SENAC – SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL

MAURO ZAMBONI

MANUAL DE BANCO DE DADOS E PROGRAMAÇÃO

(PROJETO ONGs)

ERECHIM/RS, 2020

[1 Identificação e localização do documento 7](#_Toc62654842)

[**2** **Termo de abertura do projeto** 7](#_Toc62654843)

[3 Resumo 7](#_Toc62654844)

[**4** **Introdução** 8](#_Toc62654845)

[**5** **Escopo** 8](#_Toc62654846)

[**6** **Cronograma** 9](#_Toc62654847)

[**7** **Requisitos** 9](#_Toc62654848)

[7.1 Requisitos de interface com ambiente de software e hardware 10](#_Toc62654849)

[7.2 Requisitos de execução 10](#_Toc62654850)

[7.3 Perfis de acesso 11](#_Toc62654851)

[**8** **Requisitos funcionais e não funcionais** 11](#_Toc62654852)

[8.1 Requisitos funcionais 12](#_Toc62654853)

[8.2 Requisitos não funcionais 12](#_Toc62654854)

[8.3 Resultado final do sistema 13](#_Toc62654855)

[8.4 Abrangência do projeto 14](#_Toc62654856)

[**9** **Mapeamento de requisitos com casos de uso** 14](#_Toc62654857)

[9.1 Diagramas de atividades pelos usuários 15](#_Toc62654858)

[9.2 Diagrama de cadastro de login 15](#_Toc62654859)

[9.3 Diagrama de atividades do administrador 16](#_Toc62654860)

[**10** **Modelagem de dados** 16](#_Toc62654861)

[10.1 Modelo conceitual 16](#_Toc62654862)

[10.2 Modelo lógico 17](#_Toc62654863)

[10.3 Modelo físico 18](#_Toc62654864)

[**11** **Plano financeiro** 19](#_Toc62654865)

[11.1 Estimativa dos custos 20](#_Toc62654866)

[**12** **Máquinas e equipamentos** 20](#_Toc62654867)

[12.1 Do desenvolvedor 20](#_Toc62654868)

[**13** **Softwares livres** 20](#_Toc62654869)

[14 Servidor Web Xampp 21](#_Toc62654870)

[15 Banco de dados 21](#_Toc62654871)

[15.1 MongoDB 21](#_Toc62654872)

[15.2 MySQL WorkBench 22](#_Toc62654873)

[16 Editor de texto 22](#_Toc62654874)

[16.1 Visual Studio Code 22](#_Toc62654875)

[17 Ambiente de Execução JavaScript 22](#_Toc62654876)

[17.1 Node.js 22](#_Toc62654877)

[17.2 **Npm** 23](#_Toc62654878)

[18 Express 24](#_Toc62654879)

[19 Electron 25](#_Toc62654880)

[20 Nodemon 25](#_Toc62654881)

[21 Express Session 25](#_Toc62654882)

[22 Html 25](#_Toc62654883)

[**23** **CSS** 26](#_Toc62654884)

[**24** **Bootstrap** 26](#_Toc62654885)

[**25** **Sequelize** 26](#_Toc62654886)

[**26** **Handlebars** 27](#_Toc62654887)

[**27** **jQuery** 27](#_Toc62654888)

[**28** **MySQL2** 27](#_Toc62654889)

[**29** **Body-parser** 28](#_Toc62654890)

[**30** **Popper.js** 28](#_Toc62654891)

[**31** **Package.json** 28](#_Toc62654892)

[**32** **URI** 29](#_Toc62654893)

[**33** **Roteamento** 29](#_Toc62654894)

[33.1 Req e Res 29](#_Toc62654895)

[33.2 Métodos Get e Post 29](#_Toc62654896)

[**34** **Localhost** 29](#_Toc62654897)

[35 **Algoritmo** 29](#_Toc62654898)

[35.1 Estrutura de arquivos 30](#_Toc62654899)

[35.2 Função callbacks 30](#_Toc62654900)

[35.3 CRUD 30](#_Toc62654901)

[35.4 Login 30](#_Toc62654902)

[35.5 Logoff 30](#_Toc62654903)

[35.6 Erros e ações do algoritmo 30](#_Toc62654904)

[**36** **Tela Inicial** 30](#_Toc62654905)

[36.1 Rota Página Inicial 31](#_Toc62654906)

[**37** **Tela O Projeto** 31](#_Toc62654907)

[**38** **Página de Cadastro** 32](#_Toc62654908)

[REFERÊNCIAS 34](#_Toc62654909)

**Lista de Tabelas**

[**Tabela 1 - Termo de Abertura do Projeto** 7](#_Toc62654833)

[**Tabela 2 - Cronograma** 9](#_Toc62654834)

[**Tabela 3 – Requisitos de Interface Operacional do Sistema** 10](#_Toc62654835)

[**Tabela 4 - Requisitos de Execução** 10](#_Toc62654836)

[**Tabela 5 - Perfis de Acesso** 11](#_Toc62654837)

[**Tabela 6 - Requisitos Funcionais** 12](#_Toc62654838)

[**Tabela 7 - Requisitos Não Funcionais** 13](#_Toc62654839)

[**Tabela 8 - Resultado Final do Sistema** 13](#_Toc62654840)

[**Tabela 9 - Estimativa de Custos** 20](#_Toc62654841)

**Lista de Figuras**

[**Figura 1 - Diagrama de atividades pelos usuários** 15](#_Toc62654823)

[**Figura 2 – Diagrama de cadastro de login** 15](#_Toc62654824)

[**Figura 3 – Diagrama atividades do administrador** 16](#_Toc62654825)

[**Figura 4 - Modelo Conceitual** 17](#_Toc62654826)

[**Figura 5 - Modelo Lógico** 18](#_Toc62654827)

[**Figura 6 - Modelo Físico** 19](#_Toc62654828)

[**Figura 7 - Logo Senac** 30](#_Toc62654829)

[**Figura 8 - Página Inicial** 31](#_Toc62654830)

[**Figura 9 - Página o Projeto** 32](#_Toc62654831)

[**Figura 10 - Página Cadastro** 33](#_Toc62654832)

**Lista de Quadros**

[**Quadro 1 - Computador Desenvolvedor** 20](#_Toc62654814)

[**Quadro 2 - Computador Senac** 20](#_Toc62654815)

[**Quadro 3 - Comandos NPM** 23](#_Toc62654816)

[**Quadro 4 - Comandos Express** 24](#_Toc62654817)

[**Quadro 5 - Comandos Nodemon** 25](#_Toc62654818)

[**Quadro 6 - Comandos MysSQL2** 27](#_Toc62654819)

[**Quadro 7 - Dados que compõe um Package.json** 28](#_Toc62654820)

[**Quadro 8 - Exemplo de Rota** 29](#_Toc62654821)

1. Identificação e localização do documento

Repositório em Git Hub[[1]](#footnote-1) - https://github.com/maurozamboni/ManualProjetoOngs.git

1. **Termo de abertura do projeto**

**Tabela 1 - Termo de Abertura do Projeto**

|  |  |
| --- | --- |
| Termo de Abertura | |
| Início | 11/2020 |
| Término | 04/2021 |
| Projeto | Sistema de Gerenciamento de Eventos para ONGs |
| Solicitante | Leandro Luis Bianchi - Diretor do Senac de Erechim |
| Coordenador banco de dados | Rogério Ciotti |
| Coordenador programação | André Bruscatto |
| Empresa solicitante | Não há |
| Desenvolvedor | Mauro Zamboni |

Fonte: Elaborado pelo autor.

1. Resumo

Este trabalho detalha o plano de Gerenciamento do Projeto de customização e implantação de um sistema de gerenciamento de dados para Entidades Não Governamentais (ONGs). Esse sistema de gestão otimiza as atividades operacionais e administrativas incluindo o fluxo de operações dentro do sistema de arrecadação de bens e serviços ofertados pelas entidades.

1. **Introdução**

A tecnologia da informação está transformando a gestão de negócios no meio empresarial, desta forma surge a ideia de encontrar uma alternativa de concentrar ás informações que unem quem queira fazer o bem para quem necessite ser ajudado.

