

ÍNDICE

1. INTRODUÇÃO.....	1
1.1. SITUAÇÃO PROBLEMÁTICA.....	2
1.2. HIPÓTESES	2
1.3. JUSTIFICATIVA	2
1.4. OBJECTIVOS DA PESQUISA	3
1.4.1. Objectivo Geral.....	3
1.4.2. Objectivos Específicos	3
2. METODOLOGIA DE PESQUISA	4
2.1. TIPO DE PESQUISA.....	4
2.2. CAMPO DE ESTUDO.....	4
2.3. DELIMITAÇÃO E LIMITAÇÃO DO ESTUDO	5
2.3.1. Delimitação.....	5
3. ANÁLISE DE DADOS	5
3.1. METODOLOGIA DE DESENVOLVIMENTO.....	5
3.2. FERRAMENTAS E LINGUAGENS A UTILIZAR	7
3.2.1 UML	8
3.2.3 StarUML.....	8
3.1.3 HTML, CSS e JavaScript	8
4. RESULTADOS	9
4.1 REQUISITOS FUNCIONAIS.....	9
4.2 REQUISITOS NÃO FUNCIONAIS	10
4.3 DIAGRAMA DE CASO DE USO	12
4.4 DIAGRAMA DE CLASSE.....	13
4.5 DIAGRAMA DE ENTIDADE E RELACIONAMENTO.....	14
5. CRONOGRAMA DE EXECUÇÃO	15
6. REFERÊNCIAS BIBLIOGRÁFICAS	16

1. INTRODUÇÃO

O crescimento e aperfeiçoamento da tecnologia fez com que nos últimos tempos ela se tornasse constantemente presente no cotidiano das pessoas, desde o acordar com o despertador até o enviar uma mensagem online. Com o avanço tecnológico o que não se realizava na rede, se tornou seguro e habitual para as pessoas como movimentações bancárias, compras em sites e aplicativos, agendamento de serviços e armazenamento de arquivos em servidores online, por exemplo. Antes, muitas atividades exigiam que as pessoas se deslocassem para um determinado local e aguardassem pelo atendimento dos funcionários, o que demandava grande quantidade de tempo. A automatização dessas atividades gerou comodidade e agilidade para que a população pudesse obter maior rendimento na rotina, pois operações que demandavam tempo passaram a ser feitas em poucos instantes com o uso de softwares (KOHN, MORAES, 2007).

Aplicações Web, por exemplo, disponibilizam ferramentas que permitem ao usuário do sistema executar tarefas antes complexas em um curto espaço de tempo. Desde que se tenha um dispositivo com acesso à internet é possível se conectar no sistema e fazer algum trabalho, independentemente de onde o usuário esteja (SANTOS, 2018).

O TCC, que significa Trabalho de Conclusão de Curso, é uma avaliação que acontece quando a licenciatura está chegando ao fim. Ele tem o objetivo de fazer com que o aluno demonstre o que aprendeu, desde o início dos estudos mediante um trabalho. Sendo uma parte muito importante de vários cursos, o aluno se esforça para fazê-lo da maneira correta, pois serve para aferir o conhecimento e vê por outro ângulo como foi a própria evolução durante todo o processo de aprendizagem. Desta forma, é possível saber se tudo foi absorvido e entendido corretamente.

1.1. SITUAÇÃO PROBLEMÁTICA

A elaboração do Trabalho de Conclusão de Curso (TCC) na Universidade Técnica de Angola(UTANGA) é um processo comum para os alunos que estão prestes a terminar a sua licenciatura e que queiram obter o diploma. Este processo embora sendo pessoal, envolve geralmente os mesmos elementos: coordenador, orientador, aluno, trabalho proposto e concretizado e uma mesa de júris.

Actualmente, todo o processo de acompanhamento dos trabalhos de conclusão de curso é feito manualmente, através de formulários e relatórios preenchidos pelos alunos, professores, orientadores e coordenador de TCC. Algumas informações consideradas importantes são postadas no grupo de whatsapp criado para o efeito ou são enviados por email. Essa sistemática de trabalho acaba ocasionando vários problemas, dentre eles a descentralização das informações e o grande volume de papel manuseado pelos envolvidos. No entanto desta situação advem certas dificuldades dos orientadores, professores, coordenadores de cursos e alunos, para gerir Trabalhos de Conclusão de Curso (TCC), Monografias. Os cronogramas costumam atrasar e os coordenadores e orientadores têm dificuldades para controlar o progresso dos projetos.

Como melhorar a gestão e o controle do processo dos trabalhos de conclusão do curso da UTANGA garantindo a centralização de dados e a gestão mais eficiente?

1.2. HIPÓTESES

Se implementarmos um sistema web para gestão e controle do processo dos trabalhos de conclusão de curso (TCC), irá possibilitar que alunos, professores, orientadores e coordenador de TCC tenham uma melhor interação, e informação disponível a tempo, evitando atrasos no cronograma.

