Permissão de Segurança

O sistema de permissões de segurança é um mecanismo de autorização e autenticação de usuários. Dessa forma, é possível definir regras de acessos a usuários ou grupos de usuários, além de permitir múltiplas formas de efetuar o login no sistema, utilizando as informações contidas no Banco de Dados (padrão) ou um provedor de login externo. Os provedores de login externo suportados incluem o Active Directory, SAML, OpenID-connect, Microsoft Account, Twitter, Login social (Github, Facebook, LinkedIn, Google e Cronapp), Certificados digitais e ICP Brasil, podendo também ser personalizado para qualquer provedor de login externo compatível com OAuth 2.0.

Os projetos Cronapp criados a partir da versão 2.9.6-SP.40 possuem um recurso de invalidação de tokens dos usuários. Para mais detalhes desse recurso, consulte a documentação Invalidação de tokens.

Nesta página

Assista sobre o tema no Cronapp Academy

Caso seja seu primeiro acesso ao Cronapp Academy, crie antes uma conta gratuita e matricule-se no curso abaixo.

- Aula: Permissões de segurança

Estrutura

A Permissão de Segurança do Cronapp possui uma estrutura dividida em 3 camadas: Permissionáveis (Securable), Grupos (Role) e Usuários (User). Essa estrutura possui uma pequena hierarquia, onde um **permissionável** possui as permissões do sistema e é composto por usuários e grupos de usuários; um **Grupo**, possui um ou mais usuários e pode estar associado a um ou mais permissionáveis e o **Usuário**, que pode estar dentro de um ou mais grupos e estar diretamente vinculado a um ou mais permissionáveis. A figura 1 representa essa estrutura.

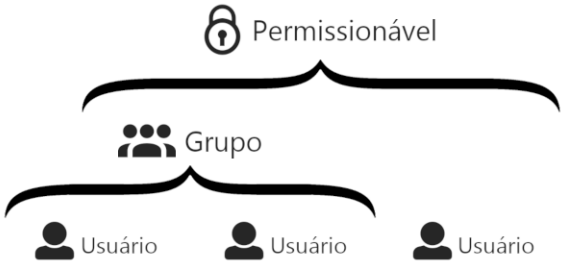

Figura 1 - Estrutura das permissões de segurança do Cronapp

Permissionável

Cada permissionável possui um conjunto de permissões (métodos de requisição HTTP) que podem ser aplicadas em arquivos específicos ou todo o conteúdo dentro de um diretório. O acesso a arquivos é feito diretamente nas configurações do permissionável, porém, o Cronapp também utiliza a lista de permissionáveis do sistema para liberar acessos a recursos dentro de uma página, relatório, bloco de programação e diversas outras funcionalidades (veja exemplos no tópico Atribuir Permissões).

Por padrão, os projetos criados no Cronapp já incluem os seguintes permissionáveis: **Public**, **Authenticated** e **Administrators**. Não sendo possível renomear ou remover esses permissionáveis. Os permissionáveis herdam permissões de outros permissionáveis, assim:

- O permissionável **Public** possui as permissões mais básicas, como acesso ao conteúdo do diretório `public` e arquivos necessários para que os usuários não logados possam visualizar o conteúdo público, exemplo de arquivos CSS e tema da aplicação.
- O permissionável **Authenticated** herda todas as permissões do permissionável **Public** e possui algumas permissões próprias, como acesso ao conteúdo do diretório `views/logged/`.
- O permissionável **Administrators** herda todas as permissões do permissionável **Authenticated** e possui algumas permissões próprias, com acesso completo ao sistema e diretório `views/admin/`.
- Todo os novos permissionáveis do sistema herdarão automaticamente as permissões do permissionável **Authenticated**.

Usuários vinculados a mais de um permissionável terá o acesso do permissionável de maior abrangência. Por exemplo, se um usuário estiver associado a dois permissionáveis, através de grupos ou diretamente, sendo que o primeiro permissionável concede acesso a parte do sistema e o segundo possua acesso completo, ao logar, esse usuário terá acesso total.

