PROJETO INTEGRADOR ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Organiza Eventos

Alunos: Thiago Nunes
Wesley Martins Silva

Professores: Aldecir Fonseca

Muriaé

2025

SUMÁRIO

1. INTRODUÇÃO

O projeto **Organiza Evento** é uma aplicação web desenvolvida utilizando o micro *framework* PHP ***AtomPHP***. Ele foi concebido para oferecer uma solução eficiente e organizada para a gestão de eventos, abrangendo funcionalidades como cadastro de pessoas, gerenciamento de usuários e controle de dados relacionados a cidades e estados. A escolha do *AtomPHP* como base para o desenvolvimento reflete a necessidade de uma estrutura enxuta, mas robusta, capaz de suportar as operações essenciais de um sistema de gerenciamento de eventos.

O *AtomPHP*, conforme descrito em seu *README.md* [1], é um *micro-framework* PHP que adota uma estrutura *MVC* (*Model-View-Controller*) básica, sistema simples de rotas, suporte a controllers e views, autoload de classes via PSR-4 (Composer), configuração por arquivos .env e pronto para integração com banco de dados. Essa arquitetura proporciona uma base sólida para o desenvolvimento de aplicações web, permitindo uma organização clara do código e facilitando a manutenção e escalabilidade do projeto.

O sistema de banco de dados, conforme evidenciado pelo arquivo script_sql.sql [2], é composto por tabelas essenciais para o funcionamento do Organiza Evento, incluindo cidade, pessoa, uf, usuario e usuariorecuperasenha. Essas tabelas armazenam informações cruciais para o gerenciamento de eventos, como dados de participantes, localização e informações de acesso de usuários.

O desenvolvimento do Organiza Evento visa otimizar processos de organização de eventos, desde o registro de participantes até a gestão de informações geográficas, proporcionando uma ferramenta centralizada e eficiente para os organizadores. A aplicação é construída com foco na usabilidade e na performance, garantindo uma experiência fluida para o usuário.

2. OBJETIVO

O objetivo principal do projeto Organiza Evento é desenvolver uma plataforma web robusta e eficiente para a gestão de eventos. Isso inclui a capacidade de cadastrar e gerenciar informações de eventos (capacidade, período, etc.), controlar dados geográficos (cidades e estados) e administrar usuários com diferentes níveis de acesso. A plataforma busca simplificar e otimizar o processo de organização de eventos, fornecendo uma ferramenta centralizada que melhora a eficiência e a comunicação entre os envolvidos.

Além disso, o projeto visa proporcionar uma experiência de usuário intuitiva e responsiva, garantindo que a navegação e a interação com o sistema sejam fluidas e acessíveis em diferentes dispositivos. A segurança dos dados e a integridade das informações são prioridades, com a implementação de mecanismos de validação e controle de acesso para proteger os dados sensíveis.

3. JUSTIFICATIVA

A crescente demanda por ferramentas digitais que facilitem a organização e gestão de eventos justifica o desenvolvimento do Organiza Evento. A complexidade inerente à coordenação de eventos, que envolve múltiplos participantes, locais e informações, torna essencial a adoção de sistemas que automatizem e centralizem esses processos. Atualmente, muitas organizações ainda dependem de métodos manuais ou de ferramentas fragmentadas, o que pode levar a erros, ineficiências e perda de informações.

O Organiza Evento surge como uma solução para esses desafios, oferecendo uma plataforma integrada que otimiza o gerenciamento de dados de pessoas, locais e usuários. Ao centralizar essas informações, o sistema reduz a carga administrativa, minimiza a

ocorrência de erros e permite que os organizadores dediquem mais tempo a aspectos estratégicos do evento. A utilização do *micro-framework AtomPHP* garante

uma base de código leve e de fácil manutenção, ideal para projetos que necessitam de agilidade no desenvolvimento e flexibilidade para futuras expansões.

4. REFERENCIAL TEÓRICO

O desenvolvimento do Organiza Evento baseia-se em conceitos e tecnologias fundamentais para a construção de aplicações web modernas e eficientes. A arquitetura *MVC* (*Model-View-Controller*), adotada pelo *AtomPHP*, é um padrão de design que promove a separação de responsabilidades, facilitando a organização do código, a manutenção e a escalabilidade do projeto. No contexto do Organiza Evento, isso se traduz em:

***Model*os (*Model*s):** Representam a lógica de negócios e a interação com o banco de dados. No projeto, os modelos como *EventoModel*, *UsuarioModel*, *CidadeModel* e *UfModel* (localizados em *app/Model0*) são responsáveis por gerenciar os dados correspondentes às suas entidades, incluindo validações e operações de persistência.

Visões (*View*s): São responsáveis pela apresentação dos dados ao usuário. Embora o projeto não detalhe explicitamente as views no *README.md* do *AtomPHP*, a estrutura app/*View*/ indica a presença de arquivos que definem a interface do usuário, como formulários de cadastro e listagens de dados.

