You:
Ideia de projeto - MVC com PHP, bootstrap e icons, vanilla js e MySQL. Composer, namespaces, .env, sistema de template estilo plates mas sem framework ou biblioteca php alguma. Versionamento com git, main, develop, feat/x feat/y, etc e merge --no-ff e pull --no-ff Questões resolvidas de concurso Empresa / ano concurso Cargo da prova questão alternativas com opção de marcar a correta usar fetch js para não dar reload desnecessário á página ou deixar escondido os itens e abrir ao ser clicado Botão ver resposta - Armazenar a resposta especifica para essa prova Botão resolução Na resolução mostrar links para estudo complementar Botão seguinte. Ao final da prova, por exemplo 100 questões, mostrar quantas acertou por exemplo 1 - A em vermelho 2 - C em azul etc e ao final - Acertou 50/100 - 40% de acerto. Criar o banco de dados para armazenar as questões Criar a pagina de registro e login. Registro deve criar um codigo temporário e enviar via e-mail. Aí com a inserção do código, registrar no banco de dados e por data de registro e um tipo de timer que desative o cadastro em 10 dias. Depois será criado uma função ou algo assim para aumentar os dias no banco. Crie o boilerplate por favor.

DeepSeek:

📋 Visão Geral

Este é um sistema completo para estudo de questões de concursos públicos, desenvolvido em PHP seguindo o padrão MVC, com MySQL, Bootstrap e JavaScript Vanilla.

🗂️ Estrutura do Projeto

text

Copy

Download

```
projeto-concursos/
├── app/
│   ├── controllers/          # Controladores da aplicação
│   ├── models/               # Modelos de dados
│   ├── views/                # Templates e views
│   ├── core/                 # Classes centrais do framework
│   └── routes/               # Definição de rotas
├── config/                   # Arquivos de configuração
├── public/                   # Ponto de entrada e arquivos públicos
├── vendor/                   # Dependências do Composer
└── README.md                 # Este arquivo
```

🚀 Instalação e Configuração

1. Requisitos do Sistema

- PHP >= 7.4
- MySQL >= 5.7
- Composer
- Git

2. Instalação

bash

Copy

Download

```
# Clone o repositório
git clone <url-do-repositorio>
cd projeto-concursos

# Instale as dependências
composer install

# Configure o ambiente
cp config/.env.example config/.env
```

3. Configuração do Banco de Dados

1. Edite o arquivo `config/.env` com suas credenciais do MySQL
2. Execute o script SQL para criar as tabelas (encontrado em `docs/database.sql`)

4. Executando a Aplicação

bash

Copy

Download

```
# Desenvolvimento
php -S localhost:8000 -t public/

# Produção (configure seu servidor web para apontar para a pasta public/)
```

🎨 Sistema de Templates

TemplateEngine - Classe estilo Plates

Nosso sistema inclui uma classe `TemplateEngine` inspirada no Plates PHP:

Recursos Principais:

- **Layouts/Master Pages**: Templates com estrutura comum
- **Partials**: Inclusão de componentes reutilizáveis
- **Escapamento automático**: Proteção contra XSS
- **Herança de templates**: Sistema de blocos e seções

Exemplo de Uso:

Controller:

php

Copy

Download

```
public function showUserProfile($userId) {
    $userModel = new User($this->db);
```

```
    $user = $userModel->readById($userId);

    return $this->template->render('profile/view', [
        'user' => $user,
        'pageTitle' => 'Perfil do Usuário'
    ]);
}
```

View (profile/view.php):

php

Copy

Download

```
<?php $this->extend('layouts/main'); ?>

<?php $this->section('content'); ?>
<div class="container">
    <h1>Perfil de <?= $this->e($user['name']) ?></h1>

    <div class="row">
        <div class="col-md-4">
            <?php $this->insert('profile/sidebar', ['user' => $user]); ?>
        </div>
        <div class="col-md-8">
            <?php $this->insert('profile/content', ['user' => $user]); ?>
        </div>
    </div>
</div>
<?php $this->endSection(); ?>
```

Métodos Disponíveis:

Método	Descrição	Exemplo
`render($template, $data)`	Renderiza um template	`$this->template->render('home')`
`e($value)`	Escapa output HTML	`<?= $this->e($userInput) ?>`
`extend($template)`	Estende um layout	`<?php $this->extend('layouts/main') ?>`
`section($name)`	Inicia uma seção	`<?php $this->section('content') ?>`
`endSection()`	Finaliza uma seção	`<?php $this->endSection() ?>`
`insert($template, $data)`	Insere um partial	`<?php $this->insert('partials/header') ?>`