1.3. JUSTIFICATIVA

Para quem passa ou já passou pelo mesmo, sabe o quão desafiador é o processo do TCC, do início até a conclusão do trabalho, é desafiante não apenas para quem o faz, mas também para quem coordena, orienta. Os desafios começam desde a escolha do tema, interação entre os intervenientes deste processo(coordenador, orientador, aluno, etc...), bem como ter a informação a tempo.

Sendo parte constituinte do processo, passando por dificuldades e observando os meus colegas passarem pelas mesmas, motivou-me a acolher este desafio. Por outra, aplicar na prática os conhecimentos adquiridos tanto dentro como fora da instituição dando resolução a um problema real tendo em mente que vai beneficiar tanto a instituição como os alunos, e a possibilidade de

conhecer outras ferramentas científicas, tem funcionado como combustível para continuar essa empreitada.

Com o sistema de gestão de trabalho de conclusão do curso (TCC) implementado facilitará os alunos dando a eles a possibilidade de enviar seu tema para aprovação ou fazer a busca por temas disponíveis para investigação, escolher tutor, dentre outras funções.

1.4. OBJECTIVOS DA PESQUISA

1.4.1. Objectivo Geral

Desenvolver um sistema web de gestão de trabalho de conclusão do curso (TCC)

1.4.2. Objectivos Específicos

A fim de alcançar o resultado esperado, foi definido alguns objectivos específicos:

- Estudar o processo básico de elaboração, desenvolvimento e apresentação de monografias;
- Compreender o funcionamento de um sistema web de gestão de TCC;
- Realizar levantamentos de requisitos;
- Elaborar a arquitetura de Software;
- Estudar metodologias de desenvolvimento de software;

2. METODOLOGIA DE PESQUISA

2.1. TIPO DE PESQUISA

O tipo de pesquisa realizada no presente trabalho foi descritiva e exploratória em relação aos objetivos, visto que, segundo Gil (1996), proporciona uma proximidade com a questão.

No sentido de, construir hipóteses, a metodologia inclui entrevistas com pessoas directamente envolvidas, sobre a forma como fazem a monografia e as experiências que tiveram no processo.

Os procedimentos de coleta dos dados supracitados, foi através de pesquisa bibliográfica e documental, com abordagem quantitativa e qualitativa, com o intuito de relacionar os dados para a interpretação.

Ao longo da construção da pesquisa, o primeiro procedimento realizado diz respeito à obtenção de nome do autor, curso, tema, resumo e ano de defesa das monografias. Em seguida, definiram-se as categorias de Engenharia Informática, para melhor evidenciar o perfil dos TCC e monografias.

Os dados analisados foram transformados em gráficos e infográficos para melhor visualização. Assim, os dados foram cruzados e interpretados tanto em quantidade como em qualidade para se constatar o Perfil dos trabalhos académicos de Engenharia da Universidade.

2.2. CAMPO DE ESTUDO

A Universidade Técnica de Angola (UTANGA) é uma universidade angolana com sede no bairro do Capolo II, Quilamba Quiaxi, na província de Luanda.

Fundada aos 7 de maio de 2007, tendo como actual reitor(a), Dr. Ilídio Pascoal Simão.

Instituições orgânicas

As unidades orgânicas ofertam os seguintes cursos:

Faculdade de Engenharias Em nível de graduação, em 2017, ministrava os cursos de: Arquitectura e Urbanismo, Engenharia de Telecomunicações e Electrónica, Engenharia de Geologia e Minas, Engenharia Civil Engenharia do Ambiente, Engenharia Informática, Engenharia de Minas.

Faculdade de Letras e Ciências Sociais Em nível de graduação, em 2017, ministrava os cursos de: Relações Internacionais Psicologia, Língua e Literatura Inglesa, Direito. Faculdade de

Gestão e Ciências Económicas Em nível de graduação, em 2017, ministrava os cursos de: Gestão Contabilidade e Finanças.

2.3. DELIMITAÇÃO E LIMITAÇÃO DO ESTUDO

2.3.1. Delimitação

A pesquisa e elaboração de trabalho científico é transversal a diversas áreas de conhecimento, tendo em conta a abrangência do assunto, delimitou-se o escopo desta investigação. Assim sendo, temos como nosso objecto de estudo a UTANGA.

3. ANÁLISE DE DADOS

3.1. METODOLOGIA DE DESENVOLVIMENTO

O Objetivo de uma metodologia é definir de forma clara quem faz o que, quando, como, e até mesmo onde, para todos os que estejam envolvidos diretamente ou não ao desenvolvimento de um software. Deve definir também qual o papel dos técnicos, dos usuários, e o da administração da empresa no processo de desenvolvimento.

As metodologias podem ser ágeis ou pesadas, mas para o presente trabalho vai se utilizar as metodologias ágeis pelas suas especialidades que serão detalhadas a seguir.

Metodologias Ágeis

Os métodos ágeis são padrões que procuram acelerar o processo de desenvolvimento de um projeto, de forma coesa e funcional BeCode, (2016).