A utilização de um sistema que agilize e administre melhor essa relação com controle de qualidade e quantidade é melhor saída nos dias atuais.

O site será do tipo responsivo que são aqueles que se adaptam ao tamanho das suas páginas ao tamanho das telas que estão sendo exibidas, como as telas de celulares e tablets, podendo desta forma ser aplicado para aplicativos web e mobile.

1. **Escopo**

A elaboração de um Sistema de Informação que coopere com a interação entre entidades não governamentais e pessoas dispostas a colaborar com os projetos destas.

O Sistema deverá fazer a divulgação e acompanhamento dos projetos publicados além de informar o resultado parcial e total dos eventos que solicitarem produtos.

Os usuários entidades não governamentais poderão publicar seus eventos inserindo logotipo, fotos, vídeos, forma de contato e ponto de recolhimento dos produtos.

Os usuários doadores poderão pesquisar todos os eventos, porém somente com login poderão realizar a doação, inserindo fotos ou vídeos caso necessitem tirar dúvidas com as entidades.

O sistema irá realizar o controle da quantidade de produtos solicitados com os doados para não realizar entradas não necessárias, assim haverá mais distribuição entre as doações.

1. **Cronograma**

**Tabela 2 - Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarefa** | | **Previsto** | **Realizado** |
| Início do projeto |  | 28/01/2021 | 28/01/2021 |
| Cronograma |  | 28/01/2021 | 28/01/2021 |
| Requisitos |  | 28/01/2021 | 28/01/2021 |
| Mapeamento de requisitos |  | 28/01/2021 | 28/01/2021 |
| Modelagem de dados |  | 28/01/2021 | 28/01/2021 |
| Planejamento financeiro |  | 28/01/2021 | 28/01/2021 |
| Máquinas e equipamentos |  | 28/01/2021 | 28/01/2021 |
| Modelagem banco de dados |  | 28/01/2021 | 28/01/2021 |
| Lista de softwares |  | 28/01/2021 |  |
| Design de telas |  | 04/02/2021 |  |
| Banco de dados |  | 15/03/2021 |  |
| Algoritmo |  | 15/03/2021 |  |
|  | Cadastro de informações de pessoas | 15/03/2021 |  |
|  | Cadastro e publicações dos eventos | 15/03/2021 |  |
|  | Controle de produtos | 15/03/2021 |  |
|  | Geração de relatórios | 15/03/2021 |  |
|  | Inclusão de imagens e vídeos | 15/03/2021 |  |
|  | Gerenciamento pelo usuário administrador de seus eventos e voluntários | 15/03/2021 |  |
|  | Mensagens em caso de falha de comunicação com o banco de dados | 15/03/2021 |  |
|  | Mostrar entidades no google mapas | 15/03/2021 |  |
|  | Mostrar pontos de coletas no google mapas | 15/03/2021 |  |
|  |  | 15/03/2021 |  |
| Entrega primeira versão |  | 15/03/2021 |  |

Fonte: Elaborado pelo autor

1. **Requisitos**

Na abordagem clássica de engenharia[[2]](#footnote-2), conjuntos de requisitos são tipicamente utilizados como informações fundamentais para a fase de [projeto](https://pt.wikipedia.org/wiki/Projeto) de um produto ou serviço, especificando as propriedades e [funções](https://pt.wikipedia.org/wiki/Fun%C3%A7%C3%B5es) necessárias (ou desejáveis) a serem consideradas no desenvolvimento do projeto em questão.

* 1. Requisitos de interface com ambiente de software e hardware

**Tabela 3 – Requisitos de Interface Operacional do Sistema**

|  |  |  |
| --- | --- | --- |
| **[RI001]** | Nível Cliente | Microcomputador, leptot ou smartphone |
| **[RI002]** | Nível Cliente | Qualquer navegador web |
| **[RI003]** | Nível Cliente | Aplicativo da Microsoft ou Libre Office |
| **[RI004]** | Nível Administrador | Microcomputador, leptop |
| **[RI005]** | Nível Administrador | Sistema operacional Windows ou Linux |
| **[RI006]** | Nível Administrador | Qualquer navegador Web |

Fonte: Elaborado pelo autor.

* 1. Requisitos de execução

**Tabela 4 - Requisitos de Execução**

|  |  |
| --- | --- |
| **[RE001]** | Tempo máximo de resposta das aplicações de 3 segundos |
| **[RE002]** | Acesso ilimitado de pessoas |
| **[RE003]** | Tempo de resposta a busca no banco de dados de 5 segundos |

Fonte: Elaborado pelo autor.

* 1. Perfis de acesso

**Tabela 5 - Perfis de Acesso**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome** | **Módulos** | **Cadastrar** | **Visualizar** | **Login** |
| **Internauta** | Tem acesso para pesquisar os eventos, doações, pontos coletas | | | |
|  | Pessoa | Não | Não | Não |
|  | Evento | Não | Sim | Não |
|  | Produto | Não | Sim | Não |
|  | Voluntariado | Não | Sim | Não |
|  | Coleta | Não | Sim | Não |
| **Administrador** | Responsável pelos eventos, coleta, recebimento das doações e voluntariados | | | |
|  | Pessoa | Sim | Sim | Sim |
|  | Evento | Sim | Sim | Sim |
|  | Produto | Sim | Sim | Sim |
|  | Voluntariado | Sim | Sim | Sim |
|  | Coleta | Sim | Sim | Sim |
| **Pessoa** | Pode ser cadastrado no sistema fazer doações, ser voluntário ou auxiliar na coleta de doações | | | |
|  | Pessoa | Sim | Não | Sim |
|  | Evento | Não | Sim | Sim |
|  | Produto | Não | Sim | Sim |
|  | Voluntariado | Sim | Sim | Sim |
|  | Coleta | Sim | Sim | Sim |

Fonte: Elaborado pelo autor.

1. **Requisitos funcionais e não funcionais**

Em primeira visita e exposição do coordenador do Senac de Erechim foram alevantados os primeiros requisitos para o desenvolvimento do Projeto. Naquela situação foi exposto que seria realizado uma plataforma para aproximar organizações não governamentais e doadores.

Sendo assim, foi identificada a necessidade de criar dois tipos de usuários, um como doador que restringe a doar objetos e o outro o administrador que deve gerenciar as doações que foram registradas na plataforma durante o evento, como organizar o recolhimento, emitir relatórios de produtos arrecadados, organização dos pontos de apoio para o recolhimento e formas de contato para esclarecer dúvidas.

* 1. Requisitos funcionais

Os requisitos funcionais para um sistema descrevem a funcionalidade ou serviços que se espera que o sistema forneça. Eles dependem do tipo de software que está sendo desenvolvido, dos usuários de software que se espera verificar e do tipo de sistema que está sendo desenvolvido (SOMMERVILLE, 2004).

**Tabela 6 - Requisitos Funcionais**

|  |  |
| --- | --- |
| **[RF001]** | Manter cadastro de informações de pessoas |
| **[RF002]** | Manter cadastro e publicações dos eventos |
| **[RF003]** | Manter controle de produtos |
| **[RF004]** | Todos os requisitos de manter será gerado relatório |
| **[RF005]** | Permitir aos usuários inclusão de imagens e vídeos |
| **[RF006]** | Permitir ao usuário administrador o gerenciamento de seus eventos e voluntários |
| **[RF007]** | Permitir exibir mensagens em caso de falha de comunicação com o banco de dados |
| **[RF008]** | Mostrar entidades no google mapas |
| **[RF009]** | Mostrar pontos de coletas no google mapas |

Fonte: Elaborado pelo autor.

* 1. Requisitos não funcionais

Os requisitos não funcionais são aqueles que não estão diretamente relacionados à funcionalidade de um sistema, eles abordam aspectos de qualidade importantes em sistema de software (CHUNG, NIXON, YU, MYLOPOULOS, 2000).

