



UNifeob
| ESCOLA DE NEGÓCIOS



2023

**PROJETO DE CONSULTORIA
EMPRESARIAL**



UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

PROJETO INTEGRADO

NOME DO PROJETO

SÃO JOÃO DA BOA VISTA, SP

JUNHO 2023

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

PROJETO INTEGRADO

NOME DO PROJETO

MÓDULO DESENVOLVIMENTO DESKTOP

Ferramentas de Desenvolvimento Web – Prof. Nivaldo de Andrade

Sistemas Operacionais – Prof. Rodrigo Marudi de Oliveira

Projeto de Desenvolvimento Web – Prof. Nivaldo de Andrade

Estudantes:

André Luiz de Melo Filho, RA 1012022101042

Lucas Rafael C. da Silva, RA 1012023100143

Oziniel Ewerton Silva, RA 1012022101458

Predro Riquelme Borsone, RA 1012022100206

SÃO JOÃO DA BOA VISTA, SP

JUNHO, 2023

1. INTRODUÇÃO

Este projeto tem como objetivo o desenvolvimento de um serviço web para gerenciamento de agenda de serviços prestados, com a finalidade de um cadastramento de clientes interessados em serviços prestados pela empresa.

2. DESCRIÇÃO DA EMPRESA

Empresa localizada na região de Mogi Guaçu-SP no ramo de prestação de serviços de manutenção corretiva e preventiva de veículos. "Auto Mecânica SMA" está localizada na Rua Rodolpho Innarelli N° 228, Jd. Herminio Bueno - Mogi Guaçu-SP, há mais de 10 anos atendendo clientes da região.

3. PROJETO DE CONSULTORIA EMPRESARIAL

Após uma entrevista com o proprietário foi identificado a necessidade de um sistema que os clientes pudessem realizar um cadastro de interesse de agendamento de serviço.

Com a proposta de desenvolvimento de uma aplicação que o próprio cliente pudesse preencher um formulário com os dados para que seja feita uma análise previa do que será realizado e assim a melhor data para agendamento do serviço, concluímos que um sistema web seria a melhor maneira de atender as necessidades da empresa.

O projeto utilizara node.js para desenvolvimento de Backend, Html 5 e Css para o desenvolvimento do Frontend, também será utilizado o framework de CSS o Bootstrap 5 para um desenvolvimento mais ágil.

4. FERRAMENTAS PARA DESENVOLVIMENTO WEB

Para o desenvolvimento desde projeto foi utilizado HTML 5 como linguagem de marcação para podermos estruturar nosso código frontend que foi desenvolvido em CSS utilizando o Framework Bootstrap.

Para o desenvolvimento do Backend foi utilizado Node.js assim podendo realizar a renderização do frontend e o desenvolvimento de toda parte de onde os dados serão armazenados.

Para auxiliar o desenvolvimento da aplicação, utilizamos alguns utilitários, são eles:

- **Nodemon:** Pacote que monitora os arquivos do aplicativo e reinicia automaticamente o servidor local sempre que ocorrer alterações nos arquivos/código, agilizando o processo de atualização e visualização das alterações.
- **Sequelize:** Usado para facilitar a interação com o banco de dados MySQL utilizado no projeto, permitindo definir os modelos, relacionamentos e consultas usando código JavaScript em vez de escrever diretamente em SQL.
- **Handlebars:** Permite a geração dinâmica de conteúdo HTML, tornando mais fácil a renderização de dados em templates pré-definidos.
- **Body-parser:** Tem como principal objetivo facilitar o processamento de dados enviados por meio de solicitações POST, PUT ou PATCH, onde os dados são

incluídos no corpo da requisição, em vez de serem passados pelos parâmetros da URL

5. Projetando a Ferramenta

Podemos definir uma aplicação web como: é um software acessado por meio de um navegador, permitindo interações com recursos e funcionalidades.

As fases principais de desenvolvimento envolvem análise de requisitos, design, desenvolvimento, testes, implantação e manutenção.

Os erros e problemas comuns incluem segurança inadequada, desempenho insatisfatório, interface confusa, falta de testes adequados e problemas na integração de terceiros.

6. Projeto de interface com o usuário

A Interação Humano-Computador (IHC) em ambientes web se concentra no estudo e projeto de interfaces entre humanos e sistemas computacionais.

