

PLATAFORMA CENTRALIZADORA DE MENSAGENS INSTANTÂNEAS

RESUMO

O objetivo deste trabalho é demonstrar os conhecimentos adquiridos ao decorrer do curso de Análise e Desenvolvimento de Sistemas. Trata-se da apresentação de parte de um sistema de uma plataforma que tem como objetivo centralizar os canais de atendimento utilizados por uma empresa para atender seus clientes. Esses canais podem ser o WhasApp, Facebook, Instagram ou até mesmo telefonia. O objetivo principal do MessageBox é promover a integração desses meios de comunicação através de uma plataforma simples e intuitiva, onde o cliente consegue realizar atendimentos simultâneos em um só lugar. Por se tratar de uma arquitetura extensa, esta apresentação terá foco apenas no processo de cadastro de clientes, onde o funcionário poderá, de forma objetiva, cadastrar novos clientes na plataforma, assim como, editar informações desses clientes e excluir cadastros que já estão obsoletos. Além disso, uma explicação sobre a conexão com a parte de telefonia será abordada. A plataforma de cadastro é desenvolvida com HTML e CSS, utilizando a linguagem de programação Javascript em alguns trechos do código e o framework Bootstrap. No back-end foi utilizado a linguagem PHP e para armazenar as informações foi conectado ao banco de dados MySQL. Para a parte de telefonia, haverá uma breve explicação de como utilizar o software Asterisk no Linux.

Palavras Chave: Asterisk, Linux. Projeto Aplicado, Mensagens instantâneas, FAQI.

¹ Graduando em Análise e Desenvolvimento de Sistemas / FAQI / Gravataí / Rio Grande do Sul/ Brasil / e-mail

² Mestre / FAQI / Gravataí / Rio Grande do Sul / Brasil / silvio.viegas@qi.edu.br

1. INTRODUÇÃO

Já pensou em ter vários aplicativos de mensagens instantâneas centralizados em apenas uma plataforma?

Imagine que você tem uma empresa que vende determinado tipo de serviço, mas seu único meio de comunicação com os clientes é através do telefone fixo e você acaba perdendo o cliente que prefere utilizar o WhatsApp ou aquele outro que encontrou seu produto no Instagram, mas não conseguiu entrar em contato por ali mesmo e desistiu de comprar.

Bem, em um mundo como o que estamos vivendo hoje, torna-se impossível pensar nossas vidas sem a tecnologia. Em uma realidade tão tecnológica, onde podemos usufruir de inúmeros aplicativos, plataformas e todo tipo de software para as mais diversas funções, a comunicação é um ponto de extrema importância neste cenário tecnológico e por isso, surgem muitas aplicações para este segmento.

Neste trabalho será apresentado a plataforma de cadastro de clientes do MessageBox. Esta plataforma é parte integrante de um conjunto de procedimentos que atuam para centralizar e suprir a necessidade de manter vários meios de comunicação em apenas um lugar.

1.1 TEMA

Desenvolvimento de um conjunto de telas de cadastro de clientes que, conectado ao banco de dados, realiza uma busca e traz como resultado o seu saldo disponível para uso na plataforma MessageBox.

1.2 DELIMITAÇÃO DO TEMA

Esta versão do MessageBox conta com um visual simples e intuitivo, onde o funcionário pode cadastrar novos clientes, editar informações e excluir dados, assim como também consegue realizar buscas pertinentes ao uso dos serviços oferecidos pelo MessageBox.

1.3 PROBLEMA

Com o aumento dos meios de comunicação oferecidos através de vários serviços e plataformas diferentes, um indivíduo que possui um negócio qualquer, tem a necessidade de ampliar sua rede de comunicação, uma vez que diversos clientes querem entrar em contato de formas diferentes, cada um utilizando a aplicação que mais lhe agrada, houve a necessidade de centralizar todas essas formas de contato em uma só plataforma, afim de captar usuários de vários aplicativos de mensagens instantâneas, tornando-os clientes em potencial.

1.4 OBJETIVO GERAL

Desenvolver as telas de cadastro para uso dos funcionários que utilizam o MessageBox para cadastrar clientes dessa plataforma que centraliza vários meios de comunicação em um só lugar.

1.5 OBJETIVOS ESPECÍFICOS

- Criação da tela inicial que apontará para as telas de cadastro, edição e exclusão de clientes.
- Receber e tratar as informações inseridas nas caixas da página e enviá-las ao banco de dados para validação.
- Manter seguros os dados dos usuários.
- Plataforma simples e intuitiva com design responsivo.
- Front-end utilizando HTML, CSS, Javascript e Bootstrap.
- Back-end utilizando PHP e MySQL.

1.6 JUSTIFICATIVA

A motivação para a realização deste trabalho implica no desejo do autor em criar um software que centralize vários meios de comunicação como aplicativos de mensagens e telefonia em uma única plataforma, afim de prospectar clientes que

busquem eficiência e qualidade na hora de vender seu produto para terceiros, assim como, conforto e agilidade para quem compra.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta sessão serão abordados assuntos sobre o crescimento dos usuários de aplicativos de mensagens instantâneas e conceitos relacionados às linguagens de programação utilizadas neste projeto.

