



SENAI Jaguariúna

Carlos Roberto de Oliveira Júnior

Danilo Alves Falciroli

Matheus Belarmino Pignata

PROJETO MASTER

Carlos Roberto de Oliveira Júnior

Danilo Alves Falciroli

Matheus Belarmino Pignata

PROJETO MASTER

Projeto de conclusão parcial de curso apresentado ao SENAI Jaguariúna, como parte dos requisitos para obtenção do título de Técnico em Desenvolvimento de Sistemas

Orientador:

Tatiana Maria Rolin.

Jaguariúna – SP

Março – 2022

Lista de Figuras

Figura 1 - DCU - [RF001].....	12
Figura 2 - DCU - [RF002].....	13
Figura 3 - DCU - [RF003].....	13
Figura 4 - DCU - [RF004].....	13
Figura 5 - Tela de login.....	12
Figura 6 - Tela de registro.....	12
Figura 7 - Tela de agendamento.....	12
Figura 8 - Tela de gerenciamento de funcionários.....	12
Figura 9 - Tela de redefinição de senha.....	12
Figura 10 - Login App.....	12
Figura 11 - Registro App.....	12
Figura 12 - Usuário App.....	12
Figura 13 - Gerenciamento de funcionários App.....	12
Figura 14 - Redefinição de senha App.....	12
Figura 15 - Modelo conceitual.....	12
Figura 16 - Modelo lógico.....	12
Figura 17 - Diagrama de classes.....	12

Lista de Abreviaturas e Siglas

API	- Application Programming Interface (Interface de programação de aplicação)
BackEnd	- Prática de programação para Servidor de Aplicação e ou Banco de Dados.
FrontEnd	- Prática de programação para cliente de programas para internet.
FullStack	- Prática de programação em todas as camadas Cliente e Servidor
Mobile	- Prática de programação para dispositivos móveis também clientes de programas para a internet.
PMO	- Project Management Office (Escritório de Projetos).
JavaScript	- Linguagem de programação para a internet.
HTML	- Hyper Text Management Language, linguagem de marcação.
CSS	- Linguagem de marcação e estilização de páginas.
Framework	- Estrutura, conjunto de códigos genéricos capaz de unir trechos de um projeto.
Node.js	- Framework que utiliza a linguagem JavaScript no backend.
ORM	- Object Relational Mapper (Mapa Objeto Relacional)
Sequelize	- recurso ORM do framework Node.js
UML	- Unified Modeling Language (Linguagem de Modelagem Unificada)
Ágil	- Metodologia Ágil (Metodologia de desenvolvimento de sistemas)
SCRUM	- Subcategoria da Metodologia Ágil. Backlog - Requisitos funcionais do projeto a serem desenvolvidos.
Sprint	- O tempo de desenvolvimento varia de duas a quatro semanas.
Gantt	- Gráfico para gestão do tempo e recursos do projeto.
MER	- Modelo Entidade e Relacionamento
DER	- Diagrama de Entidade e Relacionamento
MVC	- Modelo Visão e Controle
JSON	- Objeto JavaScript, padrão de dados de uma API
GitHub	- Repositório para códigos fonte e rede social de programadores.

Resumo

O projeto master é um projeto de internet fullstack com banco de dados relacional MySQL, Backend em Node.js, Front-end em JavaScript sem framework e aplicativo móvel desenvolvido com React Native, consumindo a API do Google Maps.

O objetivo é: fazer com que o diretor da unidade 513 do SENAI tenha ganhos em performance na hora da criação da planilha de aulas de cada semestre, fazendo com que tenha mais tempo para resolver outros problemas que possa ter durante o dia, pois tem um excessivo gasto de tempo criando as planilhas e pensando nos horários por conta própria.

Abstract

The master project is a fullstack internet project with MySQL relational database, Backend in Node.js, Front-end in JavaScript without framework and mobile application developed with React Native, consuming the Google Maps API.

The objective is: to make the director of unit 513 of SENAI have performance gains when creating the class spreadsheet for each semester, giving him more time to solve other problems he may have during the day, as he has a excessive time spent creating spreadsheets and thinking about schedules on your own.

Sumário

1. TAP (Termo de Abertura do Projeto).....	12
2. EAP (Escopo do Projeto).....	13
2.1. Requisitos não funcionais.....	13
2.2. Requisitos funcionais.....	13
3. Planejamento de custos.....	19
4. Protótipo.....	20
4.1. Protótipo do site da internet.....	20
4.2. Protótipo do Aplicativo Móvel.....	23
5. Planejamento do Banco de Dados e da API.....	28
6. Detalhe da Execução e Controle.....	30
7. Resultados.....	36
8. Bibliografia.....	36

Introdução

Nós intitulados com o(a) turma de 3DES do curso técnico em desenvolvimento de sistemas da escola SENAI Jaguariúna com conclusão do curso prevista para o mês de agosto do ano de 2022, temos nos dedicado a solução de problemas através de sistemas computacionais, também a automação de processos repetitivos utilizando recursos e bibliotecas das linguagens de programação como Python, planilhas eletrônicas e APIs disponíveis de forma gratuita através da internet.

APIs segundo o site HostGator é a interface ideal para que os programas de diferentes organizações compartilhem suas ações, ferramentas, padrões e protocolos. Gerando, assim, uma total integração.

Neste contexto, a fim de adquirirmos experiência na hora de realizar o projeto de conclusão de curso, foi realizado esse pré projeto desenvolvido por nós contando com a colaboração dos instrutores.

O projeto Masters é um projeto com intuito de gerenciar dados que sejam úteis para o coordenador geral da unidade 513 do SENAI, além de ser uma demonstração de WebDesign, Web Programming e Mobile Application.

No banco de dados que faz uso da tecnologia "MySQL" utilizando o banco de dados "MariaDB" que é "SQL Server" relacional, onde fica as informações contidas para uso dentro dos aplicativos desenvolvidos por nossa equipe.