Princípios e boas práticas de IHC, como feedback adequado, simplicidade, consistência, usabilidade, acessibilidade, testes com usuários, design responsivo e flexibilidade, são essenciais para criar interfaces intuitivas, usáveis e acessíveis em aplicações web.

Seguir essas diretrizes melhora a experiência do usuário e facilita a interação com a aplicação.

7. Linguagem de Desenvolvimento

Para o desenvolvimento do frontend utilizamos HTML 5 e CSS também foi utilizado o Framework do Bootstrap 5.

Nesta etapa do projeto foi desenvolvido o cadastramento de clientes para que possa ser realizado o agendamento posterior do serviço.

Abaixo podemos visualizar o desenvolvimento do Frontend:

```
index.html > html > body.p-3.m-0.border-0.bd-example.m-0.border-0 > nav.navbar.navbar-expand-md.bg-black.navbar-dark > div.container-fluid > div#navbarSupportedCon
You, last week | 2 authors (ozinel.silva.cla and others)
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4   <meta charset="UTF-8">
<body class="p-3 m-0 border-0 bd-example m-0 border-0">
<nav class="navbar navbar-expand-md bg-black navbar-dark">
  <div class="container-fluid">
    <a class="navbar-brand href="#"><h1 class="m-20"></h1></a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
      aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Agendamento</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Lojas</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Serviços</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">contatos</a>
        </li>
      </ul>
      <form class="d-flex" role="search">
        <input class="form-control me-2 rounded-0" type="search" placeholder="O que deseja?" aria-label="Search">
        <button class="btn btn-outline-light rounded-0" type="submit">Buscar</button>
      </form>
    </div>
  </div>
</nav>
```

```
43
44 <div class="container">
45
46   <form>
47
48     <div class="mb-3">
49
50       <div class="row g-3">
51         <div class="col-sm-7">
52           <input type="text" class="form-control" placeholder="Cidade" aria-label="City">
53         </div>
54         <div class="col-sm">
55           <input type="text" class="form-control" placeholder="Estado" aria-label="State">
56         </div>
57         <div class="col-sm">
58           <input type="text" class="form-control" placeholder="Cep" aria-label="Zip">
59         </div>
60       </div>
61
62       <div>
63         <label for="exampleFormControlTextarea1" class="form-label"> Descreva seu problema</label>
64         <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
65       </div>
66
67       <div class="mb-5">
68         <div id="envio" class="form-text">Enviar formulario:</div>
69         <button type="submit" class="btn btn-primary"> Enviar </button>
70       </div>
71
72     </form>
73   </div>
74
75 </body>
76
77 </html>
```

Abaixo podemos visualizar o desenvolvimento do backend :

```
JS index.js X
JS index.js > app.post('/add') callback
1  const express = require('express')
2  const port = 8081
3  const app = express()
4  const handlebars = require('express-handlebars')
5  const Post = require('./models/Post')
6
7
8  // Config
9  | // Template Engine
10 const exphbs = require('express-handlebars');
11 var handle = exphbs.create({
12   | defaultLayout: 'main'
13   | });
14 app.engine('handlebars', handle.engine);
15 app.set('view engine', 'handlebars');
16 //Body parser
17 app.use(express.urlencoded({extended:false}))
18 app.use(express.json())
19
20 //Rotas
21
22 app.get('/solrealizadas', function(req, res){
23   | Post.findAll().then(function(posts){
24   |   | res.render('home', {posts: posts})
25   |   | })
26   | })
27
28
29 app.get('/sol', function(req, res){
30   | res.render('index')
31   | })
```

```
31
32 })
33
34 app.post('/add', function(req, res){
35   | Post.create([
36   |   | nome: req.body.nome,
37   |   | telefone: req.body.telefone,
38   |   | email: req.body.email,
39   |   | endereco: req.body.endereco,
40   |   | cidade: req.body.cidade,
41   |   | estado: req.body.estado ,
42   |   | cep: req.body.cep,
43   |   | descri: req.body.descri
44   |   | ]).then(function(){
45   |   | res.send("Solicitação realizada com sucesso!")
46   |   | }).catch(function(erro){
47   |   | res.send("Houve um erro: " + erro)
48   |   | })
49   | })
50 })
51
52
53 app .listen(8081, function(){
54   | console.log("Servidor Rodando na url http://localhost:8081");
55   | });
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Solicitacoes Node.JS</title>
7 </head>
8 <body>
9   {{{body}}}
10 </body>
11 </html>
```

```
JS index.js JS db.js X
models > JS db.js > [?] <unknown>
1 const Sequelize = require('sequelize')
2 // Conexão com o banco de dados MySQL
3 const sequelize = new Sequelize('sistemasolicitacoes', 'root', '!@Qw159357179', {
4   host: 'localhost',
5   dialect: 'mysql'
6 })
7
8 module.exports = {
9   Sequelize: Sequelize,
10  sequelize: sequelize
11 }
```

```
JS index.js JS Post.js X
models > JS Post.js > ...
1  const db = require('./db')
2
3  const Post = db.sequelize.define('solicitacoes', {
4  nome: {
5    type: db.Sequelize.STRING
6  },
7  telefone: {
8    type: db.Sequelize.STRING
9  },
10 email: {
11   type: db.Sequelize.STRING
12 },
13 endereco: {
14   type: db.Sequelize.STRING
15 },
16 cidade: {
17   type: db.Sequelize.STRING
18 },
19 estado: {
20   type: db.Sequelize.STRING
21 },
22 cep: {
23   type: db.Sequelize.STRING
24 },
25 descricao: {
26   type: db.Sequelize.TEXT
27 }
28 })
29
30
```

