



Gerenciador de Tarefas Symfony

Guilherme Balatka





01

Introdução

02

Vantagens

03

Desvantagens

04

Exemplos

05

Diferenças

06

**Considerações
Finais**

Introdução

- Framework PHP para Desenvolvimento Web
 - >Amplamente usado para criar aplicativos web robustos e escaláveis.
- História
 - >Lançado em 2005 e desde então ganhou popularidade.
- Flexibilidade e Eficiência
 - >Oferece uma estrutura organizada para o desenvolvimento de aplicativos web.

Introdução

- Reutilização de Componentes
 - > Promove a reutilização de componentes para desenvolvimento mais rápido.
- Destaque para as Melhores Práticas
 - > Promove o uso de melhores práticas de desenvolvimento.
- Alta Qualidade e Segurança
 - > Garante alta qualidade e segurança em aplicações desenvolvidas com Symfony.

Vantagens

- Facilita o desenvolvimento web robusto.
- Utiliza arquitetura MVC (Model-View-Controller).
- Compatibilidade com bancos de dados.
- Biblioteca de componentes reutilizáveis.
- Comunidade ativa e documentação robusta.

Vantagens

- Segurança integrada
- Gerenciamento de dependências: O uso do Composer para gerenciamento de pacotes facilita a inclusão de bibliotecas e componentes de terceiros.

Desvantagens

- Curva de aprendizado íngreme para iniciantes.
- Requer mais configuração inicial em comparação com microframeworks.
- Ruim em projetos pequenos e simples.
- Requisitos de hardware e desempenho.
- Possível sobrecarga de recursos em servidores de baixo custo.

Exemplos

Criação de Database

```
C:\laragon\www\back-end\gerenciadorTarefas(master)  
λ php bin/console doctrine:database|
```

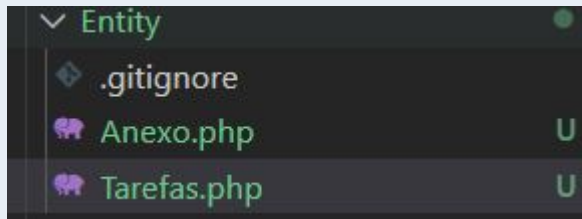
Criação da Tabela no banco

```
C:\laragon\www\back-end\gerenciadorTarefas(master)  
λ php bin/console make:entity|
```


Exemplos

Nome da Tabela

Class name of the entity to create or update (e.g. GrumpyElephant):
> tarefas|



```
<?php

namespace App\Entity;

use App\Repository\TarefasRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity(repositoryClass=TarefasRepository::class)
 */
class Tarefas
{
    /**
     * @ORM\Id
     * @ORM\GeneratedValue
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255, nullable=true)
     */
    private $nome;

    /**
     * @ORM\Column(type="text", nullable=true)
     */
    private $descricao;

    /**
     * @ORM\Column(type="date", nullable=true)
     */
    private $prazo;
}
```

Exemplos

Criação dos código sql

```
C:\laragon\www\back-end\gerenciadorTarefas(master)
λ php bin/console make:migration
```

```
<?php
declare(strict_types=1);

namespace Doctrine\Migrations;

use Doctrine\DBAL\Schema\Schema;
use Doctrine\Migrations\AbstractMigration;

/**
 * Auto-generated Migration: Please modify to your needs!
 */
final class Version20231019231131 extends AbstractMigration
{
    public function getDescription(): string
    {
        return '';
    }

    public function up(Schema $schema): void
    {
        // this up() migration is auto-generated, please modify it to your needs
        $this->addSql('CREATE TABLE anexos (id INT AUTO_INCREMENT NOT NULL, tarefa_id INT DEFAULT NULL, nome VARCHAR(255) DEFAULT NULL, ar
        $this->addSql('CREATE TABLE tarefas (id INT AUTO_INCREMENT NOT NULL, nome VARCHAR(255) DEFAULT NULL, descricao LONGTEXT DEFAULT N
        $this->addSql('CREATE TABLE messenger_messages (id BIGINT AUTO_INCREMENT NOT NULL, body LONGTEXT NOT NULL, headers LONGTEXT NOT N
        $this->addSql('ALTER TABLE anexos ADD CONSTRAINT FK_CD7EAF2C78217710 FOREIGN KEY (tarefa_id) REFERENCES tarefas (id)');
        $this->addSql('ALTER TABLE pedidos DROP FOREIGN KEY pedidos_ibfk_1');
        $this->addSql('ALTER TABLE pedidos DROP FOREIGN KEY pedidos_ibfk_2');
        $this->addSql('DROP TABLE clientes');
        $this->addSql('DROP TABLE pedidos');
        $this->addSql('DROP TABLE produtos');
    }

    public function down(Schema $schema): void
    {

