



ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO:

Docência, pesquisa e inovação tecnológica



Lilian Coelho de Freitas
(Organizadora)



Atena
Editora
Ano 2023



ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO:

Docência, pesquisa e inovação tecnológica



Lilian Coelho de Freitas
(Organizadora)



Atena
Editora
Ano 2023

Editora chefe

Profª Drª Antonella Carvalho de Oliveira

Editora executiva

Natalia Oliveira

Assistente editorial

Flávia Roberta Barão

Bibliotecária

Janaina Ramos

Projeto gráfico

Bruno Oliveira

Camila Alves de Cremo

Luiza Alves Batista

Imagens da capa

iStock

Edição de arte

Luiza Alves Batista

2023 by Atena Editora

Copyright © Atena Editora

Copyright do texto © 2023 Os autores

Copyright da edição © 2023 Atena Editora

Direitos para esta edição cedidos à Atena Editora pelos autores.

Open access publication by Atena Editora



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição *Creative Commons*. Atribuição-Não-Comercial-Não-Derivativos 4.0 Internacional (CC BY-NC-ND 4.0).

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores, inclusive não representam necessariamente a posição oficial da Atena Editora. Permitido o *download* da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

Todos os manuscritos foram previamente submetidos à avaliação cega pelos pares, membros do Conselho Editorial desta Editora, tendo sido aprovados para a publicação com base em critérios de neutralidade e imparcialidade acadêmica.

A Atena Editora é comprometida em garantir a integridade editorial em todas as etapas do processo de publicação, evitando plágio, dados ou resultados fraudulentos e impedindo que interesses financeiros comprometam os padrões éticos da publicação. Situações suspeitas de má conduta científica serão investigadas sob o mais alto padrão de rigor acadêmico e ético.

Conselho Editorial**Ciências Exatas e da Terra e Engenharias**

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto

Profª Drª Alana Maria Cerqueira de Oliveira – Instituto Federal do Acre

Profª Drª Ana Grasielle Dionísio Corrêa – Universidade Presbiteriana Mackenzie

Profª Drª Ana Paula Florêncio Aires – Universidade de Trás-os-Montes e Alto Douro

Prof. Dr. Carlos Eduardo Sanches de Andrade – Universidade Federal de Goiás

Profª Drª Carmen Lúcia Voigt – Universidade Norte do Paraná

Prof. Dr. Cleiseano Emanuel da Silva Paniagua – Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Prof. Dr. Douglas Gonçalves da Silva – Universidade Estadual do Sudoeste da Bahia
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná
Prof^o Dr^a Érica de Melo Azevedo – Instituto Federal do Rio de Janeiro
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará
Prof^o Dr^a Glécilla Colombelli de Souza Nunes – Universidade Estadual de Maringá
Prof^o Dr^a Iara Margolis Ribeiro – Universidade Federal de Pernambuco
Prof^o Dra. Jéssica Verger Nardeli – Universidade Estadual Paulista Júlio de Mesquita Filho
Prof. Dr. Juliano Bitencourt Campos – Universidade do Extremo Sul Catarinense
Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande
Prof^o Dr^a Luciana do Nascimento Mendes – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Prof. Dr. Marcelo Marques – Universidade Estadual de Maringá
Prof. Dr. Marco Aurélio Kistemann Junior – Universidade Federal de Juiz de Fora
Prof^o Dr^a Maria José de Holanda Leite – Universidade Federal de Alagoas
Prof. Dr. Miguel Adriano Inácio – Instituto Nacional de Pesquisas Espaciais
Prof. Dr. Milson dos Santos Barbosa – Universidade Tiradentes
Prof^o Dr^a Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Prof^o Dr^a Neiva Maria de Almeida – Universidade Federal da Paraíba
Prof. Dr. Nilzo Ivo Ladwig – Universidade do Extremo Sul Catarinense
Prof^o Dr^a Priscila Tessmer Scaglioni – Universidade Federal de Pelotas
Prof^o Dr Ramiro Picoli Nippes – Universidade Estadual de Maringá
Prof^o Dr^a Regina Célia da Silva Barros Allil – Universidade Federal do Rio de Janeiro
Prof. Dr. Sidney Gonçalo de Lima – Universidade Federal do Piauí
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista

Engenharia elétrica e de computação: docência, pesquisa e inovação tecnológica

Diagramação: Camila Alves de Cremo
Correção: Flávia Roberta Barão
Indexação: Amanda Kelly da Costa Veiga
Revisão: Os autores
Organizadora: Lilian Coelho de Freitas

Dados Internacionais de Catalogação na Publicação (CIP)	
E57	<p>Engenharia elétrica e de computação: docência, pesquisa e inovação tecnológica / Organizadora Lilian Coelho de Freitas. – Ponta Grossa - PR: Atena, 2023.</p> <p>Formato: PDF Requisitos de sistema: Adobe Acrobat Reader Modo de acesso: World Wide Web Inclui bibliografia ISBN 978-65-258-0946-5 DOI: https://doi.org/10.22533/at.ed.465231601</p> <p>1. Energia elétrica. 2. Computação. I. Freitas, Lilian Coelho de (Organizadora). II. Título.</p> <p style="text-align: right;">CDD 623.3</p>
Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166	

