



**SISTEMA OMNICHANNEL PARA CLÍNICA MÉDICA:  
DESENVOLVIMENTO DE PROTÓTIPO VIA ARQUITETURA SQL,  
PHP E ABORDAGEM LOW-CODE**

Autor: Sandro Delmondes dos Anjos

Professor/Orientador: Felipe Douglas

2025

Resumo:

Este artigo finaliza o Projeto de Clínica Omnichannel da matéria de Low-Code, detalhando o desenvolvimento e a análise do protótipo de uma plataforma **Omnichannel** para clínicas médicas. O projeto teve como foco demonstrar a integração de múltiplos canais de comunicação com um banco de dados centralizado. A arquitetura foi estabelecida nas Unidades 3 e 4, utilizando **SQL Workbench** para a criação da estrutura de dados, **PHP** no **VS Code** para a lógica *backend* e **XAMPP** como servidor local. A metodologia incluiu o uso de ferramentas de **Inteligência Artificial (IA)** para a abordagem *low-coding*, visando agilizar o processo de prototipagem. O estudo reflete sobre a complexidade da integração entre plataformas *low-code* (mencionadas como Bubble e Base44) e o sistema SQL customizado, concluindo que o desafio de associar a lógica *low-code* ao *backend* tradicional impacta a escalabilidade da solução.

**Palavras-chave:** Omnichannel; Low-code; SQL; PHP; Arquitetura de Dados; Extensão Universitária.

## 1. INTRODUÇÃO:

A necessidade de otimizar a experiência do paciente no setor de saúde demanda soluções tecnológicas que unifiquem os canais de atendimento. O conceito **Omnichannel** surge como resposta, garantindo que o histórico do paciente seja mantido e acessível, independentemente do ponto de contato.

O presente trabalho, desenvolvido no âmbito do Componente Curricular Projeto de Clínica Omnichannel, objetivou materializar um protótipo básico de plataforma Omnichannel, pautado na metodologia *low-coding* e na estruturação de um *backend* robusto apesar de também ter foco quando tange a metodologia low-code. O protótipo se concentra em provar a viabilidade da centralização de dados e na reflexão sobre as ferramentas de desenvolvimento rápido.

## 2. FUNDAMENTAÇÃO TEÓRICA:

1. Conceito Omnichannel: É um novo conceito focado em automatizar e simplificar a interação de usuário com atendente ou até mesmo unificar funcionalidades e serviços a fim de integrar os canais de comunicação/vendas de uma empresa.
2. O uso de arquitetura SQL em sistemas transacionais: A arquitetura baseada em SQL (Structured Query Language) é predominante em sistemas transacionais (também conhecidos como OLTP - Online Transaction Processing) devido ao uso de bancos de dados relacionais. Uma de suas vantagens do SQL é seu modelo relacional com esquema predefinido e restrições de integridade, fora que existem diversos profissionais capacitados na área!
3. Vantagens e limitações da arquitetura em Low-Code: As aplicações low-code tendem a utilizar nenhum código ou pouco código, disponibilizando ferramentas interativas e visuais (arrasta e solta) para facilitar alguns processos que precisam de uma aplicação rápida. Algumas de suas vantagens são velocidade do desenvolvimento, redução de erros e custo-benefício, mas diante de tantas vantagens a abordagem possui algumas limitações como: limitação de customização, dependência da plataforma, segurança e escalabilidade. Muitas plataformas de desenvolvimento low-code limitam o livre arbítrio do desenvolvedor, como banco de dados sem hospedagem em nuvem dentre algumas outras limitações.

## 3. METODOLOGIA E DESENVOLVIMENTO DO PROTÓTIPO

A metodologia de desenvolvimento adotada neste protótipo seguiu uma abordagem híbrida: tradicional (para a arquitetura de dados) utilizando SQL Workbench e ágil/hodierna (abordagem low-coding e uso de I.A) para a prototipagem da interface.