Alterar regras

É possível modificar as regras dos 3 permissionáveis padrão para que eles se adequem melhor as necessidades do seu sistema. Porém, é preciso ter bastante atenção ao fazer isso, pois podem abrir brechas para falhas de segurança. Além disso, essas alterações devem contemplar tanto a aplicação web quanto a mobile.

Em alguns casos, talvez seja prudente ter o permissionável **Authenticated** mais restrito, apenas com o que é realmente comum e criar permissionáveis com acessos bem específicos ao que aquele usuário ou grupo de usuários necessitarão dentro do sistema. Por exemplo, o permissionável “comprador” pode contemplar tudo o que é necessário para a funcionalidade, e se necessário, criar permissionáveis ainda mais específicos, como “Comprador - Escrita” e “Comprador - Leitura”.

Grupos

Um grupo pode conter diversos usuários, e esses usuários herdarão automaticamente os acessos dos permissionáveis dos quais esse grupo pertença. Dessa forma, um grupo é considerado um **papel** ou **fun ção**, já que pode possuir as regras necessárias para que o usuário associado execute seu papel e acesse suas funções dentro do sistema, exemplo: Vendedor, Gerente, Gestão de pessoas.

Por padrão, os projetos criados no Cronapp já incluem os seguintes Grupos:

- **Anonymous Users**: associado ao permissionável **Public**.
- **Authenticated Users**: associado ao permissionável **Authenticated**.
- **Administrators**: associado ao permissionável **Administrators**. É fundamental que sempre haja um grupo associado ao permissionável **Administrators**.

Usuários

Um usuário do sistema pode obter suas autorizações por estar associado a grupos ou diretamente a permissionáveis. Usuários criados sem vínculos com grupos ou permissionáveis receberá, de forma implícita, todas as regras do permissionável **Authenticated**.

Por padrão, os projetos criados no Cronapp contêm o usuário:

- **Administrator**: esse usuário já vem associado ao grupo **Administrators**, que por sua vez, está vinculado ao permissionável **Administrators**. É fundamental que sempre haja um usuário associado ao permissionável **Administrators**.

  Para logar com o usuário Administrador, utilize o *username* e senha "admin". Após logar, recomendamos alterar a senha desse usuário.

Estrutura de Classes

Por padrão, o sistema de permissões e segurança armazena informações de usuário em um banco de dados usando o JPA. Essa abordagem funciona bem para muitos aplicativos. Entretanto, você pode preferir usar um mecanismo de persistência ou esquema de dados diferente, dessa forma, é possível alterar e tornar o modelo extensível e personalizável.

Na figura abaixo são exibidas as classes usadas na Permissão de Segurança do Cronapp. Dependendo das configurações do seu projeto, outras três classes também serão exibidas: `AuditLog` (Log de auditoria), `Device` (Dispositivos móveis) e `InvalidatedToken` (Invalidação de tokens) (acesse as documentações Diagrama, Log de Auditoria e Invalidação de Tokens para mais detalhes sobre essas três classes).

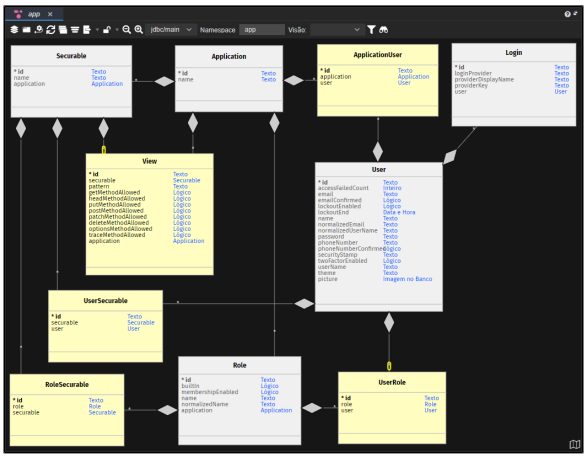

Figura 1.1 - Diagrama de dados inicial de um sistema Cronapp

O modelo de dados das permissões de segurança consiste dos seguintes tipos de entidade (Figura 1.1):

