de anotações e informações com mais eficiência e eficácia. Recursos poderosos de pesquisa podem ajudar a localizar informações em textos de imagens ou em narrações de gravações em áudio e vídeo. E ferramentas colaborativas fáceis de usar podem ajudar as equipes a trabalhar juntas com todas essas informações em blocos de anotações compartilhados, online ou offline.*
>
> *Com todas essas informações à mão, o OneNote fornece uma solução para sobrecarga de informações, permite trabalhar em equipe com mais eficácia e ajuda você a obter todas as informações de tarefas, agendas e equipes. A aparência familiar do sistema Microsoft Office facilita o uso imediato do programa, minimizando a perda de tempo e os custos com treinamento.*

O One note oferece uma aplicação robusta e completa para os seus usuários, contando com um editor de texto rico, ferramentas de desenho, tocador de áudio e de vídeo e funcionalidades de compartilhamento de conteúdo e sincronização. Tudo isso torna o aplicativo pesado e complexo, indo de encontro com a proposta de ser uma ferramenta de acesso fácil para armazenar rapidamente

os dados do usuário.

O One note também é parte integrante da suíte Microsoft Office, logo possui um preço relativamente elevado, não está disponível fora da plataforma windows e não tem seu código fonte disponível para estudo e modificação.

O One Note realiza backups automáticos de todas as alterações realizadas pelo usuário. Porém, estes backups não oferecem uma visualização rápida e acessível dos dados antigos.

EVERNOTE

> *O Evernote torna fácil lembrar das pequenas e grandes coisas de todas as partes da sua vida, usando o seu computador, celular e a web.*
>
> *Se você pensa, vê, ou experiencia algo, o Evernote pode ajudá-lo a se lembrar disso. Escreva uma nota de texto. Salve uma página da internet. Tire uma foto. Guarde uma captura de tela. O Evernote guardará tudo em segurança.*
>
> *Tudo o que você captura é automaticamente processado, indexado e pesquisável. Se quiser, você pode adicionar marcadores ou organizar notas em diferentes pastas.*
>
> *Tenha o Evernote em cada computador e celular que você usa, depois pesquise através de todas as suas notas por palavras-chave, títulos, marcadores ou mesmo localização. O Evernote também torna, magicamente, o texto impresso e manuscrito nas suas imagens pesquisável.*

O Evernote é uma opção semelhante ao One Note, com funcionalidades semelhantes. As maiores diferenças entre eles estão na interface, onde o Evernote se mostra mais moderno e fácil de usar. Outra vantagem do Evernote é a sua disponibilidade em várias plataformas, e não apenas em Windows.

PLANEJAMENTO

ESCOPO, METODOLOGIA, RECURSOS E TAREFAS

ESCOPO

A aplicação a ser desenvolvida gerenciará apenas entradas em texto simples sem formatação. Não serão gerenciados arquivos de nenhum tipo; apenas textos inseridos na interface do programa.
Os textos gerenciados nunca poderão ser excluídos e nem atualizados: quando uma entrada for "alterada" na verdade um novo registro será criado e armazenado como sendo a versão mais recente da anotação anterior.
Um módulo de sincronização será desenvolvido em paralelo, para permitir que o usuário tenha sempre suas anotações mais atualizadas disponíveis em múltiplos computadores.
Os dispositivos suportados serão apenas computadores comuns. Apesar do suporte a dispositivos móveis e uma interface web serem atraentes, eles não devem ser contemplados por enquanto.

METODOLOGIA

Todo sistema de software deve ser desenvolvido baseado em uma sólida base teórica. A escolha desta base, e os elementos que a compõem são importantíssimas para o sucesso de sua implementação. Uma escolha mal realizada pode resultar em atrasos, dificuldades no desenvolvimento, bugs, funcionalidades faltantes, ou até mesmo funcionalidades desnecessárias. Desta forma foi dado uma consideração especial na escolha das metodologias empregadas.

FREE SOFTWARE

Software Livre, ou Free Software, conforme a definição de software livre criada pela Free Software Foundation, é o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição. A forma usual de um software ser distribuído livremente é sendo acompanhado por uma licença de software livre (como a GPL ou a BSD), e com a disponibilização do seu código-fonte.

Software Livre se refere à existência simultânea de **quatro tipos de liberdade** para os usuários do software, definidas pela Free Software Foundation. Veja abaixo uma explicação sobre as 4 liberdades, baseada no texto em português da Definição de Software Livre publicada pela FSF:

As 4 liberdades básicas associadas ao software livre são:

- A liberdade de executar o programa, para qualquer propósito (liberdade nº 0)
- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade nº 1). Acesso ao código-fonte é um pré-requisito para esta liberdade.
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº 2).
- A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

UML

UML significa *Unified Modeling Language*, ou *Linguagem de Modelagem Unificada*. A UML é uma linguagem de modelagem de sistemas informatizados. Esta linguagem utiliza uma série de diagramas para descrever um sistema de software, seu comportamento, estado e seus dados.

A UML é gerida pela OMG (Object Management Group), que fornece a especificação da linguagem em seu site na internet. De acordo com esta especificação os objetivos da UML são:

> *The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.*
>
> *O objetivo da UML é de fornecer aos arquitetos de sistemas, engenheiros de software e desenvolvedores de software, ferramentas para análise, projeto e implementação de sistemas baseados em software, bem como para a modelagem de negócios e processos semelhantes.*

DESENVOLVIMENTO ORIENTADO A OBJETOS

Orientação a Objetos é um paradigma de desenvolvimento de software, se aplicando à análise, projeto e programação de sistemas de software. A Orientação a Objetos utiliza "objetos" - estruturas de dados compostos de campos de dados e métodos. Estes objetos interagem entre si através de seus métodos e armazenam dados em seus campos, para juntos formarem um sistema.

DESENVOLVIMENTO ÁGIL

Desenvolvimento Ágil é o nome atribuído a uma série de metodologias de desenvolvimento de software baseados em desenvolvimento iterativo e incremental, onde os requisitos e solução evoluem através da interação entre grupos pequenos e auto organizados. As práticas ágeis promovem planejamento adaptativo, desenvolvimento e entregas evolutivos, uma abordagem de tempo iterativa, e encoraja respostas rápidas flexíveis à mudanças.

Os quatro principais princípios do desenvolvimento ágil são:

Indivíduos e interação entre eles mais que processos e ferramentas
Software em funcionamento mais que documentação abrangente
Colaboração com o cliente mais que negociação de contratos
Responder a mudanças mais que seguir um plano

Onde os itens à direita não são ignorados ou desprezados, mas são menos valorizados do que os itens à esquerda.

RECURSOS

HUMANOS

Sendo um projeto de pequeno porte, com recursos muito limitados e uma proposta não comercial, apenas uma pessoa irá trabalhar na elaboração e desenvolvimento do sistema.

Para fins de clareza e objetividade, o único colaborador disponível receberá o título de Desenvolvedor. Este título não deve ser confundido com a posição de desenvolvedor existente em projetos mais elaborados e que é atribuída a aqueles responsáveis por implementar, e por vezes analisar funcionalidades em um sistema. O termo Desenvolvedor é utilizado aqui em um sentido muito mais amplo, e se refere à responsabilidade de desenvolver o projeto em sua totalidade, e não apenas implementar o software.

Lucas Gonçalves Bento: Aluno do curso de Sistemas de Informação, Empregado na empresa Ideais Tecnologia; o Senhor Lucas será encarregado de realizar todas as tarefas necessárias para a conclusão do projeto apresentado. As responsabilidades atribuídas ao Desenvolvedor incluem:

- Organizar e planejar o andamento do projeto, as tarefas a serem realizadas, objetivos, prazos, redigir todos os artefatos a serem criados, gerenciar os recursos disponíveis.
- Levantar os requisitos do projeto, projetar o sistema, escolher a arquitetura e tecnologias que serão utilizadas.
- Implementar o código do sistema em Java, testar as funcionalidades implementadas e desenhar a interface do programa.
- Criar uma base de dados orientada a documentos para ser utilizada pelo programa.
- Entregar e apresentar o projeto à banca examinadora.

Como muitas responsabilidades estão sendo depositadas sobre os ombros de apenas uma pessoa, o recurso mais valioso durante o andamento dos trabalhos será o tempo disponível ao Desenvolvedor, que deve ser suficiente para realizar todas as tarefas necessárias à conclusão do projeto.

Os recursos financeiros disponíveis também são escassos, compondo-se principalmente de equipamentos de posse pessoal do Desenvolvedor, e cujo custo será listado de maneira simbólica.

O pagamento pelas horas de trabalho também está listado de forma simbólica, pois nenhuma remuneração foi oferecida ao Desenvolvedor por seus trabalhos. Desta forma o custo listado representa um valor representante da remuneração recebida por ele em seu emprego regular.

FERRAMENTAS

Para o desenvolvimento do sistema, e sua documentação, foi utilizado um notebook com a seguinte configuração:

Aspire 4745-7321
Processador: Intel Core i5 450M 2.4Ghz, Cache 3MB, Hyper-Threading e Turbo Boost
Chipset: Intel HM55
Placa de Vídeo: Intel HD
RAM: 4GB DDR3
Disco Rígido: 500GB
Wireless: 802.11 b/g/n
Sistema Operacional: Windows 7 Ultimate/Ubuntu 12.04

Para ambiente de teste será utilizado um serviço de hospedagem. Ver Serviços.

SOFTWARE

Uma série de softwares são necessários para a completar o projeto, tanto para editar textos e planilhas, como diagramas, códigos fontes e outros. A lista completa segue abaixo:

- **Google Docs, Libre Office:** utilizados para redigir e editar textos, montar planilhas e criar documentos em geral.
- **Astah Comunity:** Software necessário para a criação de diagramas UML.
- **Smartsheet:** utilizado para criar um cronograma para o projeto.
- **Eclipse IDE:** programa fundamental durante o desenvolvimento. Todo o código fonte será escrito utilizando o Eclipse.
- **Java JDK:** máquina virtual e ferramentas de desenvolvimento para a plataforma Java.
- **Groovy:** linguagem dinâmica baseada na linguagem Java e executável na JVM.
- **Griffon:** Framework para desenvolvimento de aplicações desktop utilizando a linguagem Groovy.
- **OrientDB:** Banco de dados orientado a documentos.
- **Git:** sistema de versionamento distribuído, utilizado para manter os arquivos do projeto.

SERVIÇOS

Github: Serviço online de hospedagem de repositórios Git. Também é uma conhecida plataforma de compartilhamento e agregação de software livre.
Serviços Gráficos: Serviços de impressão e encadernação em copiadora comercial.

ATIVIDADES

Nome da tarefa	Início	Fim	Duração	Antecessor
Projeto Final I	**01/01/12**	**19/05/12**	**140**	
Definição do Projeto	**01/01/12**	**30/01/12**	**30**	
Objetivos	01/01/12	06/01/12	6	
Redigir uma seção de texto esclarecendo os objetivos do projeto, e o resultado esperado ao final da conclusão dos trabalhos				
Justificativa	06/01/12	16/01/12	11	
Redigir um texto dissertando sobre as razões pelas quais o projeto está sendo executado, e qual a importância do mesmo para seus possíveis clientes e seus desenvolvedores.				
Concorrentes	13/01/12	18/01/12	6	
Criar uma lista de sistemas de software com propostas e funcionalidades semelhantes ao sistema que o projeto propõe criar. Incluir comparações entre todos os concorrentes e justificar a criação do sistema diante das opções já existentes.				
Escopo	17/01/12	24/01/12	8	
Escrever um texto explicando as limitações de funcionalidade e implementação do projeto. Esclarecendo o que ele uirá e o que ele não irá fazer.				
Metodologia	25/01/12	30/01/12	6	
Redigir um guia sobre as metodologias e filosofias empregadas no projeto. Tais quais Software Livre, Análise Orientada a Objetos, Desenvolvimento Ágil e outras.				
Planejamento	**31/01/12**	**18/02/12**	**19**	**2**
Recursos	31/01/12	05/02/12	6	
Expor os recursos disponíveis e necessários para a conclusão do projeto, e como eles serão aplicados durante o andamento dos trabalhos. Inclui recursos humanos, materiais, equipamentos, serviços, software, e qualquer outra coisa que seja usada em alguma atividade do projeto				
Atividades	05/02/12	10/02/12	6	
Criar uma tabela contendo e explicando todas as atividades que precisarão ser realizadas para que o projeto seja concluído.				
Cronograma	11/02/12	18/02/12	8	10
Criar um diagrama demonstrativo das tarefas do projeto, e de como elas estão planejadas para serem realizadas no tempo limite determinado.				
Custos	11/02/12	15/02/12	5	10
Fazer uma relação de todos os custos decorrentes do projeto, tanto com recursos pessoais quanto com recursos materiais, equipamentos, serviços etc.				