Atena Editora
Ponta Grossa – Paraná – Brasil
Telefone: +55 (42) 3323-5493
www.atenaeditora.com.br
contato@atenaeditora.com.br

DECLARAÇÃO DOS AUTORES

Os autores desta obra: 1. Atestam não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao artigo científico publicado; 2. Declaram que participaram ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão.; 3. Certificam que os artigos científicos publicados estão completamente isentos de dados e/ou resultados fraudulentos; 4. Confirmam a citação e a referência correta de todos os dados e de interpretações de dados de outras pesquisas; 5. Reconhecem terem informado todas as fontes de financiamento recebidas para a consecução da pesquisa; 6. Autorizam a edição da obra, que incluem os registros de ficha catalográfica, ISBN, DOI e demais indexadores, projeto visual e criação de capa, diagramação de miolo, assim como lançamento e divulgação da mesma conforme critérios da Atena Editora.

DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código Penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access*, *desta forma* não os comercializa em seu site, sites parceiros, plataformas de *e-commerce*, ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.

O e-book intitulado “Engenharia elétrica e de computação: Docência, pesquisa e inovação tecnológica” está organizado em 12 capítulos e reúne importantes trabalhos científicos desenvolvido por pesquisadores de Norte a Sul do Brasil, que atuam em renomadas instituições de ensino e pesquisa.

Cada capítulo apresenta uma experiência única, com resultados práticos, consistentes e didáticos. Dessa forma, ao ler este livro, o leitor poderá aprofundar seus conhecimentos em desenvolvimento e teste de softwares, jogos digitais, aprendizagem de máquina, automação, geração de energia, entre outros assuntos relacionados à engenharia elétrica e de computação.

Além de uma base teórica aprofundada, nota-se que os autores de cada capítulo adotaram uma linguagem pedagógica e educativa. Assim, acredito que este livro é um excelente referencial teórico, especialmente para alunos de engenharia elétrica e de computação que estejam desenvolvendo trabalhos de conclusão de curso e que buscam exemplos de aplicações práticas para os conhecimentos teóricos estudados durante o curso. Através da reprodução dos resultados apresentados, é possível por exemplo propor melhorias, apresentar soluções alternativas para os problemas propostos ou desenvolver estudos comparativos. Assim o conhecimento científico avança.

Registro meus sinceros agradecimentos aos autores deste e-book, pelas significativas contribuições e pela parceria com a Atena Editora para tornar o conhecimento científico acessível de forma gratuita.

Aos nossos leitores, desejo um ótimo estudo, repleto de *insights* criativos e inovadores.

Lilian Coelho de Freitas

CAPÍTULO 1	1
ESTUDO E DESENVOLVIMENTO DE UMA METODOLOGIA PARA O PROCESSO DE REVISÃO EM HOMOLOGAÇÕES DE RELEASES ANDROID	
Pedro Ivo Pereira Lancellotta	
Heryck Michael dos Santos Barbosa	
João Gabriel C. Santos	
Klirssia M. Isaac Sahdo	
Janisley Oliveira De Sousa	
Abda Myrria De Albuquerque	
Roger Porty Pereira Vieira	
 https://doi.org/10.22533/at.ed.4652316011	
CAPÍTULO 2	11
ENGENHARIA DE REQUISITOS E SUA IMPORTÂNCIA NO DESENVOLVIMENTO DE <i>SOFTWARE</i>	
Henderson Matsuura Sanches	
 https://doi.org/10.22533/at.ed.4652316012	
CAPÍTULO 3	21
ALGORITMOS NÃO SUPERVISIONADOS E <i>WEB SCRAPING</i> PARA DESCOBERTA DE CONHECIMENTO DE CONHECIMENTO EM REDES SOCIAIS	
Carlos Daniel de Sousa Bezerra	
 https://doi.org/10.22533/at.ed.4652316013	
CAPÍTULO 4	38
MODELOS MENTAIS DIFUSOS PARA TOMADA DE DECISÃO SOBRE O CRESCIMENTO POPULACIONAL EM CIDADES INTELIGENTES USANDO TÉCNICAS COGNITIVAS	
Márcio Mendonça	
Caio Ferreira Nicolau	
Fabio Rodrigo Milanez	
Vicente de Lime Gonogora	
Luiz Henrique Geromel	
Marcio Aurélio Furtado Montezuma	
Rodrigo Henriques Lopes da Silva	
Marcos Antônio de Matos Laia	
Marco Antônio Ferreira Finocchio	
Renato Augusto Pereira Lima	
Edson Hideki Koroishi	
Gilberto Mitsuo Suzuki Trancolin	
André Luís Shiguemoto	
 https://doi.org/10.22533/at.ed.4652316014	
CAPÍTULO 5	57
CUSTOMIZED EXPERIENCE: DIGITAL GAMES POSSIBILITIES BEYOND	

