

# Estrutura de arquivos

Projetos criados no Cronapp possuem muitos arquivos e diretórios, todo esse conteúdo é responsável por gerar um projeto base e manter uma estrutura de fácil desenvolvimento. Dessa forma, desenvolvedores Cronapp não necessitam conhecer as diversas bibliotecas utilizadas em seu projeto, pois o Cronapp encapsula isso em recursos visuais de forma simples, permitindo rápido desenvolvimento e baixa curva de aprendizado.

Veremos aqui como está organizado a estrutura de arquivos dos projetos modelos que contém as aplicações web e mobile e quais mudanças visuais ocorrem ao habilitar/desabilitar a opção **Modo Avançado** (destaque 2 da figura 1).

A estrutura de arquivos durante o desenvolvimento do projeto é um pouco diferente da estrutura contida após compilar o projeto (arquivo \*.war), usado em produção. Veja mais detalhes em [Arquivo](#).

## Opção Modo Avançado

A árvore de arquivos ([figura 1.1](#)) exibe os diretórios e arquivos pertencentes ao seu projeto. Porém, a forma como o conteúdo será exibido dependerá da configuração da opção **Modo Avançado**. O objetivo dessa opção é simplificar ainda mais o desenvolvimento no Cronapp. Para alternar entre os modos, clique no **botão do Menu do sistema** (destaque 1 da figura 1) e mude a posição do botão alternância **Modo Avançado** (2 da figura 1). Também é possível alterar entre os modos a partir do menu do sistema (destaque 3): **Espaço de Trabalho > Modo de uso da IDE > Modo Analista** para desabilitar ou **Espaço de Trabalho > Modo de uso da IDE > Modo Avançado** para habilitar.

A alternância entre os modos pode ser feita a qualquer momento durante o desenvolvimento do projeto e **não** gera qualquer impacto no resultado final. Assim, mesmo que a estrutura de arquivos seja exibida de forma simplificada, ao exportar seu projeto, a hierarquia de arquivos estará igual ao apresentado no Modo Avançado, pois trata-se da estrutura real.

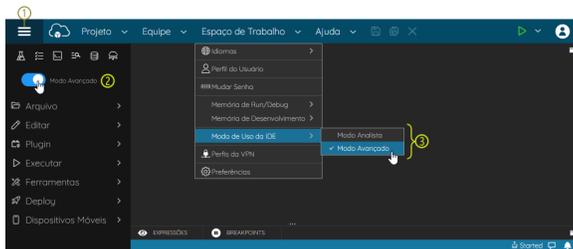


Figura 1 - Alterando entre as opções de Modo Avançado

O que muda quando alteramos o **Modo Avançado**:

- **Desabilitado:**
  - Exibe uma **Árvore de recursos**, que apresenta o conteúdo de forma simplificada, organizando os arquivos com base em suas funcionalidades. Isso significa que arquivos com propósitos semelhantes, abertos a partir da mesma ferramenta, ficam agrupados no mesmo diretório (lado esquerdo da figura 1.1).
  - Apenas arquivos com recursos **low-code** são exibidos.
  - As extensões de alguns tipos de arquivos ficarão ocultas, melhorando a legibilidade dos arquivos na árvore.
  - O caminho do diretório raiz até o arquivo é exibido em seu **endereço virtual** ou **localização**.
    - **Localização:** Formulários/Mobile/Autenticado/home
  - O usuário não será consultado quando for necessário substituir ou atualizar arquivos do projeto.
  - A aba inferior "Console do depurador" ficará recolhida após executar o projeto.
  - Não será exibida a opção "Novo Bloco de Programação Java/JavaScript (High-code)" durante a criação dos blocos de programação cliente e servidor.
- **Habilitado:**
  - Exibe uma **Árvore de arquivos**, mostrando a estrutura hierárquica real dos arquivos do projeto, incluindo todos os diretórios e seus conteúdos (lado direito da figura 1.1).

Nesta página

- [Opção Modo Avançado](#)
  - [Arquivos no modo Analista](#)
  - [Arquivos no modo Avançado](#)
    - [Diretório raiz](#)
      - [java](#)
      - [mobileapp](#)
        - [Diretório mobile](#)
        - [Diretório bile](#)
        - [Diretório www](#)
    - [Resources](#)
    - [webapp](#)
- [Menu de contexto](#)
  - [Raiz do projeto](#)
  - [Diretórios e arquivos](#)
  - [Conteúdo específico](#)
    - [Arquivo pom.xml](#)
    - [Arquivos package.json](#)
    - [Arquivos javascript e css](#)
    - [Menu Ação](#)
- [Janela de propriedades](#)
  - [Arquivo](#)
  - [Diretório](#)

Saiba mais

- [Criar projeto](#)
- [Abrir projeto](#)
- [Configurações do projeto](#)
- [Importar e exportar projetos](#)

Arquivos ocultos dos sistema ou utilizados para a execução interna do projeto podem não ser exibidos.

- Todos os arquivos e diretórios são exibidos.
- As extensões dos arquivos são exibidas.
- O caminho do diretório raiz até o arquivo é exibido em seu **endereço real**.
  - **Endereço:** `/src/main/mobileapp/www/views/logged/home.view.html`
- O usuário será consultado sempre que um arquivo necessitar ser substituído ou atualizado.
- A aba inferior "Console do depurador" será expandida automaticamente após rodar o projeto em modo *Debug*.
- Será exibida a opção "Novo Bloco de Programação Java/JavaScript (High-code)" durante a criação dos blocos de programação cliente e servidor.

As duas estruturas na figura abaixo estão com o mesmo arquivo "Bloco.blockly" selecionado. Porém, a **árvore de recursos** da esquerda (Modo Avançado desabilitado) exibe o conteúdo de forma muito mais simples e resumida que a **árvore de arquivos** da direita (Modo Avançado habilitado).

Acesse a documentação [Tour pelo Cronapp](#) para mais informações sobre os botões de atalho lateral e do campo de busca.