2.1 APLICATIVOS DE MENSAGENS INSTANTÂNEAS

De acordo com a pesquisa *Global Mobile Messaging Forecast 2021* – que aponta os dados sobre aplicativos móveis de mensagem instantânea ao redor do mundo -, apenas em 2021, cerca de 3,09 bilhões de usuários utilizaram apps de mensagens por mês em todo o mundo. Esse é um aumento de 6,1% em relação ao ano passado.

Em uma visão de ranking mundial, o Brasil ocupa o segundo lugar com 99 milhões de usuários ativos mensais, um número enorme, pois temos uma população total de 200 milhões de pessoas no total, ou seja, é quase a metade da população total.

Acompanhando este crescimento, empresas buscam meios para manter seus clientes (também usuários desses aplicativos) mais próximos do seu negócio. Uma vez que os meios de comunicação se ramificam expandindo em formas diversas que surgem através dessas plataformas, também cresce uma demanda que exige a necessidade de manter contato através do máximo possível desses aplicativos, afinal, quanto maior o número de vias de comunicação, mais clientes podem ser alcançados. Sendo assim, o MessageBox surge como uma solução para contribuir no contato entre empresa e cliente, reunindo os mais utilizados meios de comunicação e envio de mensagens instantâneas em uma única plataforma.

Um conceito interessante sobre esses aplicativos é o de ser open source. Isto significa que a forma como um app ou software é desenvolvido, está aberta e pode ser vista por qualquer pessoa. Você pode ficar com a sensação de que o fato de o código estar aberto deixe o sistema mais inseguro, mas é justamente o contrário. Por

ser aberto, tem mais gente trabalhando para melhorar a segurança dele. Isso diminui os pontos fracos e de vulnerabilidade.

2.2 LINGUAGENS DE PROGRAMAÇÃO

2.2.1 PHP

O PHP, como é conhecido hoje, é na verdade o sucessor para um produto chamado PHP/FI. Criado em 1994 por Rasmus Lerdof, a primeira encarnação do PHP foi um simples conjunto de binários Common Gateway Interface (CGI) escrito em linguagem de programação C.

Com o tempo, essa linguagem de programação se tornou muito popular. Quando vemos algumas definições sobre o que é PHP, geralmente, ele é descrito como uma linguagem do lado do servidor. Na prática, isso significa que ele é aplicado na programação que acontece em um servidor da web responsável por rodar a aplicação ou, mais frequentemente, um site.

Esse trabalho prévio permite que os elementos de uma página sejam carregados antes de serem exibidos ao usuário que acessa um site, por exemplo. O código PHP é executado no servidor que, ao ler os comandos, consegue ativar todos os elementos funcionais e de interface visual do site.

2.2.2 Javascript

JavaScript é uma linguagem de programação de alto-nível, criada no meio da década de 90, mais precisamente em 1996 pelo lendário programador Brendan Eich que, além de criar o JavaScript, foi também um dos fundadores da Mozilla Corporation.

A linguagem de programação Javascript permite ao desenvolvedor implementar diversos itens de alto nível de complexidade em páginas web, como animações, mapas, gráficos ou informações que se atualizam em intervalos de tempo padrão, por exemplo.

Javascript é a terceira camada do bolo de desenvolvimento web e front-end, junto com HTML, CSS e PHP.

2.2.3 HTML

É uma linguagem de marcação que serve para dar significado e estrutura ao conteúdo web, como, por exemplo, definir cabeçalhos, parágrafos, fazer referências ou inserir imagens e vídeos.

Figura 1 - Exemplo de código HTML

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Cadastro</title>
8   <link href="css/bootstrap.min.css" rel="stylesheet">
9 </head>
```

Fonte: O autor (2022)

2.2.4 CSS

É uma linguagem de folha de **estilos**, que tem o papel de tornar uma página apresentável na web, relacionada diretamente com o design e aparência.

Figura 2 - Exemplo de código CSS

```
width: 100%;
padding-right: calc(var(--bs-gutter-x) * 0.5);
padding-left: calc(var(--bs-gutter-x) * 0.5);
margin-right: auto;
margin-left: auto;
}