Este manifesto foi criado para valorizar os seguintes pontos:

- Indivíduos e suas interações são mais importantes que processos e ferramentas.
- Software que funciona importa mais que uma documentação abrangente.
- Colaboração com o cliente vale mais do que negociação de contratos.
- Responder às mudanças é melhor do que seguir planos

Isso não significa que processos, documentação, contratos e planos sejam descartados pelas metodologias ágeis, mas sim que eles têm uma importância secundária no processo de desenvolvimento. Quando o foco está nas pessoas, software, clientes e na resposta às mudanças, o projeto sai do planejado para a realidade com mais velocidade e qualidade.

A principal diferença das metodologias ágeis é a adaptabilidade, ao contrário de métodos preditivos. Isso significa que, durante o processo de desenvolvimento, podem haver alterações

no projeto e que estas alterações serão adaptadas de alguma forma, ao invés de serem postergadas para o fim do projeto.

Quando são utilizados métodos preditivos de desenvolvimento, é necessário um grande planejamento prévio, que é custoso e muitas vezes acaba perdendo o sentido, já que alterações são algo necessário durante o processo de desenvolvimento. Nas metodologias clássicas não existe espaço para adaptações ao longo do caminho, o que muda completamente na metodologia ágil.

Outra vantagem das metodologias ágeis é o fato de elas trabalharem com o conceito de feedback constante. Isto permite que a qualidade do software aumente a cada ciclo, já que estão sempre havendo testes e comunicações com os stakeholders do projeto (partes envolvidas: clientes, colaboradores, gestores, etc.).

As metodologias ágeis também proporcionam rápidas entregas para o cliente final, diferente da metodologia tradicional onde o software só era entregue após estar 100% pronto. Com as metodologias ágeis a todo momento, pequenas entregas são realizadas, isso para manter o caráter da adaptabilidade. Em outras palavras, com isso o cliente consegue testar a aplicação, enxergar pontos positivos, pontos a melhorar, etc.

Existem várias metodologias ágeis como Xtreme Programming, OpenUp, Scrum, Lean, dentre estas foi escolhida a metodologia OpenUp pelas suas características apresentadas abaixo.

Openup é uma metodologia que segue o Processo Unificado de engenharia de software porém com uma cultura aberta, ele utiliza a filosofia ágil a qual foca na natureza colaborativa de desenvolvimento de software. A metodologia é construída através de processos simples os quais podem ser expandidos para quaisquer tipos de softwares. O Openup foi criado como um projeto de exemplo aplicando as melhores práticas de engenharia de software do EPF, assim sendo, o Openup é um conjunto das metodologias ágeis como Scrum e XP (Extreme Programming) somado a um pouco da arquitetura do Processo Unificado, Diedrich, (2011).

A maioria dos projetos de desenvolvimento de software possui tempo médio de duração de três a seis meses com equipes variando entre três e dez pessoas. Podemos perceber que este tipo de projeto não comporta a robustez de processos de desenvolvimento como o RUP e, em sua maioria, não utiliza todos os recursos presentes em processos de desenvolvimento como o XP.

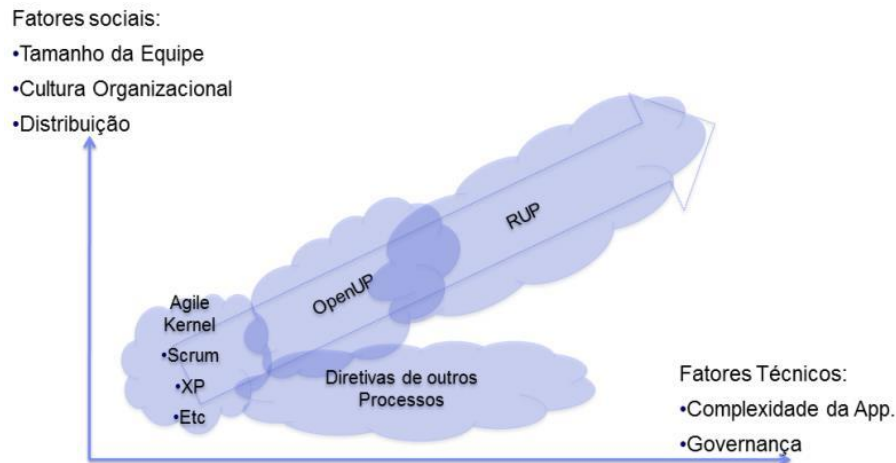


Figura 1-1 Perspectiva Openup Fonte: Diedrich, 2011

A seguir estão descritos os quatro pilares nos quais a metodologia se sustenta para trazer bons resultados e o conjunto de melhores práticas para ajudar a atingir o resultado e uma associação entre os princípios descritos e os princípios do Manifesto Ágil.

- Colaborar para alinhar os interesses e compartilhar os conhecimentos;
- Balancear as prioridades concorrentes para maximizar os valores dos stakeholders;
- Focar primariamente na arquitetura visando minimizar os riscos e planejar o processo;
- Envolver os stakeholders para obter contínuo feedback do desenvolvimento.

Depois de definida a metodologia de desenvolvimento, é necessário definir as ferramentas de desenvolvimento para a solução, esse assunto será abordado de forma profunda no próximo ponto.

