

Funcionamento do Mapa Cultural de Ananindeua



Imagem 1.1 do protótipo do Mapa Cultural de Ananindeua.

O Mapa Cultural de Ananindeua foi concebido como uma plataforma digital robusta, implementada sobre uma arquitetura *full-stack* que utiliza um *stack* de tecnologias consolidadas no mercado. O sistema explora a sinergia entre o **PHP** no *backend* para o processamento das regras de negócio e a manipulação dos dados, e um *frontend* reativo construído com **HTML5**, **CSS3** e **JavaScript** para entregar uma experiência de usuário (UX) fluida e interativa. A persistência dos dados é garantida por um banco de dados **MySQL**, cujo modelo relacional foi meticulosamente desenhado para suportar a complexidade das entidades culturais do município.

```
// incluir conexão com banco
require_once 'config/database.php';

// variáveis para mensagens
$erro = null;
$mensagem = null;

// verificar se já está logado
if (isset($_SESSION['usuario_id'])) {
    if ($_SESSION['usuario_tipo'] == 'admin') {
        header(header: 'Location: /admin/index.php');
        exit();
    } else {
        header(header: 'Location: /minhaconta.php');
        exit();
    }
}

// Processar o login
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Sanitização moderna do input
    $login = trim(string: htmlspecialchars($_POST['login'], flags: ENT_QUOTES, encoding: 'UTF-8'));
    $senha = $_POST['senha'];

    // Remove todos os caracteres não numéricos para comparar CPF
    $senhas_numeros = preg_replace(pattern: '/[^0-9]/', replacement: '', subject: $login);

    // Se precise ser um CPF (tem 3 dígitos)
    if (strlen(string: $senhas_numeros) == 11) {
        // Formatar CPF com pontos e traço
        $cpf_formatado = substr(string: $senhas_numeros, offset: 0, length: 3);
        substr(string: $senhas_numeros, offset: 3, length: 3);
        substr(string: $senhas_numeros, offset: 6, length: 3);
        substr(string: $senhas_numeros, offset: 9, length: 2);
    }
}
```

Imagem 2.1 da validação de login do sistema do Mapa Cultural.

```
$usuario = $pdo->fetch(PDO::FETCH_ASSOC);

if ($usuario && $senha == $usuario['senha']) {
    $_SESSION['usuario_id'] = $usuario['id'];
    $_SESSION['usuario_nome'] = $usuario['nome_completo'];
    $_SESSION['usuario_tipo'] = $usuario['tipo_usuario'];

    // Log para debug
    error_log(message: 'Login bem sucedido para usuário: ' . $usuario['nome_completo'],
    error_log(message: 'Tipo de usuário: ' . $usuario['tipo_usuario']);

    if ($usuario['tipo_usuario'] == 'admin') {
        error_log(message: 'Redirecionando para área admin');
        header(header: 'Location: /admin/index.php');
    } else {
        error_log(message: 'Redirecionando para minha conta');
        header(header: 'Location: /minhaconta.php');
    }
    exit();
} else {
    error_log(message: 'Falha no login para: ' . $login);
    $erro = 'login ou senha inválidos';
}

// incluir o header depois da processamento
require_once 'includes/header.php';
```

Imagem 2.2 da validação de login do sistema do Mapa Cultural.

A seguir, detalhamos a engenharia por trás de cada módulo principal do sistema.

Portal de Acesso e Onboarding (Página Inicial)

O entypoint da aplicação é um portal de engajamento dinâmico, projetado para maximizar a usabilidade e facilitar o processo de onboarding do usuário. A página serve como um hub de informações central, onde o conteúdo estático, como tutoriais e resumos institucionais, é renderizado com alta performance. A arquitetura de informação foi estrategicamente planejada para direcionar o fluxo de usuários aos módulos-chave da plataforma — Eventos, Espaços, Agentes e Oportunidades. Além disso, a página serve como uma vitrine de conhecimento, provendo acesso rápido a um knowledge base sobre as principais legislações de fomento à cultura, como as Leis Aldir Blanc e Paulo Gustavo, consolidando a plataforma como uma referência para a comunidade cultural.