Nome da Entidade	Descrição
User	Representa um usuário.
Role	Representa um grupo (função).
Login	Representa logins de provedores externos para um usuário.
Securable	Representa um permissionável.
View	Representa um objeto do tipo View ao qual você quer aplicar controle de acesso através dos métodos de requisição HTTP.
Application	Representa uma aplicação, utilizado em sistemas com multi aplicações.
RoleSecurable	Associa grupos e permissionáveis, permitindo que todos os usuários contidos em um grupo tenham permissão ao conjunto de acessos do permissionável.
UserSecurable	Associa usuários e permissionáveis, permitindo que o usuário tenha permissão diretamente ao conjunto de acessos do permissionável.
UserRole	Associa um usuário a um grupo (função).
ApplicationUser	Associa usuários e aplicações, utilizado em sistemas com multi aplicações.

Relacionamentos entre entidades

Os tipos de entidade são relacionados entre si das seguintes formas:

- Cada `User` pode ter múltiplos `Securables` associados, e cada `Securable` pode estar associado a múltiplos `Users`. Esse relacionamento muitos-para-muitos é representado pela entidade `UserSecurable`.
- Cada `User` pode ter múltiplos `Logins` associados.
- Cada `User` pode ter múltiplos `Roles` associados, e cada `Role` pode estar associada a múltiplos `Users`. Esse relacionamento muitos-para-muitos é representado pela entidade `UserRole`.
- Cada `Role` pode ter múltiplos `Securables` associados, e cada `Securable` pode estar associado a múltiplos `Roles`. Esse relacionamento muitos-para-muitos é representado pela entidade `RoleSecurable`.
- Cada `Securable` pode ter múltiplos `Views` associados.

Relacionamentos para Multi Aplicações

Os relacionamentos abaixo são específicos para a utilização de Multiplas aplicações dentro da mesma base de dados, para isso são utilizadas as tabelas `Application` e `ApplicationUser`. Os tipos de entidade são relacionados entre si das seguintes formas:

- Cada `Application` pode ter múltiplos `Role` associados.
- Cada `Application` pode ter múltiplos `Securable` associados.
- Cada `Application` pode ter múltiplos `View` associados.
- Cada `Application` pode ter múltiplos `User` associados, e cada `User` pode estar associado a múltiplos `Application`. Esse relacionamento muitos-para-muitos é representado pela entidade `ApplicationUser`.

Tabelas

Abaixo são detalhados todos os campos das tabelas usadas pelo Sistema de Segurança.

User

Usuário.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
access_failed_count	Inteiro	Número de falha de acesso seguidos.
email	Texto	E-mail do usuário.
email_confirmed	Lógico	Confirmação se o e-mail foi validado.
lockout_enabled	Lógico	Permite que o mecanismo de segurança bloqueie o acesso do usuário (por exemplo, errar a senha várias vezes).
lockout_end	Data e Hora	Horário que o usuário será desbloqueado para acessar o sistema.
name	Texto	Nome do usuário.
normalized_email	Texto	Cria uma cópia normalizada do email do usuário, garantindo que o usuário possa ter e-mail com caracteres especiais.
normalized_user_name	Texto	Cria uma cópia normalizada do user_name, garantindo que o usuário possa ter um login com caracteres especiais.
password	Texto	Senha do usuário.
phone_number	Texto	Número de telefone.
phone_number_confirmed	Lógico	Confirmação se o telefone foi validado.
security_stamp	Texto	Usado para rastrear as alterações feitas no perfil do usuário. É usado para fins de segurança quando as propriedades importantes de um usuário são alteradas, como a alteração da senha.
two_factor_enabled	Lógico	Habilita a autenticação de dois fatores.
user_name	Texto	Login do usuário.
theme	Imagem no Banco	Tema escolhido pelo usuário.
picture	Texto	Foto do usuário.