3.2. FERRAMENTAS E LINGUAGENS A UTILIZAR

A ferramenta escolhida para fazer a modelagem do sistema foi a StarUML utilizada para desenvolver os diagramas de Caso de Uso, Diagrama de Classe e o Diagrama de Atividade, e para o diagrama de entidade relacionamento, foi utilizada a ferramenta MySQL Workbench. Por ser um sistema web, optou-se por trabalhar com o trio JavaScript, CSS e HTML unidos ao Bootstrap que é um framework com código fonte aberto para o front-end, e para o back-end, a aplicação web será construída sobre a estrutura do ASP.NET MVC com a linguagem de programação C#, já para o banco de dados, será utilizado o MySQL sendo gerenciado e mapeado pelo Entity Framework.

3.2.1 UML

De acordo com Guedes (2011, p.19 até 21) a UML se tornou uma linguagem padrão de modelagem utilizada internacionalmente pela indústria de engenharia de software, cujo o objetivo é auxiliar os engenheiros a definirem características do sistema. Por mais simples que seja o sistema deve ser modelado antes de iniciar a sua implementação, pois os sistemas tendem a crescer, aumentando a sua complexidade. Assim é importante um sistema manter uma documentação detalhada para que possa ser mantido com facilidade, rapidez e ao corrigir um erro antigo não deve produzir novos erros.

Guedes (2011, p. 25) ainda afirma que a modelagem possui diversos benefícios, tais como prazos e custos. “Uma boa modelagem auxilia a estimar a complexidade de desenvolvimento de um sistema, e isso, por sua vez, ajuda – e muito – a determinar o prazo final em que o software será entregue”.

Na UML são desenvolvidos diversos diagramas que analisam sob diferentes pontos de vista o sistema como um todo ou parte dele, sendo modelados em camadas, podendo focar no comportamento de um processo específico, definição de um determinado algoritmo, estrutural do sistema (GUEDES, 2011, p. 30).

3.2.3 StarUML

É um software utilizado para modelagem ágil e precisa de sistemas, desenvolvido na Coreia do Sul pela empresa MKLabs, permitindo desenvolver diversos diagramas de modelagem de sistemas.

3.1.3 HTML, CSS e JavaScript

Em inglês HTML é Hyper Text Markup Language que em português significa, linguagem de marcação de hipertexto (SILVA,2008,p.26), o seu maior objetivo é especificar o conteúdo de páginas Web; o CSS Cascading Style Sheets na tradução folha de estilo em cascata que tem como o objetivo aprimorar o visual da página e por fim o JavaScript que especifica o comportamento da página (FLANAGAN, 2013, p.1), todos os três são interpretados pelo navegador para formar a página Web.

4. RESULTADOS

4.1 REQUISITOS FUNCIONAIS

Um requisito funcional representa algo que o sistema deve fazer, ou seja, uma função esperada do sistema que agregue valor a seus usuários, exemplos típicos incluem a emissão de relatórios e a realização e manutenção de cadastros, Xexéo, (2007).

Depois de entendido os conceitos de requisitos funcionais, eis abaixo alguns requisitos funcionais do sistema:

Tabela 3- Requisitos Funcionais do Sistema

Ref	Nome	Descrição
RF1	Efetuar login	Função que permite efetuar o login com usuário e senha e redireciona o usuário para a tela relacionada ao seu perfil que pode ser de professor ou estudante
RF2	Editar perfil	Função que permite o usuário adicionar informações adicionais ao seu perfil
RF3	Submeter a proposta de trabalho	Função que permite ao discente submeter, antes do desenvolvimento, uma proposta de trabalho que estará disponível para a visualização docente. A proposta de trabalho é composta de título, descrição, preferência de trabalho (monografia) e indicação do nome de possíveis orientadores. Após a submissão de uma proposta de trabalho, o status dele é definido como “Proposta submetida”.
RF4	Submeter a versão final do TCC	Função que permite ao discente submeter a versão final do projeto. A submissão é composta apenas do arquivo do trabalho
RF5	Visualizar propostas de trabalho	Função que permite ao docente ter acesso às propostas de trabalhos submetidas pelos discentes. As propostas serão categorizadas baseadas na indicação de orientadores e, dependendo desse fator poderão possuir as opções de aceitar ou rejeitar a orientação. Quando um trabalho for aceito por um professor, o status dele é definido como “Proposta aceita”.
RF6	Aceitar proposta	Função que permite o estudante confirmar a orientação da proposta enviada previamente, após ela ser aceita por um professor. Quando isso acontece, o docente ganha permissão de emitir o termo de aceite e o status do trabalho é definido como “Trabalho em desenvolvimento”.
RF7	Cadastrar banca	Função que permite ao orientador cadastrar qual será o dia e horário das bancas, além de cadastrar quem participará da banca. Após o cadastro da banca, o status do trabalho é definido como “Trabalho final”.
RF8	Atualizar banca	Função que permite ao orientador atualizar qual será o dia e horário das bancas, além de atualizar quem participará da banca
RF9	Visualizar banca	Função que permite ao estudante/docente visualizar quais são os docentes que compõe uma banca
RF10	Eliminar banca	Função que permite ao coordenador eliminar uma banca formada
RF11	Cadastrar usuário	Função que permite ao gestor cadastrar usuário