Imagem 3.1 da estruturação da página inicial.

Módulo de Gerenciamento de Agentes Culturais (Página de Agentes)

Este módulo representa o núcleo da rede social da plataforma, implementando funcionalidades de CRUD (Create, Read, Update, Delete) para os perfis dos agentes culturais. O *backend* foi arquitetado para suportar um modelo de dados polimórfico,

permitindo o registro de diferentes tipos de entidades (artistas individuais, coletivos, bandas, instituições) sob uma estrutura coesa. A interface de usuário disponibiliza duas formas de visualização de dados, *List View* e *Map View*, que são renderizações distintas do mesmo dataset. A funcionalidade de busca é suportada por um *engine* de filtragem facetada, que constrói dinamicamente *queries* SQL, otimizadas para consultar o banco de dados com base em múltiplos parâmetros, garantindo respostas rápidas e precisas. A visualização em mapa utiliza dados georreferenciados, processados via JavaScript para plotar os *pins* que indicam a localização de cada agente, transformando dados brutos em inteligência geográfica.

```

function agentes() {
    require_once 'config/database.php';

    // Construção da consulta base
    $sql = "
        SELECT DISTINCT a.*
        FROM usuarios a
        LEFT JOIN usuarios_areas aa ON a.id = aa.usuario_id
        LEFT JOIN areas_estacoes ae ON aa.area_id = ae.id
        LEFT JOIN usuarios_segmentos as ON a.id = as.usuario_id
        LEFT JOIN segmentos_culturais sc ON as.segmento_id = s.id
        LEFT JOIN usuarios_funcoes uf ON a.id = uf.usuario_id
        LEFT JOIN funcoes_culturais fc ON uf.funcao_id = f.id
        WHERE a.tipo_usuario = 'agente'";

    $query = $db->query($sql);
    $agents = [];

    // Aplicar filtros
    if (!empty($_GET['search'])) {
        $sql .= " AND (a.nome_completo LIKE ? OR a.mini_bio LIKE ?)";
        $searchTerm = "%" . $_GET['search'] . "%";
        $params[] = $searchTerm;
        $params[] = $searchTerm;
    }

    if (!empty($_GET['area'])) {
        $sql .= " AND aa.area_id = ?";
        $params[] = $_GET['area'];
    }

    $query = $db->query($sql, $params);
    while ($row = $query->fetch_assoc()) {
        $agents[] = $row;
    }
}

```

Imagem 4.1 da consulta de agentes com e sem filtros.

Perfil Detalhado do Agente Cultural

A página de perfil é um template dinâmico que renderiza as informações de um agente específico, recuperadas do banco de dados através de um identificador único. A riqueza de dados exibida — desde informações textuais como "Sobre" e "Áreas de Atuação" até múltiplos uploads — demonstra a complexidade e a robustez do esquema do banco de dados relacional. A seção de galerias funciona como um Repositório de Mídia Digital, com o backend em PHP gerenciando o upload e o