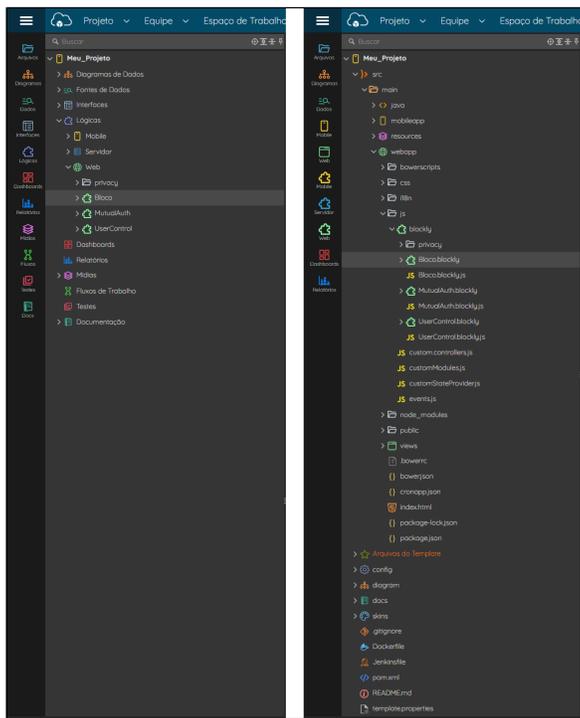


Figura 1.1 - Diferença ao visualizar os arquivos com a opção Modo Avançado desabilitada e habilitada

Ao criar ou mover arquivos, certifique-se que ele corresponde àquele diretório - por exemplo, na pasta `Blockly` só deve conter os arquivos pertencentes aos blocos de programação, assim como os arquivos do diagrama de dados devem estar na pasta `diagram` e assim por diante.

Além de melhorar a organização do seu projeto, diversas funcionalidades esperam encontrar tipos específicos de arquivos em sua pasta correspondente, caso isso não ocorra, esses arquivos podem não ser mapeados ao visualizar com o Modo Avançado desabilitado.

## Arquivos no modo Analista

Ao **desativar** o **Modo Avançado**, a árvore do projeto não exibe os arquivos na pasta raiz, apenas os diretórios dos principais recursos *low-code*. As pastas estão ordenadas com base na recomendação de desenvolvimento de projetos no Cronapp.

A pasta principal (raiz) possui o nome do projeto e um ícone que identifica se o projeto é mobile e web (ícone smartphone, figura 2), apenas web (ícone monitor) ou microsserviços (ícone engrenagem). Descrevemos abaixo a função dos diretórios que estão logo abaixo do nó raiz do projeto, quando a opção **Modo Avançado** está **desativada**.

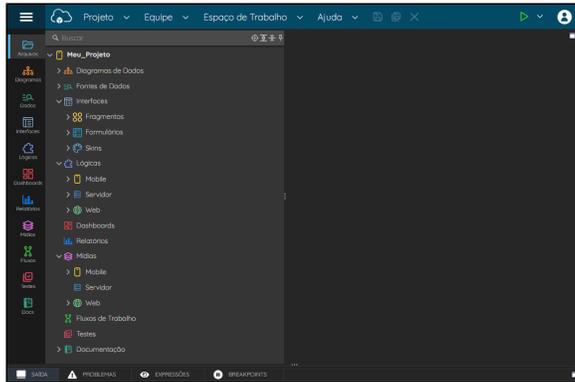


Figura 2 - Diretórios do projeto quando o Modo Avançado está desativado

Alguns dos diretórios listados abaixo são virtuais, ou seja, não existem no projeto. São visíveis apenas quando a opção **Modo Avançado** está **desativada** e tem o intuito de agrupar conteúdo com a mesma funcionalidade. Diretórios virtuais não exibem menu de contexto ao clicar com o botão direito do mouse.

- **Diagrama de dados** : contém os **Diagramas de classe** do projeto, expandindo o diagrama, é possível visualizar suas classes e atributos.
- **Fontes de Dados** : lista as **Fontes de dados** personalizadas do projeto.
- **Interfaces**: diretório virtual para agrupar os recursos relacionados aos Formulários, Fragmentos de formulários e Skins.
  - **Fragmentos**: diretório virtual para agrupar as pastas dos **Fragmentos mobile e web**.
    - **Mobile** : contém os arquivos de fragmento *mobile*.
    - **Web** : contém os arquivos de fragmento *web*.
  - **Formulários**: diretório virtual para agrupar as pastas de **Formulários** dos projetos *mobile* e *web*.
    - **Mobile** : diretório que organiza as telas da aplicação *mobile*,
    - **Web** : diretório onde ficam organizadas as páginas *web*.
  - **Skins** : contém os arquivos de **skins** criados para o projeto,
- **Lógicas**: diretório virtual para agrupar as pastas dos **blocos de programação mobile**, servidor e *web*.
  - **Mobile** : contém os arquivos de bloco de programação *mobile* (cliente).
  - **Servidor** : contém os arquivos de bloco de programação servidor (*back-end*).
  - **Web** : contém os arquivos de bloco de programação *web* (cliente).
- **Dashboards** : armazena os **Dashboards** criados no projeto.
- **Relatórios**: armazena os **Relatórios** criados no projeto.
- **Mídia**: diretório virtual para agrupar arquivos estáticos (ex.: imagens, PDFs, vídeos e outros).
  - **Mobile** : contém os arquivos do projeto utilizados na aplicação *mobile*.
  - **Servidor**: contém os arquivos do projeto mantidos no *back-end* da aplicação.
  - **Web**: contém os arquivos do projeto utilizados na aplicação *web*.
- **Fluxo de Trabalho** : contém os arquivos de fluxo de trabalho do **Cronapp BPM**.
- **Testes** : pasta reservada para os blocos de programação de **teste unitários** do projeto servidor.
- **Documentação** : diretório usado para manter os arquivos de documentação do projeto, nele você poderá criar diagramas UML, fluxogramas e diversos recursos para ajudar no desenvolvimento do projeto. Ao utilizar a ferramenta **Gerar documentação**, seu conteúdo também será criado nessa pasta.

## Arquivos no modo Avançado

Com o **Modo Avançado** **habilitado**, a árvore do projeto exibe todos os arquivos, incluindo o conteúdo *high-code*. As pastas serão ordenadas com base na hierarquia de diretórios do projeto.



Por padrão, o diretório `skins (/skins/)` não existe após criar um projeto baseado no modelo. Sendo possível tratar de duas formas:

1. Com Modo Avançado habilitado, crie um diretório (pacote) de nome "skins" na raiz do projeto, em /
  2. Com Modo Avançado desabilitado, crie ou adicione um arquivo ou diretório dentro de `Skins/`
- **.gitignore**: estão descritos os arquivos que serão ignorados na hora de fazer o `commit / push` para o git.
  - **Dockerfile**: arquivo de configuração do Docker.
  - **Jenkinsfile**: arquivo de configuração ao utilizar esteira DevOps.
  - **pom.xml**: ferramenta que gerencia os `plugins` e bibliotecas Maven do projeto.
  - **README.md**: arquivo de texto que apresenta um pequeno tutorial de importar projeto GIT. O conteúdo é exibido na tela inicial do seu projeto na página git.
  - **template.properties**: propriedades com alguns dados do projeto.

