

Curso de Análise e Desenvolvimento de Sistemas

Allan Flores de Jesus

Documentação de Desenvolvimento de Software

2Care

Sorocaba

Junho - 2018

**Allan Flores de jesus**

Documentação de Desenvolvimento de Software

2Care

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Sorocaba –FATEC, como parte dos pré-requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas

Orientador: Cristiane Palomar Mercado

Sorocaba

Junho - 2018

**Dedicatória**

Dedico este trabalho aos meus pais que com muita perseverança e paciência sempre incentivaram meus estudos e proporcionaram a educação que hoje tenho.

**Agradecimento**

Gostaria de agradecer a todos que colaboraram para com este trabalho, em especial ao Paulo que me mostrou ferramentas para ajudar no processo criativa e a Pamella que desde que falei em Trabalho de Graduação buscou me incentivar a realiza-lo.

Resumo

É fato que instituições do terceiro setor, ou seja, aquelas que atuam com fins sociais e interesses públicos possuem necessidades que não conseguem ser supridas devido a limitantes financeiros. Com recursos muitas vezes escassos e limitados, recorrem a população local em busca de recursos, sejam eles financeiros, físicos ou humanos.

Diante desta problemática, propõe-se um aplicativo para aproximar a população local das instituições do terceiro setor que ali se encontram, para isso, será desenvolvido um WebApp utilizando-se do framework Angular para a interface que será mostrada ao usuário e NodeJS com o banco de dados MongoDB para o servidor da aplicação. Assim, espera-se uma maior visibilidade da população em relação as instituições beneficentes próximas e com isso, um maior engajamento da população em ações sociais.

Palavras-chave: Instituição beneficente; Aproximar; Necessidades; Engenharia de software; Desenvolvimento;

# Lista de Figuras

Figura 1 – Visão geral da gestão de pessoas 10

Figura 2 – Enfoque sistêmico nas organizações 12

Figura 3 – Esquema da estratégia empresarial e gestão de pessoas 13

Figura 4 – Fronteiras virtuais nas organizações 24

Figura 5 – Gestão das tecnologias da informação e gestão de pessoas 25

# Lista de Tabelas

Tabela 1 – Principais mudanças na Área de Recursos Humanos (PricewaterhouseCoopers) 17

Tabela 2 – Principais mudanças na Área de Recursos Humanos (Chiavenato) 22

Tabela 3 – Resumo do Estudo de Caso 1 29

Tabela 4 – Ferramentas de T.I. e seu relacionamento com os processos da ARH (estudo de caso 1) 30

Tabela 5 – Ferramentas de T.I. e seu relacionamento com os processos da ARH (estudo de caso 2) 36

# Índice: Inserir o índice dos títulos e subtítulos.

# Introdução(nos títulos use letra arial, 16, negrito)

Entidades beneficentes são aquelas que possuem a qualificação como organização da sociedade civil com interesse público (Lei Nº 9.790, Art. 3º). No processo de obtenção desta qualificação e no processo de manutenção desta, é necessária a assinatura de um termo de parceria entre o poder público e as entidades, neste termo deve-se conter os valores esperados de receitas e despesas detalhados item por item e as remunerações e benefícios a serem pagos com os recursos vindos desta parceria (Lei Nº 9.790, Art. 10º, §2).

A estas entidades, é permitido o recebimento de doação para manutenção das atividades, por meio de recursos físicos, humanos e financeiros, ou até mesmo, ainda pela prestação de serviços intermediários de apoio a outras organizações sem fins lucrativos e a órgãos do setor público (Lei Nº 9.790, Art. 3º, Parágrafo único).

Mesmo com o auxílio financeiro do poder público e de outras organizações sem fins lucrativos como a UNESCO, muitas das entidades não conseguem cobrir todas as despesas envolvidas em suas atividades, requerendo assim doações realizadas pela comunidade local, o que muitas vezes implica em excesso de alguns itens e escassez de outros. Este fato ocorre devido ao desconhecimento da população em relação as necessidades das entidades.