armazenamento de diversos formatos de arquivo (MIME types), como imagens, vídeos e documentos. A integração com redes sociais evidencia a capacidade do sistema de se conectar a plataformas externas.

```
require_once 'config/database.php';

// Verificar se foi fornecido um ID
if (!isset($_GET['id'])) {
    header(header: 'location: agentes.php');
    exit();
}

// Buscar informações do agente
$stmt = $conn->prepare(query: "
    SELECT u.*, GROUP_CONCAT(DISTINCT a.nome) as areas_atuacao,
           GROUP_CONCAT(DISTINCT s.nome) as segmentos_culturais,
           GROUP_CONCAT(DISTINCT f.nome) as funcoes_culturais
    FROM usuarios u
    LEFT JOIN usuario_areas ua ON u.id = ua.usuario_id
    LEFT JOIN areas_atuacao a ON ua.area_id = a.id
    LEFT JOIN usuario_segmentos us ON u.id = us.usuario_id
    LEFT JOIN segmentos_culturais s ON us.segmento_id = s.id
    LEFT JOIN usuario_funcoes uf ON u.id = uf.usuario_id
    LEFT JOIN funcoes_culturais f ON uf.funcao_id = f.id
    WHERE u.id = ?
    GROUP BY u.id
");
$stmt->execute(params: [$_GET['id']]);
$agente = $stmt->fetch(mode: PDO::FETCH_ASSOC);

// Registrar visualização
if ($agente) {
    // Incremental visualizações
    $stmt = $conn->prepare(query: "
        UPDATE usuarios
        SET visualizacoes = visualizacoes + 1
        WHERE id = ?
    ");
}
```

Imagem 5.1 da busca pelo perfil do agente cultural na base de dados do sistema.

Módulo de Difusão de Eventos (Eventos Secult)

Este componente atua como a agenda cultural oficial da Secretaria de Cultura. O módulo emprega um motor de busca avançado, permitindo que os usuários executem consultas com filtros parametrizados para refinar os resultados. Assim como no módulo de Agentes, os eventos são apresentados em uma listagem paginada para otimizar o carregamento e em um mapa interativo com pins geolocalizados, oferecendo uma visualização espacial e em tempo real da atividade cultural no município. O sistema filtra e exibe apenas eventos "ativos", uma lógica de negócio processada no backend que compara as datas dos eventos com a data atual do servidor.

```

session_start();
require_once 'config/database.php';

// Preparar a consulta base
$sql = "
    SELECT e.*, u.nome_completo as criador
    FROM eventos e
    LEFT JOIN usuarios u ON e.usuario_id = u.id
    WHERE e.status = 'ativo'
";

$params = [];

// Aplicar filtros
if (!empty($_GET['search'])) {
    $sql .= " AND (e.titulo LIKE ? OR e.descricao LIKE ? OR e.local LIKE ?)";
    $searchTerm = "%" . $_GET['search'] . "%";
    $params[] = $searchTerm;
    $params[] = $searchTerm;
    $params[] = $searchTerm;
}

// Filtro por data
if (!empty($_GET['data'])) {
    switch ($_GET['data']) {
        case 'hoje':
            $sql .= " AND DATE(e.data_inicio) = CURDATE()";
            break;
        case 'semana':
            $sql .= " AND e.data_inicio BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 7 DAY)";
            break;
        case 'mes':
            $sql .= " AND e.data_inicio BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 1 MONTH)";
            break;
        case 'passados':
            // ...
    }
}

```

Imagem 6.1 da consulta por eventos da Secult com e sem filtros.

Página de Detalhe do Evento

Cada evento possui uma página de detalhe individual, populada dinamicamente a partir de uma chamada ao banco de dados. A interface exibe dados estruturados e sensíveis ao tempo, como datas e horários de início e término, que exigem um tratamento preciso de tipos de dados DATETIME ou TIMESTAMP no MySQL. A funcionalidade de compartilhamento social é implementada através de um mecanismo em JavaScript que interage com as APIs de redes sociais, ampliando o alcance e o engajamento de cada evento. O mapa embarcado renderiza as coordenadas geográficas específicas do local do evento, provendo uma utilidade prática e imediata ao usuário.

```

session_start();
require_once 'config/database.php';

if (!isset($_GET['id'])) {
    header('Location: eventos.php');
    exit;
}

// Busca de dados
$stmt = $conn->prepare("
    SELECT e.*, u.nome_completo as criador
    FROM eventos e
    LEFT JOIN usuarios u ON e.usuario_id = u.id
    WHERE e.id = ? AND e.status = 'ativo'
");

$stmt->execute([$_GET['id']]);
$evento = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$evento) {
    header('Location: eventos.php');
    exit;
}

require_once 'includes/header.php';
?>

<!-- Inclui o CSS e JS -->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>

<style>
#mapa-evento-detalle {
    height: 400px;
    width: 100%;
}

```

Imagem 7.1 da busca pelo evento específico na base de dados.