### 3.1 ARQUITETURA DE DADOS

Conforme visto nas unidades estudadas e em sala de aula, esta parte do protótipo foi crucial para estabelecer o pilar central do sistema: o banco de dados. Utilizando o SQL Workbench, foi criado a base de dados `clinica_omnichannel`. A centralização dos dados foi garantida pela interligação das seguintes tabelas:

- Pacientes: Contém o identificador do paciente e seu canal “preferido”, essencial para o conceito Omnichannel.
- Atendentes: e gerentes: Tabelas de controle de acesso e perfil.
- Conversas: Tabela principal que registra a fila de atendimento e o histórico, permitindo que a interação seja transferida entre canais sem perda de contexto.

O ambiente de backend foi simulado localmente com o XAMPP, garantindo a conectividade e a execução dos scripts em PHP e SQL.

### 3.2 LÓGICA DE SISTEMA E INTERFACES

Nesta parte do projeto, a lógica do sistema e as interfaces foram desenvolvidas em PHP (Utilizando o VS Code) para gerenciar a autenticação e o acesso à fila de atendimento.

- Painéis de Acesso: Foram criadas interfaces de login distintas, direcionando o usuário para os painéis de Atendente, Gerente e Paciente, conforme a estrutura definida no banco de dados.
- Gestão de Fila: O código em PHP foi responsável por se conectar ao SQL e buscar a lista de atendimentos (conversas), apresentando-a no painel do atendente.
- Reflexão Low-Code: A criação de algumas interfaces e funcionalidades foram auxiliadas por ferramentas de IA e Low-Code.

### 4. RESULTADOS E ANÁLISE CRÍTICA

O Projeto resultou em um protótipo totalmente funcional que valida o conceito de uma base de dados unificada para atendimento e agendamentos.

O principal aprendizado reside na conclusão sobre a metodologia low-code:

Durante o desenvolvimento, percebi que plataformas como Bubble e Base44, embora poderosas, apresentaram desafios na integração com bancos de dados SQL externos e pré-

existentes. A complexidade dessa associação se mostrou um obstáculo que impactou a escalabilidade do protótipo.

Essa vivência demonstrou que, embora o *low-code* acelere a prototipagem de interfaces, a integração com sistemas de *backend* robustos e customizados ainda exige profundo conhecimento em arquitetura e lógica de banco de dados.

## 5. CONCLUSÃO

O protótipo básico do Sistema Omnichannel para Clínica Médica foi concluído com sucesso, atingindo os objetivos de utilizar os conhecimentos de T.I para auxiliar um setor de sertão social.

A estruturação do SQL e do PHP demonstrou a **competência na especificação e implementação de sistemas computacionais**. A análise da metodologia low-code cumpriu o objetivo de **avaliar criticamente ferramentas e tendências hodiernas**, fornecendo uma base sólida para a continuidade de projetos de integração de dados em ambientes empresariais.

## REFERÊNCIA e DOCUMENTOS

1. Código utilizado para a criação da base de dados no SQL Workbench:

```
USE clinica_omnichannel;  
-- Pacientes  
CREATE TABLE pacientes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),  
    email VARCHAR(100),  
    telefone VARCHAR(20),  
    canal_preferido VARCHAR(50)  
);  
-- Atendentes  
CREATE TABLE atendentes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),
```

```

email VARCHAR(100)
);

-- Gerentes

CREATE TABLE gerentes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100),
    email VARCHAR(100)

);

-- Conversas

CREATE TABLE conversas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    paciente_id INT,
    atendente_id INT,
    canal VARCHAR(50),
    mensagem TEXT,
    data_hora DATETIME,
    FOREIGN KEY (paciente_id) REFERENCES pacientes(id),
    FOREIGN KEY (atendente_id) REFERENCES atendentes(id)

);

-- Agendamentos

CREATE TABLE agendamentos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    paciente_id INT,
    medico VARCHAR(100),
    data_hora DATETIME,
    status VARCHAR(50),
    FOREIGN KEY (paciente_id) REFERENCES pacientes(id)

);