## java

Dentro desse diretório estão contidas as camadas DAO e Entity, internacionalização, relatórios e dashboards, fonte de dados, além de diversos arquivos usados pelo servidor da aplicação.

Ao criar o projeto, os diretórios **reports** e **dashboards** não existem, para criá-los, clique nos atalhos laterais (destaques 1 e 2 da figura 3.1).

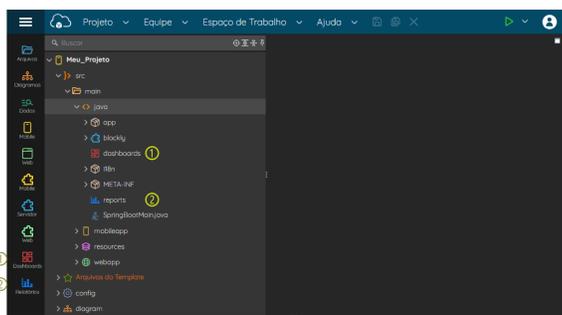


Figura 3.1 - Arquivos do Servidor da Aplicação

Os subdiretórios e arquivos da pasta **java** (Endereço: `/src/main/java/`):

- **app**: diretório do `namespace` principal do servidor.
  - **dao**: contém os arquivos responsáveis pela comunicação das entidades com o banco de dados.
  - **entity**: diretório que contém as classes das entidades da aplicação.
  - **AppConfiguration.java**: arquivo de configurações iniciais do projeto Java.
  - **populate.json**: arquivo que contém os dados para popular o banco no momento que ele é iniciado pela primeira vez. Esse conteúdo é configurado pela ferramenta de [Permissão de segurança](#).
- **blockly**: diretório que contém os arquivos com as funções de blocos de programação do servidor e os arquivos Java gerados pelos blocos. Abaixo listamos os arquivos de blocos existentes no projeto modelo e suas respectivas funções.
  - **Application.blockly**
    1. **Current**: obtém o id da aplicação em execução.
  - **RoleManager.blockly**
    1. **BeforeInsert**: trata a entidade `Role` antes de inserir o registro no banco de dados.
    2. **BeforeUpdate**: trata a entidade `Role` antes de atualizar seu registro no banco de dados.
    3. **Normalize**: trata o campo "normalizedName" da entidade `Role`.
  - **UserControl.blockly**
    1. **signUp**: obtém os dados do novo usuário para realizar sua inscrição no sistema.
    2. **resetPassword**: utilizado no processo de redefinição de senha por e-mail. Veja mais detalhes sobre essa configuração em [Recuperar senha por e-mail](#).
  - **UserManager.blockly**
    1. **BeforeInsert**: trata a entidade `User` antes de inserir o registro no banco de dados.
    2. **BeforeUpdate**: trata a entidade `User` antes de atualizar seu registro no banco de dados.

3. **BeforeDelete**: trata a entidade `User` antes de remover seu registro no banco de dados e verifica se o solicitante possui autorização para realizar essa ação.
  4. **Normalize**: trata os campos "userName" e "email" da entidade `User`.
- **dashboard**: diretório que armazena os arquivos criados a partir da ferramenta de Dashboard do Cronapp. Por padrão, essa pasta não existe quando o projeto baseado no modelo é criado, para criá-la, clique em seu atalho (destaque 1 da Figura 3.1) no menu lateral.
  - **i18n**: diretório que contém as propriedades de internacionalização no lado servidor do projeto.
  - **META-INF**: diretório relacionado aos arquivos do tipo JAR, para configurar aplicações, extensões, carregadores de classes e serviços:
    - **datasources**: diretório que contém todas as Fontes de dados personalizadas do projeto.
    - **persistence.xml**: arquivo utilizado pelo JPA.
  - **reports**: diretório que armazena os arquivos criados a partir da ferramenta de Relatórios do Cronapp. Por padrão, essa pasta não existe quando o projeto baseado no modelo é criado, para criá-la, clique em seu atalho (destaque 2 da Figura 3.1) no menu lateral.
  - **SpringBootMain.java**: classe que inicia o contexto de aplicação do SpringBoot.

## mobileapp

Diretório da aplicação *mobile* (Android e iOS).

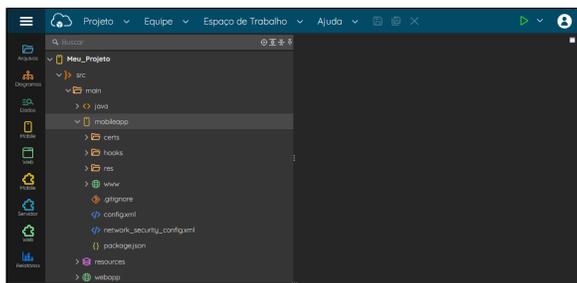


Figura 3.2 - Estrutura do diretório Códigos Fonte Mobile

Os subdiretórios e arquivos da pasta **mobile** (Endereço: `/src/main/mobileapp/`):

- **certs**: (*Certificate Enrollment Requests*) esse diretório deve conter os certificados que foram requisitados e estão com status pendentes ou já foram rejeitados.
- **hooks**: representa o diretório de scripts especiais que podem ser adicionados ao sistema para customizar comandos Cordova, que é uma estrutura apache de desenvolvimento de aplicativos móveis.
- **res**: pasta com os recursos estáticos da aplicação. Esse conteúdo é manipulado a partir da janela de Configurações dos [Dispositivos móveis](#).
  - **icon**: ícones utilizados na aplicação mobile.
  - **screen**: plano de fundo da aplicação mobile.
- **www**: armazena os arquivos da aplicação *mobile*.
- **.gitignore**: estão descritos os arquivos que serão ignorados na hora de fazer o *commit / push* para o git.
- **config.xml**: arquivo de configuração global que controla aspectos de uma aplicação Cordova. Esse conteúdo é manipulado a partir da janela de Configurações dos [Dispositivos móveis](#).
- **network\_security\_config.xml**: arquivo utilizado para a configuração de rede em aplicações mobile.
- **package-lock.json**: contém informações sobre a aplicação e sua versão. O pacote com as dependências da aplicação mobile fica no diretório [www](#).
- **package.json**: contém informações sobre a aplicação e sua versão. O pacote com as dependências da aplicação mobile fica no diretório [www](#).