RF12	Pesquisar usuário	Função que permite ao gestor pesquisar usuário
RF13	Eliminar usuário	Função que permite ao gestor eliminar usuário
RF14	Atualizar usuário	Função que permite ao gestor atualizar dados do usuário
RF15	Cadastrar turma	Função que permite ao coordenador cadastrar turma
RF16	Pesquisar turma	Função que permite ao coordenador pesquisar turma
RF17	Eliminar turma	Função que permite ao coordenador eliminar turma
RF18	Atualizar turma	Função que permite ao coordenador atualizar turma
RF19	Consultar tarefa	Função que permite ao estudante visualizar uma tarefa
RF20	Cadastrar tarefa	Função que permite ao professor/estudante criar tarefa
RF21	Eliminar tarefa	Função que permite ao professor/estudante eliminar tarefa
RF22	Finalizar tarefa	Função que permite ao professor/estudante finalizar tarefa
RF23	Cadastrar reunião	Função que permite ao professor/estudante agendar reunião
RF24	Finalizar reunião	Função que permite ao professor/estudante finalizar uma reunião
RF25	Consultar reunião	Função que permite ao professor/estudante consultar uma reunião agendada ou que já ocorreu
RF26	Cadastrar regulamento	Função que permite ao coordenador cadastrar regulamento
RF27	Atualizar regulamento	Função que permite ao coordenador atualizar regulamento
RF28	Eliminar regulamento	Função que permite ao coordenador eliminar regulamento
RF29	Consultar regulamento	Função que permite ao docente/estudante consultar regulamento
RF30	Avaliar estudante	Função que permite ao tutor e membros da banca de jurados dar uma nota de avaliação do estudante
RF31	Gerar acta	Função que permite ao coordenador gerar a acta

4.2 REQUISITOS NÃO FUNCIONAIS

Em engenharia de software, um requisito não funcional é aquele que descreve não o que o sistema fará, mas como ele fará. Assim, por exemplo, têm-se requisitos de desempenho, requisitos da interface externa do sistema, restrições de projeto e atributos da qualidade.

A avaliação dos requisitos não funcionais é feita, em parte, por meio de testes, Filho, (2008).

Para que o sistema funcione os seguintes requisitos não funcionais foram identificados:

Usabilidade

RNF1. O sistema deve ser intuitivo e fácil de navegar, qualquer usuário pode usá-lo sem ter conhecimentos avançado de informática.

Confiabilidade

RNF2. O sistema valida a coleta de dados para evitar entradas inapropriadas.

Segurança

RNF3. A integridade e confidencialidade das informações são asseguradas através de mecanismos de controlo de acesso de usuários não autorizados, através de senha e definição de acesso para cada usuário, de modo que cada um pode ter disponível somente a atividade relacionada a ele.

RNF4. O sistema garante que a exclusão de informações emita uma opção de aviso antes de executar a ação.

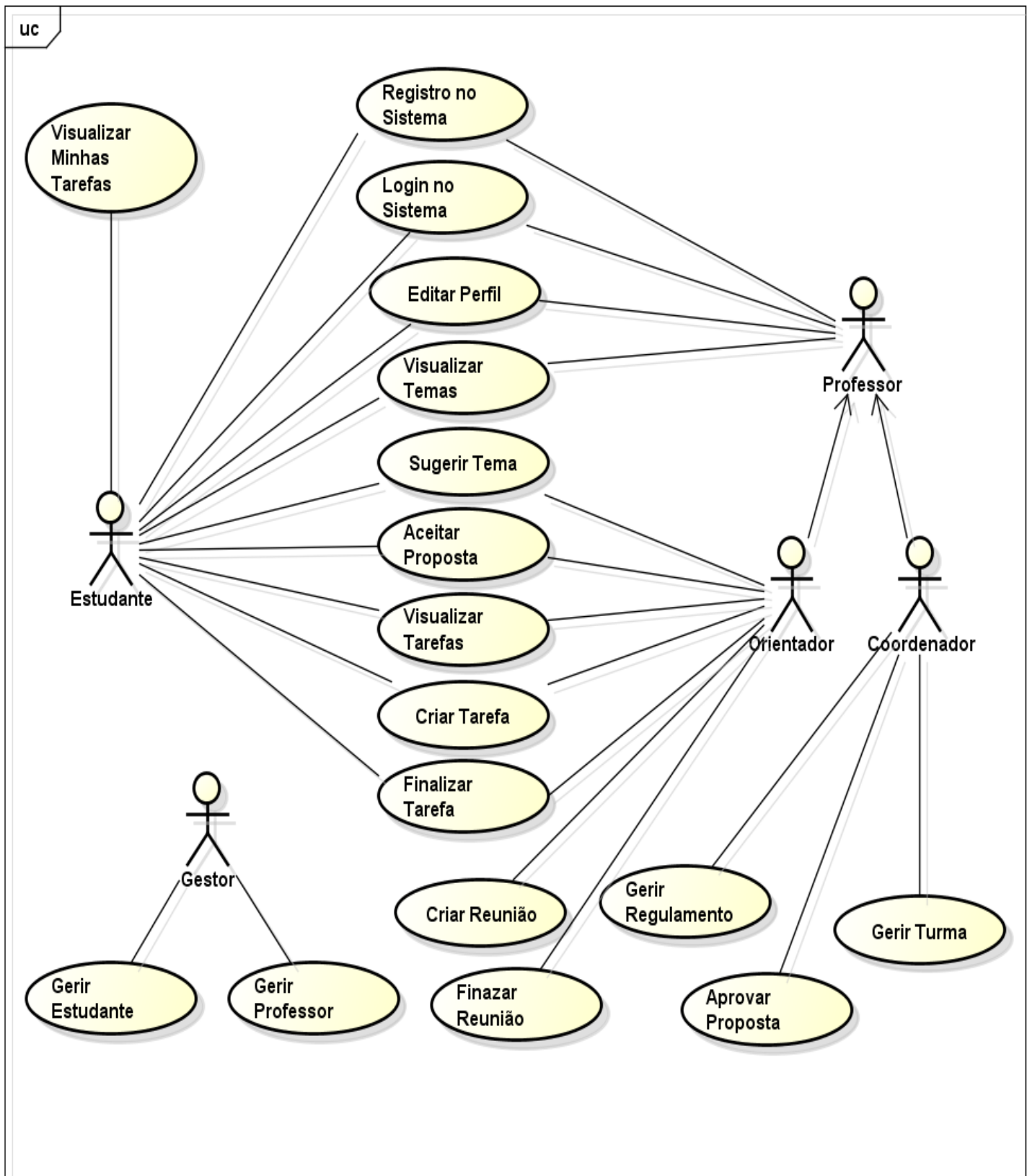
RNF5. A mensagem de erro exibida aos usuários é genérica, sem dar detalhes das informações, para não comprometer a segurança e a integridade dos dados.