```

Através do mesmo e alocação no local host utilizando XAMPP temos esta tabela:

Tabela	Linhas	Tipo	Colação	Tamanho	Sobrecarga
agendamentos	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
atendentes	1	InnoDB	utf8mb4_general_ci	16.0 KB	-
conversas	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
gerentes	1	InnoDB	utf8mb4_general_ci	16.0 KB	-
pacientes	1	InnoDB	utf8mb4_general_ci	16.0 KB	-
<b>5 tabelas</b>	<b>3</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>128.0 KB</b>	<b>0 Bytes</b>

2. Estabelecimento de conexão do banco com php:

```
<?php

$servername = "localhost";
$username = "root";
$password = ""; // Sua senha do MySQL
$dbname = "clinica_omnichannel";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {

    die("Erro na conexão: " . $e->getMessage());
}

?>
```

3. Criação da interface de login com php:

SELECT \* FROM `pacientes`

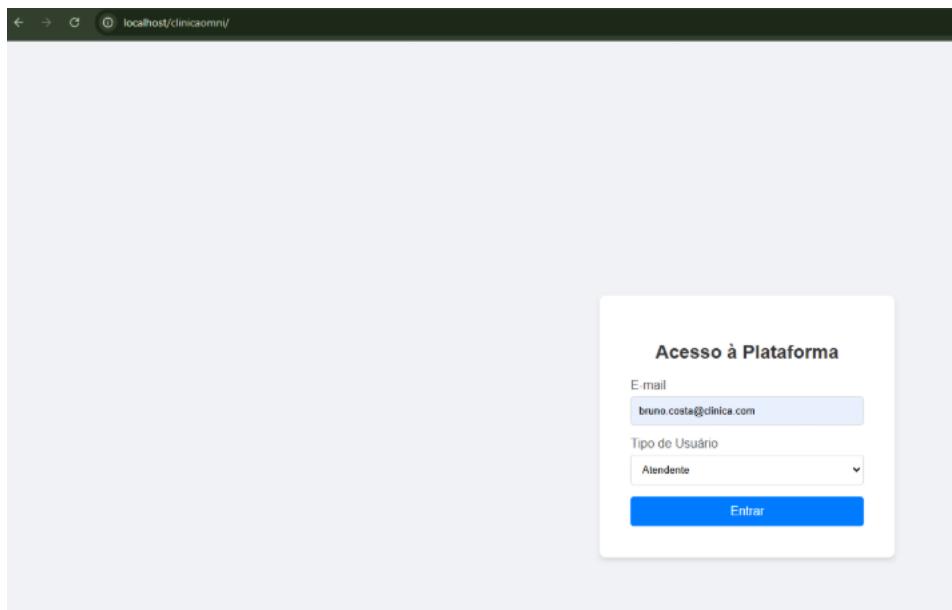
<u>id</u>	<u>nome</u>	<u>email</u>	<u>telefone</u>	<u>canal_preferido</u>
1	Ana silva	ana.silva@email.com	(11) 98765-4321	NULL

SELECT \* FROM `gerentes`

<u>id</u>	<u>nome</u>	<u>email</u>
3	Dr. Carlos Silva	gerente.carlos@clinicax.com

```
SELECT * FROM `atendentes`
```

id	nome	email
2	Bruno Costa	bruno.costa@clinica.com



Nesta tela é possível logar com as credenciais de atendente, paciente ou gerente.