Após a renderização do código podemos ver como resultado a seguinte página web:

The screenshot shows a web page for SMA Mecânica Automotiva. At the top left is the SMA logo with the text 'MECÂNICA AUTOMOTIVA'. To the right of the logo are navigation links: 'Agendamento', 'Lojas', 'Serviços', and 'contatos'. On the far right, there is a search bar with the placeholder text 'O que deseja?' and a 'Buscar' button. Below the header is a contact form with the following fields:

- Nome:
- Telefone: (with a small note: 'Digite seu telefone ex: (99)9999-9999')
- Email: (with a small note: 'Digite seu e-mail')
- Endereço:
- Cidade:
- Estado:
- Cep:
- Descreva seu problema:

At the bottom of the form, there is a label 'Enviar formulário:' and a blue 'Enviar' button.

Tela das solicitações realizadas

Lista de Solicitações:

Thu Jun 15 2023 17:27:39 GMT-0300 (Horário Padrão de Brasília)

PEDRO

19 9 99999999

rborsone3@gmail.com

SADSASDSADSAD

SADSASDSADSA

SADSASDSAD

16841208

Thu Jun 15 2023 19:23:36 GMT-0300 (Horário Padrão de Brasília)

Oziniel

19 9 99999999

oziniel@gmail.com

fovjdfogfjdfjdiogd

sdfsdfsfsfd

sdfsdfsdfs

13841205

8. SISTEMAS OPERACIONAIS

Levando em conta a necessidade descrita pelo proprietário da empresa foi idealizado um sistema web onde será preenchido pelo usuário, assim obtendo informações necessárias pra o agendamento de serviços e a organização de horários de atendimento da empresa.

O acesso a este formulário poderá ser feito através navegadores web de seus Desktop, Smartphone ou qualquer outro dispositivo que tenha instalado um navegador web e com acesso à internet.

9. COMPONENTES DE SISTEMAS OPERACIONAIS

Para o desenvolvimento de de uma aplicação web podemos utilizar alguns sistemas operacionais disponíveis no mercado atualmente como Windows, MacOS, Linux e entre outros que tem grande aceitação do mercado de desenvolvimento e também trazem ferramentas que facilitam o desenvolvimento como o Visual Studio Code e IDEs de desenvolvimento como o Node.JS.

10. GERENCIAMENTO E FUNCIONALIDADES DO SISTEMA OPERACIONAL

Durante o desenvolvimento do projeto utilizamos o Windows como sistema operacional. O sistema operacional Windows é amplamente utilizado para o desenvolvimento de aplicações web. Ele oferece suporte a várias ferramentas e ambientes de desenvolvimento, como o Visual Studio, .NET Framework e diversas IDEs populares.

O MacOS, sistema operacional da Apple, é uma escolha popular entre desenvolvedores web. Ele oferece uma interface amigável, além de uma variedade de ferramentas e ambientes de desenvolvimento, como Xcode, Visual Studio Code e Homebrew.