Outro problema a ser considerado é a falta de engajamento e acesso na busca por doações. Entidades com pouco investimento público possuem infraestrutura de telecomunicações mais limitadas, levando muitas a não possuir sites, caixas de e-mails, páginas no Facebook e afins. Isto causa a falta de visibilidade na comunidade e sociedade no geral. Além disso, alguns meios adotados como ligações telefônicas e correios podem ser custosos e demorados, assim como o pessoal para a manutenção dessas tarefas muitas vezes manuais.

Neste projeto será utilizado MEAN, nomenclatura utilizada para determinar aplicações desenvolvidas com um conjunto específico de ferramentas, são eles, MongoDB, Express, Angular e NodeJs, além disso, todo o código e documentação estarão disponíveis no Github e caracterizado como open-source, ou seja, estará disponível para todos os interessados em conhecer a fundo o projeto.

MongoDB é um banco de dados NoSQL que é alimentado por documentos no formato JSON, além disso é altamente escalável e flexível quanto as demandas de acessos.

Express é um framework para NodeJs que fornece uma camada de recursos para aplicativos web com métodos utilitários HTTP e middleware.

Angular é um framework para desenvolvimento web cross-platform, ou seja, permite o desenvolvimento para todas as plataformas, sejam elas, web, mobile web, nativas ou desktop, permitindo seu desenvolvimento utilizando apenas javascript, css e html.

NodeJs é uma plataforma de desenvolvimento para aplicações *server-side* utilizando javascript, permitindo assincronismo e escalabilidade as aplicações.

A introdução deve começar com uma apresentação geral do assunto do trabalho. Descreva o caso a ser estudado de maneira sucinta, descreva a organização onde será aplicado o estudo. Justifique a escolha do tema. Comente quais as ferramentas, tecnologias, metodologias, técnicas, modelos, etc. que serão utilizados para solucionar o problema apresentado.

Descreva também o objetivo do trabalho de forma clara. Evite colocar objetivos que não poderão ser atingidos ou mensurados, deixe claras as limitações (fronteiras) do seu trabalho, porém não justifique possíveis problemas encontrados. Os problemas deverão ser relatados nas seções seguintes, mas não devem aparecer no objetivo. Use letra Arial, tamanho 12, parágrafo com espaçamento 1,5.

# Embasamento teórico.

Esta seção deve ser utilizada quando o software desenvolvido tenha como escopo um tema que necessite uma explicação mais abrangente para o leitor. Por exemplo, no caso do software desenvolvido utilizar IoT (Internet das Coisas) para a solução do problema, apresentar os conceitos de IoT, indicando as referências utilizadas.

O nível de detalhamento desta seção deve ser discutido com o orientador. O título da seção pode ser o assunto a ser apresentado por exemplo: “Conceitos de IoT”.

Esta seção não é obrigatória para os Projetos Integrados de Engenharia de Software.

# Planejamento Inicial do Software

## Situação Atual

## (Para os Projetos Integrados usar Documento1- Concepção Inicial ES2)

### **Descrição da Situação Atual**

Descrever aqui o perfil do cliente e o contexto atual.

### **Problemas Encontrados**

Descrever os problemas que a situação atual apresenta e que devem ser solucionados. Se o trabalho envolver uma empresa real, anexar uma declaração da empresa permitindo a divulgação das informações.

### **Aplicativos Disponíveis no mercado (estado da arte)**

Caso o software a ser desenvolvido não seja voltado à um cliente específico, mas ao mercado geral (por exemplo para dispositivos móveis) descrever o que já existe no mercado e o diferencial entre estes e o software proposto neste trabalho.

## 3.2 Objetivos de Negócio do Projeto

Descrever os objetivos a serem alcançados pela implantação ou disponibilização do software. Escreva de acordo com o cenário da aplicação em questão.

