

Estrutura do Projeto TechFit - MVC com PHP e MySQL

Estrutura de Diretórios

```
techfit/
├── config/
│   └── Database.php          # Classe de conexão com MySQL
├── models/
│   ├── User.php             # Model de Usuários
│   ├── Schedule.php         # Model de Horários
│   ├── Appointment.php      # Model de Agendamentos
│   └── Workout.php          # Model de Treinos
├── controllers/
│   └── DashboardController.php # Controller principal do Dashboard
├── views/
│   └── dashboard.php         # View principal (ou usar index.php)
├── public/
│   ├── index.php            # Ponto de entrada da aplicação
│   ├── api.php              # Rotas da API REST
│   ├── css/
│   │   └── DashBoard.css    # Estilos do dashboard
│   └── js/
│       └── dashboard.js      # JavaScript para chamadas AJAX
└── database.sql              # Script de criação do banco
```

Configuração do Banco de Dados

1. Criar o Banco de Dados

Execute o arquivo `database.sql` no seu MySQL:

```
mysql -u root -p < database.sql
```

Ou importe via phpMyAdmin.

2. Configurar Credenciais

Edite o arquivo `config/Database.php`:

```
private $host = "localhost";
private $db_name = "techfit_db";
private $username = "root";
private $password = "sua_senha";
```

Arquitetura MVC

Models (Modelos)

Responsáveis pela lógica de negócio e acesso ao banco de dados:

- **User.php**: Gerenciamento de usuários, perfil e estatísticas
- **Schedule.php**: Horários disponíveis para agendamento
- **Appointment.php**: Agendamentos de treinos
- **Workout.php**: Planos de treino e exercícios

Controllers (Controladores)

Processam as requisições e coordenam Models e Views:

- **DashboardController.php**: Gerencia todas as operações do dashboard
 - `index()`: Carrega a página principal
 - `getUserData()`: Retorna dados do usuário
 - `getDashboardStats()`: Retorna estatísticas
 - `getSchedules()`: Lista horários disponíveis
 - `getAppointments()`: Lista agendamentos do usuário
 - `getWorkoutPlans()`: Lista planos de treino
 - `createAppointment()`: Cria novo agendamento
 - `confirmAppointment()`: Confirma agendamento
 - `cancelAppointment()`: Cancela agendamento
 - `updateProfile()`: Atualiza perfil do usuário

Views (Visualização)

Interface do usuário:

- **index.php**: Página principal com HTML e dados do PHP
- **dashboard.js**: JavaScript para interações dinâmicas via AJAX

Fluxo de Dados

1. Carregamento Inicial da Página

Usuário → index.php → DashboardController → Models → MySQL

↓

Retorna dados

↓

Renderiza HTML

2. Operações Dinâmicas (AJAX)

JavaScript → api.php → DashboardController → Models → MySQL

↓

Retorna JSON

↓

Atualiza interface



Como Usar

1. Instalar Dependências

- PHP 7.4+ com PDO MySQL
- MySQL 5.7+
- Servidor Apache ou Nginx

2. Configurar o Projeto

Clone ou copie os arquivos para o diretório do servidor
cd /var/www/html/techfit # ou htdocs para XAMPP

Configure as permissões
chmod -R 755 techfit/

3. Configurar Virtual Host (Opcional)

Apache:

```
<VirtualHost *:80>
    ServerName techfit.local
    DocumentRoot /var/www/html/techfit/public

    <Directory /var/www/html/techfit/public>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

4. Acessar a Aplicação

`http://localhost/techfit/public/index.php`

ou

`http://techfit.local`



Endpoints da API

Método	Endpoint	Descrição
GET	<code>api.php?action=getUserData</code>	Dados do usuário logado
GET	<code>api.php?action=getDashboardStats</code>	Estatísticas do dashboard
GET	<code>api.php?action=getSchedules</code>	Horários disponíveis
GET	<code>api.php?action=getAppointments</code>	Agendamentos do usuário
GET	<code>api.php?action=getWorkoutPlans</code>	Planos de treino
POST	<code>api.php?action=createAppointment</code>	Criar agendamento
POST	<code>api.php?action=confirmAppointment</code>	Confirmar agendamento
POST	<code>api.php?action=cancelAppointment</code>	Cancelar agendamento
POST	<code>api.php?action=updateProfile</code>	Atualizar perfil



Segurança

Implementar:

1. **Sistema de Login:** Adicionar autenticação de usuários
2. **Sessões:** Validar sessões em todas as páginas
3. **Sanitização:** Validar e sanitizar todos os inputs
4. **Prepared Statements:** Já implementado nos Models (PDO)
5. **CSRF Tokens:** Adicionar para formulários
6. **HTTPS:** Usar em produção

Exemplo de Login Básico:

```
// login.php
session_start();
if ($user->authenticate($email, $password)) {
    $_SESSION['user_id'] = $user->id;
    header('Location: index.php');
}
```










Testando a API

Usando cURL:

```
# Buscar dados do usuário
curl http://localhost/techfit/public/api.php?action=getUserData

# Criar agendamento
curl -X POST \
  http://localhost/techfit/public/api.php?action=createAppointment \
  -H 'Content-Type: application/json' \
  -d '{
    "schedule_id": 1,
    "title": "Treino de Peito",
    "date": "2024-11-15"
  }'
```

Próximos Passos

1.  Estrutura MVC implementada
2.  Banco de dados MySQL configurado
3.  API REST funcional
4.  Sistema de autenticação
5.  Validação de formulários
6.  Sistema de notificações
7.  Upload de fotos de perfil
8.  Gráficos de progresso
9.  Exportar relatórios PDF

Troubleshooting

Erro de Conexão com Banco

```
// Verifique as credenciais em Database.php
// Certifique-se que o MySQL está rodando
sudo service mysql start
```

Erro 404 nas Rotas

```
# Habilite mod_rewrite no Apache  
sudo a2enmod rewrite  
sudo service apache2 restart
```

Erro de Permissão

```
# Ajuste as permissões dos arquivos  
chmod -R 755 /var/www/html/techfit  
chown -R www-data:www-data /var/www/html/techfit
```



Recursos Adicionais

- [Documentação PDO](#)
- [Padrão MVC](#)
- [MySQL Best Practices](#)
- [PHP Security](#)

Desenvolvido para TechFit Fitness Pro 💪