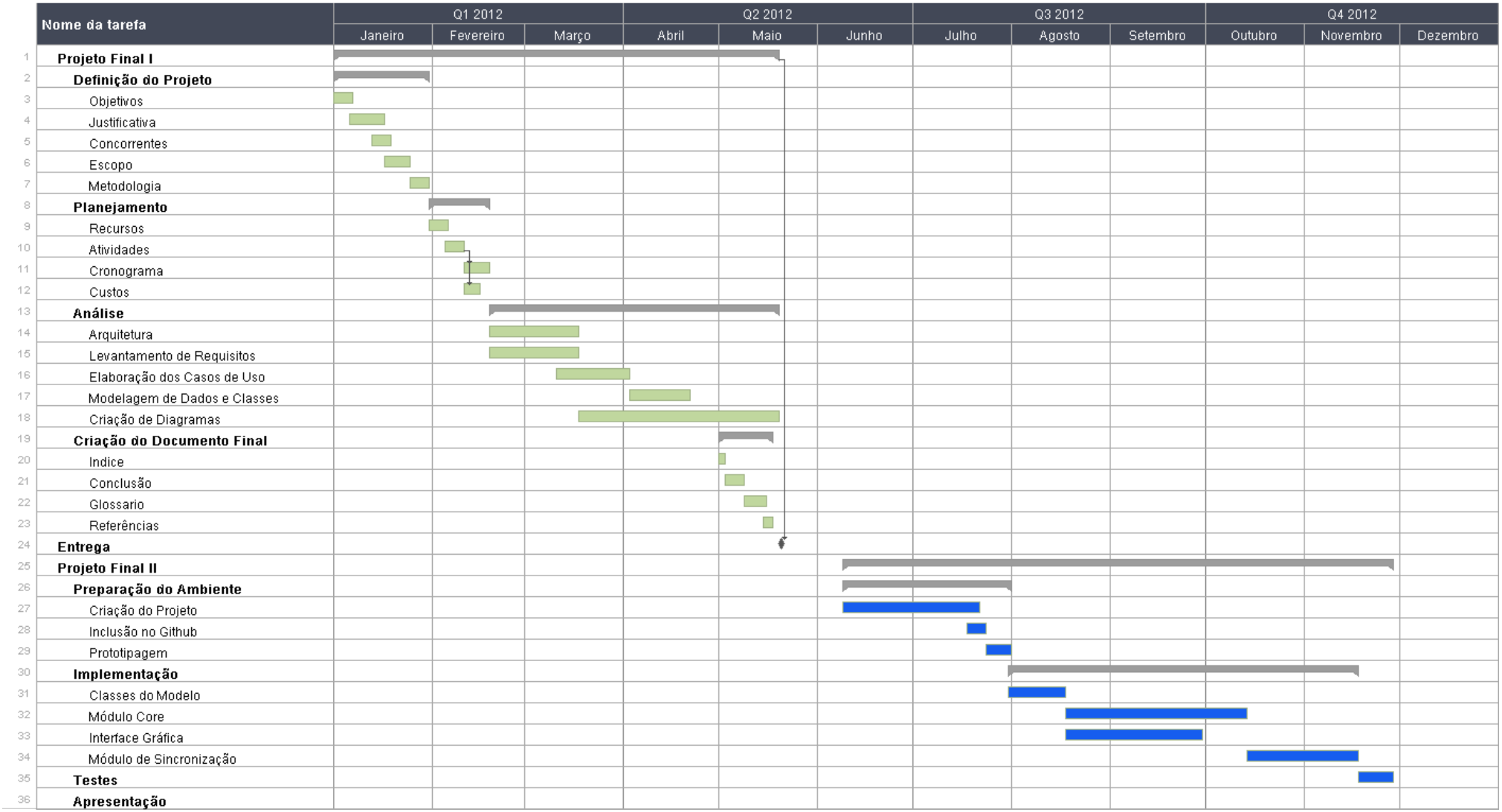
ATIVIDADES

Nome da tarefa	Início	Fim	Duração	Antecessor
Análise	**19/02/12**	**19/05/12**	**91**	
Arquitetura	19/02/12	17/03/12	28	
Escrever uma descrição geral do sistema, detalhando as tecnologias utilizadas, e os padrões empregados na implementação do software.				
Levantamento de Requisitos	19/02/12	17/03/12	28	
Realizar uma análise dos requisitos do sistema, e documentá-los.				
Elaboração dos Casos de Uso	11/03/12	02/04/12	23	
Escrever todos os casos de uso necessários para atender a todos os requisitos levantados para o sistema de acordo com os padrões da Análise Orienteda a Objetos				
Modelagem de Dados e Classes	03/04/12	21/04/12	19	5
Definir as Classes de modelo do sistema, e determinar como elas se relacionam ao banco de dados orientado a documentos.				
Criação de Diagramas	18/03/12	19/05/12	63	
Criar os diagramas referentes a casos de uso, diagrama de classes e outros diagramas disponíveis na UML que sejam aplicaveis ao sistema desenvolvido.				
Criação do Documento Final	**01/05/12**	**17/05/12**	**17**	
Indice	01/05/12	02/05/12	2	
Escrever um índice, apontando para a localização no documento de cada capítulo e subcapítulo.				
Conclusão	03/05/12	08/05/12	6	
Escrever uma conclusão para o documento, contendo considerações finais e outras observações consideradas relevantes				
Glossario	09/05/12	15/05/12	7	
Criar um dicionário de termos utilizados durante o documento, explicando seus significados, e se necessário citando referencias externas.				

ATIVIDADES

Nome da tarefa	Início	Fim	Duração	Antecessor
Projeto Final II	**09/06/12**	**28/11/12**	**173**	
Preparação do Ambiente	**09/06/12**	**31/07/12**	**53**	
Criação do Projeto	09/06/12	21/07/12	43	
Criar a estrutura do projeto para desenvolvimento. Dontendo arquivos de configuração, estrutura de diretórios bibliotecas, APIs, Frameworks etc.				
Inclusão no Github	18/07/12	23/07/12	6	
Incluir o projeto em um repositório de software livre, para ser acessado livremente pela comunidade, e qualquer outro interessado no código fonte do sistema.				
Prototipagem	24/07/12	31/07/12	8	
Criar protótipos das telas do sistema para melhor compreensão do funcionamento das funcionalidades que serão desenvolvidas.				
Implementação	**31/07/12**	**17/11/12**	**110**	
Classes do Modelo	31/07/12	17/08/12	18	
Criar as classes de modelo já levantadas anteriormente e constantes no Diagrama de Casses criado.				
Módulo Core	18/08/12	13/10/12	57	
Desenvolvimento do módulo principal do sistema, onde estará a maior parte da lógica de negócio, e responsável pela maior parte das funcionalidades.				
Interface Gráfica	18/08/12	29/09/12	43	
Criar a interface gráfica para usuários. Esta interface deverá ser desenvolvida a partir dos protótipos criados anteriormente.				
Módulo de Sincronização	14/10/12	17/11/12	35	
Desenvolver o módulo responsável por fazer a sincronização e backup dos dados entre várias instâncias clientes do sistema.				
Testes	**18/11/12**	**28/11/12**	**11**	
Realizar testes com o sistema concluido. Poderão ser feitos testes com usuários voluntários.				
Apresentação				
Apresentar e defender o projeto perante a banca examinadora.				

Cronograma


Nome da tarefa
Q1 2012
Q2 2012
Q3 2012
Q4 2012
Janeiro
Fevereiro
Março
Abril
Maio
Junho
Julho
Agosto
Setembro
Outubro
Novembro
Dezembro
1 Projeto Final I
2 Definição do Projeto
3 Objetivos
4 Justificativa
5 Concorrentes
6 Escopo
7 Metodologia
8 Planejamento
9 Recursos
10 Atividades
11 Cronograma
12 Custos
13 Análise
14 Arquitetura
15 Levantamento de Requisitos
16 Elaboração dos Casos de Uso
17 Modelagem de Dados e Classes
18 Criação de Diagramas
19 Criação do Documento Final
20 Indice
21 Conclusão
22 Glossario
23 Referências
24 Entrega
25 Projeto Final II
26 Preparação do Ambiente
27 Criação do Projeto
28 Inclusão no Github
29 Prototipagem
30 Implementação
31 Classes do Modelo
32 Módulo Core
33 Interface Gráfica
34 Módulo de Sincronização
35 Testes
36 Apresentação

CUSTOS

Pessoal			
Nome do Recurso	**Custo por Hora**	**Total de Horas**	**Custo**
Lucas Gonçalves Bento	30	504	R$ 15.120,00

Equipamento/Material			
Descrição	**Valor Unitário**	**Quantidade**	**Custo**
Notebook	R$ 1500,00	1 Unidade	R$ 1500,00
Pendrive	R$ 30,00	1 Unidade	R$ 30,00
CDs virgens	R$ 2,00	10 unidades	R$ 20,00
Material para rascunhos e anotações	R$ 30,00	1 unidade	R$ 30,00

Software			
Descrição	**Valor Unitário**	**Quantidade**	**Custo**
Google Docs	Gratuito	1 conta	$0.00
Libre Office	Gratuito	1 instalação	$0.00
Dropbox	Gratuito	1 conta	$0.00
Jude Comunity	Gratuito	1 instalação	$0.00
Eclipse JDK	Gratuito	1 instalação	$0.00
Java JDK	Gratuito	1 instalação	$0.00
Groovy	Gratuito	1 instalação	$0.00
Griffon Framework	Gratuito	1 instalação	$0.00
Github	Gratuito	1 conta	$0.00
Smart Sheet	Gratuito	1 conta	$0.00

Serviços			
Descrição	**Valor Unitário**	**Quantidade**	**Custo**
Encadernação Monografia	R$ 35,00	1 Unidade	R$ 35,00
Impressão a laser preto	R$ 0,15	200 páginas	R$ 30,00
Impressão a laser colorida	R$ 1,50	30 páginas	R$ 45,00
		TOTAL	$16,810.00

ANÁLISE

REQUISITOS E CASOS DE USO

REQUISITOS E CASOS DE USO

Os requisitos de um sistema, são funções e características que o sistema deve apresentar para se provar útil aos seus usuários.

Casos de Uso são descrições textuais de uma funcionalidade, detalhando em maior ou menor grau o seu fluxo e comportamento.

Na listagem apresentada a seguir, os casos de uso estão separados entre Funcionais e Não Funcionais. Os requisitos funcionais estão acompanhados de uma relação dos casos de uso que atendem aquele requisito, ou seja, uma lista das funcionalidades que serão implementadas para garantir que o requisito seja atendido.

REQUISITOS FUNCIONAIS

Os casos de uso funcionais descrevem funcionalidades do sistema. Geralmente é dito que eles descrevem o que se espera que o software faça.

1. **Persistir as entradas do usuário.**
   A principal função do sistema consiste em permitir que o usuário escreva as suas anotações na interface fornecida. As anotações devem ser salvas de maneira transparente.
   Casos de Uso: Inserir Anotação, Editar Anotação.

2. **Manter histórico de cada entrada.**
   O sistema deve permitir que o usuário visualize antigas versões de suas anotações. Sempre que uma anotação for “editada”, na verdade uma nova versão será criada e a antiga continuará disponível para visualização.
   Casos de Uso: Visualizar Versões, Editar Anotação, Promover Versão.

3. **Categorizar cada entrada com tags para identificação.**
   O modelo de tags - rótulos pelos quais os registros são categorizados de maneira humanamente legível - deve ser utilizado para facilitar o uso pelos usuários. Cada anotação pode ser marcada com múltiplas tags. Novas versões de uma anotação herdam as tags de seus antecessores, mas os antecessores não recebem as tags de uma versão mais recente.
   Casos de Uso: Incluir Tag, Excluir Tag

4. **Busca inteligente**
   É primordial a presença de um mecanismo de busca no aplicativo, que pesquise por nome, tags, data e conteúdo da entrada, para encontrar os registros que contenham a chave informada. A busca deve retornar qualquer registro correspondente, mesmo que seja uma versão antiga. Porém, caso mais de uma versão da mesma anotação corresponda, apenas a mais recente deve ser exibida.
   Casos de Uso: Busca Padrão, Definir Busca Padrão, Busca Avançada.

5. **Sincronização dos dados entre múltiplos computadores.**
   Uma funcionalidade importantíssima é a capacidade de sincronizar os dados locais com um servidor remoto. Desta maneira os dados do servidor remoto servirão tanto como backup como um repositório contendo sempre todos os dados mais recentes do usuário.
   Casos de Uso: Sincronizar Com servidor.

REQUISITOS NÃO FUNCIONAIS

Os casos de uso não funcionais descrevem qualidades do sistema que não se traduzem em funções realizadas pelo mesmo. Geralmente são requisitos envolvendo implementação, desempenho, interface, segurança etc.

1. **Transparência na busca.**
   A busca padrão deve ser fácil de usar. O usuário não deve ser incomodado com vários campos a serem preenchidos e opções a serem escolhidas.