Código da interface:

```
INDEX: <!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Clínica Omnichannel - Login</title>
    <style>
        body { font-family: Arial, sans-serif; background-color: #f0f2f5;
display: flex; justify-content: center; align-items: center; height: 100vh;
margin: 0; }
        .login-container { background-color: #fff; padding: 40px; border-radius: 8px; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); width: 300px; text-align: center; }
        h2 { margin-bottom: 20px; color: #333; }
        .input-group { margin-bottom: 15px; text-align: left; }
        .input-group label { display: block; margin-bottom: 5px; color: #555; }
        .input-group input, .input-group select { width: 100%; padding: 10px;
border: 1px solid #ddd; border-radius: 4px; box-sizing: border-box; }
        button { width: 100%; padding: 10px; background-color: #007bff; border: none; border-radius: 4px; color: #fff; font-size: 16px; cursor: pointer; transition: background-color 0.3s ease; }
```

```

        button:hover { background-color: #0056b3; }
        .error-message { color: red; margin-top: 10px; }
    </style>
</head>
<body>
    <div class="login-container">
        <h2>Acesso à Plataforma</h2>
        <?php
            if (isset($_GET['erro'])) {
                echo '<p class="error-message">E-mail ou tipo de usuário inválidos.</p>';
            }
        ?>
        <form action="processa_login.php" method="POST">
            <div class="input-group">
                <label for="email">E-mail</label>
                <input type="email" id="email" name="email" required>
            </div>
            <div class="input-group">
                <label for="tipo_usuario">Tipo de Usuário</label>
                <select id="tipo_usuario" name="tipo_usuario" required>
                    <option value="">Selecione...</option>
                    <option value="paciente">Paciente</option>
                    <option value="atendente">Atendente</option>
                    <option value="gerente">Gerente</option>
                </select>
            </div>
            <button type="submit">Entrar</button>
        </form>
    </div>
</body>
</html>

```

```

Processamento de login: <?php
session_start();
include("conexao.php"); // Certifica-se que estamos usando a conexão PDO

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = filter_var(trim($_POST['email']), FILTER_SANITIZE_EMAIL);
    $tipo_usuario = $_POST['tipo_usuario'];

    // Mapeamento de tipos de usuário para tabelas no banco de dados com PDO
    $tabelas_usuarios = [
        'paciente'      => ['tabela'    => 'pacientes', 'coluna_id'  => 'id',
        'coluna_nome'   => 'nome'],

```

```
'atendente' => ['tabela' => 'atendentes', 'coluna_id' => 'id',
'coluna_nome' => 'nome'],
'gerente' => ['tabela' => 'gerentes', 'coluna_id' => 'id',
'coluna_nome' => 'nome']
];

if (array_key_exists($tipo_usuario, $tabelas_usuarios)) {
    $config_tabela = $tabelas_usuarios[$tipo_usuario];
    $tabela = $config_tabela['tabela'];
    $coluna_id = $config_tabela['coluna_id'];
    $coluna_nome = $config_tabela['coluna_nome'];

    // Consulta usando PDO
    $sql = "SELECT {$coluna_id}, {$coluna_nome}, email FROM {$tabela} WHERE
email = :email";

    try {
        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':email', $email, PDO::PARAM_STR);
        $stmt->execute();
        $usuario = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($usuario) {
            // Define as variáveis de sessão com os nomes corretos para o
            dashboard.php
            $_SESSION['logado'] = true;
            $_SESSION['id_usuario'] = $usuario[$coluna_id];
            $_SESSION['nome_usuario'] = $usuario[$coluna_nome];
            $_SESSION['email_usuario'] = $usuario['email'];
            $_SESSION['tipo_usuario'] = $tipo_usuario;

            header("Location: dashboard.php");
            exit();
        } else {
            // Usuário não encontrado na tabela especificada
            header("Location: index.php?erro=1");
            exit();
        }
    } catch(PDOException $e) {
        // Logar o erro para debug e mostrar mensagem genérica ao usuário
        error_log("Erro ao processar login: " . $e->getMessage());
        header("Location: index.php?erro=2"); // Erro genérico
        exit();
    }
} else {
    // Tipo de usuário inválido
    header("Location: index.php?erro=1");
}
```

```

        exit();
    }
} else {
    // Se não for POST, redireciona para a página de login
    header("Location: index.php");
    exit();
}
?>