Os objetivos poderão definir:

- Quais serão os beneficiados pela implantação/disponibilização do SW.

Por exemplo: setores da empresa, setor específico do mercado,

população específica, etc.

- Quais os benefícios, por exemplo: mais agilidade e segurança no

processo automatizado, melhor controle dos processos, redução de custos,

etc.

## 3.3 Restrições e Riscos do Projeto Atual

Os itens abaixo podem ser baseados no (Documento2 de ES2)

3.3.1 Limitações Operacionais

Definir as limitações operacionais do software. A tabela abaixo mostra um exemplo de apresentação. Não é obrigatório utilizar o formato de tabelas, as limitações podem ser descritas como texto.

Tabela : Limitações operacionais do sistema.

|  |  |  |
| --- | --- | --- |
| LIMITAÇÃO | DESCRIÇÃO | VERSÃO |
| Sistema Operacional | Android | >= 4.0.3 (API 15) |
| Banco de Dados | SQLite (incluso com o sistema operacional Android) | >= 3.7 (API 11) |
| Métodos de Entrada | Toque em tela capacitiva. Métodos de entrada via hardware que sejam suportados pelo sistema operacional. | - |
| Armazenamento | Armazenamento local de banco de dados e arquivos de preferência compartilhada. Possibilidade de se manter um backup desse armazenamento online, via conta Google vinculada ao Google Drive. | - |

Fonte: Autoria própria.

3.3.2 Considerações Legais

Exemplo: “O sistema é um software com todos os direitos reservados. Assim sendo, é proibida a cópia e/ou reprodução deste sem a prévia autorização de seus desenvolvedores, por quaisquer meios ou processos existentes. A violação dos direitos é punível como crime, com pena de prisão e multa.”

3.3.3 Considerações de Hardware / Software / Rede

Exemplo: “O sistema será executado em um ambiente que tenha de espaço livro em disco rígido >= 2MB; precisa de, no mínimo, 4 Gb de memória RAM para um melhor desempenho. Para o backup online deve ser reservado uma conta vinculada ao Google Drive com capacidade >= 3MB”.

3.3.4 Políticas Organizacionais

Exemplo: “Para assegurar o sigilo das informações presentes no banco de dados, o acesso ao sistema só é permitido através de um ID (identificador, login) e uma senha. Outras medidas de segurança, devem ser tomadas, como a utilização de computadores com antivírus e firewall sempre atualizados, a realização de backups semanalmente.”

3.3.6 Riscos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PLANO DE RISCOS** | | | | |
|
| **Risco** | **Probab (P)** | **Impacto (I)** | **RiscoTotal (P x I)** | **Tratamento** |
|
| Cronograma sem folgas | 5 | 4 | 20 | Extensão do prazo. |
| Falta de mão de obra | 4 | 4 | 16 | Contratação de mais funcionários. |
| Funcionalidades Genéricas | 5 | 3 | 15 | Definição elaborada das funcionalidades. |
| Problemas de Integração | 4 | 3 | 12 | Fazer uma apresentação de como funcionará o projeto. |
| Problemas com sistemas já desenvolvidos | 2 | 5 | 10 | Contratação de mais funcionários. |
| Problema de saúde com os funcionários | 2 | 5 | 10 | Incentivar a boa alimentação, exercícios físicos e vacinação. |
| Tecnologia utilizada | 3 | 3 | 9 | Realizar um estudo das tecnologias já usadas pelos sistemas. |
| Falta de energia | 2 | 4 | 8 | Contratação/Compra de geradores. |
| Pagamento pelo cliente | 2 | 3 | 6 | Reservar dinheiro para pagamento dos funcionários. |
| Catástrofe Naturais | 1 | 5 | 5 | Servidor ter uma sala própria e backups guardados em lugares seguros e distintos. |
| Perda de dados | 1 | 5 | 5 | Fazer backup diariamente. |