2. **Filtragem da busca por data, tag e profundidade da versão da entrada.**
   Apesar da necessidade de simplicidade da busca padrão, deve estar disponível também uma busca avançada, para usuários que sintam a necessidade de controlar melhor os parâmetros de filtragem da busca.

3. **Simplicidade na sincronização.**
   O ato de se conectar com um servidor remoto, e sincronizar dados que podem conflitar entre si pode ser complicada para o usuário comum. Um cuidado especial deve ser dado para que esta ação seja o mais simples possível.

4. **Executar na plataforma Java.**
   Devido à maturidade da plataforma Java, sua disponibilidade em um grande número de ambientes, e o conhecimento prévio do Desenvolvedor, foi decidido que o sistema deve ser desenvolvido nesta plataforma.

5. **Praticidade de utilização, com atalhos de teclado etc.**
   É de opinião do Desenvolvedor que um dos maiores apelos para o uso ou não de uma aplicação é sua praticidade. Um usuário irá escolher frequentemente uma opção menos robusta caso ela seja mais simples e ágil. Logo é importante que a aplicação seja responsiva, rápida e tenha todas as suas funcionalidades facilmente acessíveis.

6. **Oferecer a possibilidade de Aprendizado.**
   O projeto Amnésia possui fins acadêmicos, e portanto deve contribuir para a expansão do conhecimento do Desenvolvedor. Este requisito irá impactar diretamente na escolha de tecnologias utilizadas, favorecendo a escolha daquelas que oferecerem oportunidades de aprendizado de novos conceitos, ou aprofundamento de conceitos já conhecidos.

CASOS DE USO

A seguir, uma listagem compreensiva dos casos de uso.

1. **Inserir Anotação**
   O usuário cria uma nova anotação no sistema.

2. **Editar Anotação**
   O usuário edita uma anotação já existente no sistema.

3. **Visualizar Versões**
   O usuário visualiza todas as versões existentes de uma anotação.

4. **Promover Versão**
   O usuário cria uma nova anotação a partir de uma versão antiga de uma anotação existente.

5. **Incluir Tag**
   O usuário Associa uma tag a uma anotação.

6. **Excluir Tag**
   O usuário Exclui uma Tag associada a uma Anotação

7. **Busca Padrão**
   O usuário realiza uma pesquisa na aplicação, buscando por um texto especifico.

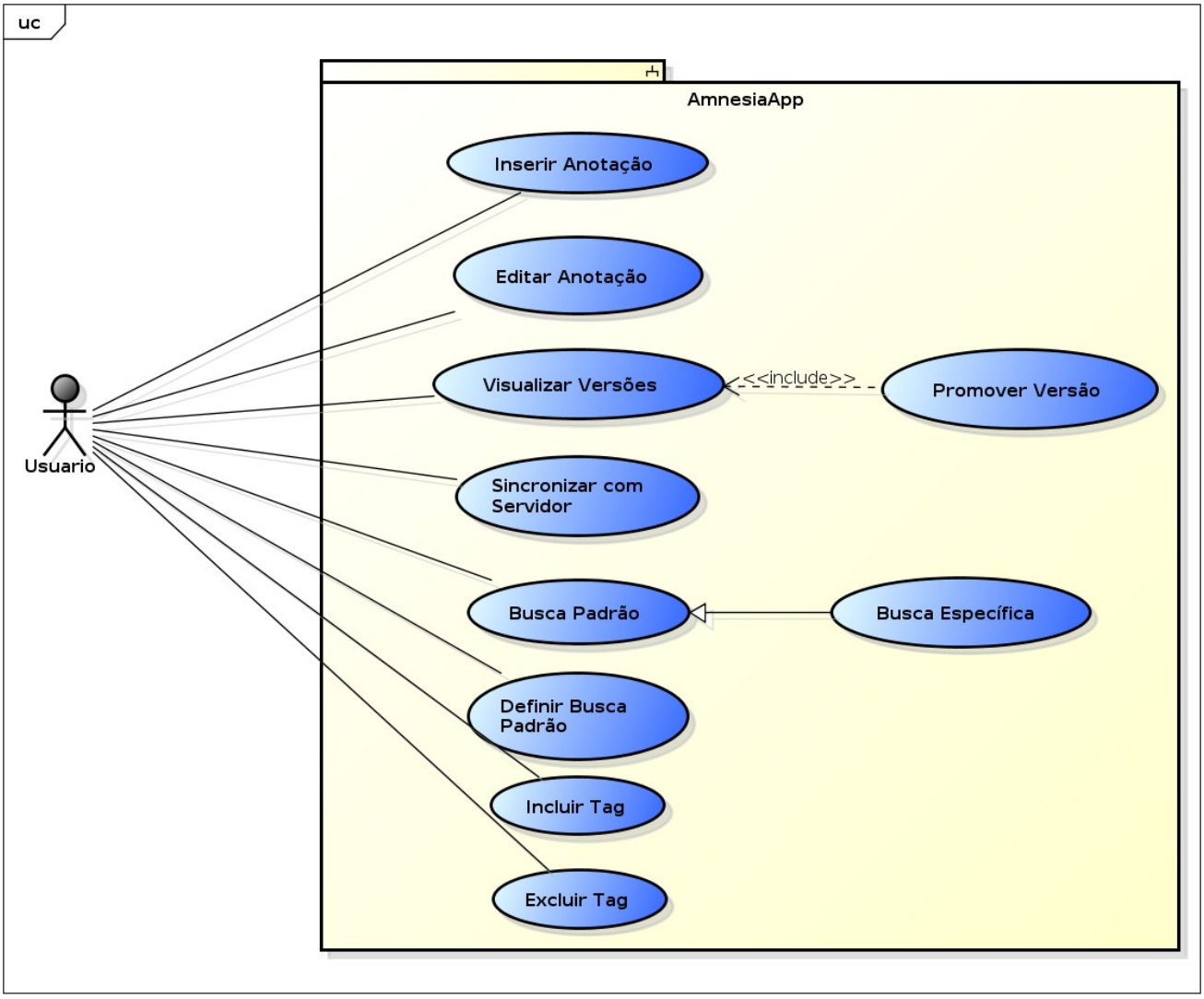
8. **Busca Avançada**
   O usuário realiza uma pesquisa na aplicação, buscando por um texto especifico, e parâmetros de filtragem personalizados.

9. **Definir Busca Padrão**
   O usuário seleciona critérios de filtragem personalizados para serem utilizados sempre que uma Busca Padrão for realizada.

10. **Sincronizar Com servidor**
    O usuário sincroniza os seus dados locais com os dados em um servidor remoto.

DIAGRAMA DE CASOS DE USO

Neste diagrama os casos de uso avantados estão representados de maneira gráfica. Pode se observar que alguns casos de uso estão relacionados. Também é possível visualizar o ator (Usuário) do sistema, e os casos de uso com os quais ele interage.



DESCRIÇÃO DOS CASOS DE USO

As descrições dos casos de uso possuem a finalidade de definir o funcionamento do sistema, detalhando cada funcionalidade da maneira como ela seria utilizada por um usuário.

O modelo utilizado para a descrição dos casos de uso do sistema Amnésia consiste em uma tabela, onde constam todas as informações relevantes ao caso de uso. Para melhor compreensão segue uma legenda para as tabelas de casos de uso:

- **Nome do Caso de Uso:** O nome do caso de uso correspondente.
- **Ator:** Referente ao usuário que ativa o caso de uso.
- **Resumo:** uma descrição resumida do caso de uso.
- **Pré-condições:**Condições necessárias para que o caso de uso seja iniciado.
- **Pós-Condições:**Condições necessárias para que o caso de uso seja considerado bem sucedido.
- **Fluxo Principal:**Descrição do fluxo normal do caso de uso.
- **Ações do Ator:**Ações do usuário durante o fluxo.
- **Ações do Sistema:**Ações do sistema durante o fluxo
- **Fluxo Alternativo:**Descrição de um fluxo alternativo do caso de uso
- **Fluxo de Exceção:**Descrição de um fluxo de erro do caso de uso.

DESCRIÇÃO DE CASO DE USO – INSERIR ANOTAÇÃO

Nome do Caso de Uso	**Inserir Anotação**
Ator	**Usuário**
Resumo	**O usuário cria uma nova anotação no sistema.**
Pré Condições	**Tela principal do programa em foco.**
Pós Condições	**Anotação Salva**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Seleciona a caixa de texto no topo da tela, referente ao nome da nova anotação e a preenche com o texto desejado.	
2- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e a preenche com o conteúdo desejado.	
3- Aciona o atalho de teclado atribuído a ação de salvar a anotação – default: CONTROL+ENTER.	
	4- Cria o registro contendo a anotação no banco de dados.
	5- Atualiza a tela, para exibir a nova anotação junto às demais, e limpa o formulário para a próxima anotação.
Fluxo Alternativo – Anotação sem título	
Ações do Ator	**Ações do Sistema**
1- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e a preenche com o conteúdo desejado.	
2- Aciona o atalho de teclado atribuído a ação de salvar a anotação – default: CONTROL+ENTER.	
	3- Cria o registro contendo a anotação no banco de dados. **O registro será criado sem um nome.**

	4- Atualiza a tela, para exibir a nova anotação junto às demais, e limpa o formulário para a próxima anotação.
Fluxo Secundário – Salvamento automático por tempo	
Ações do Ator	**Ações do Sistema**
1- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e a preenche com o conteúdo desejado, tendo ou não preenchido o campo de nome anteriormente.	
2- Espera sem digitar nado no campo, durante o tempo configurado para ativar o evento de salvamento automático, default: 30 segundos.	
	3- Valida o campo de conteúdo: não deve estar em branco.
	4- Cria o registro contendo a anotação no banco de dados, com o texto constante no formulário no momento do timeout, e o nome se ele existir.
	4- Atualiza a tela, para exibir a nova anotação junto às demais, e limpa o formulário para a próxima anotação.
Fluxo Secundário – salvamento automático por mudança de foco	
Ações do Ator	**Ações do Sistema**
1- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e a preenche com o conteúdo desejado, tendo ou não preenchido o campo de nome anteriormente.	
2- Retira o foco do campo de conteúdo, mudando-o para qualquer outro campo ou janela.	
	3- Valida o campo de conteúdo: não deve estar em branco.
	4- Cria o registro contendo a anotação no banco de dados, com o texto constante no formulário no momento da mudança de foco, e o nome se ele existir.

	4- Atualiza a tela, para exibir a nova anotação junto às demais, e limpa o formulário para a próxima anotação.

DESCRIÇÃO DE CASO DE USO – EDITAR ANOTAÇÃO

Nome do Caso de Uso	**Editar Anotação**
Ator	**Usuário**
Resumo	**O usuário edita uma anotação já existente no sistema.**
Pré Condições	**Tela principal do programa em foco, mostrando a anotação a ser editada.**
Pós Condições	**Nova versão da anotação criada**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Seleciona na tela principal a anotação que deseja editar.	
2- Seleciona a caixa de texto no topo da tela, referente ao nome da anotação e altera o texto conforme desejado.	
3- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e altera o texto conforme desejado.	
4- Aciona o atalho de teclado atribuído a ação de salvar a anotação – default: CONTROL+ENTER.	
	5- Cria o registro contendo a nova versão da anotação no banco de dados.
	6- Atualiza a tela, para exibir a nova versão da anotação no lugar da antiga.
Fluxo Alternativo – Anotação sem título	
Ações do Ator	**Ações do Sistema**
1- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e altera o texto conforme desejado.	
2- Aciona o atalho de teclado atribuído a ação de salvar a anotação – default: CONTROL+ENTER.	
	3- Cria o registro contendo a nova versão da anotação no banco de dados.

	4- Atualiza a tela, para exibir a nova versão da anotação no lugar da antiga.
Fluxo Secundário – Salvamento automático por tempo	
Ações do Ator	**Ações do Sistema**
1- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e altera o texto conforme desejado, tendo ou não alterado o nome anteriormente.	
2- Espera sem digitar nada no campo, durante o tempo configurado para ativar o evento de salvamento automático, default: 30 segundos.	
	3- Valida o campo de conteúdo: não deve estar em branco.
	4- Cria o registro contendo a nova versão da anotação no banco de dados, com o texto constante no formulário no momento do timeout, e o nome se ele existir.
	5- Atualiza a tela, para exibir a nova versão da anotação no lugar da antiga.
Fluxo Secundário – salvamento automático por mudança de foco	
Ações do Ator	**Ações do Sistema**
1- Seleciona a caixa de texto abaixo do nome, referente ao conteúdo da anotação e altera o texto conforme desejado, tendo ou não alterado o nome anteriormente.	
2- Retira o foco do campo de conteúdo, mudando-o para qualquer outro campo ou janela.	
	3- Valida o campo de conteúdo: não deve estar em branco.
	4- Cria o registro contendo a nova versão da anotação no banco de dados, com o texto constante no formulário no momento da mudança de foco, e o nome se ele existir.
	5- Atualiza a tela, para exibir a nova versão da anotação no lugar da antiga.