THEIR MECHANICS

Paula Poiet Sampedro
 Nicholas Bruggner Grassi
 Isabela Zamboni Moschin
 Vânia Cristina Pires Nogueira Valente
 Emilene Zitkus

 <https://doi.org/10.22533/at.ed.4652316015>

CAPÍTULO 673**O USO DA AUTOMAÇÃO DIGITAL PARA AGILIZAR PROCESSOS E SUPRIMIR ERROS NA EXECUÇÃO DE ROTINAS**

Geovane Griesang
 Pedro Henrique Giehl
 Mateus Roberto Algayer

 <https://doi.org/10.22533/at.ed.4652316016>

CAPÍTULO 780**HOSPITAL INTELIGENTE: UMA SIMULAÇÃO DE MONITORAMENTO DE PACIENTES UTILIZANDO INTERNET DAS COISAS**

Júlia Borges Santos
 Vinicius da Rocha Motta
 Saymon Castro de Souza
 Ciro Xavier Maretto

 <https://doi.org/10.22533/at.ed.4652316017>

CAPÍTULO 887**DESENVOLVIMENTO DE UM APLICATIVO NO AMBIENTE *APP DESIGNER* DO *SOFTWARE* MATLAB® PARA PLANEJAMENTO DE TRAJETÓRIA DO ROBÔ PUMA 560**

Eber Delgado de Souza
 Flávio Luiz Rossini
 Luiz Fernando Pinto de Oliveira

 <https://doi.org/10.22533/at.ed.4652316018>

CAPÍTULO 9110**ANÁLISE DE MOTIVAÇÃO E SATISFAÇÃO NA INSTALAÇÃO DE PAINÉIS SOLARES FOTOVOLTAICOS POR MEIO DE MAPAS COGNITIVOS FUZZY**

Márcio Mendonça
 Angelo Feracin Neto
 Carlos Alberto Paschoalino
 Matheus Gil Bovolenta
 Emerson Ravazzi Pires da Silva
 Marcio Aurelio Furtado Montezuma
 Kazuyochi Ota Junior
 Marcos Antonio de Matos Laia
 Augusto Alberto Foggiato
 Vicente de Lima Gongora

Andre Luis Shiguemoto
Francisco de Assis Scannavino Junior
Nikolas Catib Boranelli

 <https://doi.org/10.22533/at.ed.4652316019>

CAPÍTULO 10..... 126

DESENVOLVIMENTO E AVALIAÇÃO DE UM CONTROLADOR PREDITIVO
NÃO-LINEAR BASEADO EM MODELO QUASILINEAR MODIFICADO

Manoel de Oliveira Santos Sobrinho
Adhemar de Barros Fontes

 <https://doi.org/10.22533/at.ed.46523160110>

CAPÍTULO 11 140

IMPLEMENTAÇÃO DE ATERRAMENTO EM UMA RESIDÊNCIA COM DR
PARA ELIMINAR O CHOQUE ELÉTRICO

Eliandro Marquetti
Elielton Christiano de Oliveira Metz
Luciana Paro Scarin Freitas

 <https://doi.org/10.22533/at.ed.46523160111>

CAPÍTULO 12..... 156

PANORAMA DAS FONTES TÉRMICAS PARA GERAÇÃO DE ENERGIA
ELÉTRICA NO BRASIL

Bruno Knevez Hammerschmitt
Felipe Cirolini Lucchese
Marcelo Bruno Capeletti
Renato Grethe Negri
Leonardo Nogueira Fontoura da Silva
André Ross Borniatti
Fernando Guilherme Kaehler Guarda
Alzenira da Rosa Abaide

 <https://doi.org/10.22533/at.ed.46523160112>

SOBRE A ORGANIZADORA 171

ÍNDICE REMISSIVO..... 172

ENGENHARIA DE REQUISITOS E SUA IMPORTÂNCIA NO DESENVOLVIMENTO DE SOFTWARE

Data de aceite: 02/01/2023

Henderson Matsuura Sanches

Centro Universitário UDF
ORCID: 0000-0003-2354-3393

RESUMO: Diante das constantes mudanças e dos intensos avanços tecnológicos que vem ocorrendo nas últimas décadas, a informação passou a ser um recurso estratégico das empresas, sendo o objetivo deste trabalho é abordar o papel e a importância da Engenharia de Requisitos (ER) no desenvolvimento de software e definir os processos e atividades da ER, como a definição da ER, apresentar normas, a Unified Modeling Language (UML), a linguagem de destinada à modelação e documentação das fases de desenvolvimento de softwares onde o Diagrama de Caso de Uso (DCU) sendo o diagrama fundamental para qualquer desenvolvimento de software permite dar uma visão global e de alto nível do sistema, descrevendo sua visão externa e as interações com o mundo exterior, tendo como metodologia bibliográfica com o intuito de demonstrar a importância da ER no desenvolvimento de software e a ER define, sem dúvida, um dos mais importantes

conjuntos de atividades a serem realizadas em projetos de desenvolvimento de software.

PALAVRAS-CHAVE: Engenharia de Requisitos, UML, Software, ISO/IEC, Desenvolvimento.