Role

Grupo.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
builtin	Lógico	Quando verdadeiro, o registro só pode ser alterado em tempo de desenvolvimento.
membership_enabled	Lógico	Quando verdadeiro, novos membros podem ser inseridos no grupo.
name	Texto	Nome do grupo.
normalized_name	Texto	Cria uma cópia normalizada do nome do grupo (função), permitindo que o nome tenha caracteres especiais.
application	Texto	Chave estrangeira da tabela Application.

Login

A entidade Login garante que um mesmo usuário possa logar com diferentes contas. Por exemplo, um usuário pode logar e acessar na sua conta do sistema usando sua conta do Facebook, Gmail ou do próprio sistema, essa última é gravada diretamente na tabela User.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
login_provider	Texto	Provedor da autenticação externa.
provider_display_name	Texto	Login do usuário no provedor externo,
provider_key	Texto	Senha do usuário no provedor externo.
user_id	Texto	Chave estrangeira da tabela User.

Securable

Permissionável.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
name	Texto	Nome do permissionável.
application	Texto	Chave estrangeira da tabela Application.

View

Aplica o controle de acesso (métodos de requisição HTTP) à diretórios e arquivos.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
securable_id	Texto	Chave estrangeira da tabela Securable.
pattern	Texto	Página ou local com o controle de acesso.
getMethodAllowed	Lógico	Permissão para o método get.
headMethodAllowed	Lógico	Permissão para o método head.
putMethodAllowed	Lógico	Permissão para o método put.
postMethodAllowed	Lógico	Permissão para o método post.
patchMethodAllowed	Lógico	Permissão para o método patch.
deleteMethodAllowed	Lógico	Permissão para o método delete.

optionsMethodAllowed	Lógico	Permissão para o método options.
traceMethodAllowed	Lógico	Permissão para o método trace.
application	Texto	Chave estrangeira da tabela Application.

Application

Aplicações. Ver mais detalhes em Multi aplicações.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
name	Texto	Nome da aplicação.

Role Securable

Tabela de relacionamento entre Grupo e Permissionável.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
role_id	Texto	Chave estrangeira da tabela Role.
securable_id	Texto	Chave estrangeira da tabela Securable.

User Securable

Tabela de relacionamento entre Usuário e Permissionável.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
securable_id	Texto	Chave estrangeira da tabela Securable.
user_id	Texto	Chave estrangeira da tabela User.

User Role

Tabela de relacionamento entre Usuário e Grupo.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
role_id	Texto	Chave estrangeira da tabela Role.
user_id	Texto	Chave estrangeira da tabela User.

Application User

Tabela de relacionamento entre Aplicação e Usuários. Permite relacionar aplicações e usuários a partir de uma mesma base de dados, ver mais detalhes em Multi aplicações.

Coluna do Banco	Tipo	Função
id	Texto	Chave primária.
application	Texto	Chave estrangeira da tabela Application.
user	Texto	Chave estrangeira da tabela User.

Ferramenta

Para gerenciar inicialmente essa estrutura, o Cronapp possui a ferramenta de **Permissão de Segurança** (Figura 2), ela deve ser usada para cadastrar as permissões, grupos e usuários que irão alimentar o sistema assim que ele for executado pela primeira vez, criando uma base de dados inicial. Após essa etapa, não é mais necessário o seu uso, podendo realizar ajustes de Grupos e usuários em tempo de execução. Os projetos modelos do Cronapp já possuem páginas para configuração de Grupos e usuários do sistema, acessíveis apenas por usuários que possuem o permissionável "Administrators".

Na prática, sempre que alteramos a ferramenta **Permissões de segurança** (Figura 2), alimentamos o ar quivo populate,json, responsável por preencher o banco de dados no primeiro carregamento do sistema. Acesse o tópico Estrutura de Classes para mais detalhes.

O arquivo `populate.json` alimenta o Banco de dados sempre que o projeto é executado no Cronapp. Se algum dado contido no arquivo for alterado no Banco de dados, como mudar a senha do usuário Administrador de "admin" para "novaSenha", ao parar e executar (debug) o projeto novamente, a senha retornará ao valor padrão, "admin". Veja abaixo como contornar isso.