```

#### 4. Criação da interface de DASHBOARD para os 3 tipos de usuários:

```

<?php
session_start();

if (!isset($_SESSION['logado']) || $_SESSION['logado'] !== true) {
    header("Location: index.php");
    exit();
}

$nome_usuario = $_SESSION['nome_usuario'];
$tipo_usuario = $_SESSION['tipo_usuario'];
$id_usuario = $_SESSION['id_usuario'];

include("conexao.php");
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard - Clínica Omnichannel</title>
    <style>
        body { font-family: Arial, sans-serif; background-color: #f0f2f5;
margin: 0; padding: 20px; }
        .header { display: flex; justify-content: space-between; align-items: center; background-color: #fff; padding: 10px 20px; border-radius: 8px; box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); margin-bottom: 20px; }
        .header .user-info { font-size: 18px; color: #333; }
        .header .user-info span { font-weight: bold; }
        .header .logout-btn { background-color: #dc3545; color: #fff; border: none; padding: 8px 15px; border-radius: 4px; text-decoration: none; transition: background-color 0.3s ease; }
        .header .logout-btn:hover { background-color: #c82333; }
        .dashboard-container { background-color: #fff; padding: 20px; border-radius: 8px; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); }

```

```

.dashboard-container h2 { margin-bottom: 20px; color: #333; }
.card { border: 1px solid #ddd; border-radius: 8px; padding: 15px;
margin-bottom: 15px; box-shadow: 0 1px 3px rgba(0,0,0,0.1); }
.card h3 { margin-top: 0; color: #007bff; }
.data-table { width: 100%; border-collapse: collapse; margin-top:
10px; }
.data-table th, .data-table td { padding: 8px; text-align: left;
border-bottom: 1px solid #ddd; }
.data-table th { background-color: #f2f2f2; }
</style>
</head>
<body>
<div class="header">
<div class="user-info">
    Bem-vindo(a), <span><?php echo htmlspecialchars($nome_usuario);
?></span>!
    Seu tipo de usuário: <span><?php echo
htmlspecialchars ucfirst($tipo_usuario)); ?></span>
</div>
    <a href="logout.php" class="logout-btn">Sair</a>
</div>

<div class="dashboard-container">
<?php
    // Lógica para renderizar o dashboard com base no tipo de usuário
    if ($tipo_usuario === 'paciente') {
        echo "<h2>Painel do Paciente</h2>";
        echo "<p>Bem-vindo(a), Sr(a). " .
htmlspecialchars($nome_usuario) . "!</p>";

        echo '<div class="card">';
        echo '<h3>Histórico de Conversas</h3>';
        try {
            $sql_conversas = "SELECT c.canal, c.mensagem, c.data_hora,
a.nome as nome_atendente FROM conversas c LEFT JOIN atendentes a ON
c.atendente_id = a.id WHERE c.paciente_id = :paciente_id ORDER BY c.data_hora
DESC";
            $stmt_conversas = $conn->prepare($sql_conversas);
            $stmt_conversas->bindParam(':paciente_id', $id_usuario,
PDO::PARAM_INT);
            $stmt_conversas->execute();
            $conversas = $stmt_conversas->fetchAll(PDO::FETCH_ASSOC);

            if ($conversas) {
                echo "<ul>";
                foreach ($conversas as $conversa) {