ABSTRACT: Faced with the constant changes and intense technological advances that have been taking place in recent decades, information has become a strategic resource for companies, where the objective of this work is to address the role and importance of Requirements Engineering (RE) in software development and defining the ER processes and activities, such as defining the ER, presenting standards, the Unified Modeling Language (UML), the language intended for modeling and documenting the software development phases where the Use Case Diagram (UCD) being the fundamental diagram for any software development, it allows giving a global and high-level view of the system, describing its external view and interactions with the outside world, having as a bibliographic methodology in order to demonstrate the importance of RE in software development and RE defines, without a doubt, one of the most important

sets of activities to be carried out in software development projects.

KEYWORDS: Requirements Engineering, UML, Software, ISO/IEC, Development.

1 | INTRODUÇÃO

Diante dos avanços tecnológicos que vem ocorrendo nas últimas décadas, a informação passou a ser um recurso estratégico das empresas. O *software* se tornou a força motora desta nova era. Sendo Ele capaz de manipular o bem mais importante para uma empresa – a informação. A fim de evitar que o desenvolvimento de *software* seja realizado com baixa qualidade e pouca previsibilidade de custos e recursos, surgiram técnicas de engenharia de *software* (MARTINS, 2013). O desenvolvimento e a implementação de um *software* visa a melhoria do desempenho organizacional diante dos mercados com a acirrada competição. A indústria de *software* vem demonstrando crescente interesse em Engenharia de Requisitos e percebendo que o tempo utilizado no entendimento do problema é um excelente investimento (COSTA, 2018).

Segundo Pádua (1998, apud Pressman, 2003) para projetar e construir um *software* que atenda às necessidades do cliente deve-se entender o que ele realmente deseja. E para tanto, o autor sugere o uso da Engenharia de Requisitos (ER), que inclui o conjunto de tarefas que levam ao entendimento de qual será o impacto do *software* sobre o negócio, do que o cliente quer e de como os usuários finais vão interagir com o *software*.

1.1 Objetivo Geral

O objetivo deste artigo é abordar o papel e a importância da ER no desenvolvimento de *software* e definir os processos e atividades da ER.

2 | DEFINIÇÕES DE ENGENHARIA DE REQUISITOS (ER)

Antes de definir o processo de ER, inicialmente, é fundamental definir o que são requisitos.

Conforme Chichinelli e Cazarini (2003) requisito é uma condição ou uma capacidade que deve ser alcançada ou possuída por um sistema ou componente do sistema, para satisfazer um contrato, um padrão, uma especificação ou outros documentos impostos formalmente.

Para Alencar (1999) requisito é um conjunto de descrições de como o sistema pretendido deve se comportar, ou um conjunto de propriedades, de atributos do sistema e ou limitações do próprio processo de desenvolvimento do *software*.

Requisito pode ser definido como “algo que um cliente necessita”. Entretanto do ponto de vista do Engenheiro de software, requisitos pode ser definido como “algo que necessita ser projetado”. (MACAULAY, 1996, p.72).

3 | A CLASSIFICAÇÃO DOS REQUISITO

Os requisitos podem ser classificados como funcionais e não funcionais. Os requisitos funcionais dizem respeito à definição das funções que um sistema ou um componente de sistema deverá fazer, ou seja, as entradas que deverão ser transformada, e as saídas que deverão ser produzidas. Enquanto os requisitos não funcionais, dizem respeito às restrições e comportamentos que o software deverá satisfazer (TURINE e MASIEIRO,1996).

De forma simples, a diferença entre requisitos funcionais e não funcionais está no fato dos requisitos funcionais descreverem “o que” o sistema deverá fazer, enquanto os requisitos não funcionais fixam restrições sobre “como” os requisitos funcionais serão implementados.

4 | ISO/IEC E NORMAS EM ER

Durante o processo de *software* os requisitos mudam constantemente, seja em virtude do melhor entendimento do problema, com o amadurecimento da compreensão dos envolvidos acerca do que desejam que o *software* faça, ou em alterações de *hardware*, *software* ou ambiente organizacional no qual o sistema está inserido. Novos requisitos surgem à medida que novas necessidades e prioridades são descobertas por parte dos *stakeholders* ao longo do processo. Pode-se dizer desta forma, que os requisitos devem evoluir visando refletir as novas visões do problema (SOMMERVILLE, 2007).

A preocupação com a qualidade na produção de *software* inicialmente centrou foco na qualidade do produto; esta visão evoluiu e atualmente a preocupação com qualidade envolve tanto o processo de produção (o ciclo de desenvolvimento do *software*) quanto o resultado final (o produto ou *software* gerado) (SAYÃO, 2003).

Na busca de aprimoramento nas atividades do processo de desenvolvimento de *software* originou-se as normas ISO/IEC 12207 e ISO/IEC 15504, e os modelos de processo *Capability Maturity Model Integration* (CMMI) e Modelo de Processo do *Software* Brasileiro (MPS.BR). Essas normas e modelos apresentam o processo de *software* sob a ótica de seu funcionamento, mas não apresentam como estes processos devem ser definidos, sendo esta uma responsabilidade do engenheiro de *software*. Nelas é apresentado um conjunto de resultados esperados em termos de qualidade para a engenharia de requisitos (LAHOZ, 2003).