- **Em modo Debug**: recomendamos renomear ou excluir o arquivo **populate.json** (Endereço: `src/main/java/app/populate.json`).
- **Exportar projeto (`*.war`)**: basta desmarcar a opção **Auto Popular Dados** na janela **Opções de Geração de War** (mais detalhes nas documentações Importar e exportar projetos ou Serviços de Cloud).

Para abrir a ferramenta, acesse no menu do sistema **Projeto > Permissão de Segurança**, como na figura 2.

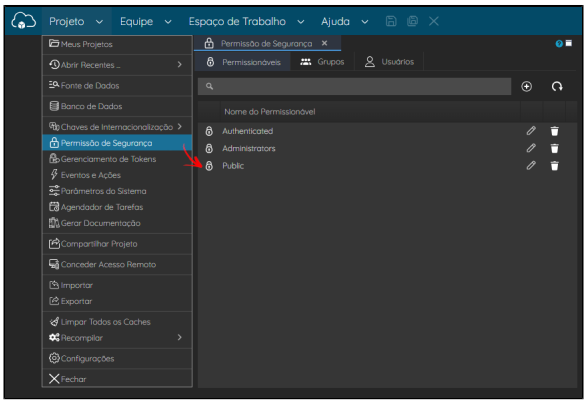

Figura 2 - Acesso a funcionalidade Permissão de Segurança

Aba Permissionáveis

Para adicionar um novo permissionável, clique no ícone "**+**" (destaque 1 da figura 2.1) para informar um nome e salvar um novo registro. Como informado anteriormente, não é possível renomear ou remover os permissionáveis nativos: **Public**, **Authenticated** e **Administrators**.

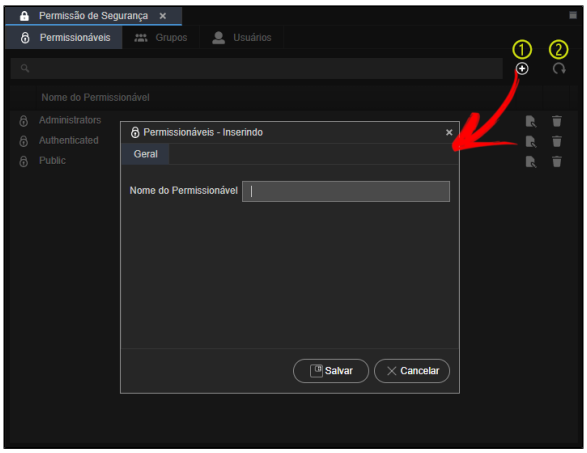

Figura 2.1 - Inserindo novo permissionável

1. **Adicionar**: abre a janela para inserir novo item;
2. **Atualizar**: recarrega a lista.

Após inserir, clique no ícone **Editar** (destaque 1 da figura 2.2) para abrir a janela de Edição, que agora possui 4 abas: **Geral**, **Visões**, **Grupos** e **Usuários**.