**Tabela 7 - Requisitos Não Funcionais**

|  |  |  |
| --- | --- | --- |
| **[RNF001]** | Usabilidade | Deverá ser operável por usuários sem a necessidade de treinamento prévio |
|  | Usabilidade | Deverá fazer uso de design responsivo na implementação de suas interfaces gráficas. Deverá se comportar adequadamente em navegadores acessados via computador, smartphone e tablet |
|  | Usabilidade | Deverá estar disponível em português |
|  | Usabilidade | As operações devem ser efetivamente utilizadas e consideradas úteis pelo usuário, fáceis de aprender e reaprender, de fácil recordação |
| **[RNF002]** | Disponibilidade | Estará disponível 24 horas por dia, 7 dias por semana, com a possibilidade de paradas pré-programadas para manutenção |
| **[RNF003]** | Portabilidade | Deverá ser suportado por todos os navegadores com suporte ao JavaScript |
| **[RFN004]** | Segurança | Apenas os cadastrados devem possuir acesso ao sistema |
|  | Segurança | Deverá fornecer mecanismos de segurança e autenticação |
|  | Segurança | Não permitirá funções exclusivas de doações em moeda corrente |
| **[RFN005]** | Implementação | Será desenvolvido utilizando a linguagem JavaScript, banco de dados MySQL, servidor de aplicação Node.js |
|  | Implementação | Deverá ser implementado utilizando um sistema de gerenciamento de dados relacional com licença open source |
| **[RFN006]** | Éticos | Não apresentará aos usuários quaisquer dados de caráter privativo, tais como informações pessoais de outros usuários |
|  | Éticos | Terá no cadastro do usuário um termo de uso do sistema |

Fonte: Elaborado pelo autor.

* 1. Resultado final do sistema

**Tabela 8 - Resultado Final do Sistema**

|  |  |
| --- | --- |
| **[RFS001]** | Aumentar e agilizar os eventos a serem publicados |
| **[RFS002]** | Crescimento de voluntários integrados aos eventos |
| **[RFS003]** | Melhora nos índice de arrecadação de doações |
| **[RFS004]** | Melhor distribuição de produtos entre as ONGs |

Fonte: Elaborado pelo autor.

* 1. Abrangência do projeto

O sistema será desenvolvido com a finalidade de auxílio ás regiões que compreendem a AMAU – Associação de Municípios do Alto Uruguai.

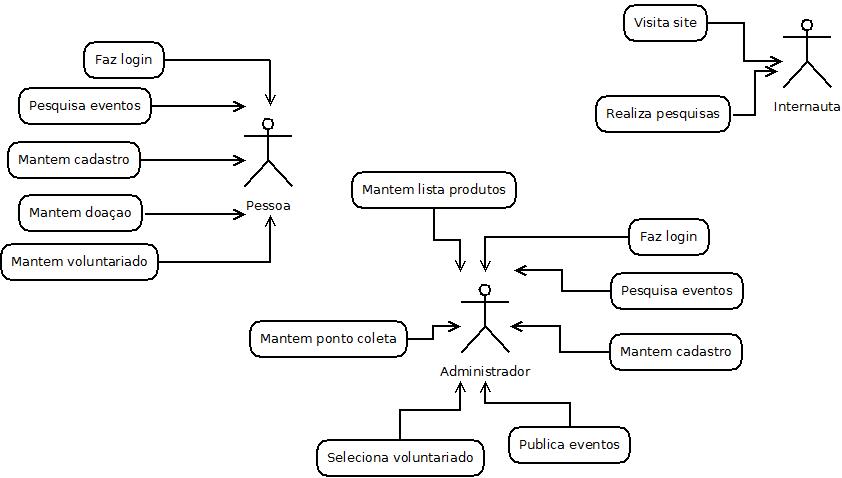
1. **Mapeamento de requisitos com casos de uso**

O diagrama de casos de uso é utilizado para descrever a funcionalidade da aplicação e a sua interação com os usuários.

Os casos de uso são apresentados utilizando o diagrama de Casos de Uso proposto pela UML[[3]](#footnote-3), sendo escrito em termos de atores, casos de uso e o sistema a ser modelado.

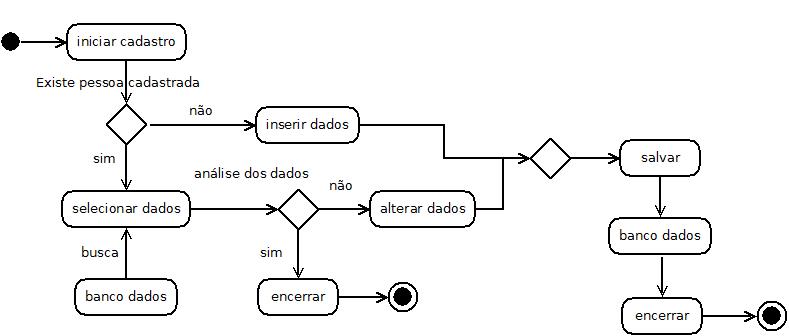
Os atores representam o papel de uma entidade externa como por exemplo um usuário, um hardware ou até mesmo um outro sistema. Os atores se comunicam com a aplicação através dos casos de uso, que representam uma sequência de ações a serem executadas.

* 1. Diagramas de atividades pelos usuários

****

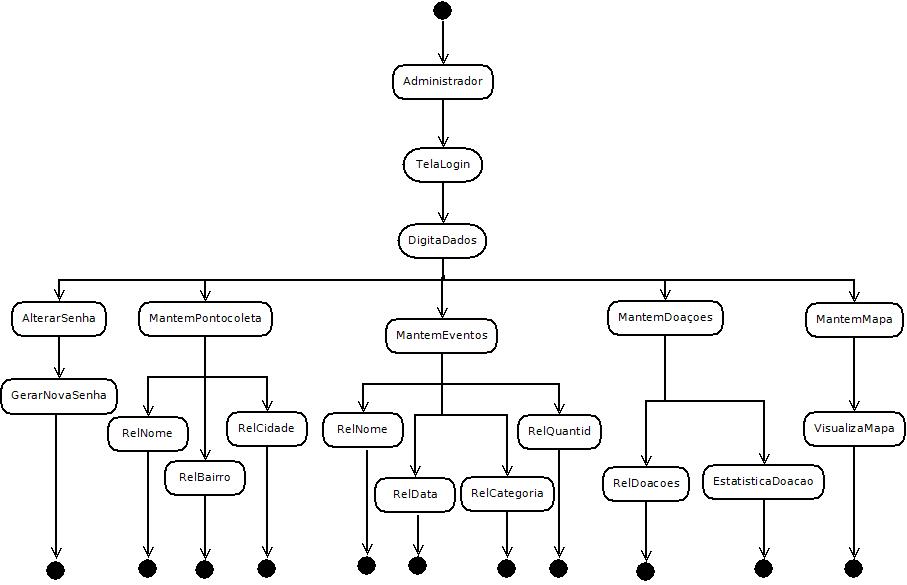
**Figura 1 - Diagrama de atividades pelos usuários**

* 1. Diagrama de cadastro de login



**Figura 2 – Diagrama de cadastro de login**

* 1. Diagrama de atividades do administrador



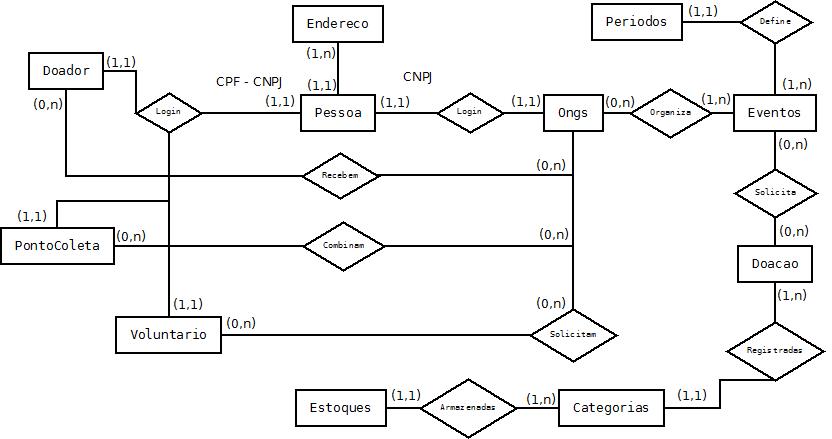
**Figura 3 – Diagrama atividades do administrador**

1. **Modelagem de dados**

Modelagem de dados[[4]](#footnote-4) é uma técnica usada para especificação das regras de negócios e as estruturas de dados de um banco de dados. Modelar dados consiste em desenhar o sistema de informações, concentrando-se nas entidades lógicas e nas dependências lógicas entre essas entidades.