O *Institute of Electrical and Electronics Engineers* (IEEE), apresenta um documento que orienta os desenvolvedores quanto às atividades de especificação de requisitos de *software*.

Na Tabela 1 é apresentado algumas ISO/IEC e suas orientações.

ISO/IEC	Orientações
ISO/IEC 9126	Define um modelo de qualidade com um conjunto de características e <i>subcaracterísticas</i> de que aferem as capacidades do <i>software</i> no atendimento aos atributos de qualidade e apresenta um conjunto de características que definem um modelo de qualidade e podem ser aplicadas a qualquer produto de <i>software</i> .
ISO/IEC 12207	É usada como referência por empresas produtoras e prestadoras de serviços de <i>software</i> em muitos países, com o objetivo de auxiliar os envolvidos na construção de <i>software</i> a definir seus papéis, e por meio de processos bem definidos, proporcionar a obtenção de um entendimento das atividades a serem executadas. Segundo esta norma os requisitos são divididos em requisitos de sistema e requisitos de <i>software</i> .
ISO/IEC 15504	É uma norma que representa um padrão internacional emergente que estabelece um <i>framework</i> para construção de processos de avaliação e melhoria do processo de <i>software</i> . Sendo uma evolução da norma ISO/IEC 12207.
ISO 14598	Guias para a avaliação de produtos de <i>software</i> , baseados na utilização prática da norma ISO 9126.
NBR ISO 9001	Sistemas de qualidade – Modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica (processo).
NBR ISO 9000-3	Gestão de qualidade e garantia de qualidade. Aplicação da norma ISO 9000 para o processo de desenvolvimento de <i>software</i> .
IEEE 830	É uma recomendação prática para escrever especificações de requisitos de <i>software</i> . Seu objetivo é descrever o conteúdo e as qualidades fundamentais para a elaboração de um bom documento de especificação de requisitos de <i>software</i> .

Tabela 1: ISO/IECs e suas Orientações.

Martins, 2013 apresenta o conjunto de requisitos a ser considerar os atributos de qualidade apresentados na IEEE 830 a seguir:

- **Correção:** um conjunto de requisitos é considerado correto se todos os requisitos representam algo que deve estar presente no sistema que está sendo desenvolvido, ou seja, os requisitos reais do usuário devem coincidir com os requisitos identificados.
- **Não-ambiguidade:** um conjunto de requisitos é considerado não ambíguo quando pode ser interpretado por todos os envolvidos em um projeto de uma única maneira.
- **Completo:** um conjunto de requisitos é dito completo quando descreve todas as demandas de interesse dos usuários. Estas demandas incluem requisitos funcionais, de desempenho, restrições, atributos e interfaces externas.
- **Consistência:** um conjunto de requisitos é dito consistente se nenhum subconjunto destes requisitos entra em conflito com os demais requisitos do sistema.
- **Classificabilidade:** um conjunto de requisitos é considerado classificável quando pode ser classificado por grau de importância e/ou estabilidade.
- **Verificabilidade:** um requisito é verificável se existe uma forma efetiva, em

termos de tempo e custo, para que pessoas ou ferramentas indiquem se um sistema cumpre o requisito.

- **Modificabilidade:** um conjunto de requisitos é modificável quando seu estilo e estrutura é tal que as alterações podem ser realizadas de forma simples e consistente com os demais requisitos.
- **Rastreabilidade:** um requisito é considerado rastreável quando se pode acompanhar a vida de um requisito em ambas as direções do processo de *software* e durante todo o seu ciclo de vida.

Martins, 2013 também nos apresenta as características de qualidade definidas pela norma ISO/IEC 9126 sendo:

- **Funcionalidade:** conjunto de atributos que evidenciam a existência de funções que satisfazem as necessidades explícitas e implícitas;
- **Usabilidade:** conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o *software*;
- **Confiabilidade:** conjunto de atributos que evidenciam a capacidade do *software* de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo estabelecido;
- **Eficiência:** conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do *software* e a quantidade de recursos usados, sob condições estabelecidas;
- **Manutenibilidade:** conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no *software*;
- **Portabilidade:** conjunto de atributos que evidenciam a capacidade do *software* de ser transferido de um ambiente para outro.

As características e subcaracterísticas definidas pela norma ISO 9126 adaptada de (ISO/IEC, 2003) são apresentadas na Tabela 2. Para cada subcaracterística é elencada uma pergunta chave que remete ao objetivo ao qual a mesma se propõe (MARTINS, 2013).