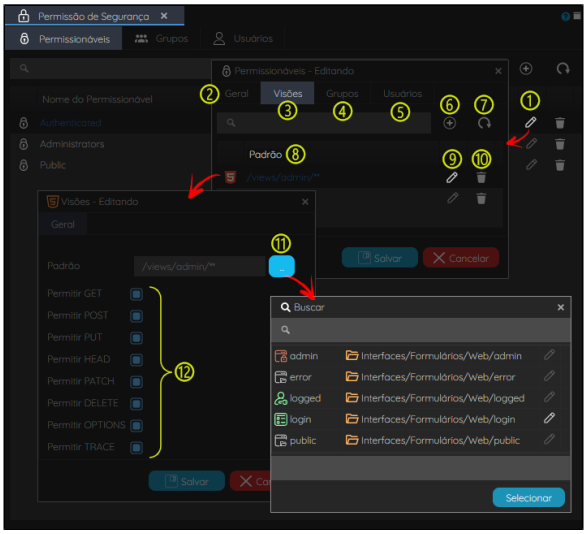

Figura 2.2 - Janelas de configuração dos permissionáveis

1. **Editar permissionável**: Abre a janela de edição.
2. **Aba Geral**: edita o nome do permissionável (Figura 2.1).
3. **Aba Visões**: permite cadastrar ou editar as views que esse permissionável terá acesso.
4. **Aba Grupos**: adiciona grupos ao permissionável. Essa aba só permite selecionar os grupos adicionados a partir da aba Grupos, criados em tempo de desenvolvimento. É fundamental que sempre haja um grupo associado ao permissionável **Administrators**.
5. **Aba Usuários**: adiciona usuários ao permissionável. Essa aba só permite selecionar os usuários adicionados a partir da aba Usuários, criados em tempo de desenvolvimento.
6. **Inserir visão**: abre a janela para informar um endereço e selecionar os métodos HTTP.
7. **Recarregar visão**: recarrega a lista.
8. **Coluna Padrão**: lista todos os endereços já cadastrados.
9. **Editar**: abre a janela para editar o endereço e alterar os métodos de requisição HTTP.
10. **Excluir**: apaga o endereço selecionado.
11. **Campo Padrão**: endereço que terá as permissões concedidas.
    Abra a janela de busca para selecionar um diretório ou arquivo. O "`/**`" ao final passa a mesma permissão para todo o conteúdo dentro do diretório (ex: `/views/<diretório>/**`).

12. **Métodos HTTP**: define os métodos de requisição HTTP autorizados para esse diretório ou conjunto de arquivos.

A janela **Buscar** (destaque 11 da figura 2.2) exibirá o conteúdo dentro do diretório **webapp** (Endereço: s`rc/main/webapp/`) da aplicação web e **www** (Endereço: `src/main/mobileapp/www/`) da aplicação mobile.

Os diretórios, arquivos e endereços exibidos na janela **Buscar** vão variar caso a opção Modo Avançado esteja habilitada ou não. Veja mais detalhes sobre esse recurso em Estrutura de arquivos.
Ao excluir um registro de permissionável, caso ele possua relacionamentos com grupos ou usuário, esses relacionamentos também serão excluídos no arquivo populate json.

Aba Grupos

Para criar um grupo, clique no ícone "**+**" (destaque 1 da figura 2.3) para informar seu nome. Após salvar, clique no ícone **Editar** (destaque 3 da figura 2.3) para abrir a janela de Edição, que agora, além da aba **Geral** exibirá a aba **Usuários**. É fundamental que sempre haja um grupo associado ao permissionável **Administrators**.

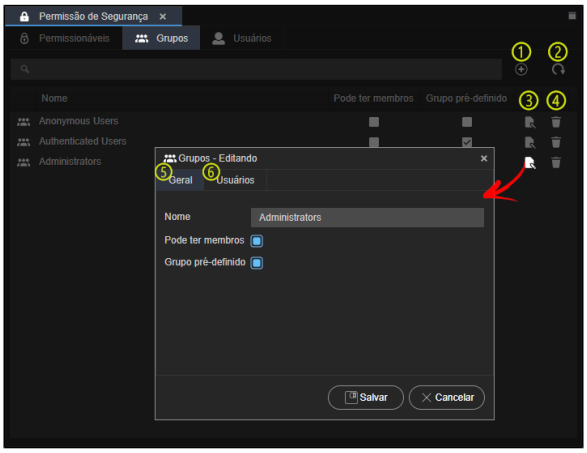

Figura 2.3 - Edição do grupo Administrators

1. **Adicionar**: abre a janela para inserir novo grupo.
2. **Atualizar**: recarrega a lista de grupos.
3. **Editar grupo**: abre a janela de edição.
4. **Excluir**: apaga o grupo selecionado. Se o grupo tiver qualquer tipo de relacionamento com permissionáveis ou usuários, o relacionamento também será excluído no arquivo populate json.
5. **Aba Geral**: edita o nome do grupo.
    - **Campo Nome**;
    - **Pode ter membros**: se ativo, o grupo poderá ter usuários associados;
    - **Grupo pré-definido**: quando ativo, não permite que o grupo seja editado em tempo de execução.
6. **Aba Usuários**: adiciona usuários ao grupo. Essa aba só permite selecionar os usuários adicionados a partir aba principal Usuários, criados em tempo de desenvolvimento.