DESCRIÇÃO DE CASO DE USO – VISUALIZAR VERSÕES

Nome do Caso de Uso	**Visualizar versões**
Ator	**Usuário**
Resumo	**O usuário visualiza todas as versões existentes de uma anotação.**
Pré Condições	**Tela principal do programa em foco, mostrando a anotação a ser visualizada.**
Pós Condições	**Versões exibidas na tela.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Seleciona a anotação desejada, clicando nela na tela.	
	2- Exibe o botão “Histórico”, sinalizando a existência de versões anteriores.
3- Ativa o botão Histórico.	
	4- Exibe as versões anteriores na tela, abaixo da versão atual.

DESCRIÇÃO DE CASO DE USO – PROMOVER VERSÃO

Nome do Caso de Uso	**Promover Versão**
Ator	**Usuário**
Resumo	**O usuário cria uma nova anotação a partir de uma versão antiga de uma anotação existente.**
Pré Condições	**Tela principal do programa em foco, mostrando a versão a ser promovida.**
Pós Condições	**Nova Anotação criada a partir da versão selecionada.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Seleciona a anotação desejada, clicando nela na tela.	
	2- Exibe o botão “Histórico”, sinalizando a existência de versões anteriores.
3- Ativa o botão Histórico.	
	4- Exibe as versões anteriores na tela, abaixo da versão atual.
5- Seleciona a versão desejada ao clicar nela.	
6- Edita a anotação, acionando o caso de uso **Editar Anotação**	
	7- Cria uma nova anotação a partir da versão editada. A nova anotação compartilha todas as versões anteriores à editada com a anotação original.
	8- Exibe a nova anotação na tela.

DESCRIÇÃO DE CASO DE USO – INCLUIR TAG

Nome do Caso de Uso	**Incluir Tag**
Ator	**Usuário**
Resumo	**O usuário Associa uma tag a uma anotação.**
Pré Condições	**Tela principal do programa em foco, mostrando a anotação que receberá a tag.**
Pós Condições	**Tag associada à anotação selecionada.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Aciona o botão **Tags** da Anotação desejada.	
	2- Ativa a edição do campo Tags da anotação.
3- Digita as tags desejadas, separadas por vírgula.	
4- Ativa o comando de Salvar (padrão: CONTROL+ENTER)	
	5- Associa as tags à anotação e grava na base de dados.
	6- Exibe as tags junto a Anotação.
Fluxo Secundário – Salvamento automático por tempo	
Ações do Ator	**Ações do Sistema**
1- Aciona o botão **Tags** da Anotação desejada.	
	2- Ativa a edição do campo Tags da anotação.
3- Digita as tags desejadas, separadas por vírgula.	
4- Espera durante o tempo configurado para salvamento automático da anotação (padrão: 30 segundos).	
	5- Valida o conteúdo do campo de tags: não deve ser vazio.

	6- Associa as tags à anotação e grava na base de dados.
	7- Exibe as tags junto à Anotação.
Fluxo Secundário – salvamento automático por mudança de foco	
Ações do Ator	**Ações do Sistema**
1- Aciona o botão **Tags** da Anotação desejada.	
	2- Ativa a edição do campo Tags da anotação.
3- Digita as tags desejadas, separadas por vírgula.	
4- Muda o foco do cursor para outra área ou janela qualquer.	
	5- Valida o conteúdo do campo de tags: não deve ser vazio.
	6- Associa as tags à anotação e grava na base de dados.
	7- Exibe as tags junto à Anotação.

DESCRIÇÃO DE CASO DE USO – EXCLUIR TAG

Nome do Caso de Uso	**Excluir Tag**
Ator	**Usuário**
Resumo	**O usuário Exclui uma Tag associada a uma Anotação**
Pré Condições	**Tela principal do programa em foco, mostrando a anotação e a tag desejadas.**
Pós Condições	**Tag desassociada da Anotação.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Aciona o botão **"X"** na tag desejada.	
	2- Apaga o relacionamento entre a Tag e a Anotação na base de dados.
	3- Atualiza a tela, e exibe a anotação sem a tag deletada.

DESCRIÇÃO DE CASO DE USO – BUSCA PADRÃO

Nome do Caso de Uso	**Busca Padrão**
Ator	**Usuário**
Resumo	**O usuário realiza uma pesquisa na aplicação, buscando por um texto especifico.**
Pré Condições	**Tela principal do programa em foco.**
Pós Condições	**Resultados da busca exibidos na tela.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Aciona o campo de busca no alto da tela, e digita nele o texto desejado.	
2- Inicia a busca clicando no botão **"Buscar"**, ou digitando ENTER.	
	3- Consulta a base de dados por Anotações que possuam o texto digitado.
	4- Ordena os resultados encontrados por relevância, baseado no campo em que o texto foi encontrado, quantas vezes foi encontrado, e se a anotação é atual ou uma versão antiga.
	3- Atualiza a tela e exibe os registros encontrados na pesquisa. Caso mais de uma versão da mesma anotação seja encontrada, apenas a mais recente é exibida.

DESCRIÇÃO DE CASO DE USO – BUSCA AVANÇADA

Nome do Caso de Uso	**Busca Avançada**
Ator	**Usuário**
Resumo	**O usuário realiza uma pesquisa na aplicação, buscando por um texto especifico, e parâmetros de filtragem personalizados.**
Pré Condições	**Tela principal do programa em foco.**
Pós Condições	**Resultados da busca exibidos na tela.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Seleciona o botão **"Busca Avançada"** na tela do programa.	
	2- Exibe os campos de filtragem para a Busca Avançada.
3- Preenche o formulário de Busca Avançada com os parâmetros de filtragem desejados.	
4- Caso o usuário proceda para concluir a busca o caso de uso **Busca Padrão** é ativado, porém contendo a filtragem selecionada.	

DESCRIÇÃO DE CASO DE USO – DEFINIR BUSCA PADRÃO

Nome do Caso de Uso	**Definir Busca Padrão**
Ator	**Usuário**
Resumo	**O usuário seleciona critérios de filtragem personalizados para serem utilizados sempre que uma Busca Padrão for realizada.**
Pré Condições	**Tela principal do programa em foco.**
Pós Condições	**Critérios de filtragem associados à busca padrão do aplicativo.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Seleciona o botão **"Busca Avançada"** na tela do programa.	
	2- Exibe os campos de filtragem para a Busca Avançada.
3- Preenche o formulário de Busca Avançada com os parâmetros de filtragem desejados.	
4- Aciona o botão **Definir Busca Padrão**.	
	5- Associa os parâmetros valores dos parâmetros presentes no formulário a todas as Buscas Padrão realizadas no futuro, até que uma nova Busca Padrão seja definida.

DESCRIÇÃO DE CASO DE USO – SINCRONIZAR COM O SERVIDOR

Nome do Caso de Uso	**Sincronizar com o Servidor**
Ator	**Usuário**
Resumo	**O usuário sincroniza os seus dados locais com os dados em um servidor remoto.**
Pré Condições	**Tela principal do programa em foco.**
Pós Condições	**Dados sincronizados com o servidor remoto.**
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1- Seleciona o botão **"Sincronizar"** na tela do programa.	
	2- Exibe o diálogo de sincronização, contendo o endereço do servidor, nome de usuário e senha.
3- preenche o formulário com os seus dados. Os dados só precisam ser preenchidos na primeira vez, e serão lembrados nas próximas vezes.	
	4- Realiza a sincronização dos dados locais com os dados do servidor.

DIAGRAMAS DE SEQÊNCIA

Os diagramas de sequência demonstram o fluxo de um caso de uso. Através destes diagramas podemos visualizar os objetos existentes dentro do software, e as interações entre eles. Estas interações, demonstradas no diagrama na forma de mensagens, são abstrações de métodos reais que devem ser implementados no código fonte do programa.

A seguir os diagramas de sequência de todos os casos de uso do projeto Amnésia.