Característica	Subcaracterística	Subcaracterística
Funcionalidade (Satisfaz as necessidades?)	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Faz o que foi proposta de forma correta?
	Interoperabilidade	Interage com os sistemas especificados?
	Conformidade	Está de acordo com as normas, leis, etc?
	Segurança de acesso	Evita acesso não autorizado aos dados?
Confiabilidade (É imune a falhas?)	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas como ele reage?
	Recuperabilidade	É capaz de recuperar dados em caso de falhas?
Usabilidade (É fácil de usar?)	Inteligibilidade	É fácil de entender o conceito e a aplicação?
	Apreensibilidade	É fácil de aprender a usar?
	Operacionalidade	É fácil de operar e controlar?
Eficiência (É fácil, rápido e “enxuto”?)	Tempo	Qual é o tempo de resposta, a velocidade de execução?
	Recursos	Quanto recurso usa? Durante quanto tempo?
Manutenibilidade (É fácil de modificar?)	Analísabilidade	É fácil de encontrar uma falha quando ocorre?
	Modificabilidade	É fácil modificar e adaptar?
	Estabilidade	Há grande risco quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade (É fácil de usar em outro ambiente?)	Adaptabilidade	É fácil adaptar em outros ambientes?
	Instabilidade	É fácil instalar em outros ambientes?
	Conformidade	Está de acordo com padrões de portabilidade?
	Substituibilidade	É fácil usar para substituir outro?

Tabela 2: Características e Subcaracterísticas da ISO/IEC 9126 (MARTINS, 2013).

5 | ENGENHARIA DE REQUISITOS NO PROCESSO DE DESENVOLVIMENTO ÁGIL

O surgimento dos métodos ágeis no cenário do desenvolvimento de *software* na atual área da engenharia de requisitos repensa alguns de seus procedimentos. Isto se deve principalmente porque estes métodos abdicam, em parte, de controles e documentação tão presentes nesta área. Os métodos ágeis em geral não mencionam como realizam, por exemplo, a documentação da especificação de requisitos, ou como mantêm a

rastreabilidade dos requisitos. Os princípios apresentados pelo manifesto ágil e discutidos por Fowler (2003, p.1) mostram que certos valores relacionados com a área de engenharia de requisitos continuam sendo importantes, como o entendimento dos requisitos, porém preocupam-se em não gerar muita documentação que, justificam, provavelmente nunca será lida.

As discussões em torno da compatibilidade dos métodos ágeis com modelos de qualidade de *software*, que possuem documentação em todos os processos, principalmente na área de requisitos, têm aumentado significativamente. Turner e Jain (2002, p.161) comentam que, apesar da existência de características distintas entre os métodos ágeis e o modelo *Capability Maturity Model Integration* (CMMI), ambos possuem planos específicos para o desenvolvimento de *software* e buscam o melhor para que a organização produza *software* com qualidade. O CMMI é um modelo de processo, criado e mantido pelo *Software Engineering Institute* (SEI), que recomenda um conjunto de “melhores práticas” para os processos de desenvolvimento de *software*. Atualmente, o CMMI é o modelo mais recomendado para empresas que pretendem exportar *software* ou serviços de TI, devido a sua ampla aceitação no mercado comprador internacional (TAURION, 2005).

O Processo de *Software* Brasileiro (MPS-BR) é uma metodologia voltada à área de desenvolvimento de sistemas e que foi criada por um conjunto de organizações ligadas ao desenvolvimento de *software*. O MPS-BR é um modelo de referência brasileiro para o processo de *software*, adequado ao perfil das empresas nacionais e compatível com os padrões de qualidade aceitos internacionalmente pela comunidade de *software*.

No que tange o método ágil *Scrum* segundo Schwaber (2002, p.2) tem como objetivo definir um processo para projeto e desenvolvimento de *software* orientado a objeto, que seja focado nas pessoas e que seja indicado para ambientes em que os requisitos surgem e mudam rapidamente. As práticas gerenciais do Scrum são: *Product Backlog*, *Daily Scrum*, *Sprint*, *Sprint Planning Meeting*, *Sprint Backlog* e *Sprint Review Meeting*.

A atividade de gestão ou gerência de requisitos é utilizada para compreender e controlar as mudanças dos requisitos de *software* (SOMMERVILLE, 2007). É importante que os requisitos de um projeto sejam acompanhados individualmente, mantendo ligações entre os requisitos dependentes, de forma que seja possível avaliar o impacto de possíveis mudanças. Para isso, um processo de gerenciamento deve ser estabelecido e mantido (SOMMERVILLE, 2011).

O planejamento é um estágio essencial no processo de gestão dos requisitos. Nesse momento deve-se decidir sobre (SOMMERVILLE, 2011):

- **Identificação de requisitos:** forma como cada requisito será identificado unicamente de modo que possa ser feita referência cruzada entre este e outros requisitos e para que ele possa ser utilizado nas avaliações de rastreabilidade.
- **Processo de gerenciamento de mudanças:** conjunto de atividades que avaliam o impacto e o custo das mudanças.

- **Políticas de rastreabilidade:** regras que definem os relacionamentos entre os requisitos e entre os requisitos e o projeto do sistema, que devem ser registrados, e como esses registros serão mantidos.
- **Apoio de ferramentas CASE:** uma ferramenta automatizada adotada como forma de apoio para a atividade de gerenciamento. Essas ferramentas trazem benefícios quanto ao armazenamento dos requisitos, o gerenciamento de mudanças e o gerenciamento de rastreabilidade.