@media (min-width: 576px) {
  .container-sm, .container {
    max-width: 540px;
  }
}
```

Fonte: O autor (2022)

2.2.5 Bootstrap

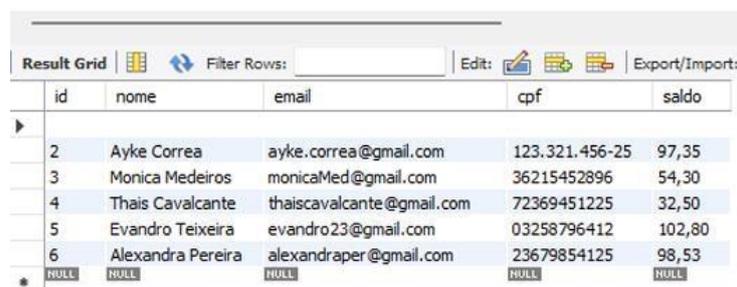
Bootstrap é um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web, usando HTML, CSS e Javascript, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo.

2.2.6 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional, que utiliza a linguagem SQL como interface. É atualmente um dos sistemas de gerenciamento de banco de dados mais populares da Oracle Corporation, com mais de 10 milhões de instalações pelo mundo.

Figura 3 - Exemplo de consulta ao banco de dados

```
2
3 • SELECT * FROM cliente;
```



id	nome	email	cpf	saldo
2	Ayke Correa	ayke.correa@gmail.com	123.321.456-25	97,35
3	Monica Medeiros	monicaMed@gmail.com	36215452896	54,30
4	Thais Cavalcante	thaiscavalcante@gmail.com	72369451225	32,50
5	Evandro Teixeira	evandro23@gmail.com	03258796412	102,80
6	Alexandra Pereira	alexandraper@gmail.com	23679854125	98,53
NULL	NULL	NULL	NULL	NULL

Fonte: O autor (2022)

2.3 FERRAMENTAS E TECNOLOGIAS

2.3.1 Visual Studio Code

O Visual Studio Code (VSCode) é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e MacOS. Ele inclui suporte para depuração, controle

de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código.

2.3.2 Xampp

Xampp é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl.

2.3.3 Asterisk

O Asterisk foi originalmente desenvolvido para Linux, mas atualmente pode ser instalado e executado em uma grande variedade de sistemas operacionais, incluindo NetBSD, OpenBSD, FreeBSD, Mac OS X, Solaris e até mesmo Microsoft Windows. É um software, baseado na licença GPL (GNU General Public License), que executa todas as funções de uma central telefônica convencional através das principais tecnologias de comunicação existentes no mercado, como por exemplo: linhas telefônicas analógicas, links de telefonia digital via placas de comunicação TDM.

Através do Asterisk é possível estabelecer uma chamada entre qualquer um desses protocolos. O Asterisk é o que chamamos de B2B User Agent, ou melhor, Back-to-Back User Agent, ele estabelece uma chamada telefônica, separando cada “perna” da chamada entre origem e destino, e monitora todo o tráfego de áudio entre esses pontos. Por isso, caso esses pontos estejam utilizando CODECs diferentes, o Asterisk consegue transcodificar o áudio sem qualquer problema.

Embora não seja apresentado neste trabalho, com ele podemos implementar chamadas por voz na plataforma MessageBox. Na figura 4 segue exemplo de um código em script dialplan para funcionar com Asterisk.

```

exten = 0,1,Verbose(1, "Caller ${CALLERID(num)} dialed the operator.")
same = n,Goto(Dial-Users,111,1)
exten = 1,1,Verbose(1, "Caller ${CALLERID(num)} dialed the Sales queue.")
same = n,Goto(External-Features,2565551200,1)
exten = 2,1,Verbose(1, "Caller ${CALLERID(num)} dialed the Customer Experience queue.")
same = n,Goto(External-Features,2565551250,1)
exten = 3,1,Verbose(1, "Caller ${CALLERID(num)} dialed Accounting and Receivables.")
same = n,Goto(Dial-Users,1100,1)
exten = 4,1,Verbose(1, "Caller ${CALLERID(num)} dialed the directory.")
same = n,Directory(example,Dial-Users)

exten = 1,1,Playback(option-is-invalid)
same = n,Goto(2565551100,start)

exten = 1,1,Playback(are-you-still-there)
same = n,Goto(2565551100,start)

; Calls from our ITSP SIP account arrive in DCS-Incoming. We should be careful
; to route calls very explicitly so as to avoid any security issues, such as
; accidentally giving outbound dial access to inbound callers.
;
; Each context includes extension pattern matching to match the inbound DID
; dialed appropriately.
[DCS-Incoming]
include = Main-IVR
include = DID-Extensions
include = External-Features

[ramais]
exten => 800,1,Goto(atendimento,s,1)

[atendimento]
exten => s,1,Answer()
exten => s,2,Background(/var/lib/asterisk/sounds/bem-vindo)
exten => s,3,Background(/var/lib/asterisk/sounds/escolha-area)
exten => s,4,WaitExten(5)
exten => s,1,Goto(suporte,s,1)
exten => s,1,Goto(desenvolvimento,s,1)
exten => s,5,Background(/var/lib/asterisk/sounds/opcao-invalida)
exten => s,6,Background(/var/lib/asterisk/sounds/despedida)
exten => s,7,Hangup