Para o BackEnd é feito uso da tecnologia "Node JS", uma framework que contém as ferramentas utilizadas para o desenvolvimento deste projeto, aqui uma prévia de cada uma das ferramentas que serão usadas durante o desenvolvimento: "Sequelize" é um "SGBD"(Sistema Gerenciador de Banco de Dados) que fica entre o Site ou Aplicação Móvel e o banco de dados fazendo cadastros, alterações, inserções e remoção dos dados dentro do banco de dados, "MySQL 2" é um sub-extensão do "Sequelize" que faz o gerenciamento de cada ação do sequelize, que seriam as ações do "CRUD" e a comunicação entre o banco de dados e o BackEnd por parte do "Sequelize", "Express" é uma ferramenta desenvolvida para fazer a conexão entre a Aplicação Móvel ou Site ao BackEnd através da rede, fazendo envio de dados e comunicação entre o servidor e o cliente, "BodyParser" extensão do "Express" para aumentar ou diminuir o limite de tráfego de dados entre o cliente e o servidor BackEnd.

Para a Aplicação Móvel foi usada uma ferramenta fornecida pelo "Node JS" conhecida como "React-Native", permitindo que o aplicativo seja utilizado tanto em Android

quanto IOS e tenha uma proximidade maior na hora de fazer o upload para alguma loja de aplicativos online, pois as lojas de aplicações móveis que temos atualmente possuem aplicativos com maior taxa de recomendação quando feito usando essa ferramenta por ser mais próxima das empresas que fornecem essas lojas para vendas, caso seja solicitado pelo cliente.

Para o Site foi utilizado as linguagens de programação "HTML", "CSS" e "JavaScript", tudo para dar uma "UX" User Experience de qualidade para as pessoas que forem fazer uso de nosso aplicativo.

Como forma de simplificar a lógica de programação foi adotada a linguagem JavaScript em todas as camadas do projeto, o framework utilizado no servidor de aplicação backend foi o Node.js. No frontend optou-se por não utilizar nenhum framework, mas sim recursos básicos das linguagens HTML, CSS e JavaScript.

Por fim, para a programação de dispositivos móveis este modelo utilizou o framework React Native que também trabalhou sobre a linguagem JavaScript.

Objetivos

Desenvolver um aplicativo que gerencie alertas de trânsito, onde os condutores de veículos possam apontar no mapa eventuais acidentes, congestionamentos ou outros eventos, assim alertando outros usuários do aplicativo.

Este projeto cumpre os seguintes objetivos específicos:

- Facilitar o gerenciamento de aulas semestrais
- Servir de modelo para o desenvolvimento de soluções semelhantes.
- Servir de exemplo de consumo de API de terceiros
- Servir de exemplo de armazenamento de imagens em banco de dados
- Servir de exemplo de utilização de mapas via internet frontend e mobile.
- Servir de exemplo de utilização de recursos ORM como “sequelize”.

Justificativa

Ao concluir o curso técnico de desenvolvimento de sistemas o aluno adquire várias competências, o desenvolvimento de um projeto de conclusão auxilia a demonstrar estas capacidades técnicas na prática, serve como forma de avaliação somativa dentro das práticas da metodologia de ensino profissionalizante adotada pelo SENAI além de gerar a oportunidade de apresentar este projeto ao corpo docente da instituição e até a membros da gestão das indústrias da cidade de Jaguariúna e região.

1. TAP (Termo de Abertura do Projeto)

Título do Projeto

Projeto Masters "Site e Aplicativo Móvel que faz o gerenciamento dos funcionários e calendário de aulas dentro da unidade 513 do SENAI.

Patrocinadores: SENAI Jaguariúna

Recursos Humanos	Carlos Roberto de Oliveira Júnior Matheus Belarmino Pignata
Gerente do Projeto	Danilo Alves Falcirolli
Patrocinador	SENAI Jaguariúna
Cliente	SENAI Jaguariúna
Prazo	31/03/2022

Local: _____ Data: ____ / ____ / ____	
Patrocinador	Ass: _____
Cliente	Ass: _____
Gerente do Projeto	Ass: _____

2. EAP (Escopo do Projeto).

O escopo de um projeto de aplicação fullstack, para a internet e para dispositivos móveis consistem de seus requisitos funcionais e não funcionais, este projeto, portanto está dividido desta forma.

2.1. Requisitos não funcionais.

[RN001] Linguagens de Programação:

[RN001.1] BackEnd: Javascript com o framework Node.js

[RN001.2] BackEnd: biblioteca ORM sequelise.

[RN001.3] FrontEnd: HTML, CSS e JavaScript sem framework.

[RN001.4] Mobile: Javascript com o framework React Native.

[RN002] FrontEnd e Mobile: Gerenciamento SENAI.

[RN003] Servidor é possível hospedagem.

[RN003.1] HEROKU <https://www.heroku.com/> ou algum serviço de nuvem.

[RN004] Recursos de Banco de Dados relacional e SQL.

[RN004.1] MySQL 10.4.11-MariaDB.

2.2. Requisitos Funcionais.

Os requisitos funcionais serão apresentados utilizando o recurso UML Diagrama de Casos de Uso e uma classificação de criticidade.

A metodologia de desenvolvimento de sistemas aplicada neste projeto é conhecida como SCRUM que é uma subcategoria da metodologia Ágil, por este motivo cada requisito aqui apresentado faz parte do backlog do projeto que será dividido em duas Sprints conforme a sua criticidade.

A criticidade está dividida em três categorias: essencial, importante e desejável definindo o nível de prioridade para cada requisito acordado neste escopo.