As informações de rastreabilidade são geralmente representadas através de matrizes de rastreabilidade, podendo relacionar os requisitos aos stakeholders que os propuseram, a outros requisitos numa relação de dependência, ou aos módulos de um projeto, nos quais esses requisitos são implementados (SOMMERVILLE, 2011).

6 | MODELAGEM DE REQUISITOS

Segundo Pádua (1998, apud Pressman, 2003) a ER, como todas as outras atividades da engenharia de *software*, precisa ser adaptada às necessidades do processo, projeto, produto e do pessoal que está desenvolvendo o trabalho. Na perspectiva do processo de *software*, a ER começa durante a atividade de comunicação e continua durante a atividade de modelagem, sem ela, o *software* tem uma alta probabilidade de não satisfazer as necessidades do cliente.

A *Unified Modeling Language* (UML), é uma linguagem de notação destinada à modelação e documentação das fases de desenvolvimento de softwares orientados a objetos sendo padrão para a elaboração da estrutura de projetos de *software* sendo empregada na visualização, na especificação, na construção e na documentação de artefatos que façam uso de sistemas complexos de *software*. Em outras palavras, na área de engenharia de *software*, a UML é uma linguagem de modelagem que permite representar um sistema de forma padronizada (com intuito de facilitar a compreensão pré-implementação).

A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida ou como projetar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre os objetos (e em certos casos a identificação dos processos).

Um diagrama que é condensa entre inúmeros autores é o Diagrama de Caso de Uso (DCU) sendo o principal diagrama para a criação dos demais assim para toda a documentação, que de acordo com (Silva, 2001), um diagrama de caso de uso descreve a relação entre atores e casos de uso de um sistema. Isso permite dar uma visão global e de alto nível do sistema, descrevendo sua visão externa e as interações com o mundo exterior. Estes diagramas são utilizados preferencialmente na fase de especificação de requisitos, podendo ser utilizados também durante a modelagem dos processos de negócio (SILVA, 2001). Na Figura 1 é apresentado um Diagrama de Caso de Uso simples onde é observado

a relação entre os atores e algumas funcionalidades de um *software* criado somente como exemplo no aplicativo *Umbrello UML Modeller*.

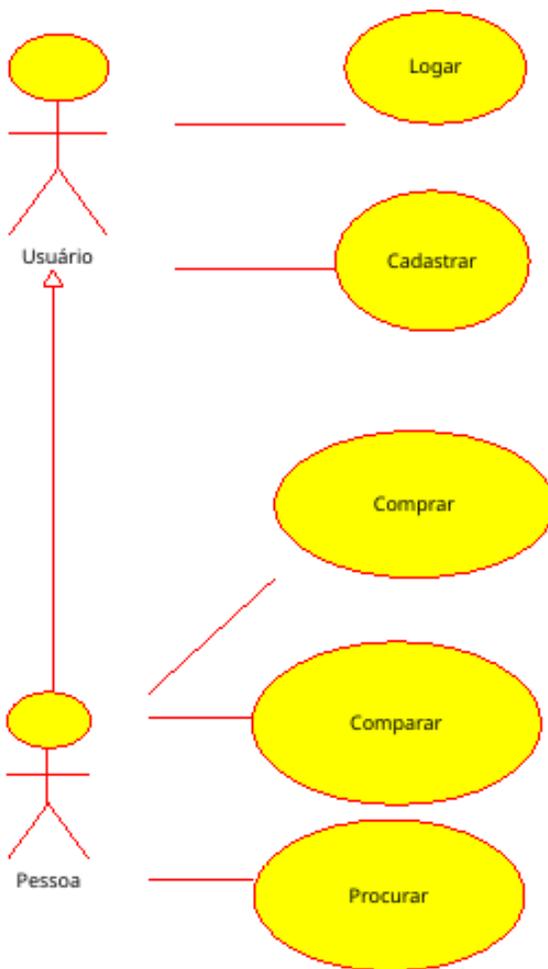


Figura 1: Diagrama de Caso de Uso Simples sendo Observado a Relação entre os Atores e Algumas Funcionalidades de um *Software*.

Fonte: Autor.

7 | METODOLOGIA

Foi realizada uma pesquisa bibliográfica, com o intuito de demonstrar a importância da ER no desenvolvimento de *software*, na busca de sobre o tema, bem como publicações e artigos científicos em torno do assunto.

8 | CONCLUSÃO

A ER define, sem dúvida, um dos mais importantes conjuntos de atividades a serem realizadas em projetos de desenvolvimento de *software*. Embora não garanta a qualidade dos produtos gerados, o DCU é fundamental permitindo dar uma visão global e de alto nível do sistema, sendo o principal diagrama para o desenvolvimento do *software*, descrevendo sua visão externa e as interações com o mundo exterior e um pré-requisito básico para que se obtenha sucesso no desenvolvimento do *software*, podendo ser utilizados também durante a modelagem dos processos de negócio. Por mais que não se deseje, os requisitos estarão sempre mudando durante o desenvolvimento do *software* e quanto melhor for o processo de ER desenvolvido por todos, menores serão os problemas encontrados.