* 1. Modelo conceitual

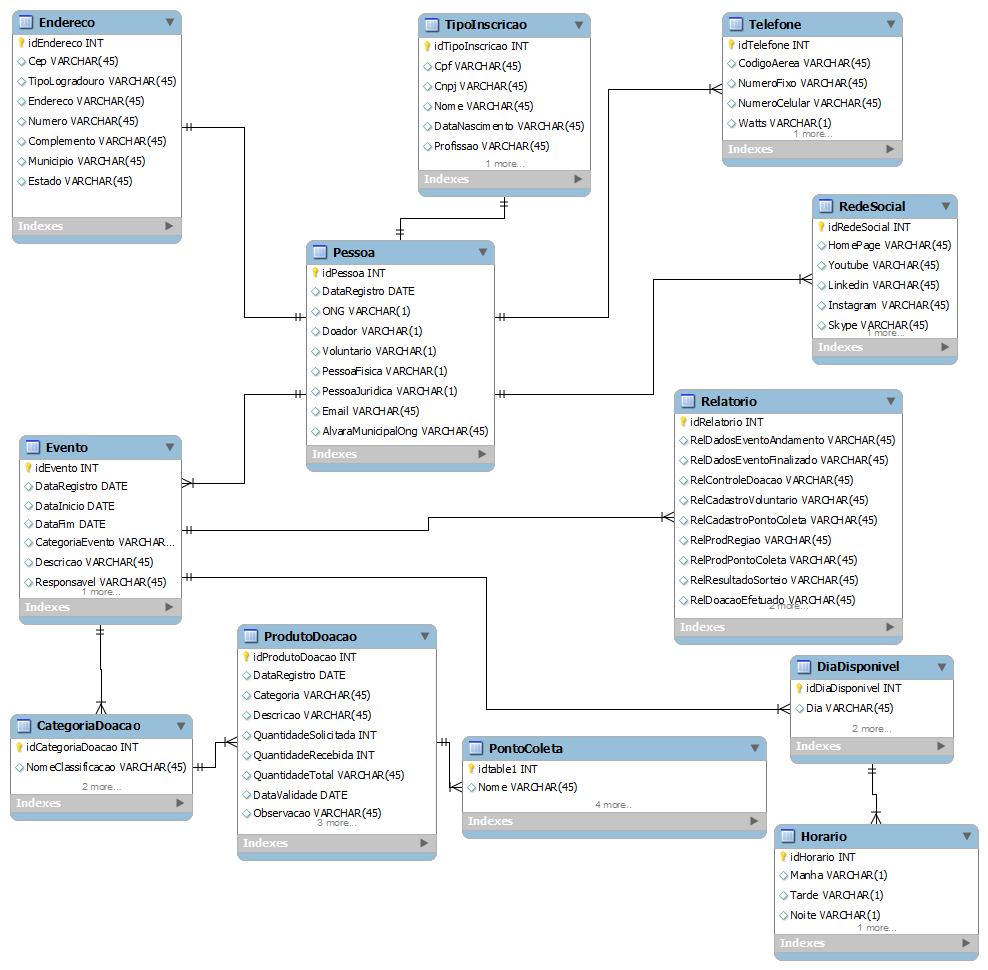
A modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco é discutir os aspectos do negócio do cliente e não da tecnologia. Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica.



**Figura 4 - Modelo Conceitual**

* 1. Modelo lógico

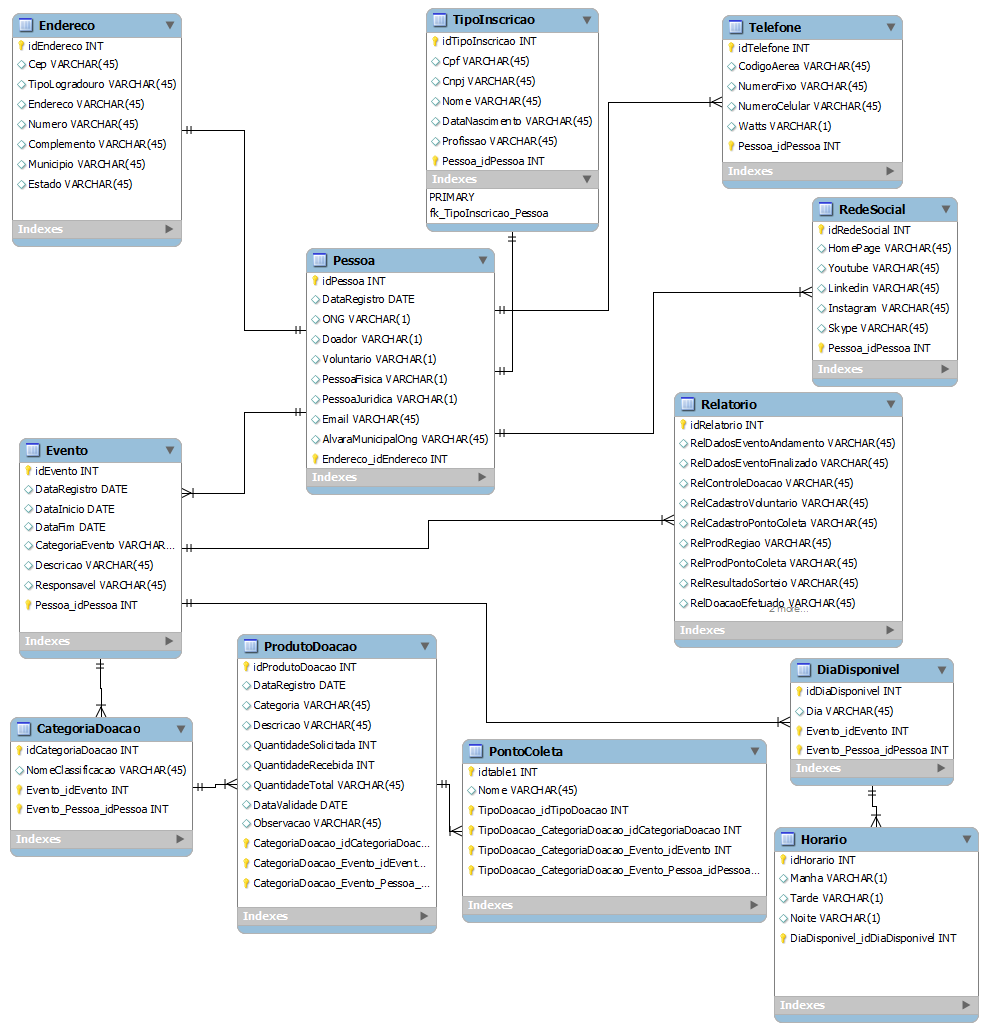
O modelo lógico já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras.



**Figura 5 - Modelo Lógico**

* 1. Modelo físico

O modelo físico fazemos a modelagem física do modelo de banco de dados. Neste caso leva-se em contas as limitações impostas pelo Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido e deve ser criado com base no modelo lógico.



**Figura 6 - Modelo Físico**

1. **Plano financeiro**

Será estimado valores de custos, porém não haverá retorno financeiro devido ser um trabalho de conclusão de curso.

Todos os aplicativos empregados no trabalho são gratuitos, para cálculo da mão de obra será levado em consideração o valor do curso que será o valor do contrato realizado em 11/05/2020 no valor total de R$ 8.228,66, considerando que este valor é do total de 3 módulos, será considerado somente um módulo respectivo da disciplina que será então o valor de R$ 2.742,89.

* 1. Estimativa dos custos

**Tabela 9 - Estimativa de Custos**

|  |  |
| --- | --- |
| **Custos** | **R$** |
| Software | 0,00 |
| Mão de obra | 2.742,89 |
| **Total** | **2.742,89** |
| Valor ressarcido | 0,00 |

Fonte: Elaborado pelo autor.