RNF6. O canal de comunicação com o servidor de banco de dados deve ser seguro.

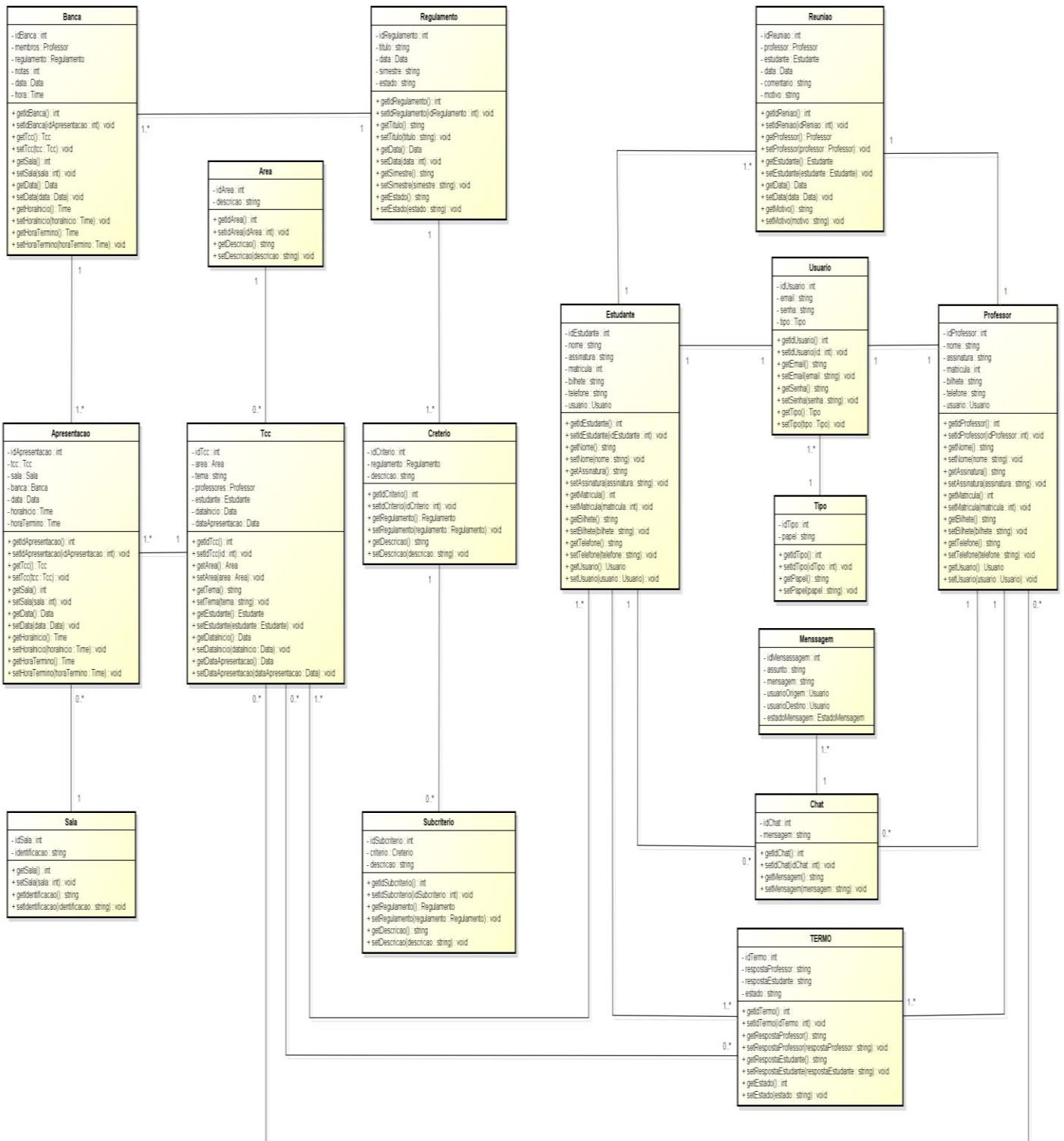
Aparência ou interface externa

RNF7. A boa organização das informações é garantida para permitir a interpretação correta.