[suporte]
exten => s,1,Playback(/var/lib/asterisk/sounds/suporte)
exten => s,2,Read(cpf,cpf,11)
exten => s,3,AGI(connection.php,${cpf})
exten => s,4,NoOp("****SALDO****")
exten => s,5,AgI(asterisk-googletts/googletts.agi,"seu saldo é de R$ ${SALDO}",pt-BR)
exten => s,6,Hangup

[teste]
exten => s,1,Background(/var/lib/asterisk/sounds/audio-sup)
exten => s,2,AGI(/var/lib/asterisk/agi-bin/teste.php)
exten => s,3,NoOp(Digite a opção/cpf/0-cadastro)

```

Fonte: O autor (2022)

O script do dialplan foi feito em um arquivo no Linux e segue uma sequência de comandos que atendem a chamada e enviam para determinados processos dentro da aplicação.

É possível gravar áudios que irão servir como suporte para orientar o usuário em suas escolhas no menu que será programado em script, assim como, conectar no dialplan um arquivo PHP com script que execute qualquer tarefa relevante para o funcionamento do mesmo, um exemplo disso, pode ser a conexão com o banco de dados, o que seria de suma importância na plataforma desenvolvida neste trabalho.

3. METODOLOGIA

Em estudos teóricos que foram realizados, observou-se a necessidade de desenvolver uma estrutura que permitisse que empresas pudessem comercializar seus produtos mantendo contato com seus clientes através de variados canais de comunicação, uma vez que a tecnologia nos dias atuais permite muitas vias de acesso e comunicação por meio de diversas plataformas e aplicativos de mensagens instantâneas. Pensando em centralizar esses serviços, foram levantados requisitos

para mapear a construção da plataforma MessageBox, com linguagem de programação utilizando paradigma orientado à objetos. Entretanto, por se tratar de uma aplicação muito extensa em sua estrutura como um todo, este trabalho trás como foco a arquitetura de front e back-end apenas da parte de cadastro de clientes e manipulação no banco de dados, não tendo a necessidade do uso de classes e orientação a objetos para que esse microserviço funcione.

3.1 PESQUISA APLICADA

Pesquisa aplicada para mensurar esforço e compreender o que será necessário para o desenvolvimento e aplicação das tecnologias necessárias.

Sobre a pesquisa científica, podemos dizer que se trata de uma atividade que cria uma relação de proximidade entre a realidade e a investigação. É o produto de uma investigação detalhada e meticulosa, e nesse aspecto, busca responder o problema proposto, baseando-se em mecanismos científicos. Utilizando de uma associação de métodos e técnicas, a pesquisa científica se constrói sobre esses elementos empregados pelos pesquisadores para alcançar e produzir novos conhecimentos ou complementar conhecimentos já existentes. Assim a pesquisa científica segue todos os processos e normas metodológicas para que a investigação seja validada e representativa.

Figura 5 - Metodologia do projeto EMK

1.1 - Área de atuação; 1.2 - Linguagem de Programação; 1.3 - Banco de dados;	1 - Definição
2 - Conhecimento	2.1 - Esocial; 2.2 - Operadoras de Saúde; 2.3 - Integração ERP Totvs RM; 2.4 - Regras de negócio;
3.1 - Moldes Estruturais ER; 3.2 - Moldes Conceituais POO;	3 - Modelagem
4 - Desenvolvimento	4.1 - Arquitetura; 4.2 - Codificação; 4.3 - Padrão de <i>Interface</i> ; 4.4 - Teste unitários e Validação;
5.1 - Análise;	5 - Resultado

Fonte: O autor (2022)

1 - Definição: Nesse primeiro segmento são definidas as especificações macros do projeto, que são decisivas no seu desenvolvimento.

1.1 - Área de atuação: Parte onde é identificado o nicho de mercado do projeto, neste caso, área de vendas e comunicação.

1.2 - Linguagem de programação: Define as linguagens utilizadas no projeto.

PHP e Javascript.

1.3 - Banco de dados: Para este projeto, foi utilizado o banco de dados MySQL.

2 - Conhecimento: Parte onde é pensado as regras do negócio e influência que terá sobre a aplicação.

2.1 - Visão do negócio: Quem o negócio irá atender, suas camadas e subcamadas para que o projeto atinja seus objetivos.

2.2 - Regras de negócio: Esse tópico foi utilizado para a compreensão macro das possíveis regras de negócio que podem estar envolvidas para benefício da empresa.

3 - Modelagem: Nesse segmento são definidas as especificações relativas à modelagem conceitual desse trabalho.

3.1 - Moldes estruturais ER: Esse diagrama é utilizado para retratar a visão da estrutura de armazenamento dos dados.

3.2 - Moldes conceituais POO: Os diagramas escolhidos para retratar o projeto foram o diagrama de atividade, visando a importação e exportação dos dados do software e o diagrama de classe para abstrair a ideia dos objetos constantes no EMK.

4 - Desenvolvimento: Parte do projeto onde são estabelecidas as tecnologias que serão utilizadas na arquitetura do projeto.

4.1 - Arquitetura: Arquitetura baseada no modelo cliente-servidor onde o cliente requisita informações no front-end e o servidor retorna através do back-end.

4.2 - Codificação: Linguagens de programação serão PHP e Javascript. HTML como linguagem de marcação e CSS como linguagem de folha de estilos.

5 - Resultado: Fazer uso da plataforma, testando seu funcionamento ao cadastrar novos clientes, assim como para editar suas informações e também deletá-las.

5.1 - **Análise:** A análise é realizada partindo da interação realizada com a aplicação e testes durante certo período para avaliação e controle de atualização e funcionamento livre de bugs.

4. DESENVOLVIMENTO

Neste capítulo será apresentado o desenvolvimento desta aplicação, o processo de criação, funcionamento do código e como o script se comunica com o base de dados para buscar as informações solicitadas pelo cliente.

4.1 PÁGINA INDEX

Explicação sobre o desenvolvimento da página index, servindo de índice para a criação das demais páginas do projeto.

Figura 6 - Código HTML da página index