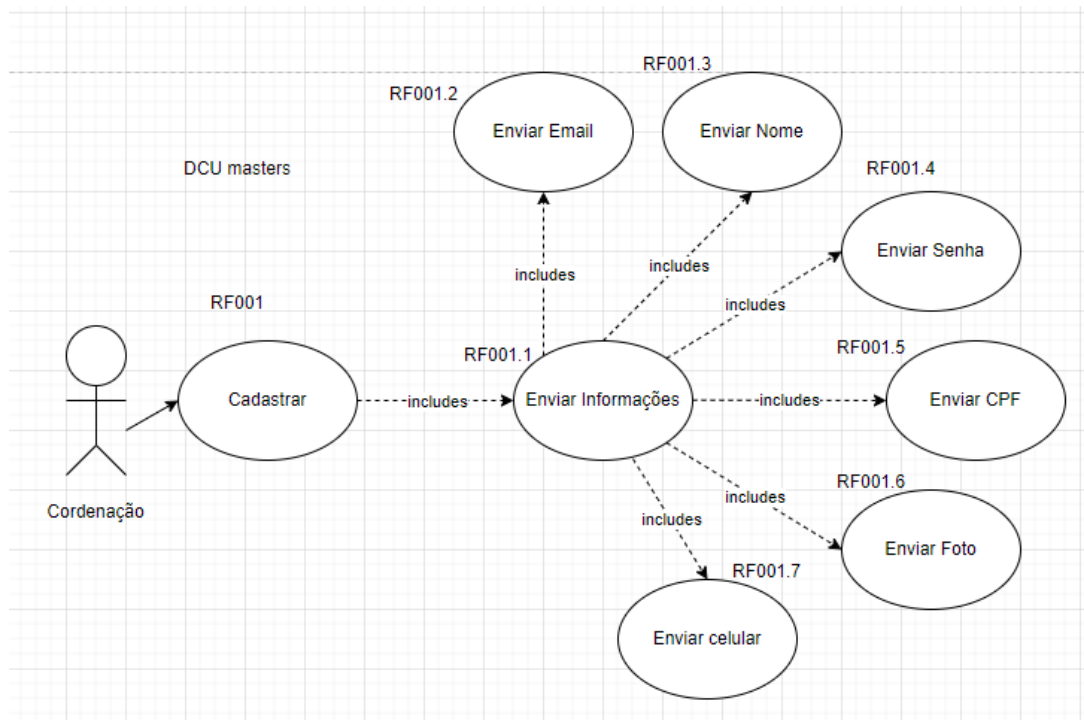


Figura 1 - DCU - [RF001]

[RF001]: Cadastrar

Tela do sistema para cadastrar-se.

Criticidade: (x) essencial () importante () desejável

[RF001.2]: Informar email.

Preencher email.

Criticidade: (x) essencial () importante () desejável

[RF001.3]: Informar nome.

Preencher nome.

Criticidade: (x) essencial () importante () desejável

[RF001.4]: Informar senha.

Preencher senha.

Criticidade: (x) essencial () importante () desejável

[RF001.5]: Informar CPF.

Preencher CPF.

Criticidade: (x) essencial () importante () desejável

[RF001.6]: Anexar uma foto.

Anexar uma imagem/foto de perfil.

Criticidade: () essencial () importante (x) desejável

[RF001.7]: Informar celular.

Preencher celular.

Criticidade: (x) essencial () importante () desejável

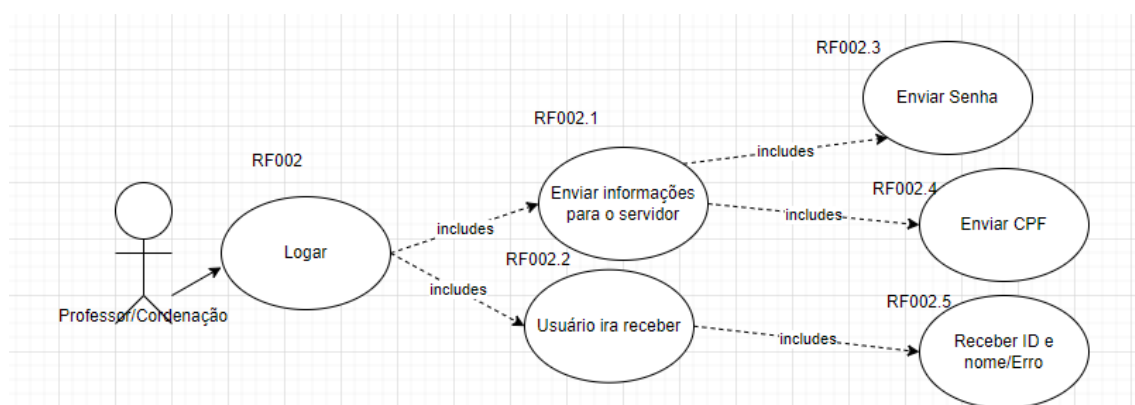


Figura 2 - DCU - [RF002]

[RF002]: Fazer Login.

Tela inicial do sistema para acessar ou cadastrar-se.

Criticidade: (x) essencial () importante () desejável

[RF002.1]: Enviar dados para o servidor.

Informar CPF e senha para acessar o sistema.

Criticidade: (x) essencial () importante () desejável

[RF002.2]: Receber dados do servidor.

O sistema irá receber informações do usuário.

Criticidade: (x) essencial () importante () desejável

[RF002.3]: Informar senha.

Informar senha.

Criticidade: (x) essencial () importante () desejável

[RF002.4]: Informar CPF.

Informar CPF.

Criticidade: (x) essencial () importante () desejável

[RF002.5]: Receber informações do servidor.

O sistema irá receber a resposta do servidor e tratá-las de acordo com a chegada de dados e/ou erros.