Módulo de Gerenciamento de Espaços Culturais

Este módulo transforma a plataforma em um ecossistema colaborativo, permitindo que os próprios usuários cadastrem e gerenciem os espaços culturais. A funcionalidade de "busca combinada" é um diferencial técnico, indicando uma lógica de backend sofisticada, capaz de construir consultas SQL complexas com múltiplas cláusulas JOIN e WHERE para cruzar diferentes critérios de pesquisa. A arquitetura de front-end, mais uma vez, demonstra sua flexibilidade ao oferecer as visualizações em lista e mapa georreferenciado para o mesmo conjunto de dados.

```

session_start();
require_once 'config/database.php';

// Preparar a consulta base
$sql = "
    SELECT e.id, e.nome, e.descricao, e.imagem_capa, e.data_cadastro,
           e.latitude, e.longitude, e.logradouro, e.numero, e.complemento,
           e.bairro, e.cidade, e.estado, e.acessibilidade, e.status,
           u.usuario_id, u.nome_completo as nome_responsavel
    FROM espacos_publicos e
    LEFT JOIN usuarios u ON e.usuario_id = u.id
    WHERE e.status = 'ativo'
";

$params = [];

// Aplicar filtros
if (!empty($_GET['search'])) {
    $sql .= " AND (e.nome LIKE ? OR e.descricao LIKE ?)";
    $searchTerm = "%" . $_GET['search'] . "%";
    $params[] = $searchTerm;
    $params[] = $searchTerm;
}

// Filtro por bairro
if (!empty($_GET['bairro'])) {
    $sql .= " AND e.bairro = ?";
    $params[] = $_GET['bairro'];
}

// Filtro por acessibilidade
if (!empty($_GET['acessibilidade'])) {
    if ($_GET['acessibilidade'] == 'sim') {
        $sql .= " AND e.acessibilidade IS NOT NULL AND e.acessibilidade != ''";
    } else {

```

Imagem 8.1 da consulta por espaços culturais com e sem filtros.

Perfil Detalhado do Espaço Cultural

A página de perfil de um espaço é um exemplo primoroso de modelagem de dados. Ela apresenta um vasto conjunto de atributos, como tipo de espaço, características de infraestrutura, horários de funcionamento e contatos, refletindo um esquema de banco de dados bem normalizado e projetado para capturar informações heterogêneas. O componente de galeria é reutilizado aqui, funcionando como um repositório de ativos digitais que compõem um portfólio multimídia completo para cada espaço cadastrado.

```

session_start();
require_once 'config/database.php';

// Verificar se foi fornecido um ID
if (!isset($_GET['id'])) {
    header('Location: espacos.php');
    exit();
}

// Buscar informações do espaço
$stmt = $conn->prepare(query: "
    SELECT e.*, u.nome_completo as nome_responsavel, u.nome_socio as nome_socio_responsavel
    FROM espacos_publicos e
    LEFT JOIN usuarios u ON e.usuario_id = u.id
    WHERE e.id = ?
");
$stmt->execute(array($_GET['id']));
$espaco = $stmt->fetch(PDO::FETCH_ASSOC);

// Se o espaço não for encontrado, redirecionar
if (!$espaco) {
    header('Location: espacos.php');
    exit();
}

// Registrar visualização
if (!isset($_SESSION['visualizacoes_espaco'])) {
    $_SESSION['visualizacoes_espaco'] = [];
}

// Verifica se o usuário já visualizou este espaço nas últimas 24 horas
$agora = time();
$ultimo_visualizacao = isset($_SESSION['visualizacoes_espaco'][$espaco['id']])
    ? $_SESSION['visualizacoes_espaco'][$espaco['id']]

```

Imagem 9.1 da busca pelo espaço cultural específico na base de dados.