Aba Usuários

Essa aba permite criar os usuários que irão alimentar o sistema em sua primeira execução. É fundamental que sempre haja um usuário associado ao permissionável **Administrators**.

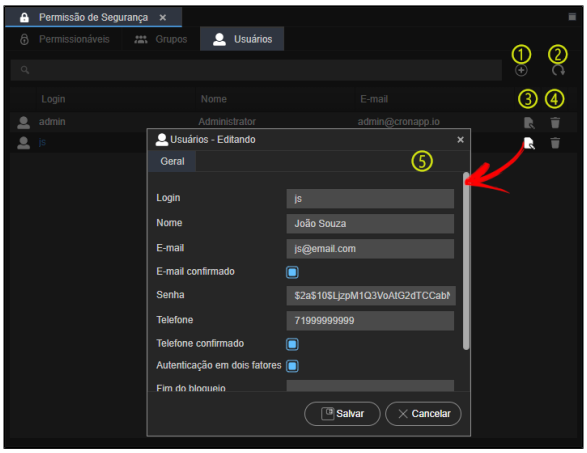

Figura 2.4 - Inserindo um novo usuário

1. **Adicionar**: abre a janela para inserir novo usuário.
2. **Atualizar**: recarrega a lista de usuários.
3. **Editar grupo**: abre a janela de edição.
4. **Excluir**: apaga o usuário selecionado. Se o usuário tiver qualquer tipo de relacionamento com permissionáveis ou grupos, o relacionamento também será excluído no arquivo populate json.
5. **Aba Geral**: possui os campos para cadastro de usuários.

Atribuir permissões

Após a criação dos permissionáveis, é possível definir as permissões de cada local dentro da aplicação e atribuir permissões para os usuários.

Em desenvolvimento

A propriedade **Segurança** está disponível em diferentes ferramentas dentro do Cronapp. O exemplo da figura 3 exibe a janela da propriedade **Segurança** do Editor de views, nela é possível exibir e habilitar um componente visual para determinados permissionáveis (acesse o tópico "Segurança" em Propriedades dos componentes visuais pra detalhes sobre os campos "Visível para", "Habilitado para" e "Renderizado Para").

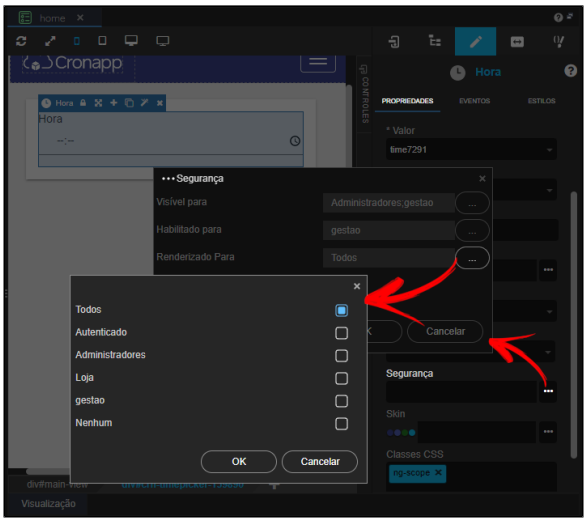

Figura 3 - Restringindo o acesso de um componente visual

Já no exemplo da figura 3.1, a propriedade **Segurança** está disponível na janela de **Propriedades do Bloco de programação**, nesse caso, é possível habilitar a execução e os recursos de CRUD ao chamar um bloco de programação.