Criticidade: (x) essencial () importante () desejável

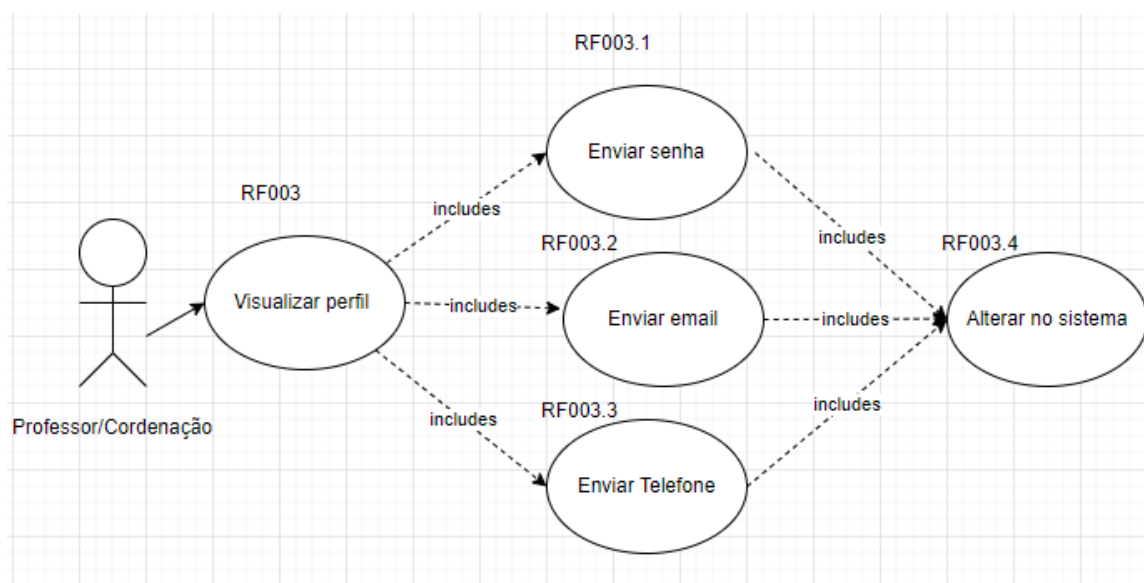


Figura 3 - DCU - [RF003]

[RF003]: Visualizar informações do usuário.

Tela onde o usuário poderá ver suas informações.

Criticidade: (x) essencial () importante () desejável

[RF003.1]: Enviar senha.

Informar senha para alteração.

Criticidade: (x) essencial () importante () desejável

[RF003.2]: Enviar email.

Informar email para alteração.

Criticidade: (x) essencial () importante () desejável

[RF003.3] Enviar telefone.

Informar telefone para alteração.

Criticidade: (x) essencial () importante () desejável

[RF003.4]: Alterar informações.

Ao clicar no botão de alterar informações, o usuário registra suas informações atualizadas.

Criticidade: () essencial (x) importante () desejável

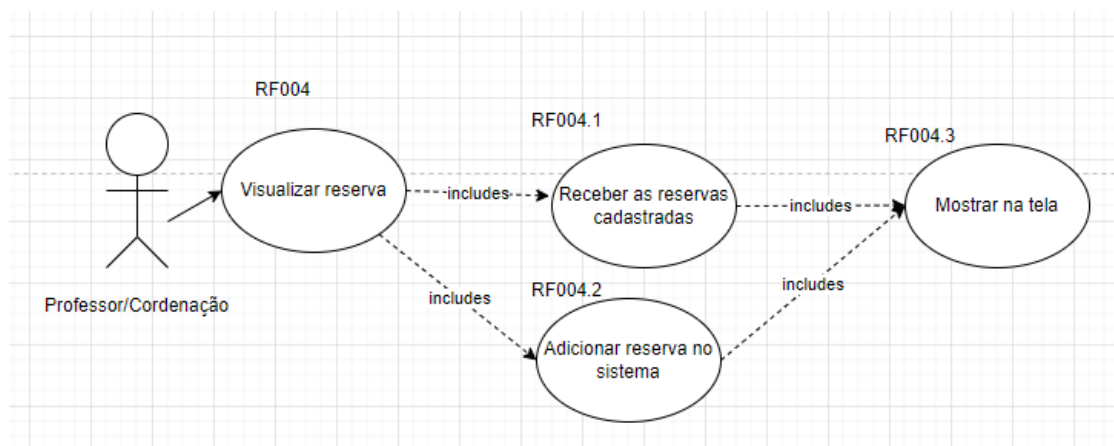


Figura 4 - DCU - [RF004]

[RF004]: Visualizar reserva realizada.

Tela onde o usuário pode visualizar as reservas feitas por outras pessoas, e realizar sua própria reserva.

Criticidade: (x) essencial () importante () desejável

[RF004.1]: Receber do servidor as reservas cadastradas.

Resposta do servidor onde todas as reservas são carregadas para o usuário.

Criticidade: (x) essencial () importante () desejável

[RF004.2]: Registrar uma nova reserva.

Ao clicar em um botão intuitivo, o usuário registra uma nova reserva.

Criticidade: () essencial (x) importante () desejável

3. Planejamento de custos.

Para calcular os custos deste projeto, basta somar as horas dedicadas, e multiplicar por um valor estipulado por hora, sabendo que foram gastas quatro horas por dia útil, do dia 13/02/2022, ao dia 31/03/2022, totalizando 33 dias úteis, temos 132 horas investidas nesse projeto. A seguir temos um exemplo de como seria calculado o custo.

Carlos - 44 horas trabalhadas.

Danilo - 44 horas trabalhadas.

Matheus - 44 horas trabalhadas.

Totalizando 132 horas de projeto e partindo do princípio que cada recurso custe R\$10,00 por hora, o custo FINAL deste projeto seria de: R\$ 1320,00.

4. Protótipo.

As imagens a seguir foram geradas utilizando a técnica de prototipagem para apresentar um esboço das telas do site da internet e do aplicativo.