Feed de Notícias (CMS - Listagem)

Este módulo constitui um sistema de gerenciamento de conteúdo (CMS) integrado à plataforma. A listagem de notícias é um feed dinâmico, onde o backend em PHP consulta a tabela de notícias no MySQL, ordenando os resultados por data de publicação em ordem decrescente (ORDER BY publication_date DESC). Para garantir a performance e a escalabilidade, a listagem é implementada com um sistema de paginação, que fraciona a carga de dados entregue ao frontend a cada requisição.

```

session_start();
require_once 'config/database.php';

// Preparar a consulta base
$sql = '
SELECT n.*, u.nome_completo as autor
FROM noticias n
LEFT JOIN usuarios u ON n.usuario_id = u.id
WHERE n.status = 'ativo'
';
$noticias = [];

// Aplicar filtros
if (!empty($_GET['search'])) {
    $sql .= ' AND (n.titulo LIKE ? OR n.subtitulo LIKE ? OR n.conteudo LIKE ?)';
    $searchTerm = "%" . $_GET['search'] . "%";
    $noticias[] = $searchTerm;
    $noticias[] = $searchTerm;
    $noticias[] = $searchTerm;
}

// Filtro por data
if (!empty($_GET['data'])) {
    switch ($_GET['data']) {
        case 'hoje':
            $sql .= ' AND DATE(n.data_publicacao) = CURDATE()';
            break;
        case 'semana':
            $sql .= ' AND n.data_publicacao BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 7 DAY)';
            break;
        case 'mes':
            $sql .= ' AND n.data_publicacao BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 1 MONTH)';
            break;
        case 'passados':
    }
}

```

Imagem 10.1 da consulta por notícias com e sem filtros.

Página de Detalhe da Notícia

Esta é a view final do CMS, responsável pela renderização de um artigo individual. O conteúdo textual, armazenado em campos do tipo TEXT ou LONGTEXT no banco de dados, é apresentado de forma estruturada. O template exibe metadados cruciais como autor e data de publicação, e reincorpora o componente de compartilhamento social em JavaScript, mantendo a consistência da experiência do usuário através da plataforma.

```

session_start();
require_once 'config/database.php';

// Verificar se o ID da notícia foi fornecido
if (!isset($_GET['id'])) {
    header(header: 'location: noticias.php');
    exit();
}

// Buscar a notícia
$stmt = $conn->prepare("
    SELECT n.*, u.nome_completo as autor
    FROM noticias n
    LEFT JOIN usuarios u ON n.usuario_id = u.id
    WHERE n.id = ? AND n.status = 'ativo'
");
$stmt->execute(array($_GET['id']));
$noticia = $stmt->fetch(PDO::FETCH_ASSOC);

// Se a notícia não existir ou estiver inativa, redirecionar
if (!$noticia) {
    header(header: 'location: noticias.php');
    exit();
}

require_once 'includes/header.php';
??

<div class="container py-5">
    <nav aria-label="breadcrumb" class="mb-4">
        <ol class="breadcrumb">
            <li class="breadcrumb-item"><a href="noticias.php">Notícias</a></li>
            <li class="breadcrumb-item active" aria-current="page">
                <?php echo htmlspecialchars(string: $noticia['titulo']); ?>
            </li>
        </ol>
    </nav>

```

Imagem 11.1 da busca pela notícia específica na base de dados.