1. **Máquinas e equipamentos**
   1. Do desenvolvedor

**Quadro 1 - Computador Desenvolvedor**

|  |  |
| --- | --- |
| Processador | AMD Processor model unknown 2.70GHz |
| Memória Ram | 4,00GB |
| Sistema Operacional | 32 bits, processador com base em x64 – Windows 8.1 Pro – 2013 |

Fonte: Elaborado pelo autor.

* 1. Do Senac

**Quadro 2 - Computador Senac**

|  |  |
| --- | --- |
| Processador | Intel(R) Core™ i5-3570 CPU 3.40GHz |
| Memória Ram | 8,00GB |
| Sistema Operacional | 64 bits, processador com base em x64 – Windows 10 Pro – 1909 |

Fonte: Elaborado pelo autor.

1. **Softwares livres**

Softwares Livre[[5]](#footnote-5)s são softwares ou programas aos quais podem ser usados, copiados, alterados e distribuídos sem restrições. Geralmente eles vêm acompanhados de uma licença com o código fonte do software, essa licença garante os direitos autorais de quem criou o software. A licença mais comum é GPL (General Public License – Licença Pública Geral).

1. Servidor Web Xampp

É um pacote com os principais servidores de código aberto do mercado, incluindo FTP[[6]](#footnote-6), banco de dados MySQL[[7]](#footnote-7) e Apache[[8]](#footnote-8) com suporte às linguagens PHP[[9]](#footnote-9) e Perl[[10]](#footnote-10). Foi escolhido por ser um software livre[[11]](#footnote-11).

1. Banco de dados

Um banco de dados[[12]](#footnote-12) é usado para armazenar informações estruturadas e organizadas de forma a permitir sua recuperação rápida e eficiente. Existem diversos Modelos de banco de dados como: Em Rede, Hierárquico, Relacional e Orientado a Objeto.

* 1. MongoDB

MongoDB[[13]](#footnote-13) é um software de banco de dados orientado a documentos livre, de código aberto e multiplataforma, escrito na linguagem C++. Classificado como um programa de banco de dados NoSQL, o MongoDB usa documentos semelhantes a JSON com esquemas.

* 1. MySQL WorkBench

MySQL Workbench[[14]](#footnote-14) é um visual [de design de banco de dados](https://en.wikipedia.org/wiki/Database_design) ferramenta que integra [SQL](https://en.wikipedia.org/wiki/SQL) [desenvolvimento](https://en.wikipedia.org/wiki/Software_development), [administração](https://en.wikipedia.org/wiki/Database_administration), [design de banco de dados](https://en.wikipedia.org/wiki/Database_design), criação e manutenção em um único [ambiente de desenvolvimento integrado](https://en.wikipedia.org/wiki/Integrated_development_environment) para o [MySQL](https://en.wikipedia.org/wiki/MySQL) sistema de banco de dados. Versão instalada 5.2.31 CE em 06/2020.

1. Editor de texto
   1. Visual Studio Code

O Visual Studio Code[[15]](#footnote-15) é um editor de [código-fonte](https://pt.wikipedia.org/wiki/C%C3%B3digo-fonte) desenvolvido pela [Microsoft](https://pt.wikipedia.org/wiki/Microsoft) para [Windows](https://pt.wikipedia.org/wiki/Windows), [Linux](https://pt.wikipedia.org/wiki/Linux) e [macOS](https://pt.wikipedia.org/wiki/MacOS). Ele inclui suporte para [depuração](https://pt.wikipedia.org/wiki/Depura%C3%A7%C3%A3o), controle [Git](https://pt.wikipedia.org/wiki/Git) incorporado, [realce de sintaxe](https://pt.wikipedia.org/wiki/Realce_de_sintaxe), complementação inteligente de código, *snippets* e [refatoração de código](https://pt.wikipedia.org/wiki/Refatora%C3%A7%C3%A3o). Ele também é customizável, fazendo com que os usuários possam mudar o tema do editor, teclas de atalho e preferências. Versão instalada 1.52.1 em 06/2020.

1. Ambiente de Execução JavaScript
   1. Node.js

Node.js[[16]](#footnote-16) trata-se de um software [open-source](https://pt.wikipedia.org/wiki/Open-source)[[17]](#footnote-17), [cross-platform](https://pt.wikipedia.org/wiki/Cross-platform)[[18]](#footnote-18), e de um runtime[[19]](#footnote-19) de [JavaScript](https://pt.wikipedia.org/wiki/JavaScript) que executa código de JavaScript a nível back-end[[20]](#footnote-20) e front-end[[21]](#footnote-21). Versão instalada 14.12.0 em 07/2020.

Conforme o Ebook “Node para Iniciantes”, pág. 13 O Node.js é composto basicamente por:

* Um runtime JavaScript (GoolgeV8);
* Uma biblioteca para I/O de baixo nível (libuv);
* Bibliotecas de desenvolvimento básico (os core modules);
* Um gerenciador de pacotes via linha de comando (NPM);
* Um gerenciador de versões via linha de comando (NVM);
* Utilitário REPL via linha de comando
  1. **Npm**

NPM[[22]](#footnote-22) (Node Package Manager) é um gerenciador de pacotes para JavaScript que permite compartilhar seu código fonte, é possível instalar, atualizar, controlar versões e dependências de pacotes e também publicar seus pacotes.

**Quadro 3 - Comandos NPM**

|  |  |
| --- | --- |
| **Comandos** | **Descrição** |
| npm -v | Mostra o número da versão do npm |
| npm install --global | Instala npm no diretório do pacote – esse comando funciona como uma ferramenta para criar arquivo package.json de um projeto |
| npm install express | Framework para aplicações web e móvel |
| npm install nome\_do\_modulo | Instala um módulo no projeto |
| npm install nome\_do\_modulo -save | Instala o módulo e adiciona-o na lista de dependências do package.json do projeto |
| npm list | Lista todos os módulos existentes no projeto |
| npm remove nome\_do\_modulo | Desinstala um módulo |
| npm update nome\_do\_modulo | Atualiza a versão do módulo |
| npm init | Inicializar o projeto |

Fonte: Elaborado pelo autor.

1. Express

Express.js[[23]](#footnote-23) é um framework para aplicações web para Node.js, lançado como software livre e de código aberto sob a Licença MIT. É feito para otimizar a construção de aplicações web e APIs.

**Quadro 4 - Comandos Express**

|  |  |
| --- | --- |
| **Comandos** | **Descrição** |
| npm install --save express | Comando que irá instalar (install) o Express e salvar o nome e a versão atual no package.json como uma dependência da aplicação (--save) |

Fonte: Elaborado pelo autor.

1. Electron

O **Electron**[[24]](#footnote-24)é um framework que nos permite criar aplicações desktop para Windows, Mac e Linux utilizando tecnologias web: **HTML, CSS** e **JavaScript.** Versão instalada V8:8.3.110.13-electron.0 em 07/2020.

1. Nodemon

O nodemon[[25]](#footnote-25) é um file watcher (observador de arquivo) que roda internamente o próprio comando node. A diferença entre usá-lo ou usar o comando node é que ele faz auto-restart da aplicação, toda vez que um arquivo do projeto for modificado.

**Quadro 5 - Comandos Nodemon**

|  |  |
| --- | --- |
| **Comandos** | **Descrição** |
| npm install –g nodemon | Comando que irá instalar (install) o Nodemon |
| nodemon server.js | Comando para rodar os scripts |

Fonte: Elaborado pelo autor.

1. Express Session

|  |  |
| --- | --- |
| **Comandos** | **Descrição** |
| npm install express-session --save | Comando que irá instalar (install) o express-session |

Fonte: Elaborado pelo autor.

1. Html

HTML[[26]](#footnote-26) (abreviação para a [expressão inglesa](https://pt.wikipedia.org/wiki/L%C3%ADngua_inglesa) HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto) é uma [linguagem de marcação](https://pt.wikipedia.org/wiki/Linguagem_de_marca%C3%A7%C3%A3o) utilizada na construção de [páginas na Web](https://pt.wikipedia.org/wiki/P%C3%A1gina_web).

Documentos HTML podem ser interpretados por [navegadores](https://pt.wikipedia.org/wiki/Navegador_(inform%C3%A1tica)). A [tecnologia](https://pt.wikipedia.org/wiki/Tecnologia) é fruto da junção entre os padrões [HyTime](https://pt.wikipedia.org/wiki/HyTime) e [SGML](https://pt.wikipedia.org/wiki/SGML).