4.1. Protótipo do site da internet.

As imagens mostram a sequência de telas que um usuário do site deve acessar para usufruir de todas as funcionalidades da aplicação.

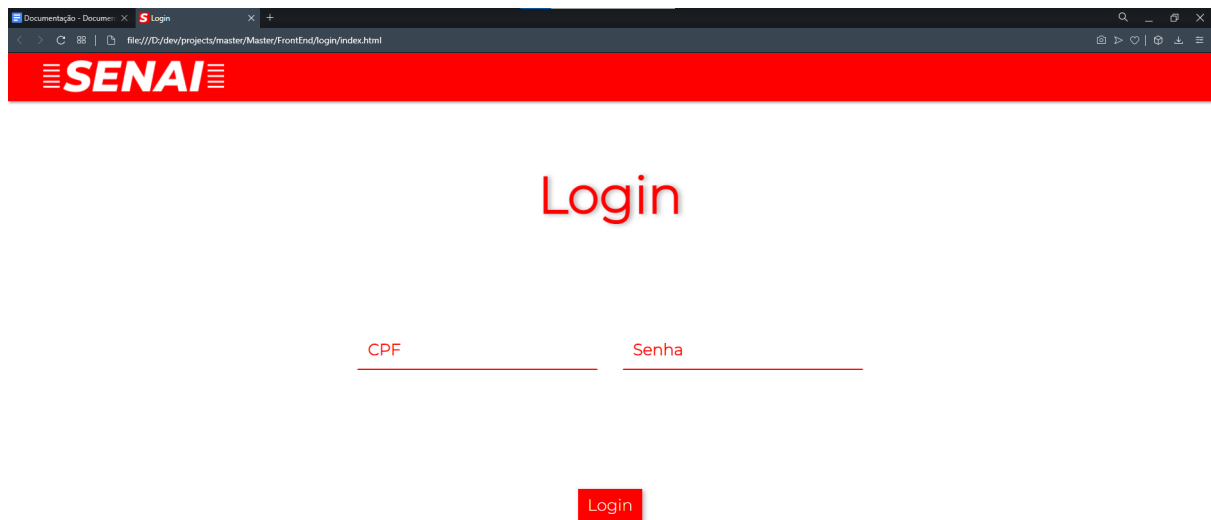


Figura 5 - Tela de login

Documentação - Documen... x S Registro x +

file:///D:/dev/projects/master/Master/FrontEnd/registro/index.html

SENAI

Registro

Nome Completo

Telefone

CPF

Senha

E-mail

Confirme a senha

Cargo

Seleccione a foto

Registrar

Voltar

Figura 6 - Tela de registro

Documentação - Documen... x S Login x +

file:///D:/dev/projects/master/Master/FrontEnd/agenda/index.html

SENAI

Curso

Sala

Data Inicial

Data Final

☐ Carlos

☐ Danilo

Gerar arquivo CSV

Figura 7 - Tela de agendamento



Figura 8 - Tela de gerenciamento de funcionários.

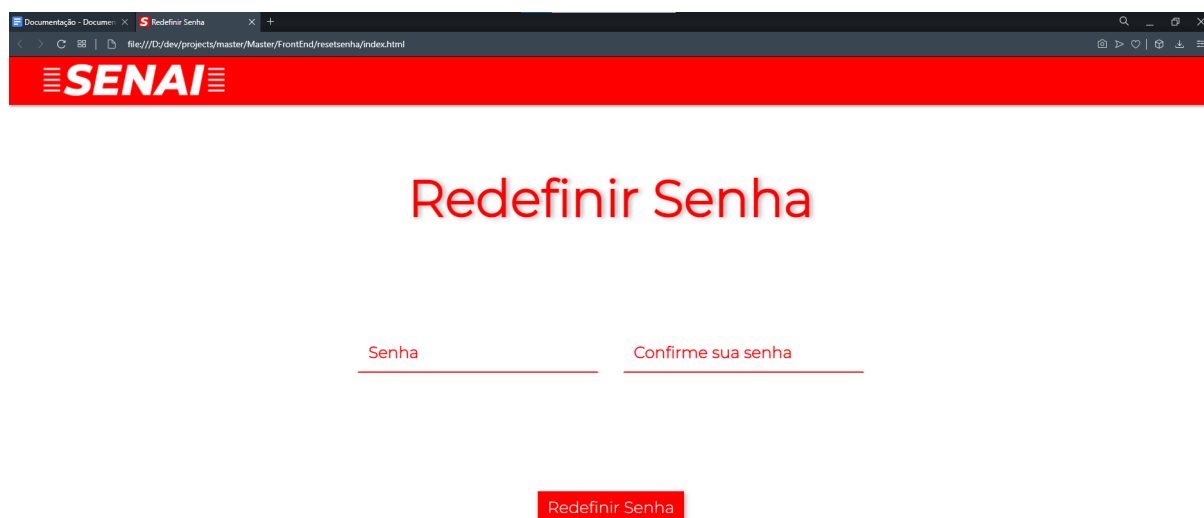


Figura 9 - Tela de redefinição de senha.

4.2. Protótipo do Aplicativo Móvel.

As imagens mostram a sequência de telas que um condutor deve acessar para usufruir de todas as funcionalidades do aplicativo móvel.



Login

CPF

Senha

Logar

Figura 10 - Login App.

Registro

502.713.148-85

Danilo Alves Falcirolli

falcirolli5531@gmail.com

(19) 98338-6189

.....

.....

Diretor de Unidade de Formação Pro... ▼

Escolha uma foto


Usuario


Registro


Usuarios

Figura 11 - Registro App.



Danilo Alves Falciroli

CPF

502.713.148-85

Alterar Informações

danilo.falciroli5531@gmc

(19) 98338-6189

Alterar informações

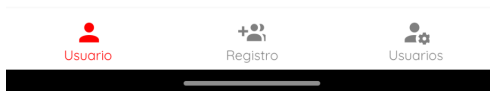


Figura 12 - Usuário App.