Indicar os riscos envolvidos no projeto. Utilizar uma Matriz de Risco conforme definido no PMBoK. Veja exemplo a seguir.

|  |  |
| --- | --- |
| **Probabilidade e Impacto** | |
| **Valor** | **Descrição** |
| 1 | Muito baixa |
| 2 | Baixa |
| 3 | Média |
| 4 | Alta |
| 5 | Muito alta |

4. Análise de Requisitos

4.1 Descrição da técnica utilizada para levantamento dos requisitos

Descrever a técnica utilizada para levantamento dos requisitos. Por exemplo, se foram utilizados questionários, análise do sistema atual (antigo) da empresa, pesquisa de mercado, etc..

Não é necessário incluir telas ou relatórios do sistema antigo. Se necessário para compreensão incluir os documentos como Apêndice.

4.2 Situação Proposta

Descrever o que será feito sem descrever como.

Descrever a proposta do que será desenvolvido em alto nível.

4.3 Requisitos Funcionais (Utilizar documento de Elaboração ES2)

São declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer (Sommerville,2000).

Descrever e numerar os requisitos funcionais. Exemplo:

RF1 - Cadastrar usuário

Este requisito refere-se ao cadastro de novos usuários no aplicativo.

O usuário só poderá entrar no aplicativo depois de se cadastrar informando

nome, e-mail e senha.

RF2 – Login no aplicativo

Este requisito refere-se ao login do usuário no aplicativo.

O usuário deverá inserir seu e-mail e senha para ter acesso ao aplicativo.

**4.3.1 Diagrama de Casos de Uso e Descrição dos Casos de Uso**

Apresentar o Diagrama de Casos de Uso e também a descrição dos mesmos usando o padrão a seguir:

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | RF4: CONSULTAR EXAMES AGENDADOS | |
| **Ator Principal** | USUÁRIO | |
| **Ator Secundário** |  | |
| **Pré-Condição** | O agendamento deve ter sido previamente cadastrado pelo usuário. | |
| **Pós-Condição** |  | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1 – O usuário acessa a agenda de exames. | |  |
|  | | 2 – Carrega todos os agendamentos registrados em ordem de data registrada (da mais recente para a mais antiga). |
| 3 – O usuário seleciona um dos agendamentos na listagem carregada para ver seus detalhes. | |  |
|  | | 4 – Exibe os detalhes do agendamento selecionado em uma janela pop-up acima da lista. |

4.4 Requisitos Não Funcionais

São aqueles que não dizem respeito, diretamente às funções específicas fornecidas pelo sistema. Eles estão relacionados a propriedades como confiabilidade, tempo de resposta e espaço em disco.

Requisitos não funcionais de processo podem solicitar o uso de uma determinada ferramenta CASE, linguagem de programação ou método de desenvolvimento.

Os requisitos não funcionais podem ser mais importantes que requisitos funcionais individuais, pois a falha em não cumprir um requisito não funcional pode tornar o sistema inútil (Sommerville,2000).

Alguns outros tipos de Requisitos Não Funcionais podem ser:

- Requisitos de Desempenho

- Requisitos de Armazenamento

- Requisitos de HW, SW e Redes

- Outros (ver anexo1 deste documento conforme Sommerville,2000)

5. Projeto Detalhado do Software

Este item poderá ter suas seções alteradas com a autorização do orientador. As modificações podem ser decorrentes do emprego de um Método de Processo de Software específico. Por exemplo, se o desenvolvimento for na área de jogos/jogos educativos o aluno poderá seguir outras metodologias por ex. Extreme Game Develoment (XGD) ou alguma sistemática indicada por algum especialista no assunto. O mesmo pode ocorrer com desenvolvimento àgil para aplicações móveis ou web.