1. **CSS**

CSS[[27]](#footnote-27) é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos escritos em uma linguagem de marcação como [HTML](https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos/). O CSS separa o conteúdo da representação visual do site.

Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

1. **Bootstrap**

Bootstrap[[28]](#footnote-28) é um [framework web](https://pt.wikipedia.org/wiki/Framework_para_aplica%C3%A7%C3%B5es_web) com [código-fonte aberto](https://pt.wikipedia.org/wiki/Software_livre_e_de_c%C3%B3digo_aberto) para desenvolvimento de componentes de interface e [front-end](https://pt.wikipedia.org/wiki/Front-end) para [sites](https://pt.wikipedia.org/wiki/Site) e [aplicações web](https://pt.wikipedia.org/wiki/Aplica%C3%A7%C3%A3o_web) usando [HTML](https://pt.wikipedia.org/wiki/HTML), [CSS](https://pt.wikipedia.org/wiki/Cascading_Style_Sheets) e [JavaScript](https://pt.wikipedia.org/wiki/JavaScript), baseado em modelos de design para a [tipografia](https://pt.wikipedia.org/wiki/Tipografia), melhorando a experiência do usuário em um site amigável e [responsivo](https://pt.wikipedia.org/w/index.php?title=Responsivo&action=edit&redlink=1).

Originalmente criado por um designer e um desenvolvedor no Twitter, tornou-se uma das estruturas front-end mais populares e projetos de código aberto no mundo.

1. **Sequelize**

Sequelize[[29]](#footnote-29) é uma ferramenta Node.js ORM (Mapeamento de Objeto Relacional) para rodar com banco de dados como o MySQL, MariaDB, SQLite, Microsoft SQL Server, etc... Faz o mapeamento de dados relacionais (tabelas, colunas e linhas) para objetos Javascript.

Permite criar, buscar, alterar e remover dados de banco de dados utilizando o método JS, além de permitir a modificação da estrutura das tabelas, tendo facilidade na criação, população e migração de banco de dados.

1. **Handlebars**

Handlebars[[30]](#footnote-30) é um compilador feito em JavaScript que pega qualquer HTML e expressão Handlebars e o compila para uma função JavaScript. Esta função JavaScript derivada então pega um parâmetro, um objeto – seu dado – e retorna uma string HTML com o valor da propriedade do objeto inserida (interpolada) dentro do HTML.

1. **jQuery**

jQuery[[31]](#footnote-31) é uma biblioteca de funções JavaScript que interage com o HTML, desenvolvida para simplificar os scripts interpretados no navegador do cliente.

1. **MySQL2**

**Quadro 6 - Comandos MysSQL2**

|  |  |
| --- | --- |
| **Comandos** | **Descrição** |
| npm install -- save mysql2 | Comando que irá instalar (install) o MySQL2 |

Fonte: Elaborado pelo autor.

1. **Body-parser**

Pacote que pode ser usado para manipular solicitações JSON. Versão 1.19.0.

1. **Popper.js**

Popper.js[[32]](#footnote-32) é um motor de posicionamento, sua finalidade é calcular a posição de um elemento para que seja possível posicioná-lo próximo a um determinado elemento de referência. Usado pelo Bootstrap. Versão 1.16.1.

|  |  |
| --- | --- |
| **Comandos** | **Descrição** |
| npm install – save popper.js | Comando que irá instalar (install) o Popper.js |

Fonte: Elaborado pelo autor.

1. **Package.json**

Package.json[[33]](#footnote-33) é uma espécie de manifesto do projeto. Ele pode fazer várias coisas, completamente não relacionadas. É um repositório central de configurações de ferramentas, por exemplo. Também é onde npm armazena os nomes e versões dos pacotes instalados.

**Quadro 7 - Dados que compõe um Package.json**

|  |  |
| --- | --- |
| name | Define o nome da aplicação ou pacote |
| version | Indica a versão atual |
| description | É um resumo da sua aplicação/pacote |
| main | Define o ponto de entrada da aplicação (o que vai ser retornado se alguém der um requeri no nosso pacote |
| scripts | Define um conjunto de scripts node para executar |
| dependencies | Define uma lista de pacotes npm instalados como dependências de produção |

1. **URI**
2. **Roteamento**

O Roteamento[[34]](#footnote-34) refere-se à determinação de como um aplicativo responde a uma solicitação do cliente por um endpoint específico, que é uma URI (ou caminho) e um método de solicitação HTTP específico (GET, POST,...).

Cada rota pode ter uma ou mais funções de manipulação, que são executadas quando a rota é correspondida.

**Quadro 8 - Exemplo de Rota**

|  |  |
| --- | --- |
| Ex.: App.METHOD (PATH, HANDLER) |  |
| app | É uma instância do express |
| METHOD | É um método de solicitação |
| PATH | É um caminho no servidor |
| HANDLER | É a função executada quando a rota é correspondida |

Fonte: [Roteamento básico no Express (expressjs.com)](https://expressjs.com/pt-br/starter/basic-routing.html).

* 1. Req e Res
  2. Métodos Get e Post

1. **Localhost**
2. **Algoritmo**

1ª Passo – iniciar o xampp

2ª Passo – iniciar o nodemon server.js

* 1. Estrutura de arquivos
  2. Função callbacks

Callback[[35]](#footnote-35), é uma função que será disparada “quando” e “se” um evento acontecer. Resumidamente é uma função para ser “chamada de volta”.

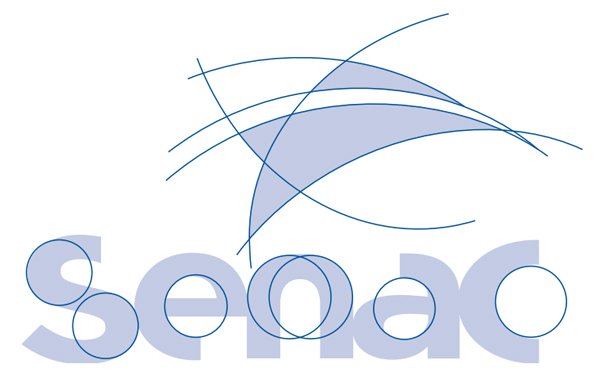
* 1. CRUD

É um acrônico para **C**reate (criar), **R**ead (ler), **U**pdate (atualizar), **D**elete (excluir). É um conjunto de operações que temos para executar nos servidores (POST, GET, PUT, DELET).

* Create (POST) – Insere um valor no banco de dados
* Read (GET) – Seleciona um ou mais valores para exibir
* Update (PUT) – Altera um valor em nosso bando de dados
* Delete (DELETE) – Remove um valor
  1. Login
  2. Logoff
  3. Erros e ações do algoritmo

1. **Tela Inicial**

Com base neste estudo do logotipo[[36]](#footnote-36) pelo autor publicado em seu post, independente de seus questionamentos realizados, surge a idea de levar esse formato circular para a tela inicial, já que a presença deste em torno da escrita “Senac”, então, Home, Cadastro, Evento, Voluntário, Pesquisa e O Projeto, receberam no momento do design uma modelagem circular, preservando a cor laranja e no cabeçalho a azul.