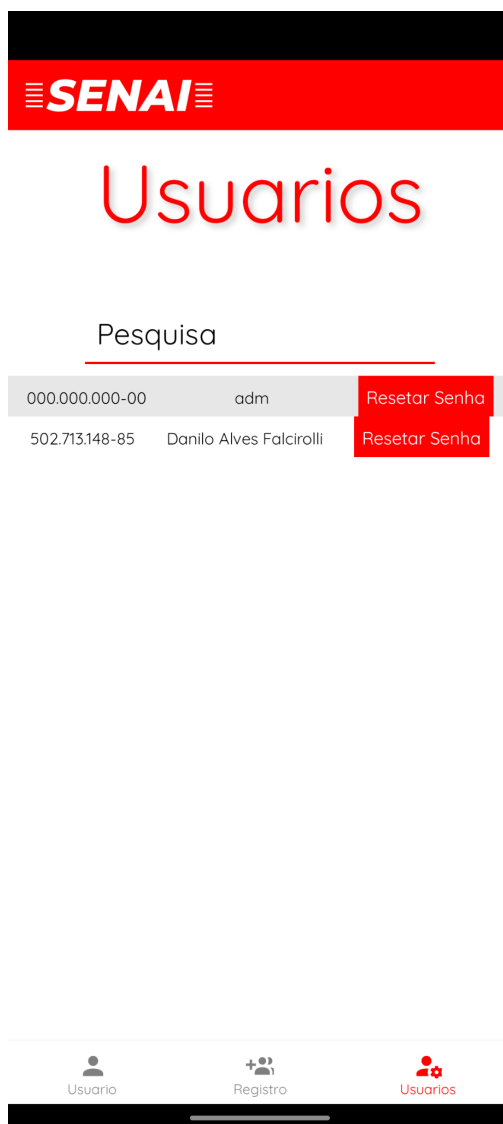


Figura 13 - Gerenciamento de funcionários App.



Resetar Senha

Senha

Conf. Senha

Salvar

Figura 14 - Redefinição de senha App.

5. Planejamento do Banco de Dados e da API.

O backend deste projeto se comporta como uma API, fornecendo dados tanto para o frontend quanto para o aplicativo móvel, esta seção documenta o projeto do banco de dados através do MER (Modelo Entidade e Relacionamento) e DER (Diagrama de Entidade e Relacionamento) conceitual e lógico.

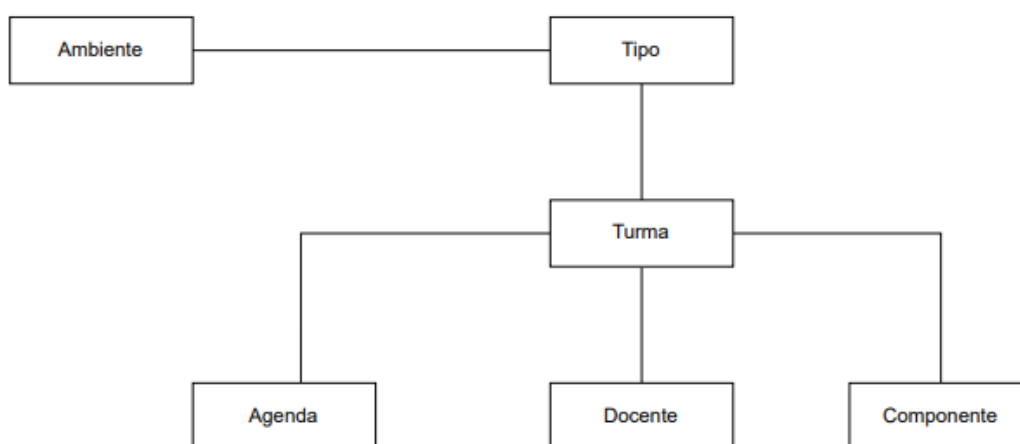


Figura 15 -Modelo conceitual.