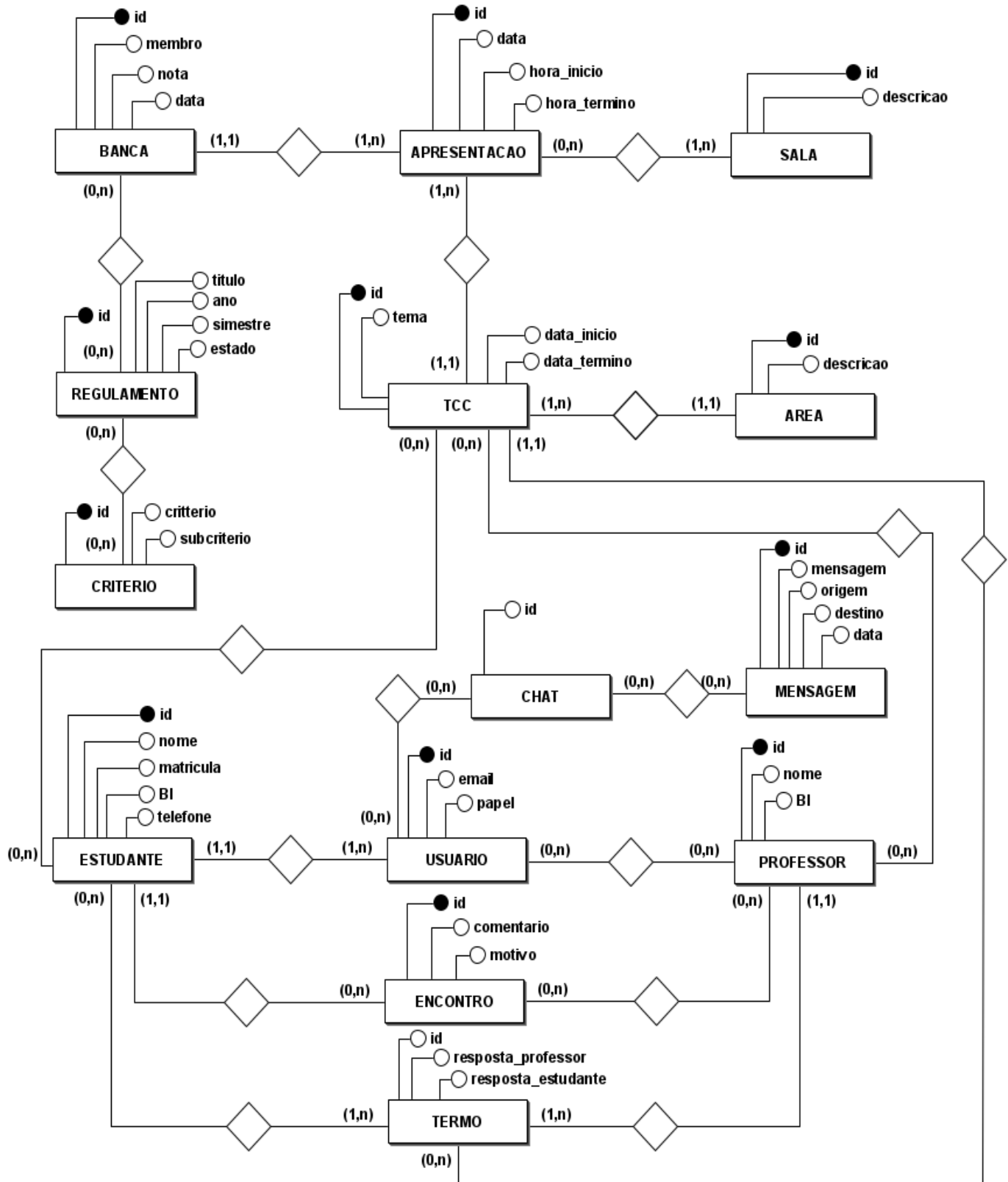
4.3 DIAGRAMA DE CASO DE USO



4.4 DIAGRAMA DE CLASSE



4.5 DIAGRAMA DE ENTIDADE E RELACIONAMENTO



5. CRONOGRAMA DE EXECUÇÃO

CRONOGRAMA						
N.º	ANO	2022/2023				
	MÊS	Out.	Nov.	Dez.	Jan.	Fev.
Planeamento	Preparação da pesquisa	X	X			
	Escolha do Tema		X			
	Formulação do problema e Objectivos		X			
	Determinação da metodologia		X			
Recolha de dados	Pesquisa bibliográfica		X	X	X	
	Ficheiros bibliográficos e de leitura		X	X	X	
Análise	Elaboração do plano definitivo			X	X	
	Revisão geral da documentação				X	
Redacção	Redacção provisória				X	
	Redacção definitiva. Digitação					X
Revisão	Revisão do manuscrito					X
	Correcções					X
	Digitação final					X

6. REFERÊNCIAS BIBLIOGRÁFICAS

Edmundo Fernandes. 2014. O impacto dos computadores na sociedade. Tecnologia em Geral. [Online] Tecnologia em Geral, 2014. [Citação: 17 de 05 de 2018.] <http://tecnologiaemgeral.blogspot.com/2014/10/o-impacto-dos-computadores-na-sociedade.html>.

BeCode. 2016. Manifesto Ágil e as TOP 3 Metodologias Ágeis de Desenvolvimento! BeCode. [Online] BeCode, 2016. [Citação: 28 de 05 de 2018.] <https://becode.com.br/manifesto-agil-e-top-metodologias-ageis/>.

Brizen, Marcos. 2011. Classificação dos Padrões de Projeto GoF. Brizen. [Online] 12 de 12 de 2011. [Citação: 13 de 03 de 2019.] <https://brizen.wordpress.com/2011/12/12/classificacao-dos-padroes-de-projeto-gof/>.

Burgués, Esteban Gracia. 2016. Aprende a Modelar Aplicaciones con UML: 2ª Edición. Espanha : ITCampos Academy, 2016. 978-1523498536.

Campos, Edmilson. 2010. Modelagem Conceitual. Rio Grande do Norte : Instituto federal de Educação, Ciência e Tecnologia, 2010.

DIEDRICH, LUCAS GUILHERME. 2011. INTEGRAÇÃO DA METODOLOGIA ÁGIL OPENUP NOS. MEDIANEIRA : UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR, 2011.

Fabri, José Augusto. 2013. Modelagem: do Problema ou do Negócio? Engenharia Software. [Online] 11 de 11 de 2013. [Citação: 18 de 02 de 2018.] <https://engenhariasoftware.wordpress.com/2013/11/11/modelagem-do-problema-ou-donegocio/>.