Controladores (*Controller*s): Atuam como intermediários entre os modelos e as visões, processando as requisições do usuário, interagindo com os modelos para obter ou manipular dados e selecionando a visão apropriada para exibir a resposta. Exemplos de controladores no Organiza Evento incluem ApiPessoa.php, Evento.php, *Login.php*, Pessoa.php, Usuario.php, entre outros (localizados em app/*Controller*/). O controlador ApiPessoa.php, por exemplo, gerencia as operações CRUD (Criar, Ler, Atualizar, Apagar) para a entidade Pessoa através de uma API RESTful.

Banco de Dados e Gerenciamento de Dados

O sistema de banco de dados do Organiza Evento é fundamental para o armazenamento e a recuperação das informações. O arquivo script_sql.sql [2] detalha a estrutura das tabelas, que incluem:

cidade: Armazena informações sobre cidades, incluindo nome, UF e código IBGE.

evento: Contém dados de eventos, como nome do evento, wiki, data de início, data de término, cidade, capacidade e imagem. Este é um dos pilares para o cadastro de eventos.

uf: Armazena as unidades federativas (estados) do Brasil, com sigla, bandeira e descrição.

usuario: Gerencia os usuários do sistema, incluindo nível de acesso, nome, email, senha e status de registro. Isso permite a criação de diferentes perfis de usuário (Super Administrador, Administrador, Usuário).

usuario: Gerencia a parte de quem somos da empresa, podendo ser alterado via área administrativa com, título, texto sobre a empresa e imagem.

usuariorecuperasenha: Utilizada para o processo de recuperação de senha, armazenando chaves de recuperação e seus status.

A interação com o banco de dados é abstraída através da classe Database.php (em core/Library/), que provavelmente utiliza PDO ou uma extensão similar para garantir a segurança e a eficiência das operações. Os modelos, como Pessoa*Model*, estendem *Model*Main (em core/Library/), que fornece métodos para interagir com o banco de dados de forma padronizada.

Validação de Dados e Segurança

A validação de dados é um aspecto crucial para a integridade do sistema. O *AtomPHP*, e consequentemente o Organiza Evento, utiliza a classe Validator.php (em core/Library/) para validar os dados de entrada. Conforme observado no Pessoa*Model*.php [3], regras de validação são definidas para cada campo, como required, min, max, email e date, garantindo que os dados inseridos no sistema estejam em conformidade com os requisitos. A validação de tokens, como visto no ApiPessoa.php, também é utilizada para proteger as rotas da API.

Gerenciamento de Sessões e Requisições

O gerenciamento de sessões é tratado pela classe Session.php (em core/Library/), que permite armazenar e recuperar informações do usuário entre as requisições. A classe Request.php (em core/Library/) é responsável por lidar com as requisições HTTP, incluindo a leitura de dados JSON enviados no corpo da requisição, como demonstrado no ApiPessoa.php .

Rotas e Estrutura de URL

O sistema de rotas do *AtomPHP*, implementado pela classe Routes.php (em core/Library/), direciona as requisições HTTP para os controladores e métodos apropriados. A estrutura de URL, como http://localhost/usuario/listar para chamar o método listar() da classe Usuario*Controller*, demonstra a clareza e a previsibilidade do roteamento no framework. Isso facilita a organização das funcionalidades e a criação de URLs amigáveis.

5. METODOLOGIA

O desenvolvimento do projeto Organiza Evento seguiu uma abordagem estruturada, baseada nos princípios do *micro-framework AtomPHP*, que adota o padrão arquitetural *Model-View-Controller* (*MVC*). Essa escolha metodológica visa promover a modularidade, a reusabilidade do código e a facilidade de manutenção, características essenciais para o desenvolvimento de aplicações web robustas e escaláveis.

5.1. Padrão *Model-View-Controller* (*MVC*)

O *MVC* é um padrão de design amplamente utilizado no desenvolvimento de software, especialmente em aplicações web, que divide a aplicação em três componentes interconectados:

***Model* (*Model*o):** Responsável pela lógica de negócios e pela interação com o banco de dados. No Organiza Evento, os modelos (localizados em app/*Model*/) encapsulam as regras de negócio e as operações de persistência para entidades como Pessoa, Usuario, Cidade e UF. Eles garantem a integridade dos dados e fornecem uma interface para que os controladores manipulem as informações.

***View* (Visão):** Encarregada da apresentação dos dados ao usuário. As visões (localizadas em app/*View*/) são responsáveis por renderizar a interface do usuário,

exibindo as informações processadas pelos controladores. Embora o conteúdo específico das visões não tenha sido analisado em detalhes, a estrutura do projeto indica que elas são construídas para proporcionar uma experiência de usuário intuitiva e responsiva.