```

4 <head>
5
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8 <title>Cadastro</title>
9 <link href="css/bootstrap.min.css" rel="stylesheet">
10 </head>
11
12 <body>
13 <nav class="navbar navbar-expand-lg bg-light">
14 <div class="container-fluid">
15 <div>
16 
17 </div>
18 <a class="navbar-brand" href="#">Cadastro</a>
19 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
20 <span class="navbar-toggler-icon"></span>
21 </button>
22 <div class="collapse navbar-collapse" id="navbarSupportedContent">
23 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
24 <li class="nav-item">
25 <a class="nav-link active" aria-current="page" href="index.php">Home</a>
26 </li>
27 <li class="nav-item">
28 <a class="nav-link" href="?page=novo">Novo Cliente</a>
29 </li>
30 <li class="nav-item">
31 <a class="nav-link" href="?page=listar">Listar Clientes</a>
32 </li>
33 </ul>

```

Fonte: O autor (2022)

Utilizando HTML e Bootstrap, a primeira parte da página index cria basicamente a tela principal da nossa plataforma de cadastro.

Figura 7 - Switch case em PHP

```

38 <div class="container">
39 <div class="row">
40 <div class="col mt-5">
41 <?php
42 include("config.php");
43 switch (@$_REQUEST["page"]) {
44 case "novo":
45 include("novo-cliente.php");
46 break;
47 case "listar":
48 include("listar-clientes.php");
49 break;
50 case "salvar":
51 include("salvar-cliente.php");
52 break;
53 case "editar":
54 include("editar-cliente.php");
55 break;
56 default:
57 print "<h1>Bem-vindo</h1>";
58 }

```

Fonte: O autor (2022)

Ainda na página index, abre-se a tag PHP, faz-se a inclusão do arquivo “config.php” e inicia um switch case para direcionar as páginas.

Figura 8 - Arquivo de conexão com banco de dados

```
config.php > ...  
1  <?php  
2  
3  define('HOST', 'localhost');  
4  define('USER', 'root');  
5  define('PASS', '0020');  
6  define('BASE', 'cadastro');  
7  
8  $conn = new MySQLi(HOST, USER, PASS, BASE);  
9  
10
```

Fonte: O autor (2022)

O arquivo “config.php” é responsável por estabelecer a conexão com o banco de dados por meio do script na figura 7.

4.2 CADASTRO DE CLIENTES

Explicação sobre o desenvolvimento do código na página de cadastro de clientes da plataforma MessageBox.

Figura 9 - Código de cadastro

```
2
3  switch ($_REQUEST["acao"]) {
4      case "cadastrar":
5          $nome = $_POST["nome"];
6          $email = $_POST["email"];
7          $cpf = $_POST["cpf"];
8          $saldo = $_POST["saldo"];
9
10         $sql = "INSERT INTO cliente (nome, email, cpf, saldo)
11         VALUES ('{$nome}', '{$email}', '{$cpf}', '{$saldo}')";
12
13         $res = $conn->query($sql);
14
15         if ($res == true) {
16             print "<script>alert('Cadastrado com sucesso');</script>";
17             print "<script>location.href='?page=listar';</script>";
18         } else {
19             print "<script>alert('Não foi possível realizar o cadastro');</script>";
20             print "<script>location.href='?page=listar-clientes.php';</script>";
21         }
22         break;

```

Fonte: O autor (2022)

No arquivo de cadastro de clientes da plataforma (figura 8) temos outro switch case onde na primeira ação encaminha para a página “cadastrar”. A requisição é feita via “POST” e armazena nas variáveis o que foi inserido pelo usuário. Em seguida é feito um *insert* no banco de dados que vai inserir informações na base “cadastro”. Também podemos ver um *if-else* que retorna um valor booleano e usa Javascript para fazer aparecer um aviso na tela, avisando sobre sucesso ou falha ao realizar o cadastro.

4.3 EDIÇÃO DE CADASTRO

Código do arquivo que edita as informações já armazenadas na base de dados. Quando existe a necessidade de reformular uma informação sobre o cliente cadastrado.

Figura 10 - Código de edição