5.1 Arquitetura da aplicação Atual (Utilizar documento elaborado em ES3)

Apresentar de maneira sucinta, qual foi o modelo arquitetural escolhido para o projeto. Por exemplo, o MVC (model, view, controller). É interessante incluir figuras facilitando o entendimento dos componentes.

5.2 Tecnologias utilizadas e APIs

**5.2.1 Tecnologias**

Descrever a tecnologia que será utilizada para desenvolvimento da aplicação, principalmente se for uma tecnologia nova. Exemplo: nova linguagem, framework, banco de dados ou hardware. Se necessário podem ser incluídas subseções. Indicar referências.

Exemplo: “As tecnologias utilizadas para o desenvolvimento do aplicativo serão relacionadas a seguir.

### **NativeScript (front-end)**

NativeScript (NativeScript, 2017) é um framework gratuito e open source para a construção de aplicações nativas para Android e iOS usando JavaScript e CSS. Segundo o fabricante, o NativeScript processa UIs (interfaces do usuário) com o mecanismo de renderização da plataforma nativa em WebViews, resultando em UI e desempenho nativos........etc, etc.”

### **Node.js (back-end)**

O Node.js (Node.js, 2017) é uma plataforma para desenvolvimento de aplicações server-side baseadas em rede utilizando JavaScript e o V8 JavaScript Engine, ou seja, com Node.js podemos criar uma variedade de aplicações Web utilizando apenas código em JavaScript.......”

**5.2.2 APIs Utilizadas**

Muitos aplicativos utilizam APIs (Application Programming Interface) já existentes. Neste caso relatar cada uma delas. Exemplo:

“Para oferecer a funcionalidade de vídeos será utilizada a YouTube API.

A YouTube API permite adicionar funcionalidades do YouTube em sites e aplicativos através de um serviço REST. A figura 5 mostra a Try this API, um console que se comunica com a YouTube API. Inserindo-se a url e os parâmetros a API apresenta a resposta......”.

5.3 Componentes do SW

Descrever os módulos da aplicação.

5.4 Diagrama de Classes

Apresentar Diagrama de Classes conforme padrão UML.

5.5 Banco de Dados

Informar o banco de dados utilizado, fabricante e versão.

5.5.1 Modelo Conceitual

Diagrama Entidade-relacionamento

5.5.2 Modelo Lógico

Definir as entidades, atributos, relacionamentos domínios e validações. Se necessário incluir um dicionário de dados com detalhamento dos atributos. O Script das tabelas pode ser colocado no Apêndice.

Se o modelo de banco de dados não for o relacional (NoSQL) apresentar a estrutura do documento.

5.6 Diagrama de Sequência

É um diagrama de comportamento dinâmico que procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem.

5.7 Diagrama Pacotes (Somente para Projeto Integrado – ES3)

O diagrama de pacotes descreve os pacotes ou pedaços do sistema divididos em agrupamentos lógicos, mostrando as dependências entre estes, ou seja, pacotes podem depender de outros pacotes. Utilizar o já definido na disciplina de Engenharia de Software 3, atualizando-o se necessário.

5.8 Diagrama Estado (Somente Projeto Integrado – ES3)

O diagrama de estado é tipicamente um complemento para a descrição das classes. Este diagrama mostra todos os estados possíveis que objetos de uma certa classe podem se encontrar, e mostra também quais são os eventos do sistemas que provocam tais mudanças.

Ele captura todo o ciclo de vida dos objetos, subsistemas e sistemas. Ele mostra os estados que um objeto pode possuir e como os eventos afetam estes estados ao passar do tempo.

Apresentar os diagramas apenas do módulo principal conforme definido na disciplinas de Engenharia de SW 3 . Operações de Inclusão, alteração e exclusão não devem ser incluídas.

5.7 Interfaces com o usuário

Apresentar aqui as Interfaces com o usuário. Pode ser *printscreen* das telas ou layout.

5.8 Relatórios

Descrever os relatórios disponibilizados.