🆕 Criando Novos Controllers

Estrutura Básica de um Controller

php

Copy

Download

```
<?php
namespace App\Controllers;

use App\Core\TemplateEngine;

class MeuNovoController extends BaseController {
    public function __construct($db) {
        parent::__construct($db);
        // Inicializações específicas do controller
    }

    // Método para ação index
```

```
    public function index() {
        $model = new MeuModel($this->db);
        $dados = $model->getAll();

        return $this->template->render('meu-template/index', [
            'dados' => $dados,
            'titulo' => 'Meu Título'
        ]);
    }

    // Método para ação com parâmetros
    public function ver($id) {
        $model = new MeuModel($this->db);
        $item = $model->getById($id);

        if (!$item) {
            // Redirecionar para página de erro
            header('Location: /erro');
            exit;
        }

        return $this->template->render('meu-template/ver', [
            'item' => $item
        ]);
    }

    // Método para processar formulários
    public function salvar() {
        if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
            header('Location: /');
            exit;
        }

        // Processar dados do formulário
        $dados = [
            'nome' => $_POST['nome'],
            'email' => $_POST['email']
        ];

        $model = new MeuModel($this->db);
        $resultado = $model->create($dados);

        if ($resultado) {
            header('Location: /sucesso?mensagem=Item criado com sucesso');
        } else {
            header('Location: /erro?mensagem=Falha ao criar item');
        }
        exit;
    }
}
```

Registrando Rotas para o Novo Controller

No arquivo `app/routes/web.php`:

php

Copy

Download

```
// Rotas para o novo controller
$router->get('/meu-recurso', [MeuNovoController::class, 'index']);
$router->get('/meu-recurso/ver/{id}', [MeuNovoController::class, 'ver']);
$router->get('/meu-recurso/novo', [MeuNovoController::class, 'novo']);
$router->post('/meu-recurso/salvar', [MeuNovoController::class, 'salvar']);

// Rota com middleware de autenticação
$router->get('/meu-recurso/editar/{id}', [MeuNovoController::class, 'editar'], ['auth']);
```

🆕 Criando Novas Views

🆕 Criando Novas Views

Estrutura de Diretórios para Views

text

Copy

Download

```
app/views/
├── layouts/                 # Layouts principais
│   ├── main.php            # Layout principal
│   └── admin.php           # Layout administrativo
├── partials/               # Componentes reutilizáveis
│   ├── header.php
│   ├── footer.php
│   ├── navigation.php
│   └── form-components/
│       ├── input.php
│       └── select.php
├── meu-template/           # Views do novo controller
│   ├── index.php
│   ├── ver.php
│   ├── novo.php
│   └── editar.php
└── errors/                 # Páginas de erro
    ├── 404.php
    └── 500.php
```

Exemplo de View (index.php)

php

Copy

Download

```
<?php $this->extend('layouts/main'); ?>

<?php $this->section('title'); ?>
<?= $this->e($titulo) ?> - Sistema de Questões
<?php $this->endSection(); ?>

<?php $this->section('styles'); ?>
<style>
    .custom-style {
        color: #3b5998;
    }
</style>
<?php $this->endSection(); ?>

<?php $this->section('content'); ?>
<div class="container">
    <h1 class="my-4"><?= $this->e($titulo) ?></h1>

    <?php $this->insert('partials/alertas'); ?>

    <div class="row">
        <div class="col-md-12">
            <div class="card">
                <div class="card-header d-flex justify-content-between align-items-center">
                    <h5 class="card-title mb-0">Lista de Itens</h5>
                    <a href="/meu-recurso/novo" class="btn btn-primary btn-sm">
                        <i class="bi bi-plus-circle"></i> Novo Item
                    </a>
                </div>
                <div class="card-body">
                    <?php if (empty($dados)): ?>
                        <div class="alert alert-info">
```

```
                        Nenhum item encontrado.
                    </div>
                <?php else: ?>
                    <div class="table-responsive">
                        <table class="table table-striped">
                            <thead>
                                <tr>
                                    <th>ID</th>
                                    <th>Nome</th>
                                    <th>Email</th>
                                    <th>Ações</th>
                                </tr>
                            </thead>
                            <tbody>
                                <?php foreach ($dados as $item): ?>
                                <tr>
                                    <td><?= $this->e($item['id']) ?></td>
                                    <td><?= $this->e($item['nome']) ?></td>
                                    <td><?= $this->e($item['email']) ?></td>
                                    <td>
                                        <a href="/meu-recurso/ver/<?= $item['id'] ?>" class="btn btn-info btn-sm'
                                            <i class="bi bi-eye"></i>
                                        </a>
                                        <a href="/meu-recurso/editar/<?= $item['id'] ?>" class="btn btn-warning b
                                            <i class="bi bi-pencil"></i>
                                        </a>
                                    </td>
                                </tr>
                                <?php endforeach; ?>
                            </tbody>
                        </table>
                    </div>
                <?php endif; ?>
            </div>
        </div>
    </div>
</div>
<?php $this->endSection(); ?>

<?php $this->section('scripts'); ?>
<script>
    document.addEventListener('DOMContentLoaded', function() {
        // Scripts específicos para esta página
        console.log('Página carregada');
    });
</script>
<?php $this->endSection(); ?>
```