```

```

        $atendente_nome = $conversa['nome_atendente'] ? "
- Atendente: " . htmlspecialchars($conversa['nome_atendente']) : "";
                echo "<li><strong>Canal:</strong> " .
htmlspecialchars($conversa['canal']) . " - <strong>Mensagem:</strong> " .
nl2br(htmlspecialchars($conversa['mensagem'])) . " (<strong>" . date('d/m/Y
H:i', strtotime($conversa['data_hora'])) . "</strong>)" . $atendente_nome .
"</li>";
            }
            echo "</ul>";
        } else {
            echo "<p>Nenhuma conversa registrada ainda.</p>";
        }
    } catch(PDOException $e) {
        error_log("Erro ao buscar conversas do paciente: " . $e-
>getMessage());
        echo "<p>Erro ao carregar histórico de conversas.</p>";
    }
    echo '</div>';

    echo '<div class="card">';
    echo '<h3>Próximos Agendamentos e Lembretes</h3>';
    try {
        $sql_agendamentos = "SELECT medico, data_hora, status FROM
agendamentos WHERE paciente_id = :paciente_id ORDER BY data_hora ASC";
        $stmt_agendamentos = $conn->prepare($sql_agendamentos);
        $stmt_agendamentos->bindParam(':paciente_id', $id_usuario,
PDO::PARAM_INT);
        $stmt_agendamentos->execute();
        $agendamentos = $stmt_agendamentos-
>fetchAll(PDO::FETCH_ASSOC);

        if ($agendamentos) {
            echo "<ul>";
            foreach ($agendamentos as $agendamento) {
                echo "<li><strong>Médico:</strong> " .
htmlspecialchars($agendamento['medico']) . " - <strong>Data/Hora:</strong> " .
date('d/m/Y H:i', strtotime($agendamento['data_hora'])) . " -
<strong>Status:</strong> " . htmlspecialchars($agendamento['status']) .
"</li>";
            }
            echo "</ul>";
        } else {
            echo "<p>Nenhum agendamento realizado ainda.</p>";
        }
    } catch(PDOException $e) {
        error_log("Erro ao buscar agendamentos do paciente: " . $e-
>getMessage()));
    }
}

```

```

        echo "<p>Erro ao carregar agendamentos.</p>";
    }
    echo '</div>';

    echo '<div class="card">';
    echo '<h3>Simular uma Mensagem Omnichannel</h3>';
    echo '<p>Este é um protótipo, mas aqui você poderia enviar uma
mensagem que apareceria no painel do atendente.</p>';
    echo '<form action="enviar_mensagem.php" method="POST">';
    echo '<div class="input-group">';
    echo '<label for="canal">Canal de Comunicação</label>';
    echo '<select id="canal" name="canal" required>';
    echo '<option value="WhatsApp">WhatsApp</option>';
    echo '<option value="Email">E-mail</option>';
    echo '<option value="Chat">Chat do Site</option>';
    echo '<option value="Instagram">Instagram Direct</option>';
    echo '<option value="Facebook">Facebook Messenger</option>';
    echo '</select>';
    echo '</div>';
    echo '<div class="input-group">';
    echo '<label for="mensagem">Mensagem</label>';
    echo '<textarea id="mensagem" name="mensagem" rows="4"
required></textarea>';
    echo '</div>';
    echo '<button type="submit">Enviar Mensagem</button>';
    echo '</form>';
    echo '</div>';

} elseif ($tipo_usuario === 'atendente') {
    echo "<h2>Painel do Atendente</h2>";
    echo "<p>Bem-vindo(a), " . htmlspecialchars($nome_usuario) .
"! Gerencie as conversas com os pacientes.</p>";

    echo '<div class="card">';
    echo '<h3>Caixa de Entrada Unificada</h3>';
    echo '<p>Aqui você visualiza e responde a todas as
solicitações, de todos os canais.</p>';
    try {
        $sql_conversas_atendente = "SELECT p.nome as
nome_paciente, c.canal, c.mensagem, c.data_hora, c.atendente_id, c.id AS
conversa_id FROM conversas c JOIN pacientes p ON c.paciente_id = p.id WHERE
c.atendente_id IS NULL OR c.atendente_id = :atendente_id ORDER BY c.data_hora
DESC LIMIT 10";
        $stmt_conversas_atendente = $conn-
>prepare($sql_conversas_atendente);
        $stmt_conversas_atendente->bindParam(':atendente_id',
$id_usuario, PDO::PARAM_INT);

