

Curso de Análise e Desenvolvimento de Sistemas

Kelvin Cluxnei

Documentação de Desenvolvimento de Software

Título: Mãos que alimentam

Sorocaba/SP

Junho 2020

**Kelvin Cluxnei**

Documentação de Desenvolvimento de Software

Título: **Mãos que alimentam**

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

Orientador: Nome do orientador

Sorocaba/SP

Junho 2020

**Dedicatória**

Exemplo de dedicatória

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

Agradecimentos

Neste item podem ser incluídas todas as pessoas que de alguma forma colaboraram para a realização do trabalho: amigos, pais, professores, coordenadores etc. O texto abaixo é apenas um exemplo.

A Deus, que segura minhas mãos e guia meus passos.

À minha família, minha maior bênção: meus filhos fulano, beltrano e ciclano, minha mulher Beltrana.

Aos meus pais, Maria e João, pela dedicação e amor infinito.

Aos amigos que colaboraram das mais diversas formas na elaboração deste trabalho: Mariana, Sérgio, Juliana.....

Ao José da Silva, pelas informações preciosas sobre os processos do setor de RH

À Profª Rita Souza, pela orientação precisa e dedicação incansável, etc.

Resumo

1,3 bilhão de toneladas são desperdiçadas ou se perdem ao longo das cadeias produtivas de alimentos, ou seja, 30% de toda a comida produzida por ano no mundo.

Nações Unidas para a Alimentação e a Agricultura (FAO – ONU)

A necessidade da diminuição do desperdício de alimentos bem como a união de potenciais entidades doadoras e receptoras sempre vai existir e atualmente o número de pessoas que passam fome na região de Sorocaba é grande. Já existem muitas entidades doadoras que fazem a diferença na região de Sorocaba, que podem ser potencializadas pelo trabalho proposto, reconhecendo com facilidade as demandas de necessidades e conhecendo outras instituições com o mesmo Objetivo. Será desenvolvido um aplicativo totalmente gratuito que é o modelo principal e um painel de controle para monitorar e gerenciar o aplicativo e uma página para divulgação do mesmo, que terá como objetivo fazer a ponte do compartilhamento de alimentos para quem precisa, através das entidades doadoras e também será desenvolvido uma página para apresentação, divulgação e marketing.

Palavras-chave: Aplicativo, Alimentos, Doação, Web, Mobile.

# 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, 14, negrito)

1,3 bilhão de toneladas são desperdiçadas ou se perdem ao longo das cadeias produtivas de alimentos, ou seja, 30% de toda a comida produzida por ano no mundo.

Nações Unidas para a Alimentação e a Agricultura (FAO – ONU)

A necessidade da diminuição do desperdício de alimentos bem como a união de potenciais entidades doadoras e receptoras sempre vai existir, atualmente o número de pessoas que passam fome na região de Sorocaba é grande.

O grupo Amizadaria Solidária nasceu em 2015 através de um encontro de mães e filhos, o qual se voluntariaram a cozinhar, levar alimento, calor humano e esperança às pessoas em situação de rua na cidade de Sorocaba e região.

O trabalho voluntário, sem fins lucrativos e apolíticos, do grupo junto aos carentes, vai desde alimentação, distribuição de roupas e cobertores, abrangendo, entre outras atividades, encaminhamento de pessoas com dependência para tratamento em comunidades terapêuticas e buscando também manter com doações estas comunidades.

Assim como o grupo Amizadaria outras iniciativas podem ter o trabalho facilitado e organizado com o uso do aplicativo proposto, reconhecendo com facilidade as demandas de necessidades e conhecendo outras instituições com o mesmo Objetivo.

No lado do servidor (backend) será utilizado o framework Laravel escrito na linguagem PHP, com banco de dados relacional Mysql/MariaDB bem como o site de apresentação, divulgação e marketing .

No painél de controle (frontend), será utilizado o pacote Laravel/AdminLTE para integração com o backend e estruturação, HTML, CSS com preprocessador SCSS, e JS com preprocessador baseado em NODE.JS arquitetados pelo Laravel Mix (webpack).

O aplicativo será desenvolvido utilizando a tecnologia do Facebook: React Native, framework que possibilita a criação de aplicativos tanto pra Android quando para IOS com o mesmo código.

Se comunicará atravéz de uma API escrita em Laravel.

A sistema administrativo irá rodar em um servidor Linux com apache2, Mysql e PHP 7.4.

Atualmente o aplicativo web Comida Invisível atua na mesma área, porém não na região de Sorocaba, funciona assim: quem quer doar alimentos próprios para o consumo, mas sem valor comercial, cadastra-se na página e informa o que tem, indicando a validade, data e a forma de entrega. Feito isso, a doação aparece como disponível para as entidades interessadas, que distribuem ou preparam comida nas proximidades. Se a doação for aceita, o doador confirma se fará a entrega ou se vai aguardar a retirada.