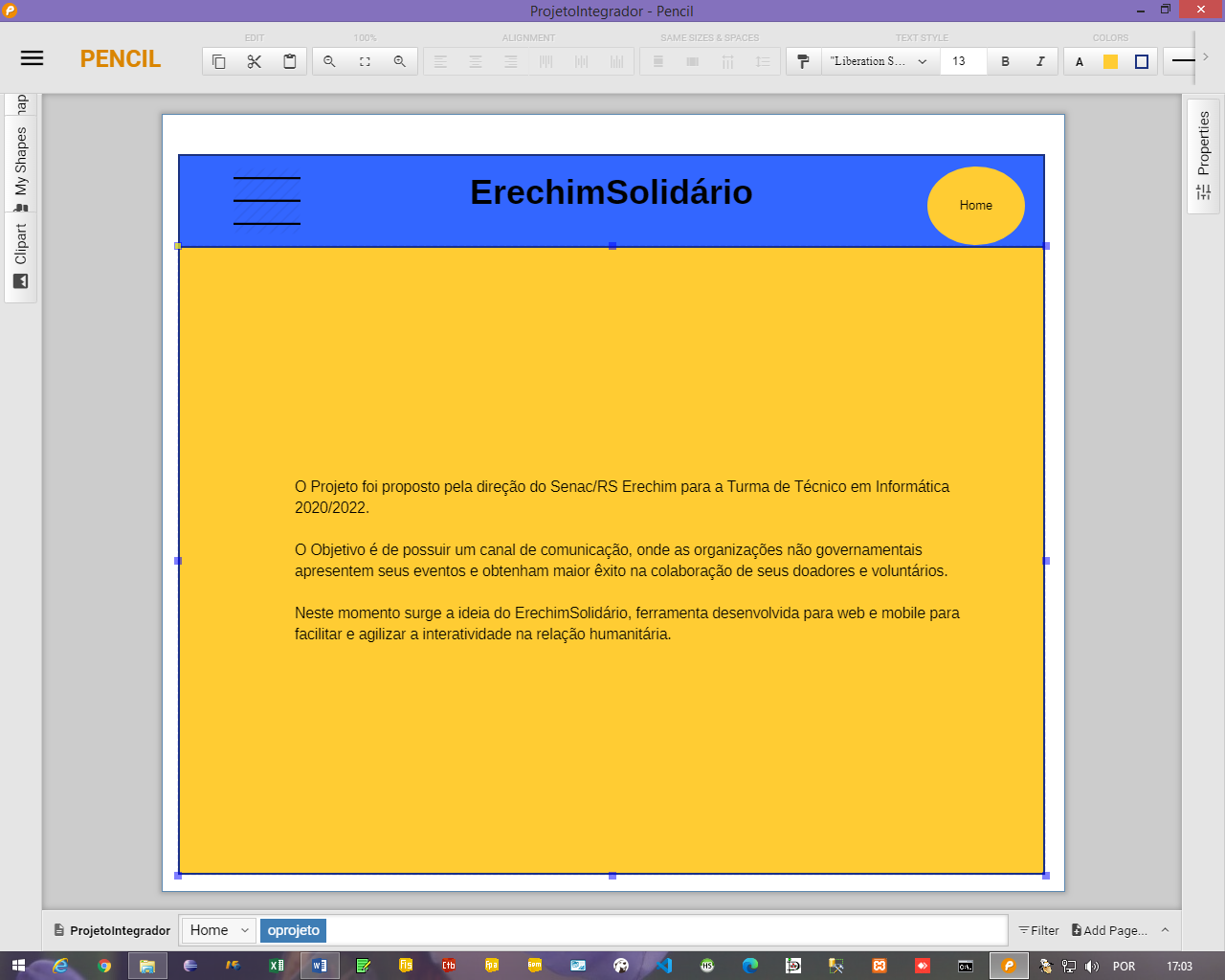
**Figura 7 - Logo Senac**



**Figura 8 - Página Inicial**

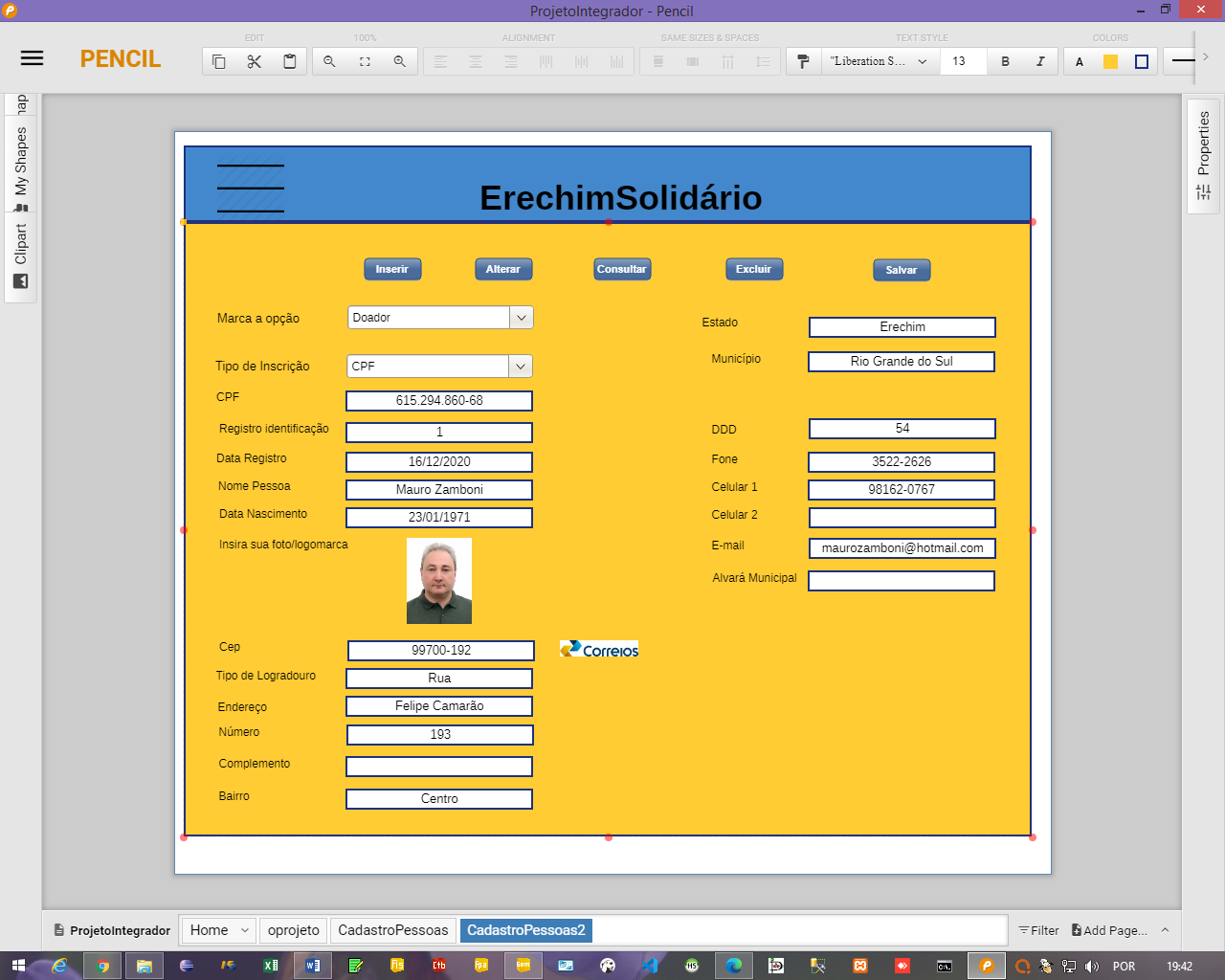
* 1. Rota Página Inicial

1. **Tela O Projeto**



**Figura 9 - Página o Projeto**

1. **Página de Cadastro**



**Figura 10 - Página Cadastro**

REFERÊNCIAS

IHRIG, Colin J. **Pro Node.js para Desenvolvedores**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2014.

Sommerville, Ian. **Engenharia de Software**. Tradução Ivan Bosnic e Kalinnka G. de O. Gonçalves; revisão técnica Kechi Hirama. – 9. ed. – São Paulo: Person Prentice Hall, 2011.

**Um Guia do Conhecimento em Gerenciamento de Projetos** (Guia PMBOK)/Project. - Sexta Edição – Newton Square, PA: Project Management Institute, 2017.

.