O Linux é uma escolha muito comum para desenvolvimento web, especialmente entre desenvolvedores mais experientes. Diversas distribuições Linux, como Ubuntu, Fedora e CentOS, oferecem uma ampla gama de ferramentas e ambientes de desenvolvimento.

Além desses sistemas operacionais, existem outras opções menos comuns, como FreeBSD e Chrome OS, que também podem ser usados para o desenvolvimento web, embora sejam menos populares.

11. GERENCIAMENTO DE HARDWARE PELO SISTEMA OPERACIONAL

A escolha entre bancos de dados locais e em nuvem pode ter implicações significativas em termos de segurança, atualização e fluxo de dados. Aqui estão algumas considerações sobre esses aspectos, bem como possíveis vantagens e desvantagens de cada abordagem:

Segurança:

Bancos de dados locais: Os bancos de dados locais estão sob o controle direto da organização ou usuário. Isso permite a implementação de medidas de segurança personalizadas e maior controle sobre o acesso aos dados. No entanto, a segurança depende da implementação correta de práticas de segurança, como criptografia, controle de acesso e proteção contra vulnerabilidades locais.

Bancos de dados em nuvem: Os bancos de dados em nuvem geralmente são fornecidos por provedores de serviços, que são responsáveis pela segurança física e lógica dos dados. Eles geralmente implementam medidas de segurança avançadas, como criptografia de dados em repouso e em trânsito, firewalls e monitoramento de segurança. No entanto, a segurança depende em parte da confiança no provedor de serviços e da adoção de práticas de segurança adequadas pelos usuários.

Atualização:

Bancos de dados locais: Com bancos de dados locais, a responsabilidade pelas atualizações de software e patches de segurança recai sobre a organização ou usuário. Isso permite um controle total sobre o momento e o processo de atualização. No entanto, pode exigir esforços significativos para manter os sistemas atualizados e garantir a compatibilidade com novas versões de software.

Bancos de dados em nuvem: Os bancos de dados em nuvem são atualizados regularmente pelo provedor de serviços. Isso garante que as correções de segurança e as melhorias sejam implementadas de forma rápida e eficiente, sem a necessidade de intervenção direta do usuário. No entanto, as atualizações automáticas podem levar a interrupções temporárias no serviço ou a alterações não desejadas na funcionalidade.

Fluxo de dados:

Bancos de dados locais: Com bancos de dados locais, o fluxo de dados geralmente ocorre dentro da infraestrutura local da organização. Isso pode oferecer maior controle sobre a largura de banda, latência e requisitos de conformidade. No entanto, o acesso remoto aos dados pode ser limitado e pode ser necessário implementar soluções adicionais para permitir o acesso a dados fora do ambiente local.

Bancos de dados em nuvem: Com bancos de dados em nuvem, os dados são armazenados em servidores remotos e o acesso a eles é realizado pela internet. Isso permite acesso fácil e rápido aos dados de qualquer lugar, facilitando a colaboração e o compartilhamento de informações. No entanto, o fluxo de dados depende da conexão com a internet, o que pode ser limitante em termos de velocidade, confiabilidade e segurança.

Em resumo, a escolha entre bancos de dados locais e em nuvem tem implicações em termos de segurança, atualização e fluxo de dados. Os bancos de dados locais oferecem maior controle, mas exigem esforços adicionais para segurança e atualizações.

12. CONCLUSÃO

O desenvolvimento de uma aplicação web de agendamento de clientes para uma oficina mecânica proporciona benefícios significativos, como automação do processo de agendamento, melhor experiência para os clientes e otimização do fluxo de trabalho interno. A aplicação pode ser acessada facilmente por meio de navegadores web e oferece conveniência tanto para os clientes quanto para a equipe da oficina. A segurança dos dados é essencial, e a escolha entre um banco de dados local ou em nuvem deve ser feita considerando as necessidades específicas da oficina. Em resumo, a aplicação web de agendamento de clientes melhora a eficiência, a conveniência e a experiência geral na oficina mecânica.

13. REFERÊNCIAS

Essa parte está reservada para as referências, as quais devem estar metodologicamente discriminadas em ordem alfabética e corresponder às citações realizadas ao longo dos textos.

A utilização da metodologia científica é obrigatória e deve ser utilizado o Manual UNIFEOB para Trabalhos Acadêmicos ou as Normas da ABNT.