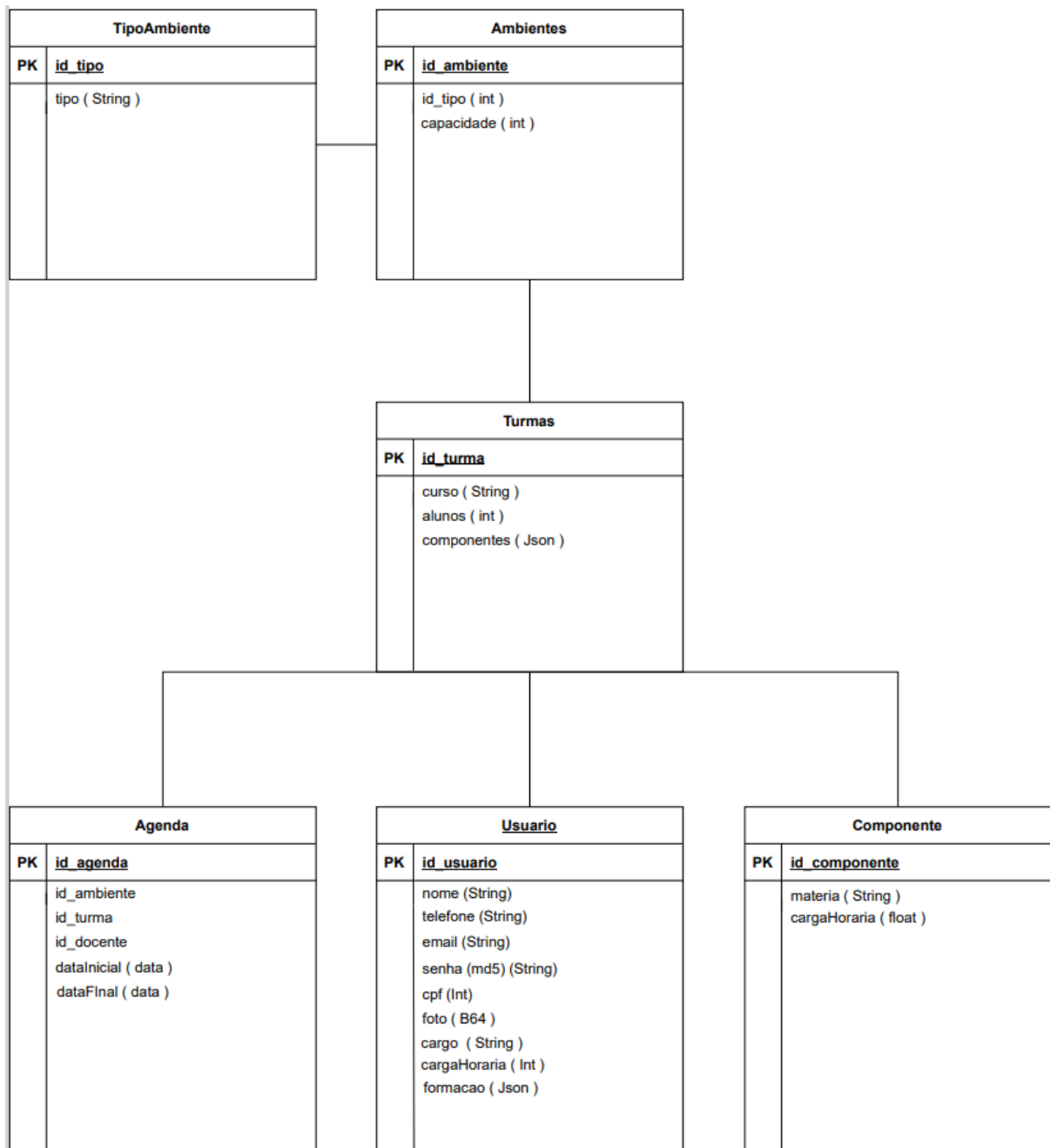


Figura 16 -Modelo Lógico.

O diagrama de classes a seguir apresenta o projeto das classes da camada modelo do padrão de projetos MVC (Modelo Visão e Controle) do backend ou API.

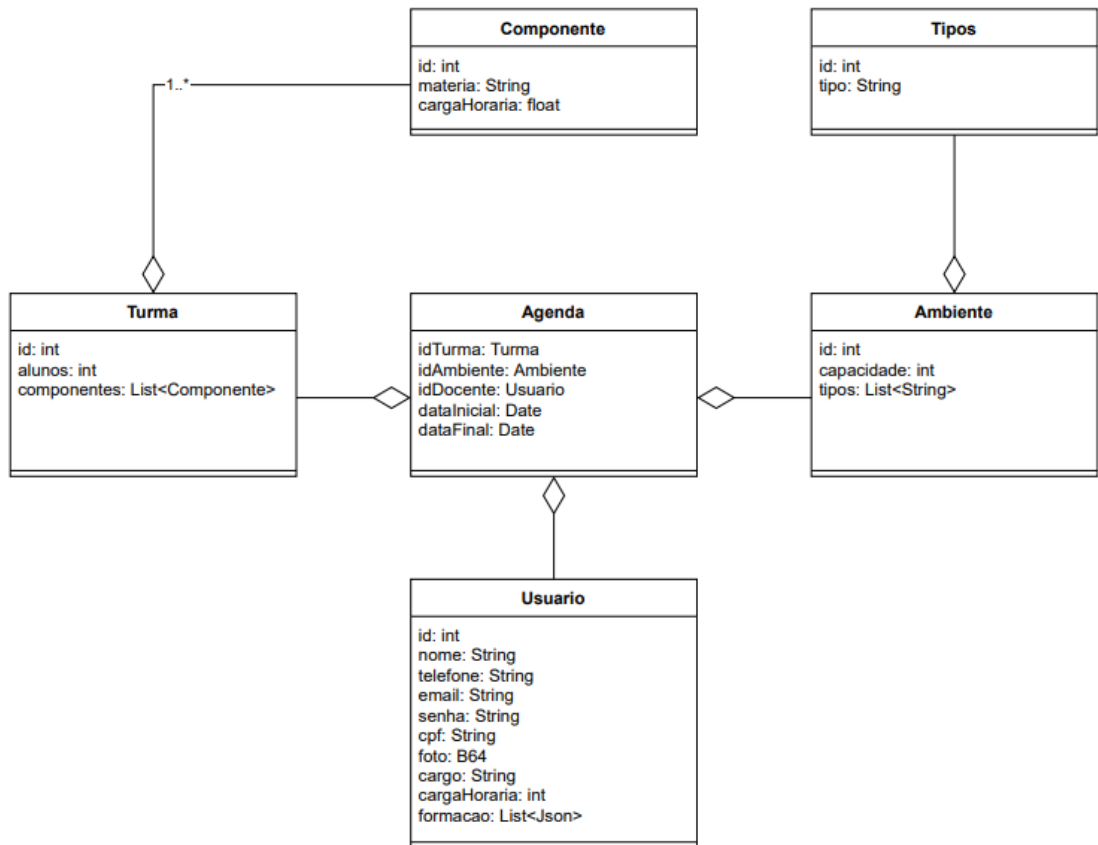


Figura 17 - Diagrama de classes.

6. Detalhes da Execução e Controle.

A seguir segue o dicionário de dados no formato JSON que é o padrão de dados oficial de troca de informações entre APIs.