```

23     case "editar":
24         $nome = $_POST["nome"];
25         $email = $_POST["email"];
26         $cpf = $_POST["cpf"];
27         $saldo = $_POST["saldo"];
28
29         $sql = "UPDATE cliente SET nome='{$nome}', email='{$email}', cpf='{$cpf}', saldo='{$saldo}'
30             WHERE cpf=" . $_REQUEST["cpf"];
31
32         $res = $conn->query($sql);
33
34         if ($res == true) {
35             print "<script>alert('Editado com sucesso');</script>";
36             print "<script>location.href='?page=listar';</script>";
37         } else {
38             print "<script>alert('Não foi possível Editar');</script>";
39             print "<script>location.href='?page=listar';</script>";
40         }

```

Fonte: O autor (2022)

Agora o switch case entra na ação de editar, recebendo e armazenando os valores inseridos pelo usuário e, em seguida, realizando um “update” no banco de dados, onde as informações cadastradas poderão ser editadas. Mais uma vez se utiliza um *if-else* que retorna um valor booleano e usa Javascript para fazer aparecer um aviso na tela, avisando sobre sucesso ou falha ao realizar a edição.

4.4 EXCLUSÃO DE CLIENTES

Código do arquivo que exclui informações armazenadas no banco de dados e remove o cliente.

Figura 11 - Código de exclusão

```

43     case "excluir":
44
45         $sql = "DELETE FROM cliente WHERE cpf=" . $_REQUEST["cpf"];
46
47         $res = $conn->query($sql);
48
49         if ($res == true) {
50             print "<script>alert('Excluído com sucesso');</script>";
51             print "<script>location.href='?page=listar';</script>";
52         } else {
53             print "<script>alert('Não foi possível excluir');</script>";
54             print "<script>location.href='?page=listar';</script>";
55         }
56
57         break;
58     }
59

```

Fonte: O autor (2022)

O switch case entra na ação de excluir informações, recebendo e armazenando os valores inseridos pelo usuário e, em seguida, realizando um *delete* no banco de dados, onde as informações cadastradas poderão ser deletadas. Mais uma vez se utiliza um *if-e/else* que retorna um valor booleano e usa Javascript para fazer aparecer um aviso na tela, avisando sobre sucesso ou falha ao realizar a exclusão do cadastro.

4.5 LISTAGEM DE CLIENTES

Código que mostra a lista de clientes cadastrados de forma a visualizar o que se pretende editar ou excluir.

Figura 12 - Código de listagem de clientes