1. GitHub é uma plataforma de hospedagem de código-fonte e arquivos com [controle de versão](https://pt.wikipedia.org/wiki/Sistema_de_controle_de_vers%C3%B5es) usando o [Git](https://pt.wikipedia.org/wiki/Git). Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou [Open Source](https://pt.wikipedia.org/wiki/Open-source) de qualquer lugar do mundo. Disponível: <https://pt.wikipedia.org/wiki/GitHub/>. Acesso em 18/12/2020. [↑](#footnote-ref-1)
2. **Disponível: <**[Requisito – Wikipédia, a enciclopédia livre (wikipedia.org)](https://pt.wikipedia.org/wiki/Requisito)**> Acesso em 11/12/2020.**  [↑](#footnote-ref-2)
3. **UML é um acrônimo para a expressão Unified Modeling Language. Pela definição de seu nome, vemos que a UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos. Disponível: <**[Diagramas de Caso de Uso: O que é UML ? (devmedia.com.br)](https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408) **> Acesso em 11/12/2020.**  [↑](#footnote-ref-3)
4. Disponível: <[Modelagem de dados: modelo conceitual, modelo lógico e físico (luis.blog.br)](https://www.luis.blog.br/modelagem-de-dados-modelo-conceitual-modelo-logico-e-fisico.html)>. Acesso em 21/01/2021. [↑](#footnote-ref-4)
5. Disponível:<[Software Livre - Informática - InfoEscola](https://www.infoescola.com/informatica/software-livre/)> Acesso em 21/01/2021. [↑](#footnote-ref-5)
6. File Transfer Protocol (Protocolo de Transferência de Arquivos) é um protocolo padrão/genérico independente de hardware sobre um modo de transferir arquivos e também é um programa de transferência. Disponível: <https://pt.wikipedia.org/wiki/File\_Transfer\_Protocol>.Acesso em 10/12/20020. [↑](#footnote-ref-6)
7. Sistema de gerenciamento de banco de dados. Disponível: <https://pt.wikipedia.org/wiki/MySQL>.Acesso em 10/12/20020. [↑](#footnote-ref-7)
8. O Servidor HTTP Apache, é o servidor web livre. É um servidor do tipo HTTPD, compatível com o protocolo HTTP. Disponível: <https://pt.wikipedia.org/wiki/Servidor\_Apache>.Acesso em 10/12/20020. [↑](#footnote-ref-8)
9. Personal Home Page (Página Pessoal) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capaz de gerar conteúdo dinâmico na World Wide Web (WWW). [↑](#footnote-ref-9)
10. Practical Extraction and Report Language (Extração Prática e Linguagem de Relatório) é uma família de duas linguagens de programação multiplataforma. [↑](#footnote-ref-10)
11. É o software que concede liberdade ao usuário para executar, acessar e modificar o código fonte, e redistribuir cópias com ou sem modificações. Disponível: <https://pt.wikipedia.org/wiki/Software\_livre>. Acesso em 10/12/20020. [↑](#footnote-ref-11)
12. Disponível:< [Programação de Banco de Dados - parte 1 (linhadecodigo.com.br)](http://linhadecodigo.com.br/artigo/396/programacao-de-banco-de-dados-parte-1.aspx)> Acesso em 21/01/2021. [↑](#footnote-ref-12)
13. Disponível:<[MongoDB – Wikipédia, a enciclopédia livre (wikipedia.org)](https://pt.wikipedia.org/wiki/MongoDB)> Acesso em 21/01/2021. [↑](#footnote-ref-13)
14. Disponível:<<https://en.wikipedia.org/wiki/MySQL_Workbench> > Acesso em 21/01/2021. [↑](#footnote-ref-14)
15. Disponível:<[Introdução ao Visual Studio Code (devmedia.com.br)](https://www.devmedia.com.br/introducao-ao-visual-studio-code/34418)> Acesso em 21/01/2021. [↑](#footnote-ref-15)
16. Disponível:<<https://pt.wikipedia.org/wiki/Node.js>>. Acesso em 21/01/2021. [↑](#footnote-ref-16)
17. É um código fonte que é disponibilizado livremente para possível modificação e redistribuição [↑](#footnote-ref-17)
18. Diz-se multiplataforma um programa ou sistema que pode ser executado em mais do que uma plataforma, como o Mozilla Firefox, ou que executa programas ou sistemas de mais de uma plataforma, por exemplo o MAME. [↑](#footnote-ref-18)
19. Tempo de execução ou runtime, é o período em que um programa de computador permanece em execução. Disponível:<<https://pt.wikipedia.org/wiki/Tempo_de_execu%C3%A7%C3%A3>>. Acesso em 21/01/2021. [↑](#footnote-ref-19)
20. Como o nome sugere, o desenvolvedor back-end trabalha na parte de “trás” da aplicação. Ele é o responsável, em termos gerais, pela implementação da regra de negócio. Em uma aplicação web, este desenvolvedor, quando focado, não toca na parte visual da aplicação. Por lidar com a regra de negócio, as vezes um programador de sistemas, como de aplicações comerciais e até científicas, pode ser chamado de desenvolvedor back-end. Disponível:< [O que é front-end e back-end? - Blog da TreinaWeb](https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/)>. Acesso em 21/01/2021. [↑](#footnote-ref-20)
21. O desenvolvedor front-end é responsável por “dar vida” à interface. Trabalha com a parte da aplicação que interage diretamente com o usuário. Disponível:< [O que é front-end e back-end? - Blog da TreinaWeb](https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/)>. Acesso em 21/01/2021. [↑](#footnote-ref-21)
22. Disponível:<[NPM - Conhecendo os principais comandos - dfilitto](https://dfilitto.com.br/desenvolvimento/node-js/npm-conhecendo-os-principais-comandos/)>. Acesso em 25/01/2021. [↑](#footnote-ref-22)
23. Disponível: < [Express.js – Wikipédia, a enciclopédia livre (wikipedia.org)](https://pt.wikipedia.org/wiki/Express.js)>. Acesso em 25/01/2021. [↑](#footnote-ref-23)
24. Disponível:<<https://www.devmedia.com.br/conhece-o-electron/38040>>. Acesso em 21/01/2021. [↑](#footnote-ref-24)
25. Disponível:< [Node.js ou Nodemon? | Underground WebDev (udgwebdev.github.io)](https://udgwebdev.github.io/nodejs-ou-nodemon/)>. Acesso em 26/01/2021. [↑](#footnote-ref-25)
26. Disponível:< <https://pt.wikipedia.org/wiki/HTML>>. Acesso em 13/12/2020. [↑](#footnote-ref-26)
27. Disponível:< <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>>. Acesso em 13/12/2020. [↑](#footnote-ref-27)
28. Disponível:<https://getbootstrap.com/docs/5.0/about/overview/>. Acesso em 18/12/2020. [↑](#footnote-ref-28)
29. Disponível:< [Configurando o ORM Sequelize no NodeJS com ExpressJS (rocketseat.com.br)](https://blog.rocketseat.com.br/nodejs-express-sequelize/)>. Acesso em 25/01/2021. [↑](#footnote-ref-29)
30. Disponível:<https://github.com/ericdouglas/traduz-ai/blob/master/handlebars/001-aprenda-tudo-sobre-handlebars.md>. Acesso em 26/01/2021. [↑](#footnote-ref-30)
31. Disponível:< <https://pt.wikipedia.org/wiki/JQuery>>. Acesso em 26/01/2021. [↑](#footnote-ref-31)
32. Disponível:<<https://www.npmjs.com/package/popper.js/v/1.14.3>>. Acesso em 26/01/2021. [↑](#footnote-ref-32)
33. Disponível:<<https://www.luiztools.com.br/post/o-guia-completo-do-package-json-do-node-js/?gclid=CjwKCAiAu8SABhAxEiwAsodSZN_X9PnKnVXecdJbH9fKzLCBM6za6948X-cXynNog8SnXKm1MqStDRoCR04QAvD_BwE>>. Acesso em 25/01/2021. [↑](#footnote-ref-33)
34. Disponível:< [Roteamento básico no Express (expressjs.com)](http://expressjs.com/pt-br/starter/basic-routing.html)>. Acesso em 25/01/2021. [↑](#footnote-ref-34)
35. Disponível: <https://www.luiztools.com.br/post/programacao-assincrona-em-nodejs-callbacks-e-promises/>>. Acesso em 27/01/2021. [↑](#footnote-ref-35)
36. Disponível:<[O novo logotipo do Senac e a responsabilidade do ensino universitário (b9.com.br)](https://www.b9.com.br/31240/o-novo-logotipo-do-senac-e-a-responsabilidade-do-ensino-universitario/)>. Acesso em 18/12/2020. [↑](#footnote-ref-36)