DIAGRAMA DE SEQUÊNCIA - INSERIR ANOTAÇÃO

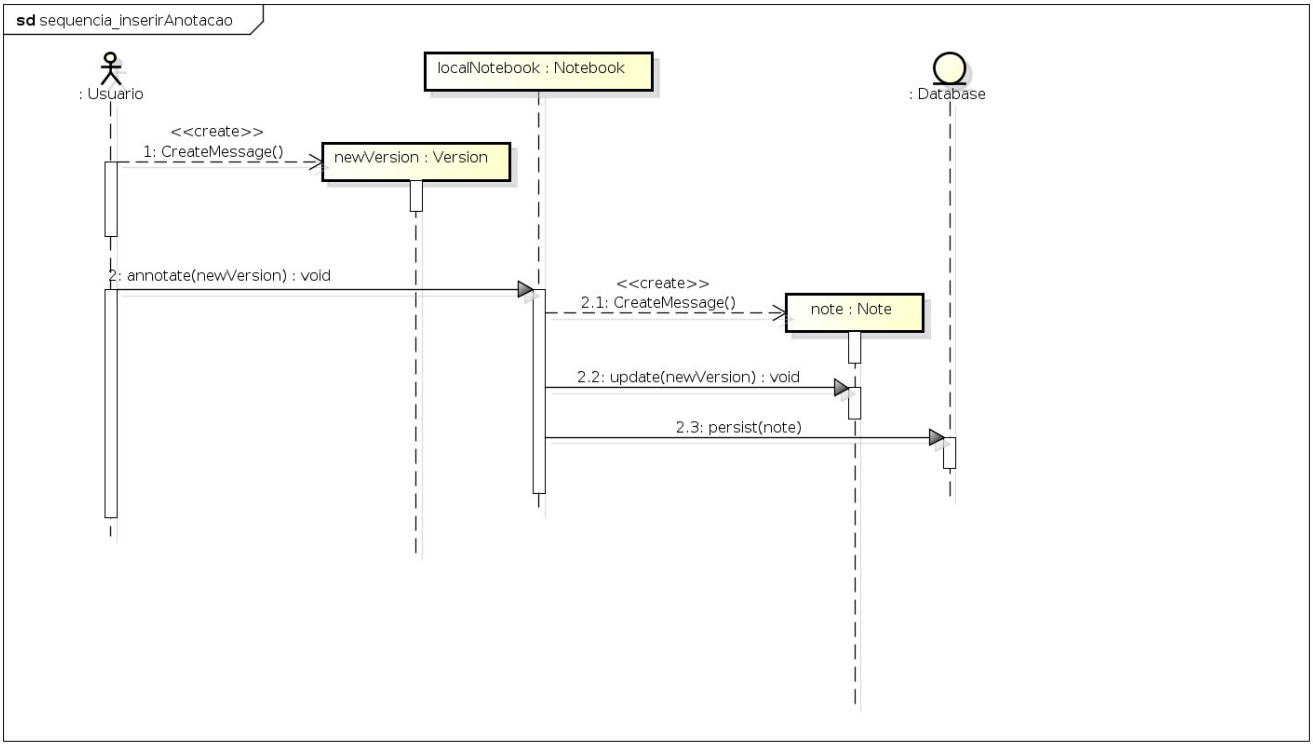


DIAGRAMA DE SEQUÊNCIA – EDITAR ANOTAÇÃO

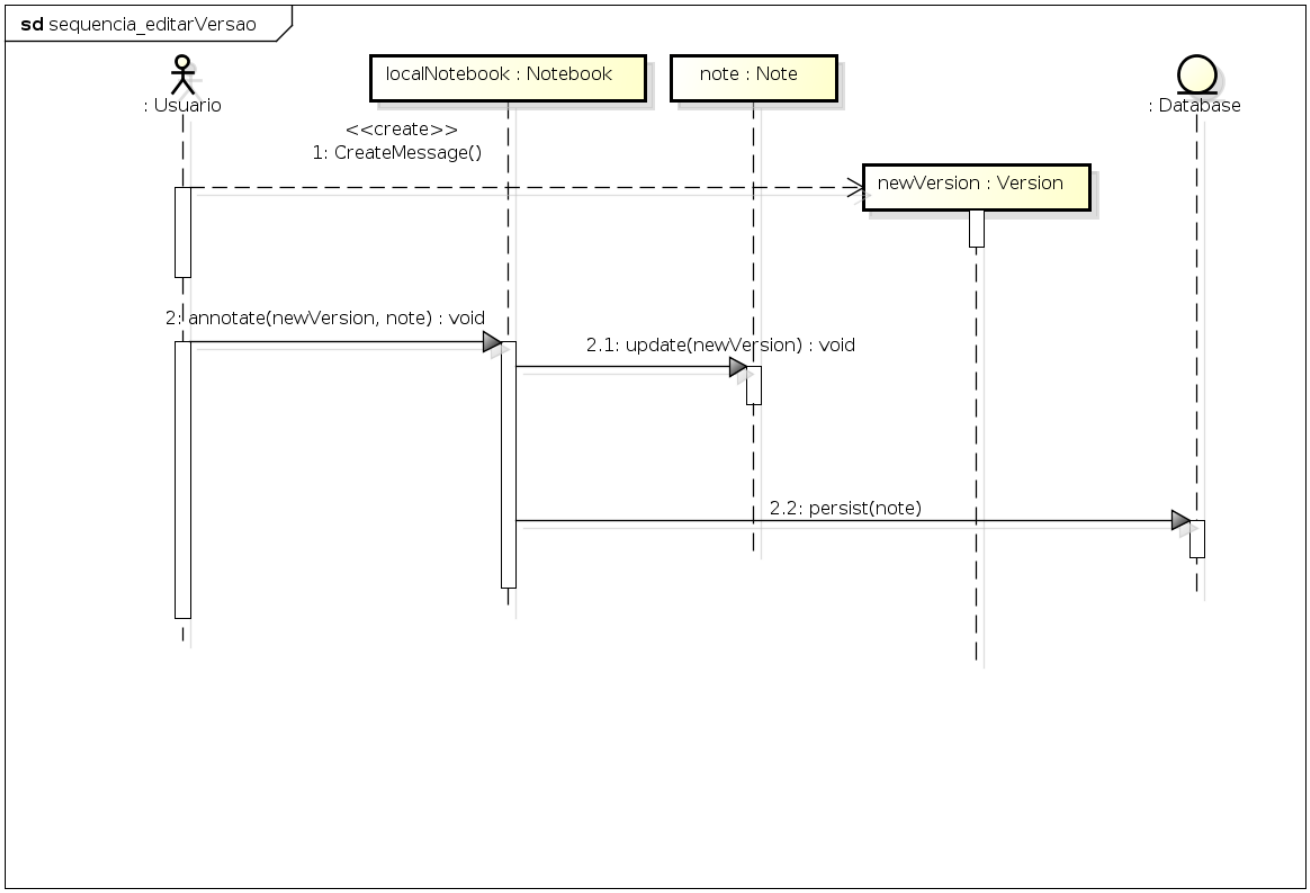


DIAGRAMA DE SEQUÊNCIA – VISUALIZAR VERSÕES

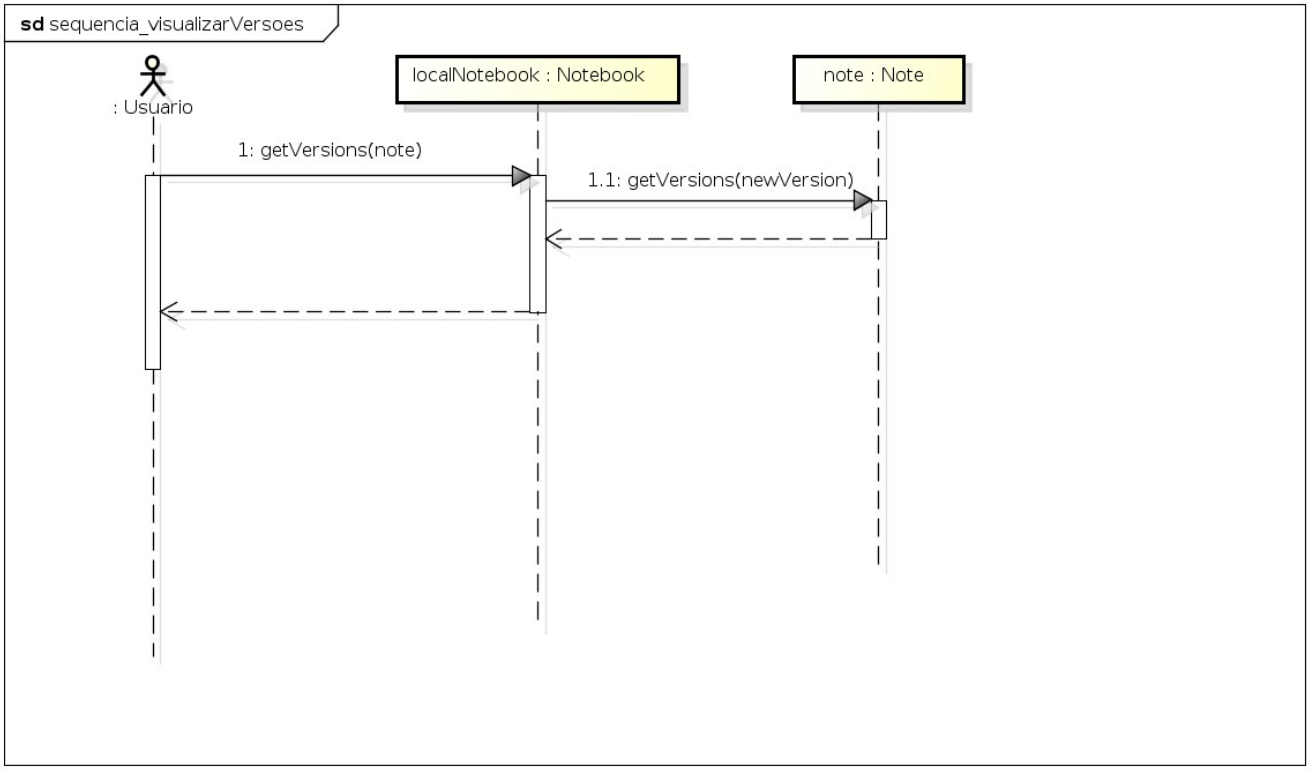


DIAGRAMA DE SEQUÊNCIA – PROMOVER VERSÃO

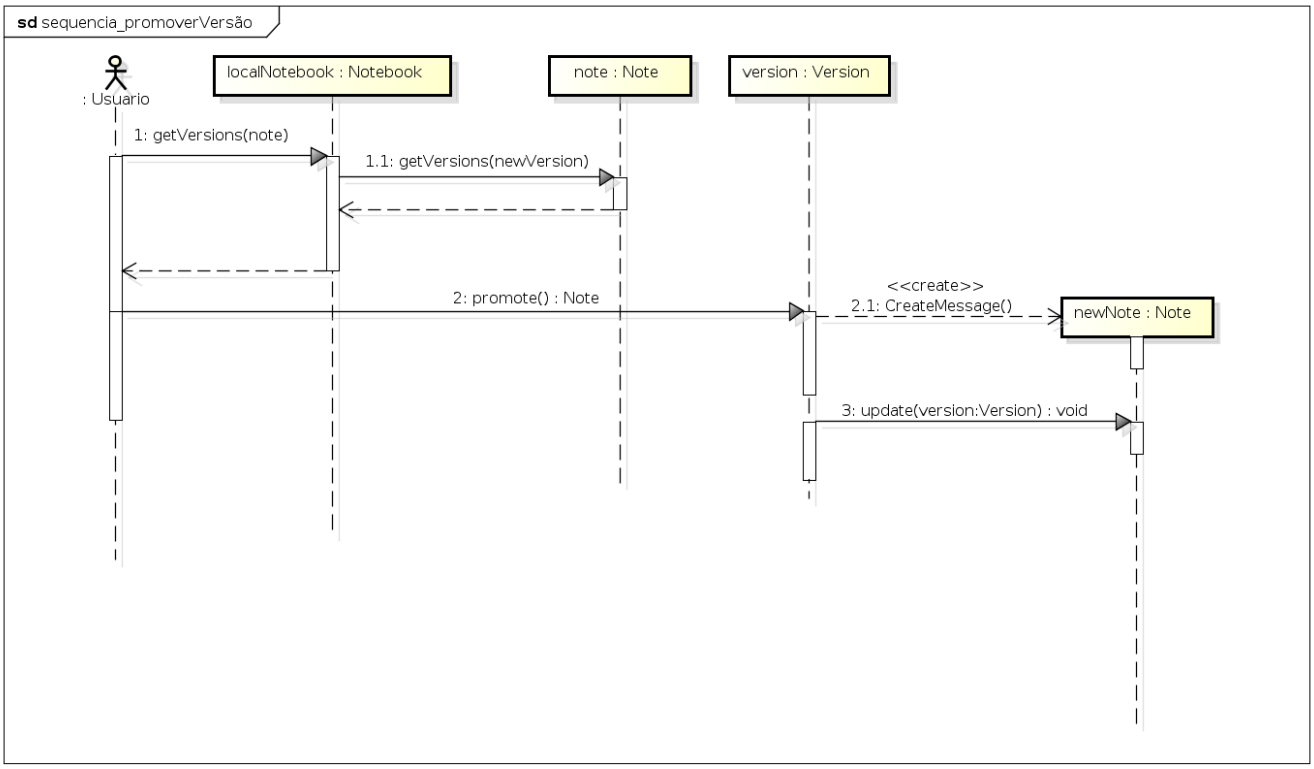


DIAGRAMA DE SEQUÊNCIA – INCLUIR TAG

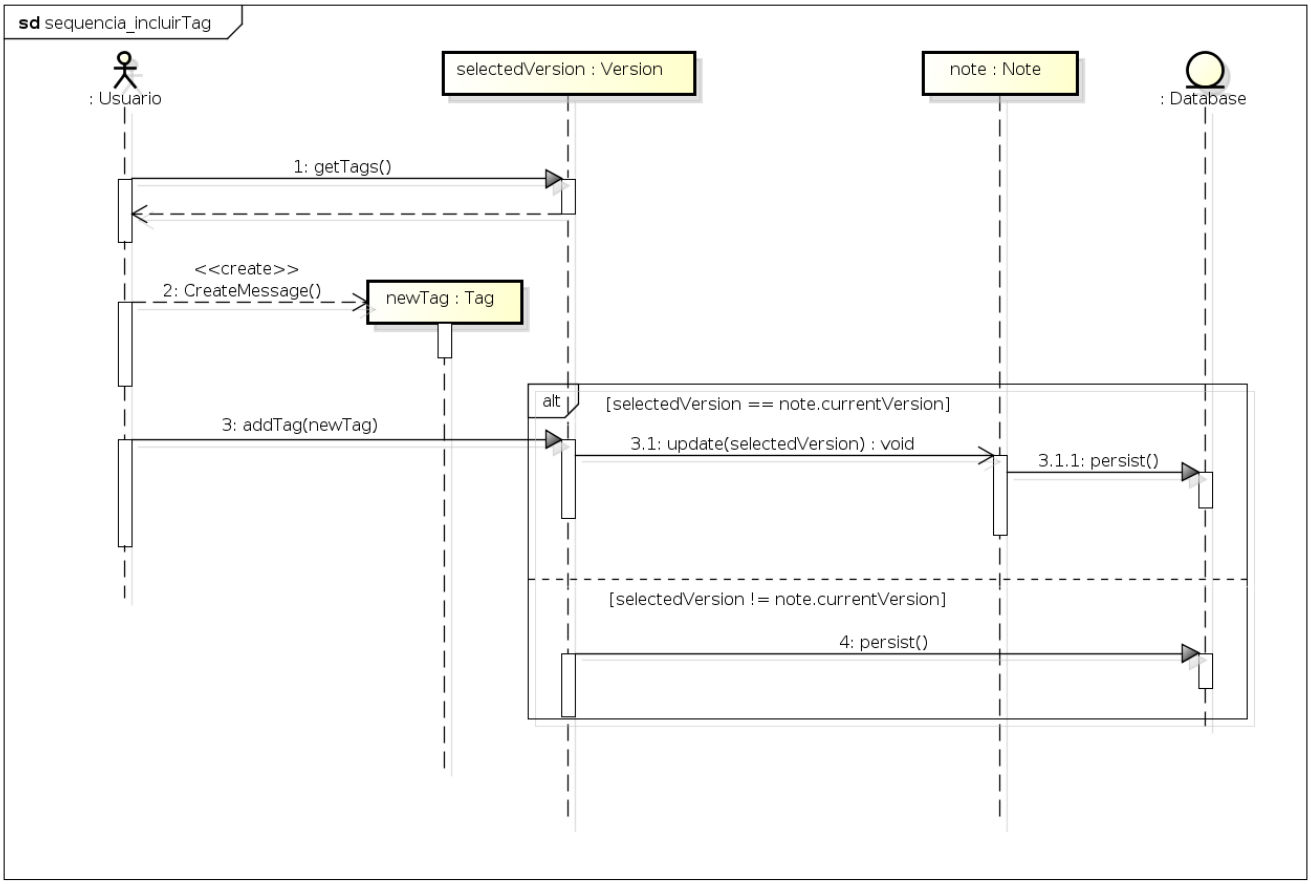


DIAGRAMA DE SEQUÊNCIA – EXCLUIR TAG

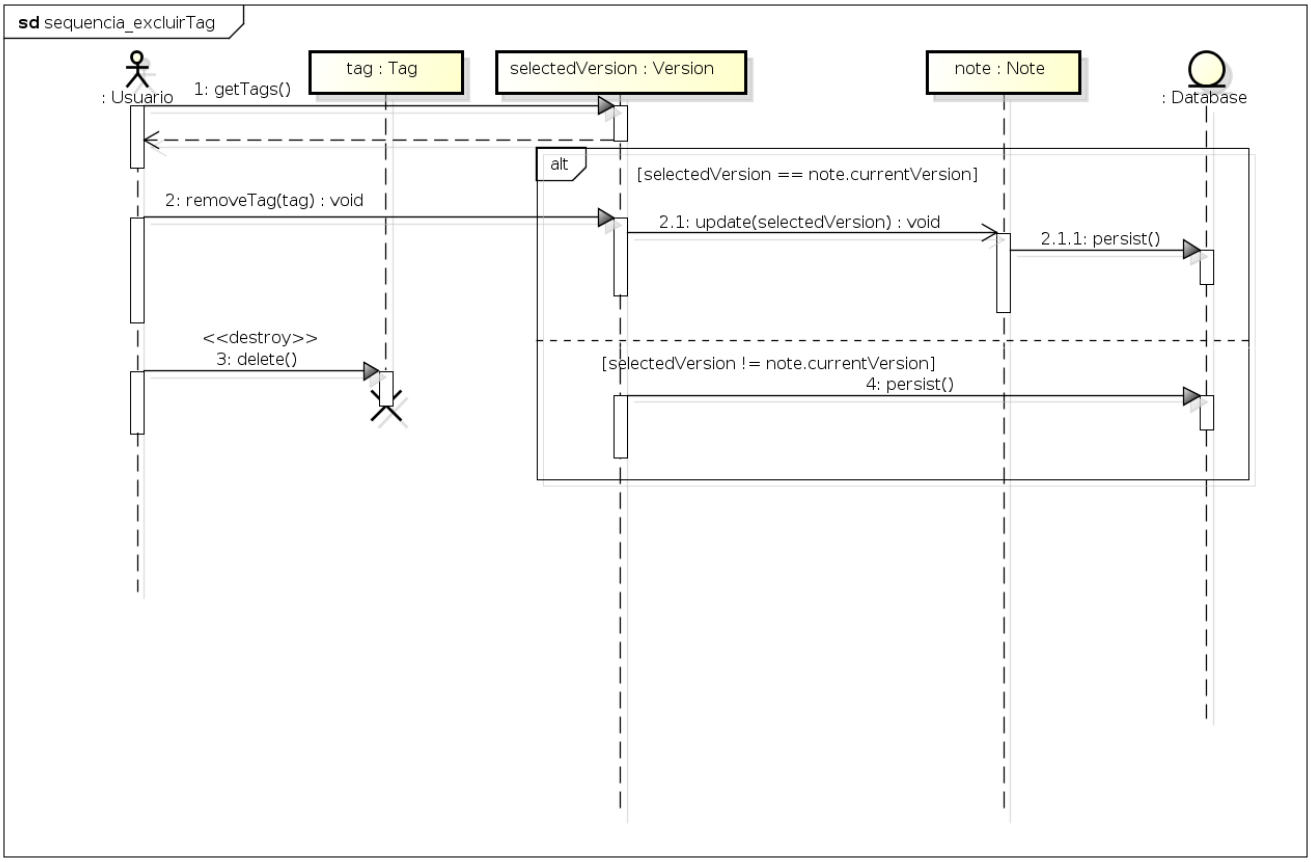


DIAGRAMA DE SEQUÊNCIA – BUSCA PADRÃO

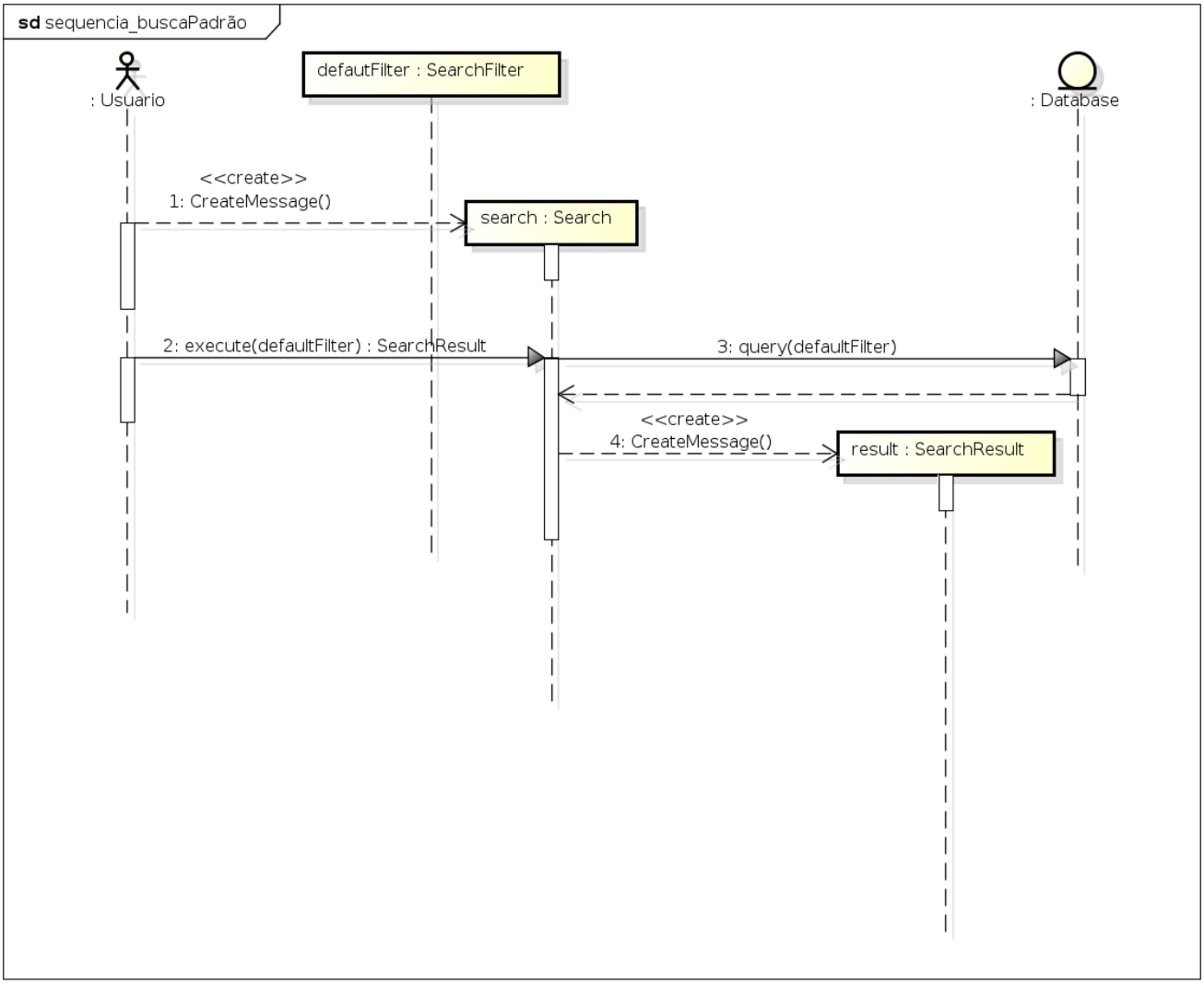


DIAGRAMA DE SEQUÊNCIA – BUSCA AVAÇADA

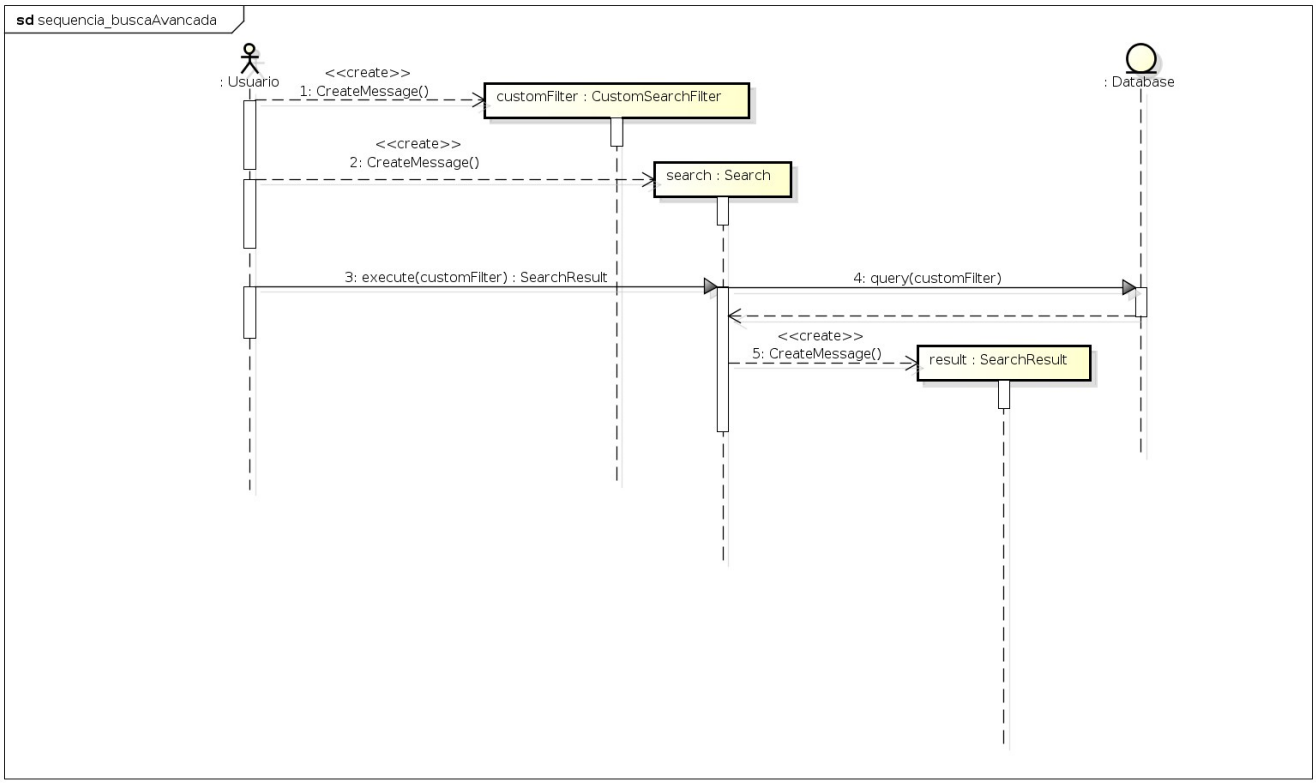


DIAGRAMA DE SEQUÊNCIA – DEFINIR BUSCA PADRÃO

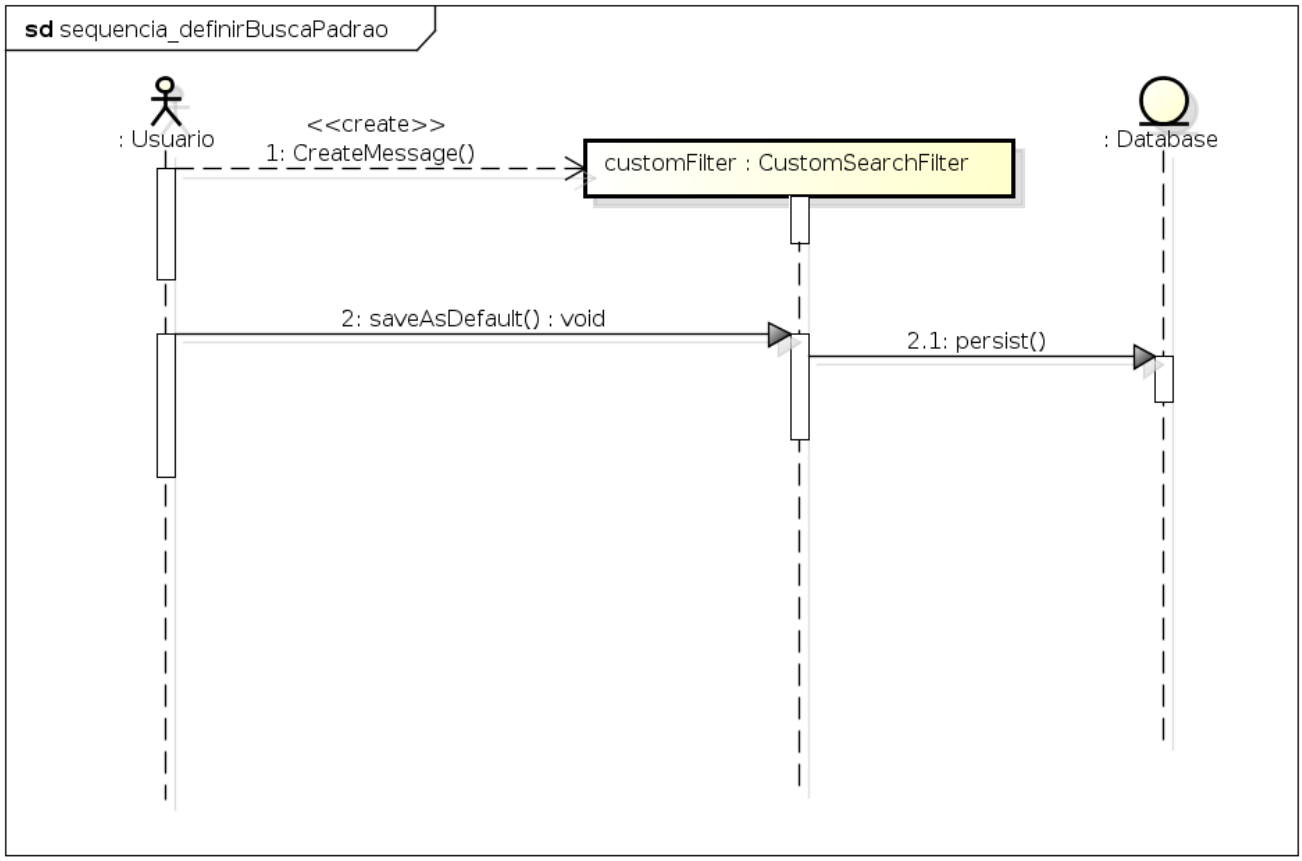
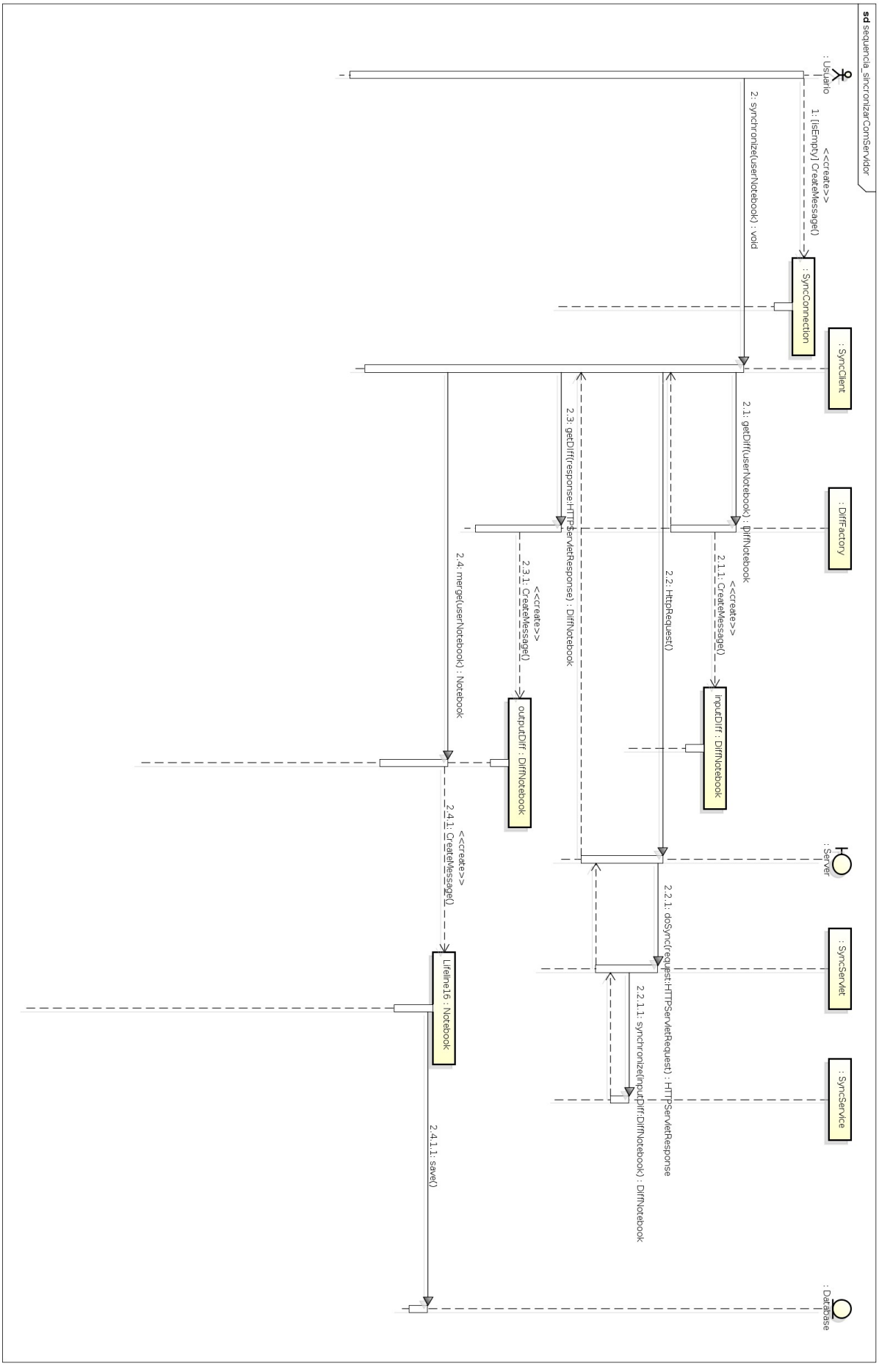


DIAGRAMA DE SEQUÊNCIA – SINCRONIZAR COM O SERVIDOR



ARQUITETURA

ARQUITETURA

A arquitetura de um sistema informatizado é um termo que se refere ao modo como ele é construído, quais componentes ou elementos são utilizados, em qual plataforma ele é executado e como ele se relaciona com outros softwares.

Uma arquitetura bem definida e robusta resulta em um software que também é robusto e eficiente, além de facilitar a sua manutenção.

É importante que a arquitetura de um sistema seja planejada de acordo com as funções e objetivos que se espera do programa. Uma tecnologia ou padrão mal escolhido pode resultar em dificuldades e atrasos no desenvolvimento, além de por em risco o resultado final.

AMBIENTE

PLATAFORMA

Uma Plataforma de Computação pode ser definida como um sistema de hardware e software que possibilitam a execução de aplicações. Assim escolher uma plataforma equivale a escolher em quais dispositivos a aplicação funcionará, e que tipo de sistema operacional e outras pacotes de software devem estar previamente instalados para que ele execute.

Por motivos de familiaridade do Desenvolvedor, o sistema Amnésia será desenvolvido na plataforma Java. Para que a aplicação execute nesta plataforma será necessário que o dispositivo possua uma JVM (Java Virtual Machine) instalada. A JVM abstrai o hardware onde está instalada, fazendo com que as aplicações não precisem se comunicar diretamente com o hardware ou o sistema operacional, apenas com a JVM. Essencialmente a JVM funciona como uma máquina dentro de outra máquina, onde os sistemas podem ser executados de maneira isolada da máquina real.