```

```

$stmt_conversas_atendente->execute();
$conversas_atendente = $stmt_conversas_atendente-
>fetchAll(PDO::FETCH_ASSOC);

if ($conversas_atendente) {
    echo '<ul>';
    foreach ($conversas_atendente as $conversa) {
        $atribuido_a = $conversa['atendente_id'] ? "Você
está atendendo esta conversa." : "<b>Pendente na Fila de Atendimento.</b>";
        echo "<li><strong>Paciente:</strong> " .
        htmlspecialchars($conversa['nome_paciente']) . " (Canal: " .
        htmlspecialchars($conversa['canal']) . ")<br><strong>Última Mensagem:</strong>
" . nl2br(htmlspecialchars($conversa['mensagem'])) . " (<i>" . date('d/m/Y
H:i', strtotime($conversa['data_hora'])) . "</i>) - " . $atribuido_a .
"</li>";
    }
    echo '</ul>';
} else {
    echo "<p>Nenhuma conversa recente para exibir.</p>";
}
} catch(PDOException $e) {
    error_log("Erro ao buscar conversas para atendente: " .
$e->getMessage());
    echo "<p>Erro ao carregar conversas recentes.</p>";
}
echo '</div>';

echo '<div class="card">';
echo '<h3>Interações (Simulação)</h3>';
echo '<p>Ações como "agendar consulta" ou "ver prontuário"
seriam acessadas por aqui, integrando com outros sistemas da clínica.</p>';
echo '<button>Acessar Agenda Médica</button> <button>Ver
Prontuário do Paciente</button>';
echo '</div>';

} elseif ($tipo_usuario === 'gerente') {
    echo "<h2>Painel do Gerente</h2>";
    echo "<p>Bem-vindo(a), " . htmlspecialchars($nome_usuario) .
"! Aqui você pode monitorar o desempenho da equipe e ter acesso a relatórios
consolidados.</p>";

    echo '<div class="card">';
    echo '<h3>Dashboard de Desempenho</h3>';
    try {
        $stmt_total_conv = $conn->query("SELECT COUNT(*) AS total
FROM conversas");
        $total_conv = $stmt_total_conv->fetchColumn();
    }
}

```

```

        $stmt_conv_pendentes = $conn->query("SELECT COUNT(*) AS
total FROM conversas WHERE atendente_id IS NULL");
        $total_pendentes = $stmt_conv_pendentes->fetchColumn();
        $stmt_total_agend = $conn->query("SELECT COUNT(*) AS total
FROM agendamentos");
        $total_agend = $stmt_total_agend->fetchColumn();

        $sql_desempenho_atendentes = "SELECT a.nome, COUNT(c.id)
AS total_atendimentos FROM atendentes a LEFT JOIN conversas c ON a.id =
c.atendente_id GROUP BY a.nome ORDER BY total_atendimentos DESC";
        $stmt_desempenho = $conn-
>prepare($sql_desempenho_atendentes);
        $stmt_desempenho->execute();
        $desempenho_atendentes = $stmt_desempenho-
>fetchAll(PDO::FETCH_ASSOC);

        echo "<p><strong>Total de Conversas Registradas:</strong>
" . $total_conv . "</p>";
        echo "<p><strong>Conversas Pendentes na Fila:</strong> " .
$total_pendentes . "</p>";
        echo "<p><strong>Total de Agendamentos
Realizados:</strong> " . $total_agend . "</p>";

        if ($desempenho_atendentes) {
            echo "<h4>Produtividade por Atendente:</h4>";
            echo '<table class="data-
table"><thead><tr><th>Atendente</th><th>Atendimentos
Finalizados</th></tr></thead><tbody>';
            foreach ($desempenho_atendentes as $item) {
                echo "<tr><td>" . htmlspecialchars($item['nome']) .
" </td><td>" . $item['total_atendimentos'] . "</td></tr>";
            }
            echo '</tbody></table>';
        }
    } catch(PDOException $e) {
        error_log("Erro ao buscar métricas para gerente: " . $e-
>getMessage());
        echo "<p>Erro ao carregar relatórios.</p>";
    }
    echo '</div>';

    echo '<div class="card">';
    echo '<h3>Supervisão e Gestão de Filas</h3>';
    try {
        $sql_fila = "SELECT p.nome as nome_paciente, c.canal,
c.mensagem, c.data_hora, c.atendente_id, c.id AS conversa_id FROM conversas c