```

Exemplos

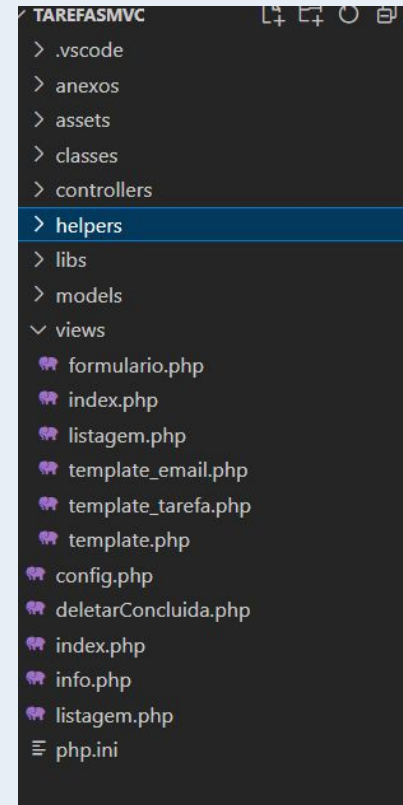
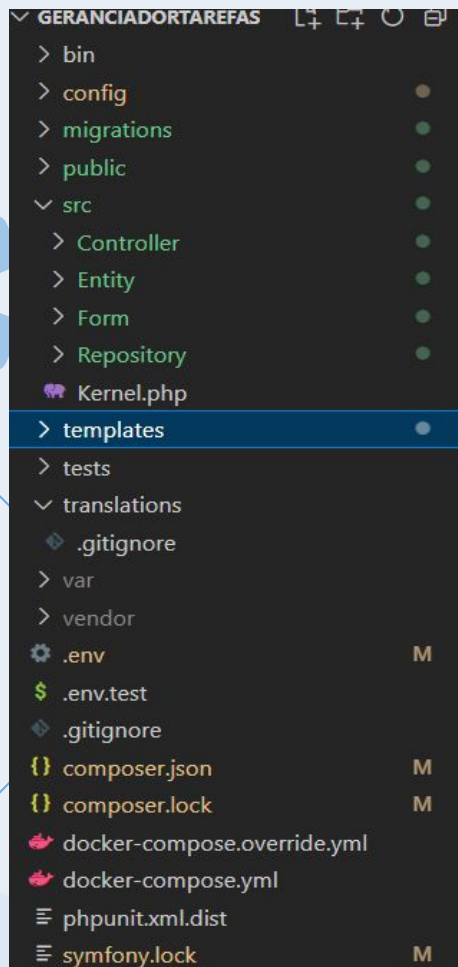
Criação da Tabela no Banco de dados

```
C:\laragon\www\back-end\gerenciadorTarefas(master)  
λ php bin/console make:migration:migrate|
```

Gera o CRUD completo

```
C:\laragon\www\back-end\gerenciadorTarefas(master)  
λ php bin/console make:crud|
```

Comparação



Comparação

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="tarefas.css">

  <title>Gerenciador de Tarefas</title>
</head>
<body>
  <div="bloco_principal">
    <h1>Tarefa: <?php echo $tarefa['nome'];?></h1>
    <p><a href="tarefas.php">Voltar para a lista de tarefas</a></p>

    <p><strong>Concluída:</strong>
    | <?php echo traduz_concluida($tarefa['concluida']);?>
    </p>

    <p><strong>Descrição:</strong>
    | <?php echo ($tarefa['descricao']);?>
    </p>

    <p><strong>Prazo:</strong>
    | <?php echo traduz_data_para_tela($tarefa['prazo']);?>
    </p>

    <p><strong>Prioridade:</strong>
    | <?php echo traduz_prioridade($tarefa['prioridade']);?>
    </p>