FILHO, ANTONIO MENDES DA SILVA. 2008. Artigo Engenharia de Software 3 - Requisitos Não Funcionais. Devmedia. [Online] 2008. [Citação: 12 de 03 de 2019.] <https://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-naofuncionais/9525>.

Fonseca, Daniel. 2016. Conceitos básicos sobre Metodologias Ágeis para Desenvolvimento de Software (Metodologias Clássicas x Extreme Programming). DevMedia. [Online] 2016. [Citação: 28 de 05 de 2018.] <https://www.devmedia.com.br/conceitos-basicos-sobre-metodologias-ageis-paradesenvolvimento-de-software-metodologias-classicas-x-extreme-programming/10596>.

Junior, Delmir Peixoto de Azevedo. 2008. Definição de requisitos de software baseada numa arquitetura de modelagem. São Paulo : editoracubo, 2008.

Koopmans, Regan. 2017. Entendendo os Padrões de Design GRASP. Medium. [Online] 24 de 09 de 2017. [Citação: 12 de 03 de 2018.] <https://medium.com/@ReganKoopmans/understanding-the-grasp-design-patterns-2cab23c7226e>.

Nakagawa, Elisa Yumi. 2013. Arquitetura de Software:. São Paulo : Universidade de São Paulo, 2013.

NPM. 2019. angular-ide. NPM. [Online] 12 de 02 de 2019. [Citação: 12 de 03 de 2019.] <https://www.npmjs.com/package/angular-ide>.

Paradigm, Visual. 2018. Ideal Modeling Diagramming Tool for Agile Team Collaboration. Visual Paradigm. [Online] Visual Paradigm, 2018. [Citação: 15 de 02 de 19.] <https://www.visual-paradigm.com/>.

Pearson Prentice Hall. 2011. Modelagem. Nova Jersey : Prentice Hall, 2011.

Silva, Flávio. 2006. Modelagem de Software. 2006.

Tigris.org. 2009. Open Source Software Engineering Tools. Tigris.org. [Online] Tigris.org, 2009. [Citação: 15 de 02 de 19.] <http://argouml.tigris.org/>.

Videira, Alberto Manuel Rodrigues da Silva e Carlos Alberto Escalera. 2001. UML, Metodologias e Ferramentas CASE. Porto - Lisboa : s.n., 2001. 972-8426-36-4.

Xexéo, Geraldo. 2007. Modelagem de sistemas de informação - Da análise de requisitos ao modelo de interface. Brazil : s.n., 2007.