🔄 Trabalhando com JOINS no Model

Exemplo de Model com JOINs

php

Copy

Download

```
<?php
namespace App\Models;

class MeuModel extends Model {
    protected $table = "minha_tabela";
    protected $primaryKey = "id";

    public function getWithRelations($id) {
        $query = "SELECT mt.*,
                         u.nome as usuario_nome,
                         u.email as usuario_email,
```

```
                    c.nome as categoria_nome
                FROM {$this->table} mt
                LEFT JOIN usuarios u ON mt.usuario_id = u.id
                INNER JOIN categorias c ON mt.categoria_id = c.id
                WHERE mt.id = :id";

        $stmt = $this->conn->prepare($query);
        $stmt->bindValue(':id', $id, \PDO::PARAM_INT);

        try {
            $stmt->execute();
            return $stmt->fetch(\PDO::FETCH_ASSOC);
        } catch (\PDOException $e) {
            error_log("Error getting item with relations: " . $e->getMessage());
            return false;
        }
    }

    public function getAllWithRelations($filters = [], $limit = null, $offset = null) {
        $query = "SELECT mt.*,
                    u.nome as usuario_nome,
                    c.nome as categoria_nome
                FROM {$this->table} mt
                LEFT JOIN usuarios u ON mt.usuario_id = u.id
                INNER JOIN categorias c ON mt.categoria_id = c.id
                WHERE 1=1";

        $params = [];

        // Filtros
        if (!empty($filters['categoria_id'])) {
            $query .= " AND mt.categoria_id = :categoria_id";
            $params[':categoria_id'] = $filters['categoria_id'];
        }

        if (!empty($filters['usuario_id'])) {
            $query .= " AND mt.usuario_id = :usuario_id";
            $params[':usuario_id'] = $filters['usuario_id'];
        }

        if (!empty($filters['search'])) {
            $query .= " AND (mt.titulo LIKE :search OR mt.descricao LIKE :search)";
            $params[':search'] = '%' . $filters['search'] . '%';
        }

        $query .= " ORDER BY mt.criado_em DESC";

        if ($limit) {
            $query .= " LIMIT :limit";
            $params[':limit'] = $limit;

            if ($offset) {
                $query .= " OFFSET :offset";
                $params[':offset'] = $offset;
            }
        }

        $stmt = $this->conn->prepare($query);

        foreach ($params as $key => $value) {
            $stmt->bindValue($key, $value, is_int($value) ? \PDO::PARAM_INT : \PDO::PARAM_STR);
        }

        try {
            $stmt->execute();
            return $stmt->fetchAll(\PDO::FETCH_ASSOC);
        } catch (\PDOException $e) {
            error_log("Error getting items with relations: " . $e->getMessage());
            return false;
        }
    }

    public function getWithPagination($page = 1, $perPage = 10, $filters = []) {
        $offset = ($page - 1) * $perPage;
```

```
        $offset = ($page - 1) * $perPage;
        $items = $this->getAllWithRelations($filters, $perPage, $offset);

        $total = $this->count($filters);
        $totalPages = ceil($total / $perPage);

        return [
            'items' => $items,
            'pagination' => [
                'current_page' => $page,
                'per_page' => $perPage,
                'total' => $total,
                'total_pages' => $totalPages,
                'has_prev' => $page > 1,
                'has_next' => $page < $totalPages
            ]
        ];
    }
}
```

Usando o Model no Controller

php

Copy

Download

```
public function listar() {
    $page = $_GET['page'] ?? 1;
    $perPage = 10;

    $filters = [];
    if (!empty($_GET['categoria_id'])) {
        $filters['categoria_id'] = $_GET['categoria_id'];
    }

    if (!empty($_GET['search'])) {
        $filters['search'] = $_GET['search'];
    }

    $model = new MeuModel($this->db);
    $resultado = $model->getWithPagination($page, $perPage, $filters);

    // Obter categorias para o filtro
    $categoriaModel = new CategoriaModel($this->db);
    $categorias = $categoriaModel->read();

    return $this->template->render('meu-template/index', [
        'items' => $resultado['items'],
        'pagination' => $resultado['pagination'],
        'categorias' => $categorias,
        'filters' => $filters
    ]);
}
```

🎯 Views Disponíveis

Layouts

- `layouts/main.php` - Layout principal da aplicação
- `layouts/admin.php` - Layout para área administrativa

Partials

- `partials/header.php` - Cabeçalho do site
- `partials/footer.php` - Rodapé do site