A plataforma Java oferece diversos benefícios para o desenvolvimento, os principais sendo:

- **Portabilidade.** as aplicações Java podem ser executadas em vários tipos de dispositivos diferentes, desde que o dispositivo tenha instalada uma JVM.
- **Maturidade.** A plataforma Java é comprovadamente estável confiável e rápida, devido aos muitos anos de desenvolvimento que possui.
- **Grande base de componentes.** Devido a sua ampla utilização, existem diversos frameworks, APIs e componentes prontos para o uso em qualquer aplicação.

Visando também o aprendizado, e a produtividade durante a programação do código, foi decidido que apesar da utilização da plataforma Java, a linguagem Java em si não será utilizada no desenvolvimento do código. A principal linguagem no desenvolvimento do Amnésia será a linguagem Groovy, que fornece recursos modernos e dinâmicos de Orientação a Objetos e produtividade. A linguagem Groovy também é muito semelhante a linguagem Java, sendo construída em cima desta, portanto a curva de aprendizado não deve ser íngreme demais a ponto de atrasar o andamento do projeto. Desta forma estão garantidos tanto o aprendizado quanto o cumprimento de prazos e o sucesso no desenvolvimento da aplicação.

BANCO DE DADOS

Os dados do sistema, consistem basicamente de informações pessoais do usuário e dados registrados por ele. Estes dados serão armazenados com a ajuda de um sistema de banco de dados - DBMS(Database Management System). Ao contrário da prática comum na área de desenvolvimento, de atrelar os sistemas criados a um sistema de banco de dados relacional, o sistema Amnésia será construído tendo como base um banco de dados orientado a documentos.

Apesar de os sistemas de bancos de dados relacionais serem ideais em aplicações complexas onde os dados possuem múltiplos relacionamentos entre si, os requisitos do sistema Amnésia demandam que outras características sejam levadas em consideração.

Possuindo um modelo de dados bastante reduzido, o Amnésia não precisa manter registro de extensos relacionamentos entre os seus dados. Os principais requisitos que um banco de dados deve atender para servir ao sistema são:

- **Rapidez no registro e recuperação de dados**, pois a aplicação deve ser muito ágil e responder instantaneamente aos comandos do usuário.
- **Escalabilidade**, já que o uso contínuo ao longo do tempo pode resultar em uma quantidade muito grande de registros de tamanho avantajado a serem gerenciados.