```

listar-clientes.php > ...
1  <h1>Listar Clientes</h1>
2  <?php
3  $sql = "SELECT * FROM cliente ORDER BY nome";
4  $res = $conn->query($sql);
5  $qtd = $res->num_rows;
6
7  if ($qtd > 0) {
8      print "<table class='table table-hover table-striped'>";
9      print "<tr>";
10     print "<th>NOME</th>";
11     print "<th>E-MAIL</th>";
12     print "<th>CPF</th>";
13     print "<th>SALDO</th>";
14     print "<th>Ações</th>";
15     print "</tr>";
16     while ($row = $res->fetch_object()) {
17         print "<tr>";
18         print "<td>" . $row->nome;
19         print "</td>";
20         print "<td>" . $row->email;
21         print "</td>";
22         print "<td>" . $row->cpf;
23         print "</td>";
24         print "<td>" . $row->saldo;
25         print "</td>";
26         print "<td>";
27         <button onclick="\location.href='?page=editar&cpf=" . $row->cpf . "';"
28             class='btn btn-success'>Editar</button>
29
30         <button onclick="\if(confirm('Tem certeza que deseja excluir?'))
31             {location.href='?page=salvar&acao=excluir&cpf=" . $row->cpf . "'};
32             else{false;}" class='btn btn-danger'>Excluir</button>
33
34         </td>";
35     print "</tr>";
36     }
37     print "</table>";
38 } else {
39     print "<p class='alert alert-danger'>Não encontrou resultados!</p>";
40 }
41 ?>

```

Fonte: O autor (2022)

O código e listagem de clientes começa fazendo um *select* no banco de dados, ordenando a listagem por nome. O que foi selecionado na listagem é quebrado em linhas e armazenado na variável. Logo usa-se um *if-else* como uma condição e dentro do bloco usa-se um *while* que tem como parâmetro o retorno da linha atual do conjunto de resultados em forma de objeto. Essa parte do código termina com os alertas em Javascript e, caso o resultado booleano retorne falso, um alerta avisando que não foram encontrados resultados é disparado.

5. RESULTADOS

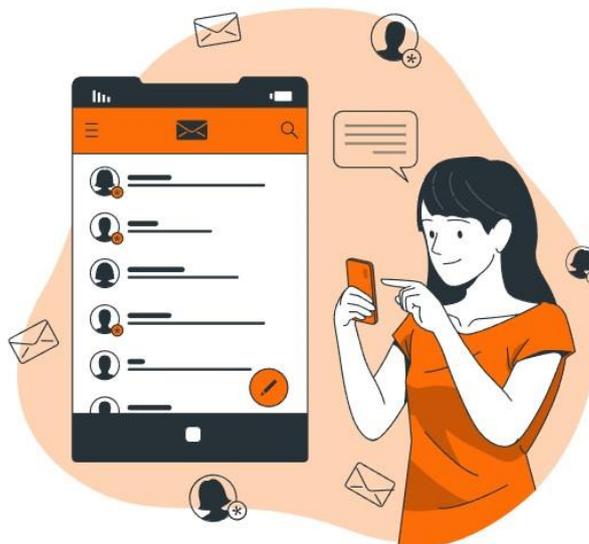
Neste capítulo iremos abordar sobre a sequência final do trabalho, mostrando a parte visual do projeto e os resultados obtidos através dos códigos e conexões. Teremos uma ideia mais sólida do funcionamento da plataforma de cadastro do MessageBox, suas funcionalidades e o seu layout.

5.1 TELA INICIAL DE CADASTRO

A tela inicial de cadastro do MessageBox é basicamente o menu para onde o usuário irá explorar os recursos que necessita.

Figura 13 - Tela inicial de cadastro

Bem-vindo



Fonte: O autor (2022)

Mapeando a tela, temos os links home, Novo cliente e Listar Clientes. São links de acesso a essas páginas para realização das tarefas escolhidas pelo usuário.

5.2 PÁGINA NOVO CLIENTE

A tela “Novo cliente” possui quatro campos de preenchimento, nome, e-mail, cpf e saldo, informações básicas de uso restrito a consultas específicas para verificar o saldo do cliente.

Figura 14 - Página de cadastro para novo cliente

message.BOX Cadastro Home Novo Cliente Listar Clientes

Cadastrar novo cliente

Nome
Patrick Correa

E-mail
patrick.correa986@gmail.com

CPF
123.321.456-25

Saldo
158,50

Cadastrar

Fonte: O autor (2022)

Ao preencher os campos com as informações e clicar sobre o botão “enviar”, as mesmas serão validadas e enviadas ao banco de dados, onde serão armazenadas para futuras consultas.

5.3 PÁGINA DE LISTAGEM DE CLIENTES

A tela “Listar clientes” tem a função de exibir na tela uma lista com todos os clientes que estão cadastrados no sistema do banco de dados.

Figura 15 - Página de listagem de clientes

messageBOX Cadastro de Clientes Home Novo Cliente Listar Clientes

Listar Clientes

NOME	E-MAIL	CPF	SALDO	Ações
Adriano Pereira	adriano.per@gmail.com	896.854.236-87	38,66	Editar Excluir
Alexandra Domingues	aledomingues@gmail.com	528.642.978-26	122,30	Editar Excluir
Bruna Barcelos	brubar@outlook.com	362.254.856-32	98,42	Editar Excluir
Cássio Silva	cassilva@yahoo.com	225.663.951-47	16,80	Editar Excluir
Daiane Vasconcelos	adayvas@outlook.com	036.852.413-73	25,95	Editar Excluir
Emerson Vieira	vieiraemerson@outlook.com	023.236.854-79	48,65	Editar Excluir
Fábio Mello	odemello@gmail.com	163.658.943-21	96,37	Editar Excluir
Gisele Mendes	gilzinha@gmail.com	256.397.851-32	157,80	Editar Excluir
Maria Castro	mariacastro@hotmail.com	036.658.415-23	23,78	Editar Excluir
Patrick Correa	mackhuller@outlook.com	123.321.456-25	158,50	Editar Excluir
Thais Helena	thais.cal@gmail.com	032.569.458-82	97,35	Editar Excluir

Fonte: O autor (2022)

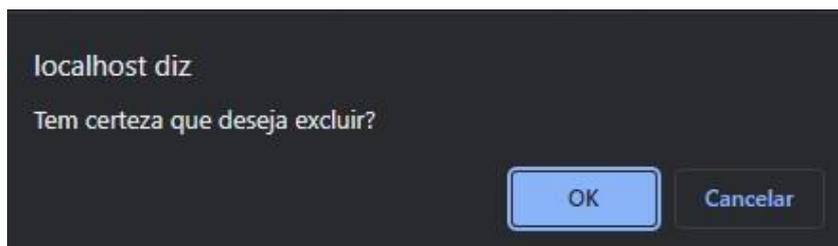
Além da lista de clientes dispostos em ordem alfabética, no canto direito da tela podemos ver os botões “editar” e “excluir” onde é possível fazer a edição das informações dos clientes e também a exclusão dos mesmos.

