



Curso PHP

Do XLSX ao CMS (aula 17)

Autorização



O conceito de autorização

Autorização é o processo de dar a alguém a capacidade de acessar um recurso. Essa definição pode não parecer muito clara, mas muitas situações na vida real podem ajudar a ilustrar o que a autorização significa para que você possa aplicar esses conceitos aos sistemas de computador. ([Auth0](#))

A autorização é o processo de determinar se deve conceder ou negar aos usuários o direito de acessar os recursos. A autorização opera seguindo um conjunto de regras e políticas predefinidas. Essas regras são normalmente gerenciadas por um sistema de controle de acesso que estabelece permissões com base nos requisitos de conformidade da organização. Quando um usuário está tentando acessar um recurso, o sistema de autorização avaliará suas permissões e as políticas predefinidas da organização antes de permitir que o usuário acesse o recurso. ([Keeper Security](#))



Tipos de controles de acesso

- **RBAC:** define permissões com base no papel e nas funções do usuário dentro da organização.
- **ReBAC:** se concentra na relação entre o usuário e o recurso.
- **ABAC:** modelo de autorização é uma forma mais detalhada de controle de acesso porque avalia o assunto, o recurso, a ação e o ambiente.
- **DAC:** os proprietários de recursos assumem a responsabilidade de decidir como seus recursos serão compartilhados.
- **MAC:** gerencia permissões de acesso com base na sensibilidade do recurso e no nível de segurança do usuário.

([Keeper Security](#))

Roles

Papéis de usuário



Papéis de usuário

Roles, ou papéis de usuário, referem-se às diferentes permissões e responsabilidades atribuídas a usuários em um sistema, aplicativo ou plataforma.

Esses papéis são fundamentais para a gestão de acesso e segurança, permitindo que cada usuário tenha acesso apenas às funcionalidades e informações necessárias para suas atividades.

No contexto de desenvolvimento web, a definição clara de User Roles é essencial para garantir que a aplicação funcione de maneira eficiente e segura.

([LBO dev](#))

```
enum Role: string
{
    use EnumSupport;

    case ADMINISTRADOR = 'administrador';
    case CONTRIBUIDOR = 'contribuidor';
    case VISITANTE = 'visitante';
}
```

```
namespace App\Middlewares;

use App\Enums\Role;

class RequerRoleContribuidor
{
    public function __invoke(): bool
    {
        return hasRole(Role::ADMINISTRADOR, Role::CONTRIBUIDOR);
    }
}
```

```
Router::put(
    uri: '/agendamentos/{agendamento}',
    action: [AgendamentoController::class, 'atualizar'],
    middlewares: [
        AcessoRestrito::class,
        RequerRoleContribuidor::class
    ]
);
```

Autorização com *roles*

Permissions

Permissões



Permissões

As permissões são um **conjunto de direitos concedidos a usuários ou grupos específicos** em um sistema de computador.

Essas **permissões permitem que esses usuários realizem determinadas ações** que normalmente não seriam permitidas a outros usuários.

Essas **permissões são essenciais para garantir a segurança e a integridade dos dados armazenados em um sistema**, pois controlam quem pode acessar, modificar ou excluir informações sensíveis.

([Netvistos](#))

```

enum Permission: string
{
    use EnumSupport;

    case MANTER_PERFIL = 'manter_perfil';
    case MANTER_AGENDAMENTOS = 'manter_agendamentos';
    case MANTER_DISCIPLINAS = 'manter_disciplinas';
    case MANTER_ATIVIDADES = 'manter_atividades';
    case MANTER_PERIODOS = 'manter_periodos';
    case MANTER_USUARIOS = 'manter_usuarios';
    case MANTER_PERMISSOES = 'manter_permissoes';
    case MANTER_HORARIOS = 'manter_horarios';
}

Router::put(
    uri: '/agendamentos/{agendamento}',
    action: [AgendamentoController::class, 'atualizar'],
    middlewares: [
        AcessoRestrito::class,
        Permission::MANTER_AGENDAMENTOS,
    ]
);

```

```

enum Role: string
{
    use EnumSupport;

    case ADMINISTRADOR = 'administrador';
    case CONTRIBUIDOR = 'contribuidor';
    case VISITANTE = 'visitante';

    public function permissions(): array
    {
        return match ($this) {
            self::VISITANTE => [
                Permission::MANTER_PERFIL,
            ],
            self::CONTRIBUIDOR => [
                // ...
            ],
            self::ADMINISTRADOR => [
                // ...
            ],
        };
    }
}

```

Autorização com *permissions*

Regras de negócio



Regras de negócio

A regra de negócio traduz uma necessidade do negócio - como validações, restrições e etc - em regras lógicas, permitindo que desenvolvimento, produto e negócio de uma empresa se alinhem com relação a estas necessidades, se guiem por elas e apliquem as regras da forma mais clara possível, fazendo com que o desenvolvimento e o crescimento do produto possam fluir da melhor forma.

([Juliana Amoasei](#))

```
namespace App\Middlewares;  
  
use App\Models\Usuario;  
  
class NaoPodeEditarAdminOuSiMesmo  
{  
    public function __invoke(Usuario $usuario): bool  
    {  
        return $usuario->id !== 1  
            && $usuario->isNot(session()->get('usuario'));  
    }  
}
```

Impedindo que usuários realizem ações danosas



Próximos passos: auditoria

- Lançamento de exceptions
- Adição/alteração de registros
- Autenticação