6. Implementação

Indicar o repositório onde o código fonte pode ser acessado.

7. Projeto de Teste

Descrever o plano de testes.

8. Instalação do software

Fornecer informações sobre a instalação do software desenvolvido, assim como dos softwares complementares a serem instalados para o funcionamento do sistema.

Aqui também podem ser especificadas informações adicionais sobre o software, informações sobre sua utilização, backups, monitoramento, etc

9. Análise dos Resultados (se houver teste real com o usuário)

Caso o software já esteja em uso poderá ser incluída uma análise dos resultados .

# 10. Conclusão

# Este item é muito importante. Faz o fechamento, concluindo as ideias. Esta etapa sintetiza todo o trabalho realizado e fornece uma resposta para a questão apresentada. Pode também levantar hipóteses e refletir sobre cada objetivo proposto.

A conclusão deverá apresentar um resumo de tudo o que foi feito. Poderão ser inseridos argumentos que mostrem quais objetivos foram atingidos e os resultados obtidos.

Referências

< Item obrigatório. Lista numerada em ordem alfabética **>**

Como apresentar a bibliografia: exemplos

**IMPORTANTE**: UTILIZAR A FERRAMENTA MORE (Mecanismo Online para Referências) da UFSC – [www.more.ufsc.br](http://www.more.ufsc.br)

Baseada nas normas da ABNT

**Inclua o MORE em suas referências**

MORE: Mecanismo online para referências, versão 2.0. Florianópolis: UFSC Rexlab, 2013. Disponível em: ‹ http://www.more.ufsc.br/ › . Acesso em: XX XXX XXXX.

**de AQUINO, Cleber Pinheiro** .- *Administração de Recursos Humanos . São Paulo : Atlas, 1992.*

**BERNARDO, André. A História do Gerenciamento de Projetos.** Responsabilidade do autor do vídeo. YouTube, 2013. Duração: 5min52seg. Disponível em:<<https://www.youtube.com/watch?v=le0GTYjlvl4>>. Acesso em: abril de 2017.

**CASTRO, Alfredo Pires de.; dos REIS, Almiro (neto) ; et alli** - *Manual de Gestão de Pessoas e Equipes. São Paulo : Editora Gente, 2003.*

**CHIAVENATO, Idalberto** *- Recursos Humanos Edição Compacta . São Paulo : Atlas, 2002. 7ª edição.*

**CRUZ, Tadeu** *- Sistemas de Informação: Tecnologia da Informação e a empresa do século XXI. Ed. Atlas, 1998.*

**DRUCKER, Peter F***. - Fator Humano e Desempenho. Ed. Livraria Pioneira, 1997. 3ª Edição.*

**FISCHER, André Luiz; NOGUEIRA, Arnaldo José França Mazzei** *– As Pessoas na Organização. São Paulo : Editora Gente, 2002, 2ª edição.*

**GATES, Bill** *. - A* estradadofuturo*. São Paulo : Companhia das Letras, 1995*

**GUFFEY, Mary E.** - *APA style electronic formats*, originalmente publicado em Business Communication Quarterly, Mar., pp. 59-76, [<http://www.westwords.com/GUFFEY/apa.html>](http://www.westwords.com/GUFFEY/apa.html) Acesso em: abril de 2017

**KEEN, P. G. W***. – Guia Gerencial para a Tecnologia da Informação. Ed. Campus 1996. 2ª Edição.*

**OLIVEIRA, Jayr Figueiredo de** – Sistemas de Informação: Um Enfoque Gerencial Inserido no Contexto Empresarial e Tecnológico. Ed. Érica, 2000.