```

```

JOIN pacientes p ON c.paciente_id = p.id WHERE c.atendente_id IS NULL ORDER BY
c.data_hora ASC LIMIT 5";
        $stmt_fila = $conn->prepare($sql_fila);
        $stmt_fila->execute();
        $fila_atendimento = $stmt_fila-
>fetchAll(PDO::FETCH_ASSOC);

        if ($fila_atendimento) {
            echo "<h4>Conversas Pendentes na Fila:</h4>";
            echo '<table class="data-
table"><thead><tr><th>Paciente</th><th>Canal</th><th>Mensagem</th><th>Data/Hor
a</th><th>Ação</th></tr></thead><tbody>';
            foreach ($fila_atendimento as $item) {
                echo "<tr><td>" .
htmlspecialchars($item['nome_paciente']) . "</td><td>" .
htmlspecialchars($item['canal']) . "</td><td>" .
nl2br(htmlspecialchars($item['mensagem'])) . "</td><td>" . date('d/m/Y H:i',
strtotime($item['data_hora']));
                "</td><td><button>Redirecionar</button></td></tr>";
            }
            echo '</tbody></table>';
        } else {
            echo "<p>Não há conversas pendentes no momento.</p>";
        }
    } catch(PDOException $e) {
        error_log("Erro ao buscar fila de atendimento: " . $e-
>getMessage());
        echo "<p>Erro ao carregar a fila de atendimento.</p>";
    }
    echo '</div>';

} else {
    echo "<p>Erro: Tipo de usuário não reconhecido.</p>";
}
?>
</div>
</body>
</html>

```

## Painel do Atendente:

Bem-vindo(a), **Bruno Costa**! Seu tipo de usuário: **Atendente** Sair

**Painel do Atendente**  
Bem-vindo(a), Bruno Costa! Gerencie as conversas com os pacientes.

**Caixa de Entrada Unificada**  
Aqui você visualiza e responde a todas as solicitações, de todos os canais.  
Nenhuma conversa recente para exibir.

**Integrações (Simulação)**  
Ações como "agendar consulta" ou "ver prontuário" serão acessadas por aqui, integrando com outros sistemas da clínica.

[Acessar Agenda Médica](#) [Ver Prontuário do Paciente](#)

## Painel do Paciente:

Bem-vindo(a), **Ana Silva**! Seu tipo de usuário: **Paciente** Sair

**Painel do Paciente**  
Bem-vindo(a), Sr(a). Ana Silva!

**Histórico de Conversas**  
Nenhuma conversa registrada ainda.

**Próximos Agendamentos e Lembretes**  
Nenhum agendamento realizado ainda.

**Simular uma Mensagem Omnichannel**  
Este é um protótipo, mas aqui você poderá enviar uma mensagem que apareceria no painel do atendente.

Canal de Comunicação WhatsApp WhatsApp

Mensagem:   
[Enviar Mensagem](#)

## Painel do Gerente:

Bem-vindo(a), **Dr. Carlos Silva**! Seu tipo de usuário: **Gerente** Sair

**Painel do Gerente**  
Bem-vindo(a), Dr. Carlos Silva! Aqui você pode monitorar o desempenho da equipe e ter acesso a relatórios consolidados.

**Dashboard de Desempenho**

Total de Conversas Registradas: 0	Conversas Pendentes na Fila: 0	Total de Agendamentos Realizados: 0
Produtividade por Atendente:		
Atendente	Atendimentos Finalizados	
Bruno Costa	0	

**Supervisão e Gestão de Fila**  
Não há conversas pendentes no momento.

## 5. Criação da função de logout

```
<?php
```

```
session_start();
session_unset();
session_destroy();

header("Location: index.php");
exit();
?>
```