### Diretório mobile www

Subdiretório que armazena diversos recursos da aplicação *mobile*, como telas, imagens, os blocos de programação que geram código fonte JavaScript, internacionalização e mais.

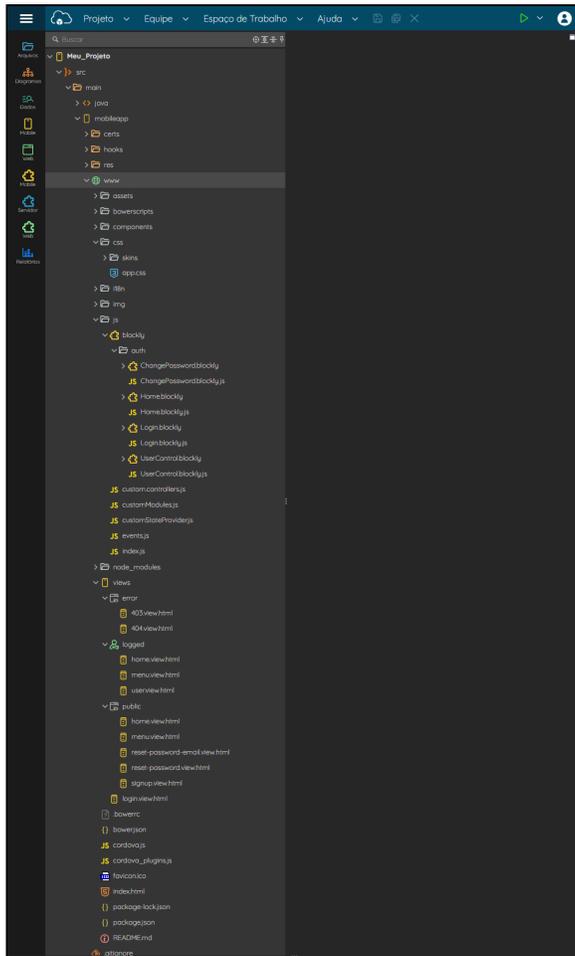


Figura 3.3 - Estrutura da pasta www da aplicação mobile

Os subdiretórios e arquivos da pasta **www** (Endereço: `src/main/mobileapp/www/`):

- **assets**: diretório responsável por manter arquivos estáticos da aplicação mobile.
- **bowerscripts**: arquivos referentes ao Bower, uma ferramenta que ajuda no gerenciamento das dependências do *front-end*.
- **components**: diretório onde são mantidos os arquivos de **fragmentos** da aplicação *mobile*.

Por padrão, o diretório **components** (`/src/main/mobileapp/www/components/`) não existe após criar um projeto baseado no modelo. Sendo possível tratar de duas formas:

1. Com Modo Avançado habilitado, crie um diretório (pacote) de nome "components" em `/src/main/mobileapp/www/`
2. Com Modo Avançado desabilitado, crie ou adicione um arquivo ou diretório dentro de `Interface/Fragmentos/Mobile/`

- **css**: local onde ficam armazenados as estilizações em CSS.
  - **skin**: diretório com os arquivos `*.css` gerados pela ferramenta [Editor de Skins](#).

Por padrão, o diretório **skin** (`/src/main/mobileapp/www/css/skin/`) não existe após criar um projeto baseado no modelo e será criado automaticamente após gerar o primeiro arquivo de Skin do projeto.

- **app.css**: arquivo `*.css` com as classes geradas pelo editor visual de estilos.
- **i18n**: diretório onde ficam os arquivos de internacionalização da aplicação *mobile*.
- **img**: local usado para armazenar imagens usadas na aplicação.
- **js**: diretório que mantém os arquivos blockly e JavaScript responsáveis por tratar as rotas Angular, autenticação e regras de negócios do sistema.
  - **blockly**: diretório que contém os arquivos com as funções de blocos de programação cliente e os arquivos JavaScript gerados pelos blocos. Abaixo listamos os arquivos de blocos existentes no projeto modelo e suas respectivas funções.
    - **auth**: diretório que agrupa as regras de negócios da autenticação nas aplicações mobile.

- **ChangePassword.blockly**
    1. `changePassword`: executa as validações necessárias e altera a senha do usuário.
    2. `closeModal`: fecha o modal que permite o usuário alterar sua senha na tela `menu.view.html`.
    3. `openModal`: abre o modal que permite o usuário alterar sua senha na tela `menu.view.html`.
  - **Home.blockly**
    1. `change`: responsável por direcionar o usuário para a tela definida no campo "Rota Principal Mobile" nas [Configurações do Projeto](#).
  - **Login.blockly**
    1. `login`: responsável por fazer as validações necessárias e direcionar os dados passados pelo usuário para o login no sistema.
  - **UserControl.blockly**
    1. `isValidSignup`: executado a partir da função `signup`, realiza as primeiras validações dos dados do novo usuário.
    2. `resetPassword`: utilizado no processo de redefinição de senha por e-mail na aplicação mobile. Veja mais detalhes sobre essa configuração em [Recuperar senha por e-mail](#).
    3. `signup`: obtém os dados do novo usuário para realizar as primeiras validações e direciona para as regras de negócio do servidor da aplicação.
  - **node\_modules**: local onde ficam armazenadas diversas bibliotecas usadas pelo Cronapp apenas no período de desenvolvimento do projeto. Durante as atualizações do Cronapp, o conteúdo dessa pasta é alterado ou substituído, por isso, não recomendamos atualizar ou adicionar nada manualmente.
  - **views**: diretório que organiza as telas da aplicação *mobile*:
    - **error**: formulários de tratamento de erros da aplicação.
      - **403**: o usuário é direcionado para essa tela caso tente acessar algo que ele não possua autorização. Nesta tela há três botões: **Tentar novamente**, recarrega a tela; **Voltar**, retorna a tela anterior e **Ir ao início**, volta pra a tela *home* da aplicação.
      - **404**: o usuário é direcionado para essa tela caso tente acessar uma tela que não exista na aplicação. Nesta tela há três botões: **Tentar novamente**, recarrega a tela; **Voltar**, retorna a tela anterior e **Ir ao início**, volta pra a tela *home* da aplicação.
    - **logged**: contém as telas com permissão de acesso apenas para os usuários logados no sistema.
      - **home**: tela principal, acessível após o *login* na aplicação mobile.
      - **menu**: tela exibida sempre que o ícone sanduíche do componente [Cabeçalho](#) for selecionado. Possui o modal para alteração de senhas.
      - **user**: acessível apenas por usuários com perfil administrador, sendo possível exibir, editar e excluir os usuários do sistema.
    - **public**: telas sem restrição de acesso.
      - **home**: tela pública principal.
      - **menu**: tela exibida sempre que o ícone sanduíche do componente [Cabeçalho](#) das telas públicas for selecionado.
      - **reset-password-email**: utilizada para a recuperação de senha do usuário a partir do e-mail cadastrado.
      - **reset-password**: após receber um e-mail com o código de autenticação OTP, o usuário poderá informar o código e a nova senha.
      - **signup**: permite que o usuário faça seu próprio cadastro.
    - **login.view.html**: por padrão é a primeira tela exibida para o usuário. Acesse a documentação [Alterar página/tela inicial](#) caso queira mudar a tela inicial da sua aplicação.
  - **.bowerrc**: pacote de gerenciamento dos componentes *mobile*, bibliotecas, *frameworks*.
  - **bower.json**: arquivo JSON de gerenciamento das dependências do projeto.
  - **cordova.js**: arquivo JavaScript com as principais funções do Cordova.
  - **cordova\_plugins**: arquivos JavaScript que exporta os *plugins* do Cordova.
  - **favicon.ico**: ícone do Cordova.
  - **index.html**: contém as referências de *plugins*, recursos JavaScript e CSS usados nas telas da aplicação mobile.
  - **package-lock.json**: contém os metadados da aplicação e as dependências NPM (Node Package Manager) exigidas pela aplicação.
  - **package.json**: contém as versões exatas das dependências NPM (Node Package Manager) exigidas pela aplicação.
- As dependências NPM podem ser recompiladas a partir da opção "Iniciar Projeto" no menu de contexto deste arquivo ou via menu do sistema: **Projeto > Recompilar > Recompilar NPM Mobile**.
- **README.md**: arquivo de texto com descrição padrão dos recursos.