- **Facilidades de busca aos dados**, pois a funcionalidade de pesquisa é essencial à aplicação.
- **Funcionar embarcado**. O usuário não deve precisar instalar e configurar um sistema de banco de dados para utilizar o Amnésia.

Estes requerimentos poderiam ser facilmente ser preenchidos por um banco de dados relacional, porém outro fator determinou na utilização de um sistema alternativo: a natureza acadêmica do projeto, que possui como fim o estudo e aprendizado por parte do Desenvolvedor. Este fator foi decisivo na escolha de um modelo alternativo que apresenta mais oportunidades de aprendizado.

Portanto foi escolhido um DBMS orientado a documentos, pois fornece as oportunidades de aprendizado desejadas e trabalha bem e rapidamente com grandes quantidades de registros de grande tamanho.

Foram analisados diferentes DBMSs orientados a documentos, e no fim foi escolhido o sistema OrientDB. Apesar de não ser o mais conhecido e renomado dos DBMSs orientados a documentos, o OrientDB foi o único que apresentou a importantíssima característica de poder ser facilmente implantado de forma embarcada dentro de uma aplicação, desta maneira permitindo uma instalação simples para o usuário. Outra característica positiva do OrientDB é o fato dele executar na plataforma Java e possuir APIs para esta plataforma, facilitando assim a integração com a aplicação, que será desenvolvida em Java e Groovy.

MODELO DE DADOS

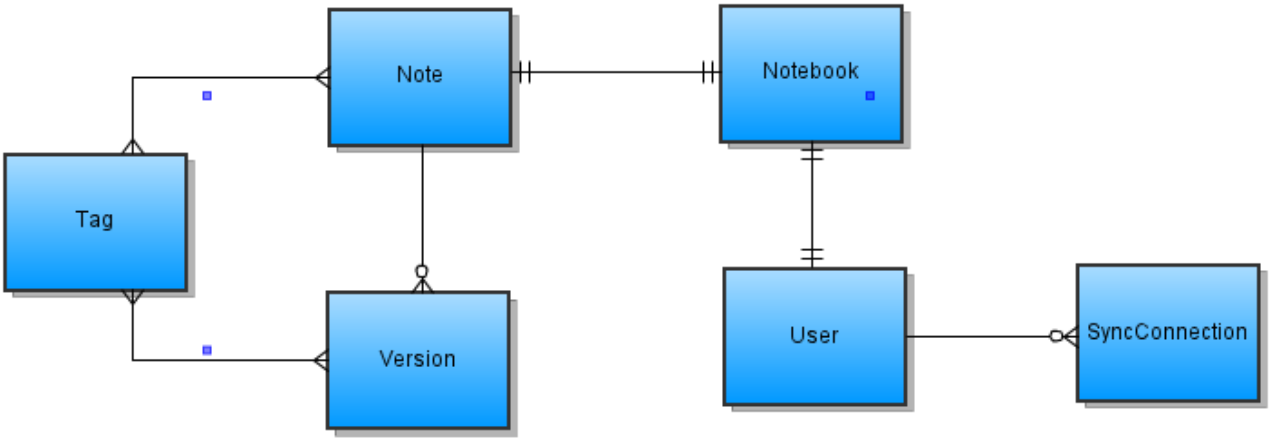
O Modelo de Dados de um sistema é a definição do conjunto informações que devem ser armazenadas, gerenciadas e processadas pelo sistema. O Modelo geralmente é apresentado na forma de uma abstração gráfica contendo figuras que representam os registros ou documentos do sistema e ligações que demonstram como estes registros de relacionam ou se comunicam entre si.

Os Modelos de Dados são geralmente divididos em três categorias:

- **Modelo conceitual** - Representa as regras de negócio sem limitações tecnológicas ou de implementação, contendo apenas a visão de negócio.
- **Modelo Lógico** - Leva em conta limites impostos por algum tipo de tecnologia de banco de dados. (banco de dados hierárquico , banco de dados relacional ,etc.).
- **Modelo Físico** - Leva em consideração limites impostos pelo SGBD (Sistema Gerenciador de Banco de dados) e pelos requisitos não funcionais dos programas que acessam os dados.