//Enviado pelo FrontEnd (Pagina login)

```
{  
  "cpf": "24516263",  
  "senha": "senha"  
}
```

//Receberá de volta Front End (Pagina login)

```
{  
  "cpf": "352.353.255-31",  
  "nome": "João da silva",  
  "email": "josesilva@gmail.com",  
  "telefone": "(19) 12516-2632",  
  "cargo": "Professor",  
  "foto": "base64",  
  "resetsenha": false  
}
```

// Enviado pelo FrontEnd (Página do usuário) Update de informações

```
{  
  "id": 1,  
  "telefone": "(15) 21285-2528",  
  "email": "josesilva267@gmail.com",  
}
```

// Receberá de volta no FrontEnd (Página do usuário) Update de informações

```
{  
  "id": 1,  
  "telefone": "(15) 21285-2528",  
  "email": "josesilva267@gmail.com",  
}
```

```
}
```

```
// Enviado pelo FrontEnd ( Página de registro )
```

```
{
```

```
  "cpf": "352.353.255-31",  
  "nome": "José da silva",  
  "email": "josesilva@gmail.com",  
  "telefone": "(19) 12516-2632",  
  "cargo": "Professor",  
  "foto": "base64",  
  "senha": "37fgu58fgf4ss8",  
  "cargaHoraria": 20,  
  "formacao": "TI em geral"
```

```
}
```

```
// Receberá de volta no FrontEnd ( Página de registro )
```

```
{
```

```
  "status": 200 || { "msg": "registrado" }
```

```
}
```

```
// Envia pelo FrontEnd ( Página de funcionários )
```

```
{
```

```
  link do back
```

```
}
```

```
//Receberá de volta no FrontEnd ( Página de funcionários )
```

```
{
```

```
  "id": 1,  
  "cpf": "352.353.255-31",  
  "nome": "José da silva",
```

```
}
```

```
// Enviara pelo FrontEnd ( Página de funcionários )
```

```
{
```

```
  "id": 1,  
  "senha": "123456789",
```



```
}  
//Receberá de volta no FrontEnd ( Página de funcionários )  
{  
    "status": 200 || { "msg": "alterado"}  
}  
  
// Enviara pelo FrontEnd ( Página de Cadastro Geral ) Componente GET  
{  
  
}  
//Receberá de volta no FrontEnd ( Página de Cadastro Geral ) Componente GET  
{  
    id: 1,  
    materia: "FPOO",  
    carga_horaria: 150  
}  
// Enviara pelo FrontEnd ( Página de Cadastro Geral ) Componente POST  
{  
    materia: "SOP",  
    carga_horaria: 75  
}  
//Receberá de volta no FrontEnd ( Página de Cadastro Geral ) Componente POST  
{  
    id: 3,  
    materia: "SOP",  
    carga_horaria: 75  
}  
// Enviara pelo FrontEnd ( Página de Cadastro Geral ) Componente PUT  
{  
    Query: "&id=1",  
    materia: "FPOO",  
    carga_horaria: 300  
}
```

```
//Receberá de volta no FrontEnd ( Página de Cadastro Geral ) Componente PUT
{
    id: 1,
    materia: "FPOO",
    carga_horaria: 300
}

// Enviara pelo FrontEnd ( Página de Cadastro Geral ) Componente Delete
{
    Query: "&id=1",
}

//Receberá de volta no FrontEnd ( Página de Cadastro Geral ) Componente Delete
{
    HTTPStatus: 410
}

// Enviara pelo FrontEnd ( Página de Cadastro Geral ) Ambiente GET
{

}

//Receberá de volta no FrontEnd ( Página de Cadastro Geral ) Ambiente GET
{
    id: 1,
    capacidade: 30,
    tipoAmbiente: {
        id: 1,
        tipo: "Laboratorio"
    }
}

// Enviara pelo FrontEnd ( Página de Cadastro Geral ) Ambiente POST
{
    id_tipo : 1,
    capacidade : 30
}
```

//Receberá de volta no FrontEnd (Página de Cadastro Geral) Ambiente POST

```
{  
    id: 1,  
    id_tipo: 1,  
    capacidade: 30  
}
```

// Enviara pelo FrontEnd (Página de Cadastro Geral) Ambiente PUT

```
{  
    Query: "&id=1",  
    id_tipo: 2,  
    capacidade: 20  
}
```

//Receberá de volta no FrontEnd (Página de Cadastro Geral) Ambiente PUT

```
{  
    id: 1,  
    id_tipo: 2,  
    capacidade: 20  
}
```

// Enviara pelo FrontEnd (Página de Cadastro Geral) Ambiente Delete

```
{  
    Query: "&id=1",  
}
```

//Receberá de volta no FrontEnd (Página de Cadastro Geral) Ambiente Delete

```
{  
    HTTPStatus: 410  
}
```

7. Resultados.

O projeto será apresentado em aula para os alunos do primeiro e terceiro módulo do curso de Desenvolvimento de Sistemas no primeiro semestre do ano de 2022, para alguns membros do corpo docente e servirá de modelo para futuras turmas.

A execução do sistema não cumpriu todos os requisitos classificados como essenciais e importantes deste projeto.

Os arquivos fonte deste projeto estão compartilhados no GitHub através do endereço de internet a seguir: <https://github.com/daniloafalciori/lli/Master> este é um repositório público para consulta dos alunos e professores desta instituição.

Este projeto não foi hospedado em nenhum serviço privado, público ou de nuvem, apenas na rede local da sala de aula desta instituição de ensino.

8. Biografia.

BIBLIOGRAFIA Brasil: HostGator, Disponível em: [Link](#), acessado em 31/03/2022.

Brasil: HostGator, Disponível em: [Link](#), acessado em 31/03/2022.

Brasil: Le Wagon, Disponível em: [Link](#), acessado em 31/03/2022.

PMBOK Guide, 6a. Edição, 2017. 762 p.

Brasil: DEVMEDIA, Disponível em: [Link](#) , acessado em 31/06/2022