## Resources

Diretório servidor que contém arquivos estáticos e de configurações do servidor.

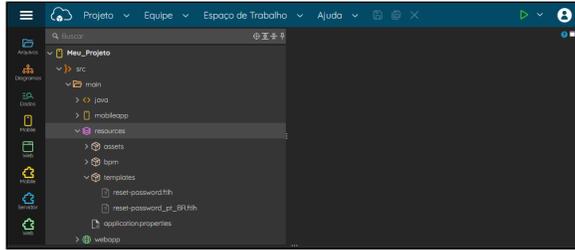


Figura 3.4 - Estrutura da pasta resources

Os subdiretórios e arquivos da pasta **resources** (Endereço: `/src/main/resources/`):

- **assets**: diretório responsável por manter os arquivos estáticos e de mídia do servidor.

Por padrão, o diretório **assets** (`/src/main/resources/assets/`) não existe após criar um projeto baseado no modelo. Sendo possível tratar de duas formas:

1. Com Modo Avançado habilitado, crie um diretório (pacote) de nome "assets" em `/src/main/resources/`
2. Com Modo Avançado desabilitado, crie ou adicione um arquivo ou diretório dentro de Mídias/Servidor/

- **bpm**: contém os arquivos de fluxo de trabalho do Cronapp BPM.

Por padrão, o diretório **bpm** (`/src/main/resources/bpm/`) não existe após criar um projeto baseado no modelo. Sendo possível tratar de duas formas:

1. Com Modo Avançado habilitado, crie um diretório (pacote) de nome "bpm" em `/src/main/resources/`
2. Com Modo Avançado desabilitado, crie ou adicione um arquivo ou diretório dentro de Fluxo de Trabalho/

- **templates**: diretório responsável por manter alguns arquivos de templates do projeto. Por padrão, essa pasta contém os templates dos **e-mails enviados na recuperação de senha** da aplicação.
- **application.properties**: arquivo que contém algumas definições das **Configurações do Projeto**.

## webapp

Diretório que armazena a estrutura da aplicação *web*, como as páginas, os blocos de programação que geram código fonte JavaScript, internacionalização e outros.