O diferencial do aplicativo em relação ao Comida Invisível é a facilidade do mundo mobile, estará disponível tando pra Android quando para IOS, e com as portas abertas para uma versão Web, além da região de atuação.

Será desenvolvido um aplicativo totalmente gratuito que é o modelo principal e um painel de controle para monitorar e gerenciar o aplicativo e uma página para divulgação do mesmo, que terá como objetivo fazer a ponte do compartilhamento de alimentos para quem precisa, através das entidades doadoras.

Também será desenvolvido uma página para apresentação, divulgação e marketing.

Será necessário uma hospedagem com servidor Linux para manter API e o painel do sistema.

A proposta é que o aplicativo rode na região de Sorocaba inicialmente somente entre entidades Jurídicas (Organizações) e não diretamente com as entidades Físicas.

Com o aplicativo finalizado, é esperado um crescimento da área de atuação das doações, não somente na região de Sorocaba mas também no resto do país.

# 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 em que o software desenvolvido utilizar IoT (Internet das Coisas) para a solução do problema, apresentar os conceitos de IoT, indicando as referências utilizadas conforme padrão ABNT.

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”.

Obs: Ao inserir uma imagem é obrigatório numerar, descrever e indicar a fonte como nos exemplos abaixo. Também é necessário fazer referência ao elemento gráfico (figura, tabela, quadro, gráfico) no texto:

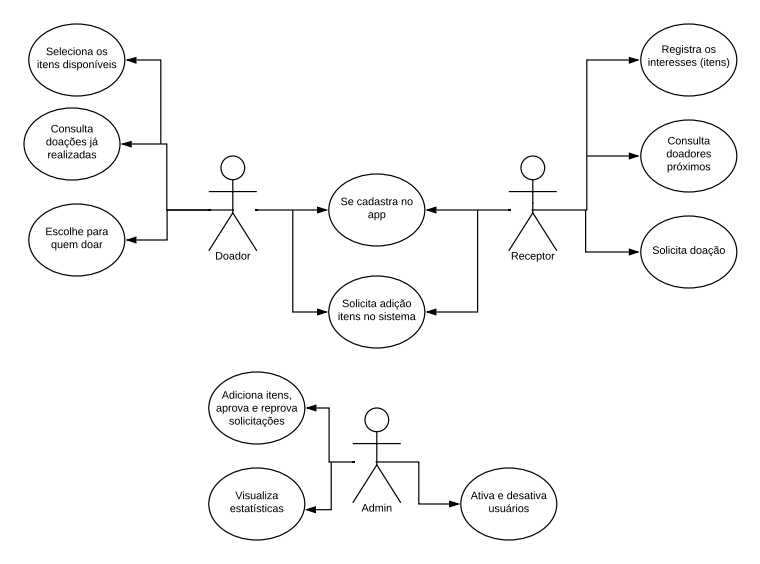
Na Figura 1 apresenta-se o Site da Amazon.

Figura 1 - Site Amazom.com em 1995



Fonte: Disponível em: https://www.businessinsider.com

Figura *2* - Diagrama de Casos de Uso



Fonte: elaborada pelo autor.

# 3. Análise de Requisitos

# 3.1 Visão geral do Produto

Descrever o tema do projeto e os seus objetivos gerais. Suas funcionalidades deverão ser indicadas de forma resumida. A visão geral pode fornecer informações sobre um possível caminho para solucionar o problema já identificado.

# 3.2 Descrição e Delimitação do problema

Deverá ser descrito: Qual é o problema a ser solucionado? Qual o público alvo?

Definir o escopo do projeto. Delimitar o que será feito e o que não será. Deixar claro a ideia central do projeto, o domínio do problema. Exemplo:

“Atualmente o processo de controle de estágio supervisionado na Faculdade XXXX não é informatizado. Os relatórios de estágio são escritos em papel e são entregues somente no final da validação das horas de estágio para o coordenador de curso/estágio. O maior problema em torno deste processo é a falta de mecanismos para gerenciamento dos estágios em andamento e dos já concluídos. Os dados dos alunos e respectivos estágios são digitados em planilhas eletrônicas. Não existe também um meio fácil para a elaboração de relatórios. Além disso, a segurança destas informações não é garantida uma vez que permanecem armazenadas nas máquinas dos funcionários responsáveis.

A implantação do Sistema Gerenciador de Estágio Supervisionado, permitirá ao setor de estágio da Faculdade XXXX, cadastrar as informações, obter dados e levantamentos de relatórios mais rapidamente, além de permitir que o próprio aluno informe o andamento de seu estágio por meio de relatórios preenchidos de forma online. Este sistema será uma aplicação web permitindo acesso aos alunos, coordenadores de curso e funcionários do setor de estágio.