Controller **(Controlador):** Atua como um intermediário entre o *Model*o e a Visão, recebendo as requisições do usuário, processando-as, interagindo com o *Model*o para obter ou atualizar dados e selecionando a Visão apropriada para exibir a resposta. Os controladores do Organiza Evento (localizados em app/*Controller*/), como ApiPessoa.php, Evento.php e Usuario.php, gerenciam o fluxo da aplicação e orquestram as interações entre os demais componentes.

5.2. Desenvolvimento Orientado a Componentes

O uso de um *micro-framework* como o *AtomPHP* incentiva o desenvolvimento orientado a componentes, onde cada parte da aplicação (modelos, controladores, bibliotecas, helpers) é desenvolvida de forma independente e modular. Isso facilita a colaboração entre desenvolvedores, a realização de testes unitários e a integração contínua de novas funcionalidades. As bibliotecas e helpers (localizados em core/Library/ e core/Helper/) fornecem funcionalidades reutilizáveis, como validação de dados (Validator.php), gerenciamento de sessões (Session.php) e manipulação de requisições (Request.php), que são essenciais para o funcionamento do sistema.

5.3. Gerenciamento de Dependências com Composer

O projeto utiliza o Composer para o gerenciamento de dependências, conforme indicado pelo arquivo composer.json . O Composer é uma ferramenta fundamental no ecossistema PHP que permite declarar as bibliotecas das quais o projeto depende e as instala automaticamente. Isso garante que todas as dependências necessárias estejam disponíveis e que o ambiente de desenvolvimento seja consistente, facilitando a instalação e a execução do projeto em diferentes ambientes.

5.4. Banco de Dados Relacional

O sistema de banco de dados relacional, com o esquema definido no script_sql.sql [2], é a base para o armazenamento persistente dos dados do Organiza Evento. A escolha de um banco de dados relacional permite a organização dos dados em tabelas com relacionamentos bem definidos, garantindo a integridade e a consistência das informações. A interação com o banco de dados é realizada através de uma camada de abstração fornecida pelo *AtomPHP*, que simplifica as operações CRUD e protege contra vulnerabilidades como injeção de SQL.

6. RESULTADOS

O projeto Organiza Evento, ao ser implementado, espera alcançar os seguintes resultados e benefícios:

Centralização e Otimização da Gestão de Eventos: A plataforma oferece um ponto centralizado para o gerenciamento de todas as informações relacionadas a eventos, desde o cadastro até a administração e dados geográficos. Isso elimina a necessidade de múltiplas ferramentas e processos manuais, otimizando o tempo e os recursos dos organizadores.

A API de pessoas (ApiPessoa.php) facilita a integração com outros sistemas e a automação de processos relacionados a dados de indivíduos.

Controle de Acesso e Segurança: A gestão de usuários com diferentes níveis de acesso (Super Administrador, Administrador, Usuário) garante que apenas pessoas autorizadas possam acessar e manipular informações sensíveis. Mecanismos de validação de dados e recuperação de senha contribuem para a segurança geral do sistema.

Flexibilidade e Escalabilidade: A arquitetura baseada em *micro-framework* e o padrão *MVC* proporcionam uma base flexível para futuras expansões e a adição de novas funcionalidades. O projeto pode ser facilmente adaptado para atender a diferentes tipos e tamanhos de eventos, garantindo sua relevância a longo prazo.

Melhora na Experiência do Usuário: Embora não detalhado nas análises, a estrutura do projeto sugere um foco na criação de uma interface de usuário intuitiva e responsiva. Uma boa experiência do usuário é crucial para a adoção e o sucesso da plataforma, incentivando a utilização contínua por parte dos organizadores e participantes.

Redução de Erros e Inconsistências: A automação de processos e a validação de dados minimizam a ocorrência de erros humanos e inconsistências nas informações. Isso resulta em dados mais precisos e confiáveis, essenciais para a tomada de decisões estratégicas na organização de eventos.

7. REFERÊNCIA BIBLIOGRAFIA

FONSECA, Aldecir. *Repositório GitHub do AtomPHP*. Disponível em: https://github.com/aldecirfonseca/atomphp. Acesso em: 21 jun. 2025, às 08h45.

SCRIPT SQL do Projeto Organiza Evento. Disponível em: /home/ubuntu/Organiz Evento/script_sql.sql. Acesso em: 21 jun. 2025, às 14h30.

CÓDIGO-FONTE do PessoaModel.php. Disponível em: /home/ubuntu/Organiza Evento/app/Model/PessoaModel.php. Acesso em: 21 jun. 2025, às 17h20.

GAMMA, Erich et al. *Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos*. 1. ed. Bookman, 2000.

FOWLER, Martin. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.

PHP The Right Way. Disponível em: https://phptherightway.com/. Acesso em: 21 jun. 2025, às 10h10.

MYSQL DOCUMENTATION. Disponível em: https://dev.mysql.com/doc/. Acesso em: 21 jun. 2025, às 19h55.