3. **setPreference**: oculta a barra de cookies, define as preferências do usuário e oculta o modal.
  4. **checkCookieAnalytics**: verifica se o usuário permitiu o uso da opção de análise nos cookies.
  5. **checkCookieEssentials**: verifica se a opção de cookies essenciais, que é obrigatório, está definido como verdadeiro.
  6. **checkCookies**: verifica se o usuário já autorizou os cookies essenciais para não exibir mais o modal.
- **MutualAuth.blockly**: funções utilizadas para o recurso de [Autenticação mútua](#).
    1. **login**: executa o procedimento necessário para o *login* de um usuário a partir da autenticação mútua.
    2. **signup**: executa o procedimento necessário para o cadastro de um usuário a partir da autenticação mútua.
  - **UserControl**
    1. **resetPassword**: utilizado no processo de redefinição de senha por e-mail na aplicação web. Veja mais detalhes sobre essa configuração em [Recuperar senha por e-mail](#).
    2. **signup**: obtém os dados do novo usuário, realiza as primeiras validações e direciona para as regras de negócio no servidor da aplicação.
    3. **isValidSignup**: executado a partir da função `signup`, realiza as primeiras validações dos dados do novo usuário.
- **node\_modules**: local onde ficam armazenadas diversas bibliotecas usadas pelo Cronapp apenas no período de desenvolvimento do projeto. Durante as atualizações do Cronapp, o conteúdo dessa pasta é alterado ou substituído, por isso não recomendamos atualizar ou adicionar nada manualmente.
  - **public**: diretório com recursos de acesso público.
    - **assets**: diretório responsável por manter arquivos estáticos da aplicação web.
  - **views**: diretório onde ficam as páginas *web*:
    - **admin**: contém as páginas com permissão de acesso apenas para os usuários com perfil de administrador do sistema.
      - **auditlog.search**: exibe o histórico de eventos ocorridos por entidades, blocos de programação e fonte de dados. Veja outros detalhes no tópico "Página Log de auditoria" em [Projeto modelo](#).
      - **role**: permite gerenciar os grupos relacionados à [Permissão de Segurança](#) do sistema.
      - **user**: permite gerenciar os usuários do sistema. Veja outros detalhes no tópico "Página usuário" em [Projeto modelo](#).
    - **error**:
      - **403**: o usuário é direcionado para essa página caso tente acessar algo que ele não possua autorização. Nesta página há três botões: **Tentar novamente**, recarrega a página; **Voltar**, retorna a página anterior e **Ir ao início**, volta pra a página *home* da aplicação.
      - **404**: o usuário é direcionado para essa página caso tente acessar uma página que não exista na aplicação. Nesta página há três botões: **Tentar novamente**, recarrega a página; **Voltar**, retorna a página anterior e **Ir ao início**, volta pra a página *home* da aplicação.
    - **Logged**: contém páginas acessíveis apenas por usuários logados no sistema.
      - **home**: página principal, acessível após o *login* na aplicação web. Possui o modal para alteração de senhas e algumas características que permitem que parte do seu conteúdo sejam visíveis em outras páginas da aplicação web. Veja outros detalhes no tópico "Página home" em [Projeto modelo](#).
    - **public**: contém páginas sem restrição de acesso.
      - **privacy**: diretório.
        - **cookies**: contém a página utilizada nas definições de Cookies do usuário. Para mais detalhes sobre esse recurso, acesse a documentação [Cookies](#). Possui o modal de preferências do usuário.
      - **home**: página pública principal.
      - **reset-password**: página utilizada na redefinição de senha do usuário. Veja mais detalhes sobre essa configuração em [Recuperar senha por e-mail](#).
      - **signup**: permite que o usuário faça seu próprio cadastro.
    - **login.view.html**: por padrão é a primeira tela exibida para o usuário. Acesse a documentação [Alterar página/tela inicial](#), caso queira mudar a tela inicial da sua aplicação. Possui o modal para informar o e-mail do usuário e redefinir sua senha. Veja mais detalhes sobre essa configuração em [Recuperar senha por e-mail](#).
  - **.bowerrc**: pacote de gerenciamento dos componentes *web*, bibliotecas, *frameworks*.
  - **bower.json**: arquivo JSON de gerenciamento das dependências do projeto.
  - **index.html**: contém as referências de *plugins*, recursos JavaScript e CSS usados nas telas da aplicação web.
  - **package-lock.json**: contém os metadados da aplicação e as dependências npm (Node Package Manager) exigidas pela aplicação.
  - **package.json**: contém as versões exatas das dependências npm (Node Package Manager) exigidas pela aplicação.

# Menu de contexto

Existem algumas ações que podem ser feitas ao clicar com o botão direito do mouse na raiz do projeto (nome do projeto) ou em diretórios e arquivos. O conteúdo exibido no menu de contexto pode variar com o tipo de projeto (servidor, *mobile* e *web*, microsserviço ou *high-code*) ou se a opção **Modo Avançado** está habilitada ou não.

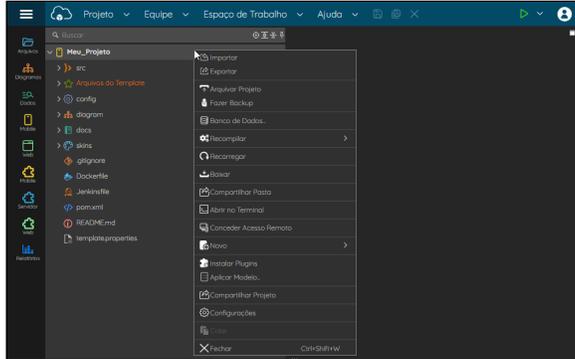


Figura 4 - Menu de contexto da raiz do projeto

## Raiz do projeto

As opções abaixo são exibidas ao acessar o menu de contexto da raiz do projeto.

- **Importar:** importa um projeto Cronapp em formato \*.zip ou recursos OpenApi/Swagger e SOAP.
- **Exportar:** exporta o projeto em um dos seguintes formatos: Git, war, zip ou [bibliotecas Cronapp](#).
- **Arquivar Projeto:** arquivava o projeto, seus arquivos serão mantidos no sistema, porém suas funcionalidades estarão desativadas até abri-lo novamente.
- **Fazer Backup:** cria uma cópia do projeto no estado atual. Acesse o *backup* através da funcionalidade [Gerenciamento de Backup](#).
- **Banco de Dados:** abre a ferramenta de configuração do banco de dados do projeto.
- **Recompilar:** opções de compilação. Essas opções também podem ser acessadas a partir do menu do sistema **Projeto > Recompilar**.
  - **Recompilar Projeto:** compila o lado servidor do projeto.
  - **Recompilar e Reabrir Projeto:** compila o lado servidor, e em seguida, fecha e abre o projeto.
  - **Recompilar Maven:** permite executar comandos Maven. O comando padrão força a resolução das dependências Maven (`pom.xml`).
  - **Recompilar NPM Web:** compila as dependências NPM da aplicação web (`package.json`).
  - **Recompilar NPM Mobile:** compila as dependências NPM da aplicação mobile (`package.json`).
- **Recarregar:** carrega novamente a árvore de arquivos ou diretório selecionado.
- **Baixar:** faz o download do conteúdo selecionado, no caso de diretórios, o mesmo é compactado em formato \*.zip.
- **Compartilhar Pasta:** abre uma janela que disponibiliza um *link* que permite que o usuário faça o download dessa pasta. Também é possível enviar esse *link* por *e-mail*.
- **Abrir no Terminal:** abre o diretório selecionado no terminal Linux.
- **Conceder acesso remoto:** abre uma janela que gera um *link* para permitir o acesso remoto por outro usuário, o Cronapp solicitará sua autorização quando o novo usuário abrir o projeto. Também é possível enviar o link de compartilhamento por *e-mail*, basta inserir os *e-mails* que deseja enviar o link.
- **Novo:** cria ou realiza o upload de um novo arquivo ao diretório selecionado.
  - **Fazer upload:** faz *upload* de um arquivo para o diretório selecionado.
  - **Fazer upload de um zip:** faz *upload* de um \*.zip para o diretório selecionado, esse processo descompacta automaticamente o conteúdo.
  - **Fazer upload de um link:** permite realizar o *upload* de um arquivo passando o seu endereço. No caso de arquivo compactado (\*.zip), o conteúdo será descompactado no diretório selecionado.

Utilize essa opção para obter conteúdos gerados a partir da funcionalidade [Compartilhar arquivos ou pastas](#).

A depender do diretório selecionado, alguns arquivos podem não ser visualizados caso a opção **Modo Avançado** esteja desabilitado.