Módulo de Oportunidades (Listagem)

Funcionando como uma central de editais e chamadas públicas, este módulo é vital para o desenvolvimento profissional da comunidade artística. O sistema automatiza a curadoria de conteúdo ao exibir apenas as oportunidades "ativas". Essa lógica é executada no backend, que filtra os registros com base no campo "prazo da oportunidade", garantindo que os usuários visualizem apenas informações relevantes e acionáveis. A sua função é democratizar e centralizar o acesso a informações que são, por natureza, descentralizadas.

```
session_start();
require_once 'config/database.php';

// Preparar a consulta base
$sql = "
    SELECT o.*, u.nome_completo AS criador
    FROM oportunidades o
    LEFT JOIN usuarios u ON o.usuario_id = u.id
    WHERE o.status = 'ativo'
";

$params = [];

// Adicionar filtros
if (!empty($_GET['search'])) {
    $sql .= " AND (o.titulo LIKE ? OR o.descricao LIKE ? OR o.tipo LIKE ?) ";
    $searchTerm = "%" . $_GET['search'] . "%";
    $params[] = $searchTerm;
    $params[] = $searchTerm;
    $params[] = $searchTerm;
}

// Filtro por tipo de oportunidade
if (!empty($_GET['tipo'])) {
    $sql .= " AND o.tipo = ? ";
    $params[] = $_GET['tipo'];
}

$sql .= " ORDER BY o.data_cadastro DESC ";

// Executar a consulta
$stmt = $conn->prepare($sql);
$stmt->execute($params);
$oportunidades = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Imagem 12.1 da consulta por oportunidades com e sem filtros.

Página de Detalhe da Oportunidade

A página de detalhe da oportunidade exibe um conjunto de dados altamente estruturado, incluindo descrição, requisitos, prazos e links externos para inscrição. A gestão de prazos é um requisito técnico crítico, e o sistema o manipula com precisão. A presença de um link de inscrição externo posiciona o Mapa Cultural como uma plataforma hub, que não apenas provê informação, mas também conecta seus usuários ao ecossistema cultural mais amplo, direcionando o tráfego de forma inteligente.

```

session_start();
require_once 'config/database.php';

if (!isset($_GET['id'])) {
    header('Location: oportunidades.php');
    exit;
}

// Buscar oportunidade
$stmt = $conn->prepare('SELECT o.*, u.nome_completo as criador
FROM oportunidades o
LEFT JOIN usuarios u ON o.usuario_id = u.id
WHERE o.id = ? AND o.status = "ativo"');

$stmt->execute([$_GET['id']]);
$oportunidade = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$oportunidade) {
    header('Location: oportunidades.php');
    exit;
}