REFERÊNCIAS

COSTA, E. C. DA. Importância Da Engenharia De Requisitos No Processo De Desenvolvimento De Sistemas De Informação. **Revista Interface Tecnológica**, v. 15, n. 1, p. 203–214, 2018.

LAHOZ, Carlos H. N.; SANTANNA, Nilson. Os Padrões ISO/IEC 12207 e 15504 e a Modelagem de Processos da Qualidade de Software. In: III Workshop dos Cursos de Computação Aplicada do INPE, 2003, São José dos Campos. Anais do III WORKCAP, 2003. p. 43-48

MACAULAY, L. Requirements Engineering. Berlin: Springer Verlag, 1996.

MARTINS, R. D. PROPOSTA DE UM PROCESSO DE ENGENHARIA DE REQUISITOS PARA O NUSIS. 2013.

PÁDUA, Fabiana Serralha Miranda de. A importância da técnica de modelagem organizacional EKD no desenvolvimento de diagramas Use Case. 2003. 130f. Dissertação (Mestrado em Engenharia de Produção) – Universidade de São Paulo, São Carlos, 2003.

SAYÃO, Miriam; STAA Arnd V.; LEITE, Julio C. S. de P. Qualidade em Requisitos. Rio de Janeiro: Departamento de Informática, PUC-Rio, 2003 (Monografia).

SILVA, Alberto M. R.; VIDEIRA, Carlos A. E. UML - Metodologias e Ferramentas CASE. Lisboa: Centro Atlântico, 2001.

SOMMERVILLE, Ian. Engenharia de Software, 8a edição. São Paulo: Pearson Addison-Wesley, 2007.

SOMMERVILLE, Ian. Engenharia de Software, 9a edição. São Paulo: Pearson Prentice Hall, 2011.

TAURION, Cezar. Software embarcado: oportunidades e potencial de mercado. Rio de Janeiro: Brasport, 2005.

A

Algoritmo doc2vec 30, 34, 35

Aterramento 140, 141, 142, 145, 146, 149, 150, 151, 152, 153, 154

C

Choque elétrico 140, 141, 142, 143, 144, 149, 152, 155

Cidades inteligentes 38, 39, 40, 41, 42, 43, 45, 48, 53

Clusterização 37

Controle preditivo não-Linear 127

D

Design 40, 56, 57, 58, 59, 61, 64, 65, 71, 72, 138

Digital games 57, 58, 68, 69, 70

Dispositivo residual 140

Dispositivos móveis 1

E

Energia eólica 111, 114, 117, 157

Energias não renováveis 157

Energia solar 111, 112, 113, 114, 116, 117, 118, 123, 124, 125

Energias renováveis 114, 157, 158, 167

Energia térmica 157, 158, 159

Engenharia de requisitos 11, 12, 13, 16, 17, 20

F

Fontes térmicas 156, 157, 158, 159, 160, 163, 167, 168

Fuzzy cognitive maps 39, 40, 49, 53, 54, 55, 56, 112, 125

G

Game customization 58

Garantia de qualidade 1, 8, 14

H

Homologação de releases Android 1

Hospital inteligente 80, 82, 85

I

Inserção automática 73

Interligação de programas 73

ISO/IEC 11, 12, 13, 14, 15, 16, 20

L

Layout de inclusão facilitada 73

M

Mapas cognitivos fuzzy 39, 110, 111, 118

Matlab 87, 88, 95, 108, 109

Modelos bilineares 126, 127, 128

P

Painéis fotovoltaicos 111, 113, 117, 122

Processos 2, 8, 11, 12, 13, 14, 17, 18, 20, 47, 73, 75, 88, 89, 90, 127, 137, 159, 161, 162, 164, 166

Puma 560 87, 88, 89, 90, 91, 92, 94, 96, 97, 99, 100, 107, 108

Q

Qualidade de software 1, 4, 17, 20

R

Robô 49, 87, 88, 89, 90, 91, 92, 93, 97, 99, 101, 102, 104, 107, 108, 109

Robótica 87, 88, 89, 90, 93, 107, 108, 109, 120

S

Satisfação do cliente 111, 122

Sistemas inteligentes de computação 39

Software 1, 2, 4, 5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 41, 54, 61, 73, 74, 76, 77, 78, 83, 87, 89, 90, 95, 96, 99, 107, 118, 122

T

Teste de software 1

U

UML 11, 12, 18, 19, 20

User experience (UE) 58, 59, 61, 62, 64, 70, 72

V

Virtual things 80

W

Web of things 80, 81, 83, 86

Web scraping 21, 22, 37

ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO:

Docência, pesquisa e inovação tecnológica

-  www.atenaeditora.com.br
-  contato@atenaeditora.com.br
-  [@atenaeditora](https://www.instagram.com/atenaeditora)
-  www.facebook.com/atenaeditora.com.br

ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO:

Docência, pesquisa e inovação tecnológica

- 🌐 www.atenaeditora.com.br
- ✉ contato@atenaeditora.com.br
- 📷 @atenaeditora
- 📘 www.facebook.com/atenaeditora.com.br