- `partials/footer.php` - Rodapé do site
- `partials/navigation.php` - Navegação principal
- `partials/alertas.php` - Sistema de alertas/mensagens
- `partials/pagination.php` - Componente de paginação

Auth

- `auth/login.php` - Página de login
- `auth/register.php` - Página de registro
- `auth/verify.php` - Página de verificação de email

Questions

- `questions/list.php` - Listagem de questões
- `questions/detail.php` - Detalhe de uma questão
- `questions/filters.php` - Componente de filtros

Exams

- `exams/start.php` - Início de um simulado
- `exams/question.php` - Questão individual durante o simulado
- `exams/result.php` - Resultado do simulado

Errors

- `errors/404.php` - Página não encontrada
- `errors/500.php` - Erro interno do servidor

📊 Exemplo de Uso de JOINs nas Views

php

Copy

Download

```
// No Controller
public function showQuestion($id) {
    $questionModel = new Question($this->db);
    $question = $questionModel->getWithDetails($id);

    return $this->template->render('questions/detail', [
        'question' => $question
    ]);
}

// Na View (questions/detail.php)
<div class="question-header">
    <span class="badge bg-primary"><?= $this->e($question['organ_acronym']) ?></span>
    <span class="badge bg-secondary"><?= $this->e($question['exam_year']) ?></span>
    <span class="badge bg-info"><?= $this->e($question['role_name']) ?></span>
</div>

<h3>Questão <?= $this->e($question['question_number']) ?></h3>
<p><?= nl2br($this->e($question['question_text'])) ?></p>
```

```
<!-- Mostrar alternativas -->
<?php $alternatives = json_decode($question['alternatives'], true); ?>
<?php foreach ($alternatives as $alt): ?>
<div class="alternative">
    <strong><?= $this->e($alt['letter']) ?>)</strong>
    <?= $this->e($alt['text']) ?>
</div>
<?php endforeach; ?>
```

🚨 Tratamento de Erros

Error Handling no Controller

php

Copy

Download

```
public function show($id) {
    try {
        $model = new MeuModel($this->db);
        $item = $model->getById($id);

        if (!$item) {
            throw new \Exception('Item não encontrado', 404);
        }

        return $this->template->render('meu-template/ver', [
            'item' => $item
        ]);
    } catch (\Exception $e) {
        error_log($e->getMessage());

        if ($e->getCode() === 404) {
            return $this->template->render('errors/404', [
                'message' => $e->getMessage()
            ], 404);
        }

        return $this->template->render('errors/500', [
            'message' => 'Erro interno do servidor'
        ], 500);
    }
}
```

🔧 Comandos Úteis

bash

Copy

Download

```
# Instalar dependências
composer install

# Atualizar autoloader
composer dump-autoload

# Executar testes (quando implementados)
composer test

# Verificar estilo de código
composer check-style

# Corrigir estilo de código
composer fix-style
```

📝 Próximos Passos

1. Implementar testes unitários
2. Adicionar sistema de cache
3. Implementar internacionalização (i18n)
4. Adicionar sistema de notificações
5. Criar API RESTful
6. Implementar sistema de upload de arquivos
7. Adicionar documentação da API

🤝 Contribuindo

1. Faça o fork do projeto
2. Crie uma branch para sua feature (`git checkout -b feature/AmazingFeature`)
3. Commit suas mudanças (`git commit -m 'Add some AmazingFeature'`)
4. Push para a branch (`git push origin feature/AmazingFeature`)
5. Abra um Pull Request

📄 Licença

Este projeto está sob a licença MIT. Veja o arquivo `LICENSE` para mais detalhes.

🆘 Suporte

Se você encontrar problemas ou tiver dúvidas:

1. Verifique a documentação
2. Procure em issues existentes
3. Crie uma nova issue com detalhes do problema

Nota: Esta documentação será atualizada conforme o projeto evolui. Mantenha-se atualizado com as últimas mudanças.

You:
Comece a criar para mim por favor. Adapte a isso se necessário: Por exemplo TRT tem prova para cargo A, cargo, B, etc e ano x, y e z. Então tem que refatorar essas tabelas e depois usaremos joins ou algo assim para resgatar os dados. Aí podemos mostrar concursos por cargo - Ex. Contador, aí mostraria todas as provas, por ano mostraria todas de 2022 e assim por diante, por empresa... todas a provas

You:
Faça todos os passos acima.

You:
Crie um Routes e as rotas a parte tipo no arquivo web.php. Não se esqueça do composer, namespaces. Foi criado um Model abstrato para o CRUD, insert, delete, update e read e readbyid ou algo assim? Se não crie e adapte o que vimos até agora?

You:
Crie o Readme.md e arquivo de ajuda principalmente para novos controllers e views e uso dos joint etc. Ah, todas as views foram criadas? Tem a classe estilo plates para lidar com as views - com template master, partials, etc?