Por exemplo, ao realizar o upload de um arquivo com extensão `*.html.view` para o diretório **Diagrama de dados** (Localização `/Diagrama de dados/`), este ficará oculto. Isso ocorre porque, com o **Modo Avançado** desabilitado, esse diretório só exibe arquivos com extensão `*.umlcd`.

- **Diretório/pacote**: cria uma pasta no local selecionado.
- **<Tipos de arquivos>**: a lista com os tipos de arquivos pode variar com o status da opção **Modo Avançado** e o diretório selecionado.

Embora seja possível criar qualquer tipo de arquivo em qualquer diretório, é altamente recomendável manter cada tipo de arquivos nos diretórios planejados. Isso assegurará uma interação mais eficiente entre os arquivos do projeto e as ferramentas do Cronapp

- **Modo Avançado Habilitado**: exibirá diversos tipos de arquivos para criar um arquivo nesse formato.
- **Modo Avançado Desabilitado**: exibirá o tipo de arquivo específico para o diretório selecionado. Por exemplo, o menu de contexto do diretório **Diagrama de dados** (Endereço `/Diagrama/`) exibirá apenas a opção de arquivos de diagramas de dados (extensão `*.umlcd`).
- **Outros**: utilize essa opção para criar arquivos em outros formatos, é necessário informar o nome do arquivo e sua extensão (exemplo: `nome.txt`).
- **Instalar Plugins**: abre a janela para selecionar e adicionar um *plugin* ao projeto. Essas opções também podem ser acessadas a partir do menu do sistema **Plugin > Instalar Plugins**.
- **Aplicar Modelo**: abre a tela de seleção do tipo de projeto e temas para retornar o projeto ao modelo inicial.

Essa opção irá sobrescrever diversos arquivos e poderá danificar o projeto. Se deseja mudar apenas o tema do projeto, utilize as ferramentas de **Temas e Skins**.

- **Compartilhar Projeto**: abre uma janela que disponibiliza um *link* que permite que outro usuário copie o projeto para sua conta. Também é possível enviar esse link por *e-mail*.
- **Configuração**: abre a janela de configurações gerais do projeto.
- **Fechar**: fecha o projeto.

## Diretórios e arquivos

O menu de contexto nos arquivos e diretórios possuem algumas diferenças em relação a raiz do projeto, veja abaixo os itens não apresentados na lista da **raiz do projeto**.

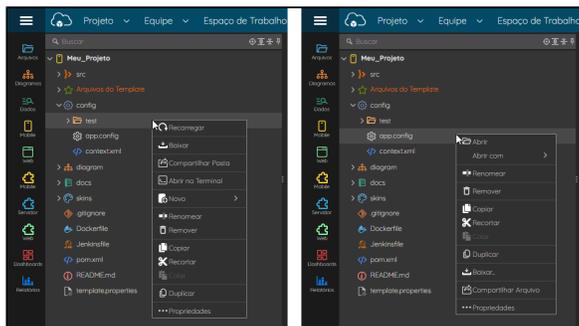


Figura 4.1 - Menu de contexto de um diretório e um arquivo

### Menu dos diretórios:

Se o projeto estiver com o **Modo Avançado** desabilitado, alguns diretórios não exibirão o menu de contexto. Esses diretórios são virtuais e não existem no projeto real, são exibidos apenas para agrupar recursos com a mesma funcionalidade.

- **Renomear**: renomeia o diretório selecionado.
- **Remover**: exclui o diretório selecionado.
- **Copiar**: copia para a área de transferência o diretório selecionado.
- **Recortar**: recorta para a área de transferência o diretório selecionado.
- **Colar**: cola o conteúdo da área de transferência dentro do diretório selecionado.
- **Duplicar**: duplica o diretório selecionado.
- **Propriedades**: exibe as propriedades do diretório selecionado.

## Menu dos arquivos:

- **Abrir:** abre o arquivo selecionado. Equivalente ao duplo clique.
- **Abrir com:** permite abrir arquivos a partir de um editor de texto ou outras [ferramentas](#).
- **Renomear:** renomeia o arquivo selecionado.
- **Remover:** exclui o arquivo selecionado.
- **Copiar:** copia para a área de transferência o arquivo selecionado.
- **Recortar:** recorta para a área de transferência o arquivo selecionado.
- **Colar:** cola o arquivo da área de transferência no mesmo diretório do arquivo selecionado.
- **Duplicar:** cria uma cópia do arquivo no mesmo diretório.
- **Propriedades:** exibe as propriedades do arquivo.

## Conteúdo específico

### Arquivo pom.xml

O arquivo **pom.xml** (Endereço: /pom.xml) só é exibido quando a opção **Modo Avançado** estiver **ativada** e possui um item a mais em seu menu de contexto.

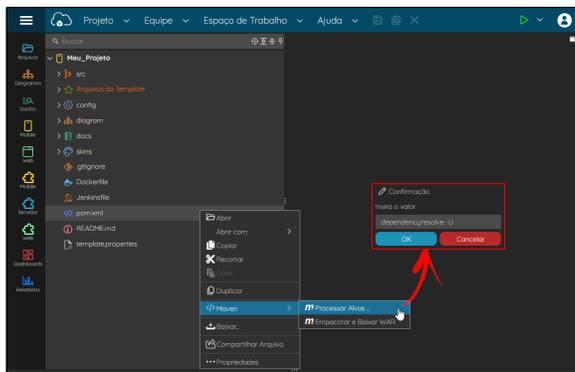


Figura 4.2 - Ações específicas para o Maven

- **Maven:** submenu com opções para o gerenciamento de dependências [Maven](#).
  - **Processar alvos:** abre uma janela para executar comandos do Maven. O comando padrão força a resolução das dependências.

Essa opção também pode ser encontrado no menu do sistema: **Projeto > Recompilar > Recompilar Maven**.

- **Empacotar e baixar WAR:** compacta os arquivos Maven e faz o download do arquivo \*.war gerado.

### Arquivos package.json

Os arquivos **package.json** das aplicações web e mobile só são exibidos quando a opção **Modo Avançado** estiver **ativada**. Seu menu de contexto possui a opção **Iniciar Projeto** para compilar as dependências NPM da aplicação selecionada.