    <h2>Anexos</h2>

    <?php if(!empty($anexos) &&(count($anexos) > 0)) : ?>
    <table>
      <tr>
        <th>Arquivo</th>
        <th>Opções</th>
      </tr>
    </table>
    </?>
  </div>
</body>
</html>
```

Template Tarefa
sem framework

Comparação

Template Tarefa com Symfony

```
{% extends 'base.html.twig' %}

{% block title %}Gerenciador de Tarefas{% endblock %}

{% block body %}
<h1>Tarefas</h1>

<table class="table">
  <thead>
    <tr>
      <th>Id</th>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Prazo</th>
      <th>Prioridade</th>
      <th>Concluída</th>
      <th>Ação</th>
    </tr>
  </thead>
  <tbody>
    {% for tarefa in tarefas %}
      <tr>
        <td>{{ tarefa.id }}</td>
        <td>{{ tarefa.nome }}</td>
        <td>{{ tarefa.descricao }}</td>
        <td>{{ tarefa.prazo ? tarefa.prazo|date('d/m/Y') : '' }}</td>
        <td>{{ tarefa.prioridade }}</td>
        <td>{{ tarefa.concluida ? 'Sim' : 'Não' }}</td>
        <td>
          <a href="{{ path('app_tarefas_show', {'id': tarefa.id}) }}"></a>
          <a href="{{ path('app_tarefas_edit', {'id': tarefa.id}) }}"></a>
          <a href="{{ path('app_anexo_new', {'tarefaId': tarefa.id}) }}"></a>
          <a href="{{ path('app_anexo_index', {'tarefaId': tarefa.id}) }}" class="btn btn-secondary">Lista de Anexos</a>
          <a href="{{ path('app_send_email', {'tarefaId': tarefa.id}) }}" class="btn btn-primary">Enviar E-mail</a>
        </td>
      </tr>
    {% endfor %}
  </tbody>
</table>

```

Comparação

```
<?php

$exibir_listagem = true;
$tem_erro = false;
$erros_validacao = [];

$tarefa = new Tarefa();
$tarefa->setPrioridade(1);

if($_POST){

    if(array_key_exists('nome', $_POST) && $_POST['nome'] != ''){
        $tarefa->setNome($_POST['nome']);
    }else{
        $tem_erro = true;
        $erros_validacao['nome'] = 'O nome da Tarefa é obrigatório';
    }

    if(array_key_exists('descricao', $_POST) ){
        $tarefa->setDescricao($_POST['descricao']);
    }else{
        $tarefa['descricao'] = '';
    }

    if(array_key_exists('prazo', $_POST) && strlen($_POST['prazo']) > 0){
        if(valida_data($_POST['prazo'])){
            $tarefa->setPrazo(traduz_data_br_para_objeto($_POST['prazo']));
        }else{
            $tem_erro = true;
            $erros_validacao['prazo'] = 'O prazo informado não é uma data válida!';
        }
    }

    $tarefa->setPrioridade($_POST['prioridade']);
}
```

Controller Tarefa
sem framework

Comparação

Controller Anexo com Symfony

```
class AnexoController extends AbstractController
{
    private $entityManager;

    public function __construct(EntityManagerInterface $entityManager)
    {
        $this->entityManager = $entityManager;
    }

    /**
     * @Route("/{tarefaId}", name="app_anexo_index", methods={"GET"}, requirements={"tarefaId"="\d+"})
     */
    public function index(AnexoRepository $anexoRepository, int $tarefaId): Response
    {
        $anexos = $anexoRepository->findBy(['tarefa' => $tarefaId]);

        return $this->render('anexo/index.html.twig', [
            'anexos' => $anexos,
            'tarefaId' => $tarefaId,
        ]);
    }

    /**
     * @Route("/new/{tarefaId}", name="app_anexo_new", methods={"GET", "POST"})
     */
    public function new(Request $request, int $tarefaId): Response
    {
        $anexo = new Anexo();
        $form = $this->createForm(AnexoType::class, $anexo);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            // Obtenha o arquivo de anexo do formulário
            $file = $form->get('arquivo')->getData();

            // Gere um nome de arquivo único
            $fileName = md5(uniqid()).'.'.$file->guessExtension();
        }
    }
}
```


The background is a light blue gradient. On the left side, there are two overlapping gears: a solid blue one on top and a white one with a blue outline below it. In the bottom right corner, there are stylized white clouds and a small blue gear outline.

OBRIGADO PELA ATENÇÃO