require_once 'includes/header.php';
?>

<div class="container py-4">
    <nav aria-label="breadcrumb" class="mb-4">
        <ol class="breadcrumb">
            <li class="breadcrumb-item"><a href="oportunidades.php">Oportunidades</a></li>
            <li class="breadcrumb-item active" aria-current="page">
                <?php echo htmlspecialchars(string: $oportunidade['titulo']); ?>
            </li>
        </ol>
    </div>

```

Imagem 13.1 da busca pela oportunidade específica na base de dados.

Diferenciais do Mapa Cultural de Ananindeua

Além de sua rica gama de funcionalidades, a plataforma do Mapa Cultural de Ananindeua se distingue por uma arquitetura de segurança e infraestrutura de nível superior, projetada para garantir resiliência, performance e confiabilidade. Estes diferenciais não são apenas características adicionais, mas pilares fundamentais que asseguram a integridade dos dados e a continuidade da operação.

Camada de Segurança com Autenticação de Dois Fatores (2FA)

Em um claro compromisso com a segurança da informação e a proteção das contas dos agentes culturais, a plataforma implementa um robusto mecanismo de **Autenticação de Dois Fatores (2FA)**. Ao exigir um código de uso único enviado por e-mail a cada tentativa de acesso, o sistema adiciona uma camada crítica de verificação de identidade. Essa abordagem mitiga drasticamente os riscos de acesso não autorizado, mesmo em caso de comprometimento de senhas, alinhando a



plataforma às melhores práticas de cibersegurança do mercado e garantindo a tranquilidade dos usuários.

Canal de Comunicação Assíncrono e Integrado

Para otimizar a comunicação entre os usuários e a gestão cultural do município, foi implementado um **sistema de comunicação direta e assíncrona** com a Secretaria de Cultura (SECULT). Esta funcionalidade permite que dúvidas, sugestões e outras demandas sejam enviadas diretamente através de um formulário na plataforma, que aciona um *script de backend* para disparar e-mails de forma automatizada. Isso não apenas centraliza e organiza o fluxo de comunicação, mas também cria um registro formal das interações, garantindo que nenhuma solicitação se perca e que a gestão possa responder de maneira ágil e eficiente.

Estratégia de Alta Disponibilidade (High Availability - HA) com Suporte 24/7

A plataforma foi concebida sob um paradigma de operação contínua. O compromisso com o **suporte 24/7** significa que há um monitoramento proativo e ininterrupto da infraestrutura, garantindo que anomalias sejam detectadas e corrigidas em tempo real. Esta estratégia de alta disponibilidade (HA) é crucial para manter o sistema sempre ativo e acessível para a comunidade cultural, minimizando o tempo de inatividade (*downtime*) e assegurando que os recursos da plataforma estejam perpetuamente à disposição de seus usuários.

Infraestrutura de Hospedagem de Alta Performance e Resiliência

A fundação do Mapa Cultural reside em uma infraestrutura de *hosting* de ponta, baseada em um **Servidor Privado Virtual (VPS)** da Hostinger. A especificação técnica deste servidor foi cuidadosamente selecionada para exceder as demandas de performance, escalabilidade e segurança da aplicação:

- **Virtualização KVM:** Garante recursos de hardware verdadeiramente isolados e dedicados, eliminando a contenção de recursos (*"noisy neighbor effect"*) e proporcionando performance consistente.
- **Processamento Superior:** Equipado com 8 vCPUs AMD EPYC, o servidor oferece um poder de processamento massivo, capaz de lidar com um alto



volume de requisições simultâneas, processamento de *uploads* e consultas complexas ao banco de dados sem degradação de performance.

- **Memória e Armazenamento de Alta Velocidade:** Com 32GB de RAM, as operações em memória são executadas com extrema rapidez. Os 400GB de armazenamento em SSD NVMe oferecem taxas de leitura e escrita ordens de magnitude superiores aos SSDs tradicionais, resultando em tempos de carregamento de página e acesso a dados drasticamente reduzidos.
- **Escalabilidade e Conectividade:** A alocação de 32TB de transferência de dados mensais assegura que a plataforma possa suportar um tráfego massivo de usuários sem gargalos.
- **Segurança de Rede:** A infraestrutura é blindada com proteção nativa contra ataques de negação de serviço distribuído (DDoS) e permite a configuração de regras de firewall granulares. O acesso ao servidor é protegido por chaves SSH, o método mais seguro para administração remota.
- **Plano de Contingência:** A existência de backups automáticos e *snapshots* cria um plano de recuperação de desastres robusto, permitindo a restauração rápida do sistema em caso de qualquer incidente.

Identidade Digital e Credibilidade com Domínio Próprio

A utilização do domínio mapaculturalananindeua.com.br confere ao projeto uma identidade digital profissional e de alta credibilidade. Um domínio próprio não só facilita a memorização e o acesso direto pelos usuários, mas também reforça a autoridade e o caráter oficial da plataforma, sendo um fator crucial para a construção de confiança junto à comunidade cultural e ao público em geral. A configuração de um **DNS reverso (rDNS)**, associada à infraestrutura, melhora a reputação dos e-mails disparados pelo sistema, garantindo maior taxa de entrega e evitando que sejam classificados como *spam*.

Ananindeua/PA, 30 de julho de 2025.

RRS
TECHNOLOGY
LTDA:552760960
00107

Assinado de forma
digital por RRS
TECHNOLOGY
LTDA:55276096000107
Dados: 2025.07.30
12:12:12 -03'00'

RAFAEL RODRIGO SITUBA DE SOUZA
DIRETOR DA R R S TECHNOLOGY LTDA