Essa opção também pode ser encontrado no menu do sistema:



- **Projeto > Recompilar > Recompilar NPM Web**
- **Projeto > Recompilar > Recompilar NPM Mobile**

### Arquivos javascript e css

Os arquivos **Javascript** (\*.js) e **Cascading Style Sheets** (\*.css) das aplicação *web* e *mobile* só são acessíveis quando a opção **Modo Avançado** estiver **ativada**. Seus menus de contexto possui opções para adicionar ou remover a referência do arquivo selecionado no arquivo **index.html** das aplicações:

- **Mobile:** /src/main/mobileapp/www/index.html
- **Web:** /src/main/webapp/index.html (destaque 1 da figura 4.3)

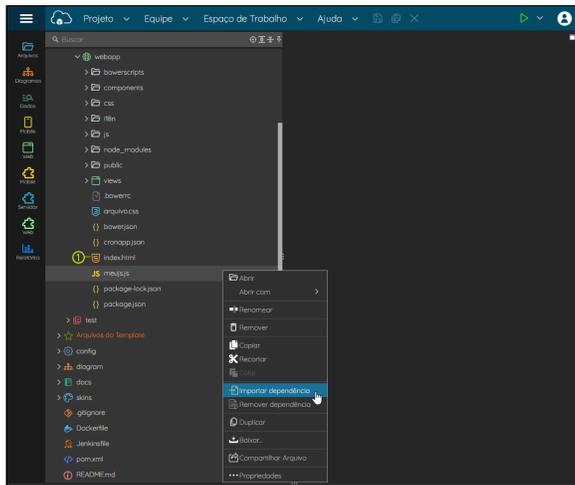


Figura 4.3 - Ações para importar e remover dependências

- **Importar dependências:** inclui a referência do arquivo selecionado no **index.html** da aplicação.
  - **Javascript:** a referência ficará em uma tag `<script>` dentro do `<body>`.
  - **css:** a referência ficará em uma tag `<link rel="stylesheet">` dentro do `<head>`.
- **Remover dependência:** exclui a referência do arquivo selecionado no `index.html` da aplicação.

## Menu Ação

Essa opção do menu de contexto permite gerar recursos a partir das classes do **Diagrama de Dados** (destaque 1 da figura 4.4) e **Fonte de dados** (destaque 2) localizados na árvore de recursos ou arquivos.

- **Diagrama de dados** - Localização: Diagrama de dados/<Namespace>/ .
- **Fonte de dados** - Localização: Fonte de Dados/ .

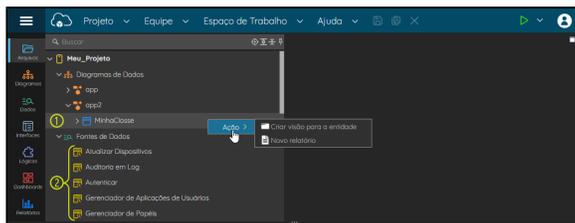


Figura 4.4 - Opção "Ação" disponível nas classes do Diagrama e Fonte de dados

- **Ação:** lista de ações disponíveis.
  - **Criar visão para a entidade:** abre a janela do [Assistente de CRUD view para a Entidade](#).
  - **Novo relatório:** abre uma janela onde é possível selecionar os campos (atributos) e definir um nome para o relatório. Ao final, um relatório é gerado no diretório **Relatório** (Localização: `Relatório/`).

## Janela de propriedades

Na janela de propriedades dos arquivos e diretórios é possível acessar informações e definir algumas características sobre o elemento selecionado. Essa opção é útil quando for necessário saber o endereço real de arquivos e diretórios.

## Arquivo

A janela de propriedades do arquivo será acessível independentemente da configuração do [Modo Avançado](#) (figura 5).

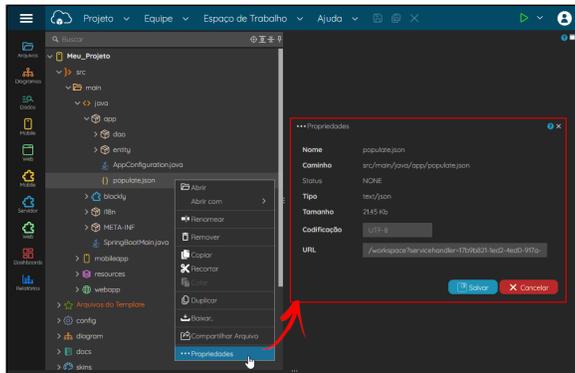


Figura 5 - Janela de propriedade dos arquivos

- **Nome:** nome completo, incluindo a extensão.
- **Caminho:** endereço do arquivo desde a raiz do projeto.
- **Status:** quando o projeto está **versionado**, esse campo exibe o status git atual (exemplo: none, new, added, changed).
  - O status "none" representa um arquivo não versionado.
- **Tipo:** **MIME Type** do arquivo,
- **Tamanho:** espaço ocupado pelo arquivo.
- **Codificação:** define a **codificação de caracteres** do arquivo. É recomendável sempre utilizar o padrão UTF-8.
- **URL:** [ recurso depreciado ] gera um endereço único para compartilhamento do arquivo. Para essa função, utilize a opção **Compartilhar Arquivo**.

## Diretório

Alguns dos diretórios listados quando a opção **Modo Avançado** está **desativada** são virtuais, ou seja, não existem no projeto. Dessa forma, não é possível acessar seu menu de contexto ou propriedade.

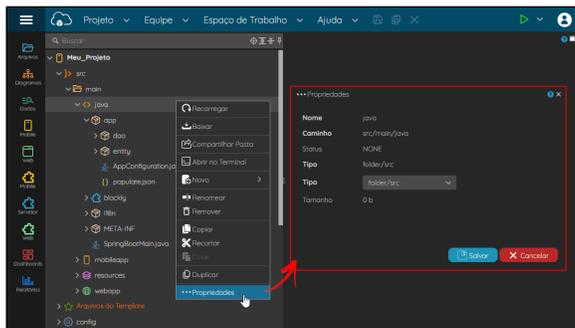


Figura 5.1 - Janela de propriedade dos diretórios

- **Nome:** nome real do diretório.
- **Caminho:** endereço do diretório desde a raiz do projeto.
- **Status:** quando o projeto está **versionado**, esse campo exibe o status git atual (exemplo: none, new, added, changed).
  - O status "none" representa um diretório não versionado.
- **Tipo:** exibe o **MIME Type** atual definido para o diretório.
- **Tipo (caixa de seleção):** altera o MIMEType do diretório e seu ícone na árvore de recurso ou arquivos.
- **Tamanho:** espaço ocupado pelo diretório (ignorando seu conteúdo).