**SANTOS, Fernando César Almada***. - Estratégia de Recursos Humanos: Dimensões Competitivas. São Paulo: Atlas, 1999a.*

**SANTOS, Fernando César Almada***. - Estratégia de Recursos Humanos: Dimensões Corporativaqs. São Paulo: Atlas, 1999b.*

**TACHIZAWA, Takeshy; et alli***. - Gestão com pessoas: uma abordagem aplicada às estratégias de negócios. Ed. FGV, 2001.*

**TOLEDO, Flávio de.** *- O que são recursos humanos. São Paulo : Ed. Brasiliense, 1993a*

**TOLEDO, Flávio de.** *- O que são recursos humanos II. São Paulo : Ed. Brasiliense, 1993b*

**TORREÃO, Paula. História do Gerenciamento de Projetos**, 2007. Disponível em:< <https://pontogp.wordpress.com/2007/04/23/historia-do-gerenciamento-de-projetos/>>. Acesso em: abril de 2017.

**VASCONCELLOS, Eduardo ; MARCOVITCH, Jacques**. *Gerenciamento da Tecnologia: um instrumento para a competitividade empresarial. Ed. Edgard Blücher Ltda, 1997.*

**WALTON, Richard E***. – Tecnologia de Informação: O uso de TI pelas empresas que obtêm vantagem competitiva. São Paulo: Atlas, 1998*

**Anexos**

É opcional. Documentos agregados à obra para fins de comprovação de dados ou ilustração.

**Glossário**

É um item opcional. Trata-se de uma listagem que contém as palavras ou termos técnicos desconhecidos utilizados no texto, com seus significados. A lista deve ser em ordem alfabética.

**Exemplo:**

**SGBD** – Sistema Gerneciador de Banco de Dados. Software que gerencia e proporciona o armazenamento de dados, permitindo consultas aos dados armazenados e garantindo sua integridade.

**Sistemas de Informação Gerencial** ou **ERP** – **E**nterprise **R**esource **P**lanning ou software de planejamento de recursos empresariais. É um software que procura integrar todas as áreas da empresa, desde o chão de fábrica até a alta administração, procurando otimizar processos e garantir confiabilidade das informações.

**Workflow** – Software que procura gerenciar e descrever o fluxo de dados entre as tarefas e processos da organização.

**Apêndice**

É opcional – São documentos agregados à obra para fins de apoio à argumentação. Nesta parte são incluídos os questionários, entrevistas, tabulação de dados, etc.

# Mais orientações sobre os padrões a serem usados .

# Padrões de Formatação:

# 1. Títulos use letra Arial ou Times New Roman, 16, negrito

* 1. **Subtítulos, Arial ou** Times New Roman**, tamanho 14, negrito**

**Corpo do texto:** Todo o corpo do texto deverá estar formatado com letra Arial ou Times New Roman tamanho 12. Espaçamento entre linhas 1,5.

**Itálico:** Deve ser usado nas palavras de outros idiomas. Esta orientação não se aplica às expressões latinas apud e et al.

**Formatação da página:** Margens: Direita e inferior: 2cm / Esquerda e superior: 3cm

Espaçamento entre linhas 1,5

**Referências para elaboração deste documento**

**Borges, Daniella A. Franceschinelli.**  Material da disciplina de Engenharia de Software III , Fatec Sorocaba, 2017

**IFSC,2018 -** Dicas para escrita de texto cientifico. Disponível em **:** <https://wiki.sj.ifsc.edu.br/wiki/index.php/Dicas_para_escrita_de_texto_cient%C3%ADfico> Acesso em: 25/04/2018

**Medeiros, Ernani Sales de.** Desenvolvendo Software com UML. Makron Books – São Paulo, 2004

**Munhoz, Levi Rodrigues**. Material da disciplina de Engenharia de Software II, Fatec Sorocaba, 2017

**Normas ABNT.** Disponível em [https://www.normaseregras.com/normas-abnt/](https://www.normaseregras.com/normas-abnt/%20%20)  Acesso em: 17/04/2018

**Sommerville, Ian*.*** Engenharia de Software. Ed. Addison Wesley - São Paulo, 2003 **Nenhuma entrada de índice remissivo foi encontrada.**