5.4 PÁGINAS PARA EDITAR E EXCLUIR CLIENTES

Ao clicar no botão verde “editar”, uma página (figura 17) será aberta com as informações do cliente para que sejam editadas conforme a necessidade. Uma consulta utilizando “UPDATE” é realizada no banco de dados e assim que as mudanças são feitas, as novas informações são salvas.

Quando clicamos no botão vermelho “excluir” no canto direito da tela, nenhuma nova página é exibida, apenas é disparado um alerta perguntando se o usuário quer realmente excluir aquele registro (figura 16).

Figura 16 - Alerta de exclusão



Fonte: O autor (2022)

Figura 17 - Página de edição de clientes

The screenshot shows the "message.BOX" web application interface. The header includes the logo "message.BOX" and navigation links: "Cadastro de Clientes", "Home", "Novo Cliente", and "Listar Clientes". The main content area is titled "Editar cliente" and contains a form with the following fields: "Nome" (Patrick Correa), "E-mail" (mackhuller@outlook.com), "CPF" (123.321.456-25), and "Saldo" (158,50). A blue "Enviar" button is located at the bottom of the form.

Fonte: O autor (2022)

Assim, de forma simples e objetiva, o usuário consegue realizar cadastros de novos clientes, assim como, editar as informações cadastradas na base de dados e também as excluir. Com a implementação do dialplan com Asterisk, tornou-se possível que o próprio cliente, ao realizar uma ligação para a central de atendimento do MessageBox, consiga, através da inserção do seu número de CPF nas teclas do aparelho, realizar uma consulta para saber seu saldo disponível para uso na plataforma. Isso torna-se possível devido a integração feita com o script em PHP entre o banco de dados e o dialplan.

6. CONSIDERAÇÕES FINAIS

Como resultados obtidos no projeto sobre a implantação da plataforma de cadastro de clientes do MessageBox, vimos que de forma muito clara e objetiva é possível desenvolver uma solução para manter salvas de forma segura as informações dos clientes cadastrados.

Com relação ao desenvolvimento, não seria possível ter uma aplicação coesa e segura sem o paradigma de orientação a objetos, pois o uso de classes e métodos seria essencial, entretanto, para a criação da plataforma de cadastro, foi possível manter uma unidade independente do sistema completo, sem uso de orientação a objetos e ainda assim, funcional e segura.

Como solução objetiva para uso diário de usuários e clientes, a plataforma de cadastro funcionou de forma rápida e bem fluída, possibilitando desenvoltura nos processamentos pertinentes ao que se propõe cumprir. O mesmo resultado pôde ser apurado nas iterações requisitadas via dialplan através do Asterisk, onde as requisições feitas por voz responderam de forma satisfatória.

Ainda que o que foi apresentado seja apenas uma parte pequena de tudo que pode ser criado na plataforma, podemos pensar a respeito de todas as possibilidades que podem ser implementadas como recursos para gerar resultados muito satisfatórios, o que poderia render ainda mais valor para a aplicação como um todo.

Mesmo que a plataforma de cadastro seja apenas uma parte de toda a arquitetura que envolve a ideia geral do projeto MessageBox, descrito em passagens de páginas anteriores, foi possível aplicar de forma efetiva e satisfatória os aprendizados adquiridos através das disciplinas do curso e o vislumbre de um futuro para a aplicação também pode ser explorado e executado.

7 REFERÊNCIAS

CARTER, Jim. **Apache HTTP server Project**. Disponível em: https://httpd.apache.org/ABOUT_APACHE.html. Acesso em: 12 set. 2022.

CORREA, Carlos. **O que é Javascript?** Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-javascript>. Acesso em: 23 ago. 2022.

EIS, D. **Uma breve história do CSS - Artigos sobre HTML, JavaScript, CSS e desenvolvimento web**. Disponível em: <https://tableless.com.br/uma-breve-historiando-css>. Acesso em: 18 ago. 2022.

HOFFMAN, C. **What Is Open Source Software, and Why Does It Matter?** Disponível em: <https://www.howtogeek.com/129967/htg-explains-what-is-opensource-software-and-why-you-should-care>. Acesso em: 6 set. 2022.

OTTO, Mark. **Bootstrap (framework front-end)**. Disponível em: [https://pt.wikipedia.org/wiki/Bootstrap_\(framework_front-end\)](https://pt.wikipedia.org/wiki/Bootstrap_(framework_front-end)). Acesso em: 21 ago. 2022.

PEREIRA, Jonas. **WhatsApp aumenta base de usuários em 41%**. Disponível em: <https://www.tecmundo.com.br/software/239556-whatsapp-aumenta-baseusuarios-engajados>. Acesso em: 12 ago. 2022.

SIERRA, K; BATES B. **Use a cabeça! Java**. Rio de Janeiro, Alta Books, 2010.

TOVILO, Ilija. **PHP 8.1.5 released**. Disponível em: <https://pt.wikipedia.org/wiki/PHP>. Acesso em: 21 ago. 2022.