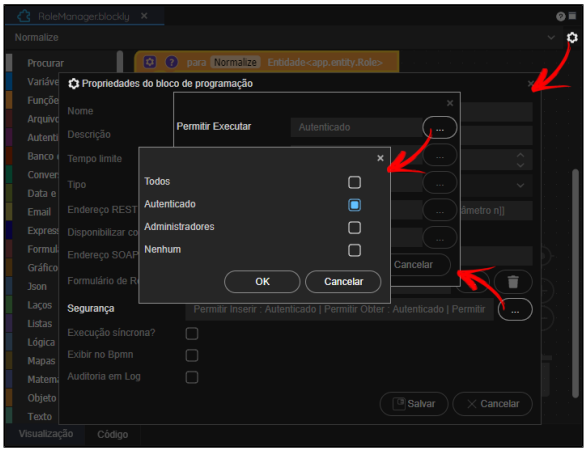

Figura 3.1 - Limitando acesso a um bloco de programação servidor a partir do permissionável

Em execução

Após logar com o perfil de administrador na aplicação, é possível visualizar o menu **Admin**, que dá acesso as páginas **Usuários** e **Grupos** (Figura 3.2). Através delas será possível definir permissionáveis aos grupos ou permissionáveis e grupos aos usuários.

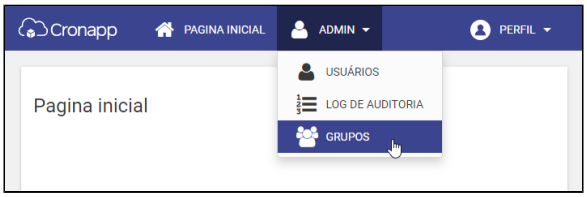

Figura 3.2 - Menu Administrativo do sistema

Ao criar ou editar um usuário do sistema, é possível definir um ou mais **Grupos** ou selecionar diretamente um ou mais **Permissionáveis** (Figura 3.3). O campo **Aplicações** permite selecionar apenas as aplicações que o usuário terá acesso, caso a base de dados possua Multi aplicações.

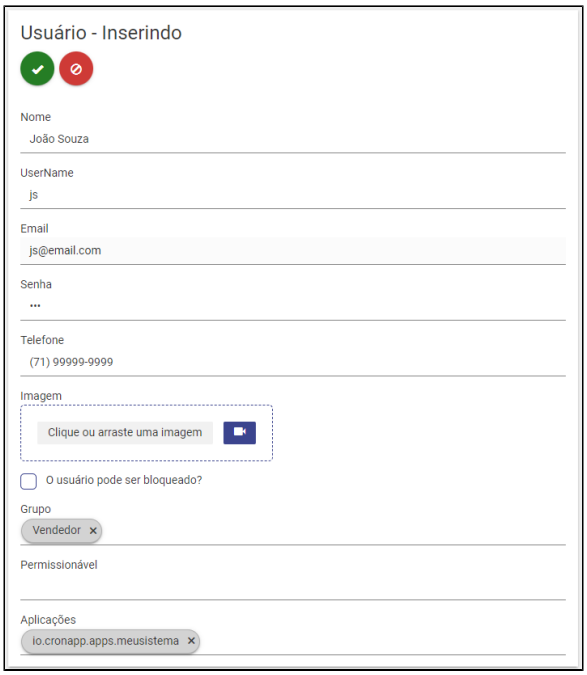

Figura 3.3 - Na edição do usuário é possível definir grupos (Funções) ou Permissionável

Ao acessar a página **Grupos**, é possível criar e editar grupos, definir quais permissionáveis estão relacionados a esse grupo e os usuários que fazem parte (Figura 3.4).

Figura 3.4 - Seleção de permissionáveis e usuários na edição do grupo

Bloquear usuário

Para impedir que um usuário tenha acesso ao sistema, é possível remover a aplicação (`Application`) associada ao usuário na tabela `ApplicationUser` (mais detalhes em Multi Aplicações) caso a permissão de acesso seja explicita, ou retirar os grupos (Roles) associadas ao usuário através da tabela (`UserRoles`).

O recomendado é que tenha apenas um grupo (`Role`) que dá permissão de acesso de um usuário a um sistema específico.

É importante destacar que toda permissão de acesso ao sistema é ignorada se o usuário for administrador geral, isto é, ele ter o permissionável (Securable) **Administrator** associado.