O Sistema não será um ambiente para divulgação de estágio pelas empresas.”

3.3 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. Se necessário incluir documentos incluí-los no Apêndice.

Também poderão ser descritas ferramentas existentes no mercado com funcionalidades semelhantes e que tenham sido utilizadas como base para a definição do projeto. Justificar a escolha do tema.

# 3.4 Requisitos de Software

3.4.1 Requisitos Funcionais

Descrever quais são os requisitos funcionais da aplicação a ser desenvolvida. Os requisitos funcionais do sistema definem as funções que o sistema deve oferecer. Expressam o comportamento de um software, são as necessidades apontadas pelo cliente, ou seja, o que ele quer que o sistema faça. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer (Sommerville,2000).

Os requisitos deverão ser descritos e numerados como o exemplo abaixo.

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.

3.4.2 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, segurança e espaço em disco.

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)

Os requisitos não funcionais deverão ser descritos e numerados tal como os requisitos funcionais.

**3.4.3 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. Atenção: numerar como “Quadro” e não “Figura”.

Quadro 1. Caso de uso – Consultar Exames Agendados

|  |  |  |
| --- | --- | --- |
| **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 |

4. 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 Development(XGD) ou alguma sistemática indicada por algum especialista no assunto. O mesmo pode ocorrer com desenvolvimento ágil para aplicações móveis ou web.

4.1 Arquitetura da aplicação Atual

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

4.2 Tecnologias utilizadas e APIs

Descrever as tecnologias que serão utilizadas para desenvolvimento da aplicação, principalmente se for uma tecnologia nova. Exemplo: nova linguagem, framework, banco de dados, API ou hardware. Se necessário podem ser incluídas subseções. Indicar referências.

Exemplo:

* OpenCV

OpenCV[[1]](#footnote-2), também chamado de Open Source Computer Vision, é uma biblioteca de visãocomputacional. Inicialmente, foi desenvolvida pela Intel, mas hoje é mantida por uma ampla comunidade de programadores independentes, empresas e universidades, sob a licença aberta BSD. O desenvolvimento está ativo, com o último lançamento estável em julho de 2019.

* 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......etc etc....

4.3 Modelo de dados

**4.3.1 Modelo Conceitual**

Apresentar o modelo de dados que foi utilizado na aplicação indicando o tipo de banco de dados utilizado para prover a persistência dos dados (relacional, não relacional). Poderá ser usado o Diagrama Entidade-Relacionamento (DER).

4.3.2 Modelo Lógico

Definir as entidades, atributos, relacionamentos domínios e validações. Se for necessário incluir um dicionário de dados com detalhamento dos atributos, abrir uma nova subseção. 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 agregado.

4.3.3 Diagrama de Classes

Deverá ser utilizado se o desenvolvimento utilizar orientação a objetos.

4.4 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. Somente os processos mais relevantes na aplicação deverão ser representados.

4.5 Diagrama de Atividades

O Diagrama de Atividades é um diagrama comportamental (que especifica o comportamento do software), e através dele podemos modelar partes do comportamento de um software. Este diagrama deverá ser utilizado para documentar o aspecto funcional (não estrutural) do software, quando é necessário representar o fluxo da informação que o software trabalhará.

4.6 Diagrama Estado e Diagrama de Pacotes

Estes diagramas devem ser incluídos caso o orientador solicite.

4.7 Interfaces com o usuário

Apresentar aqui as interfaces com o usuário acompanhada de uma pequena explicação esclarecendo aspectos do uso. Pode ser *printscreen* das telas ou layout elaborado por alguma ferramenta.

4.8 Relatórios e documentos

Descrever os relatórios ou documentos gerados pelo software.

5. Implantação

Indicar o repositório onde o código fonte pode ser acessado. 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.

# 6. 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

< Este é um item obrigatório. Lista numerada em ordem alfabética **>**

**Atenção:**

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

**Inclua também esta ferramenta em suas referências da forma:**

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

**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.*

**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.*

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

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 de agregados à obra para fins de apoio à argumentação. São documentos elaborados pelo autor. Nesta parte são incluídos os questionários, entrevistas, tabulação de dados, etc.

Anexos

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

# Padrões de formatação s serem utilizados:

# 1. Títulos use letra Arial ou Times New Roman, 14, negrito

* 1. **Subtítulos, Arial ou** Times New Roman**, tamanho 12, 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**

**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](https://wiki.sj.ifsc.edu.br/wiki/index.php/Dicas_para_escrita_de_texto_científico) Acesso em: 25/04/2018

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

**Normas ABNT.** Disponível em https://www.normaseregras.com/normas-abnt/ Acesso em: 17/04/2018

**Sommerville, Ian*.*** Engenharia de Software. Ed. Addison Wesley - São Paulo, 2003

1. Disponível em <[https://opencv.org](https://opencv.org/)> [↑](#footnote-ref-2)