Neste documento será abordado apenas o modelo conceitual do sistema Amnésia, apesar de a tecnologia a ser empregada já estar definida.

DIAGRAMA DE ENTIDADE RELACIONAMENTO



O diagrama de Entidade Relacionamento (ER) é a representação gráfica do Modelo de dados. nele constam todos os tipos de dados que serão armazenados na base de dados da aplicação. O diagrama ER possui uma equivalência com o diagrama de classes, porém, o diagrama de classes possui entidades extras que não são persistidas na base de dados, mas que são importantes para o funcionamento do sistema. Neste diagrama ER consta uma representação mais simples das entidades, apenas para indicar quais delas serão persistidas na base. Para uma visão mais completa, incluindo os campos que cada entidade possui, por favor consulte o diagrama de classes.

ANOTAÇÕES E VERSIONAMENTO

Uma anotação é um registro, geralmente textual, criado para consulta posterior. Este registro pode conter apontamentos, observações ou informações sobre algum tópico. Uma anotação geralmente é feita com o propósito de guardar uma informação a qual a pessoa precisa se lembrar, mas não confia em sua própria memória para isto.

A natureza simples e informal das anotações requer que elas sejam mutáveis, pois as informações a serem lembradas podem mudar. Um exemplo disso seria uma receita culinária, onde a pessoa depois de seguir a receita algumas vezes, pode decidir experimentar e modificar algum ingrediente ou instrução. A nova receita deve ser anotada então no lugar da anterior.

Porém a necessidade de se guardar a informação antiga ainda existe: algum familiar da pessoa pode não gostar da receita nova e pedir uma cópia da original. Neste caso, a menos que a pessoa tenha tido a presença de espírito de guardar uma cópia do registro original, a sua recuperação é impossível, e as informações contidas nela estão perdidos.

Este exemplo simples e mundano pode ser extrapolado e aplicado a outros tipos de informação: uma lista de diretórios contendo arquivos de log, informações de conexão em alguma aplicação, instruções para se resolver um problema específico, decisões tomadas em uma reunião. Todos estes exemplos podem ser modificados no futuro, mas a sua mudança não elimina a importância de se guardar as informações antigas.

Para manter um histórico das anotações, a aplicação Amnésia realiza um versionamento das mesmas. As versões antigas são salvas no banco de dados, e são facilmente recuperáveis através de pesquisas, ou simplesmente selecionando uma anotação e acionando a exibição do histórico – a aplicação por padrão apenas exibe a versão mais recente disponível. O conteúdo da versão mais recente é transferido para o registro da anotação em si para maior rapidez na consulta, mas todas as versões possuem seu próprio registro.

As versões de uma anotação nunca podem ser alteradas. Sempre que um usuário alterar uma versão, mesmo que não seja a mais recente da anotação, na realidade estará criando uma nova versão para tomar o lugar da mais recente.

BUSCAS E TAGS

Anotações geralmente são feitas a partir de um contexto específico, por exemplo um projeto, uma aula ou uma reunião. A natureza simplificada de uma anotação faz com que as pessoas geralmente não incluem nela nenhuma informação referente ao contexto ao qual ela pertence. Isso torna muito difícil a recuperação da anotação posteriormente – a pessoa pode encontrar um endereço anotado, mas será que este endereço realmente pertence à pessoa procurada?

Além disto o meio tradicional de se persistir anotações (notas escritas em papel), torna difícil até mesmo encontrar a informação quando ela é necessária, pois ela pode estar em qualquer uma das páginas de um ou mais cadernos, ou até mesmo folhas soltas, sem nenhum tipo de estrutura ou indexamento. O substituto digital mais comum (arquivos de texto simples) também não fazem muito para auxiliar na consulta de algum dado importante pois não possuem nenhum recurso especifico para indexação ou categorização do seu conteúdo.

Esta falta de recursos deixa inteiramente por conta do usuário a tarefa de organizar suas anotações de modo que elas sejam fáceis de se recuperar, e suas informações possam ser contextualizadas e entendidas. Por mais que o usuário se esforce, isso acaba trazendo muito trabalho e é propenso a causar enganos.

SINCRONIZAÇÃO

Sincronização de dados é um processo através do qual duas cópias do mesmo conjunto de dados são comparados, e as diferenças entre eles são replicados de um para o outro seguindo alguns critérios. O objetivo da sincronização é estabelecer consistência entre as duas cópias, para que ambas realmente contenham os mesmos dados.

Um dos objetivos da aplicação Amnésia é disponibilizar os dados do usuário em múltiplos dispositivos. Não seria útil para o usuário possuir diferentes conjuntos de dados em cada computador onde ele instale o Amnésia, portanto se torna necessário fazer a sincronia dos dados entre as múltiplas instalações da aplicação.

Outro ponto interessante da sincronização é a função de backup dos dados. Essencialmente, toda vez que os dados forem sincronizados, uma cópia perfeita deles está sendo guardada de maneira segura em outra máquina, possivelmente em outra localidade geográfica. Desta forma a sincronização fornece uma maneira segura de preservar as anotações do usuário caso ocorra algum acidente com seus dados locais.

DIFFNOTEBOOK

Uma diff é um documento criado a partir da comparação entre outros dois documentos. Quando esta comparação é feita, todas as diferenças encontradas são então registradas na diff. Um DiffNotebook é uma entidade do sistema Amnésia que serve para indicar as modificações realizadas em um Notebook de usuário.

Os DiffNotebooks não são necessariamente criados a partir da comparação entre dois notebooks, e podem ser gerados a partir da consulta de todas as modificações realizadas a partir de uma data específica. Isso é possível por causa da natureza do sistema, onde nenhum dado nunca é apagado ou modificado, logo todas as diferenças encontradas no DiffNotebook são de natureza aditiva, ou seja, registros a serem adicionados a outro notebook.

SYNCSERVER

Para realizar a sincronização do Notepad do usuário é necessária a criação de um servidor de sincronização. A versão inicial do servidor será um módulo simples contendo apenas um webservice RESTful que realizará a sincronização dos Notepads de alguns usuários previamente configurados. A principio funcionalidades de registro e manutenção de contas de usuários não estão previstas para esta versão do servidor.

O SyncServer possuirá uma cópia do Notebook, e quando for requisitado, irá comparar esta cópia com um DiffNotebook fornecido. Os registros contidos no DiffNotebook serão adicionados ao Notebook do servidor, e caso necessário, outro DiffNotebook será retornado, contendo registros que deverão ser adicionados ao Notebook remoto, na máquina que requisitou a sincronização com o servidor.

CONCLUSÃO

CONSIDERAÇÕES FINAIS

A experiência de se desenvolver um projeto do início ao fim é de suma importância na área de TI, e o curso de Sistemas de Informação se propõe a garantir que os graduados possuam a capacidade de realizar esta tarefa.

O desenvolvimento do projeto Amnésia proporciona diversas oportunidade de aprendizado e crescimento. As várias etapas de um projeto desta natureza permitem ao exercício e fixação de muitos conceitos estudados durante o curso de graduação na área de análise, projeto e desenvolvimento e também a descoberta de novos tópicos de interesse. Durante o desenvolvimento foi dada muita importância à busca por estas oportunidades, pois na opinião do desenvolvedor a busca pelo aprendizado é a pedra fundamental da experiência acadêmica e também essencial para o sucesso no mercado de TI.

O sistema criado durante o projeto tem um foco um pouco diferente do que é normalmente visto durante o curso. A intenção é a de demonstrar criatividade e capacidade de se desenvolver algo diferente e único.

GLOSSÁRIO

Software Livre

Software livre, segundo a definição criada pela Free Software Foundation é qualquer programa de computador que pode ser usado, copiado, estudado e redistribuído sem restrições.

Software Proprietário

Software proprietário ou não livre é aquele cuja cópia, redistribuição ou modificação são em alguma medida restritos pelo seu criador ou distribuidor. A expressão foi cunhada em oposição ao conceito de software livre.

REST

A REST (Transferência do Estado Representacional) é pretendida como uma imagem do design da aplicação se comportará: uma rede de websites (um estado virtual), onde o usuário progride com uma aplicação selecionando as ligações (transições do estado), tendo como resultado a página seguinte (que representa o estado seguinte da aplicação) que está sendo transferida ao usuário e apresentada para seu uso.

Banco de dados

Banco de dados (ou base de dados), é um conjunto de registros dispostos em estrutura regular que possibilita a reorganização dos mesmos e produção de informação. Um banco de dados normalmente agrupa registros utilizáveis para um mesmo fim.

Banco de dados Orientado a Documentos

Bancos de dados orientados a documentos são baseado em documentos XML ou JSON que podem ser localizados pelo seu id único ou por qualquer registro que tenha no documento, e contém todos os dados de um registro.

Desenvolvimento Ágil

No desenvolvimento ágil, os projetos adotam o modelo iterativo e em espiral (figura 1.5). Neste processo, todas as fases descritas no modelo em cascata são executadas diversas vezes ao longo do projeto, produzindo ciclos curtos que se repetem ao longo de todo o desenvolvimento, sendo que, ao final de cada ciclo, sempre se tem um software funcional, testado e aprovado. Os ciclos são chamados de iterações e crescem em número de funcionalidades a cada repetição, sendo que, no último ciclo, todas as funcionalidades desejadas estarão implementadas, testadas e aprovadas.

Java	O Java é uma linguagem de programação multiplataforma, com uma sintaxe até certo ponto parecida com o C++, porém com bibliotecas diferentes. Os programas em Java podem ser executados em qualquer sistema operacional, desde que o interpretador esteja instalado.
Groovy	Groovy é uma linguagem de programação orientada a objetos desenvolvida para a plataforma Java como alternativa à linguagem de programação Java. Groovy possui características de Python, Ruby e Smalltalk.
JVM	Máquina virtual Java (do inglês Java Virtual Machine - JVM) é um programa que carrega e executa os aplicativos Java, convertendo os bytecodes em código executável de máquina. A JVM é responsável pelo gerenciamento dos aplicativos, à medida que são executados. Graças à máquina virtual Java, os programas escritos em Java podem funcionar em qualquer plataforma de hardware e software que possua uma versão da JVM, tornando assim essas aplicações independentes da plataforma onde funcionam.
OrientDB	OrientDB é um sistema de gerenciamento de bancos de dados orientados a documentos disponível para a plataforma Java, e que pode ser implantado de maneira embarcada em uma aplicação que rode nesta plataforma.
JSON	JSON (JavaScript Object Notation - Notação de Objetos JavaScript) é uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação JavaScript , Standard ECMA-262 3a Edição -Dezembro - 1999 . JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados.
UML	A UML - Unified Modeling Language - é um modelo de linguagem para modelagem de dados orientado a objetos, usada para especificar, construir, visualizar e documentar um sistema de software.

Anotação	Um apontamento, contendo comentários, observações, lembretes ou outras informações simples. No projeto Amnésia se refere a um registro gerenciado pelo sistema e suas versões.
Versão	Uma Versão é o resultado de uma edição realizada no conteúdo de uma Anotação, que resulta no versionamento da mesma, onde passam a existir a versão atual editada e a versão original anterior.
Tag	Uma tag é uma palavra-chave (relevante) ou termo associado com uma informação (ex: uma imagem, um artigo, um vídeo) que o descreve e permite uma classificação da informação baseada em palavras-chave.
Diff	Diff é um termo utilizado na computação, aplicado ao resultado da comparação entre dois documentos. Esta comparação resulta em um terceiro documento contendo as diferenças (diff) entre os dois documentos comparados.

BIBLIOGRAFIA

LARMAN, Craig. *Utilizando UML e Padrões*. 3 ed. São Paulo. Bookman. 2007.

GAVINHO, Ana. *OO Conceitos*. Rio de Janeiro. Universidade Estácio de Sá. 2003 - Material cedido em formato digital pela professora Patrícia Fiúza.

WEBSITES

FOWLER, Martin et al. Manifesto para Desenvolvimento Ágil de Software. em <http://agilemanifesto.org/iso/ptbr/> Acessado em 26 de Maio de 2012.

STALLMAN, Richard. *O Manifesto GNU*. Free Software Foundation. 2001 em <http://www.gnu.org/gnu/manifesto.pt-br.html> Acessado em 26 de Maio de 2012.

Object Management Group (OMG). *Especificação da linguagem UML*. 2012 em <http://www.omg.org/technology/documents/profile_catalog.htm> Acessado em 26 de Maio de 2012.

Microsoft One Note 2010. Microsoft. 2010. em <http://office.microsoft.com/pt-br/onenote/> Acessado em 26 de Maio de 2012.

Remember Everything with Evernote. Evernote Corp. em <http://evernote.com/> Acessado em 26 de Maio de 2012.

Comparison of notetaking software. Wikipedia. em <http://en.wikipedia.org/wiki/Comparison_of_notetaking_software> Acessado em 26 de Maio de 2012.