

Camada de persistência

Como principais recursos, essa funcionalidade permite gerar as camadas **Entity** e **DAO** a partir das alterações feitas no Diagrama de dados, além de definir as modificações no banco de dados a partir da próxima reinicialização do sistema.

Fique atento ao banco de dados selecionado (destaque 1 da figura 1), pois, a depender da configuração, essa ação poderá excluir todos os dados do seu banco!

Gerar camada de persistência

Ao clicar no ícone de atalho (seta da Figura 1) **Gerar camada de persistência**, uma janela de configuração será aberta, permitindo selecionar as classes, informar o nome do pacote e selecionar o tipo de geração que ocorrerá no banco de dados selecionado (destaque 1 da Figura 1).

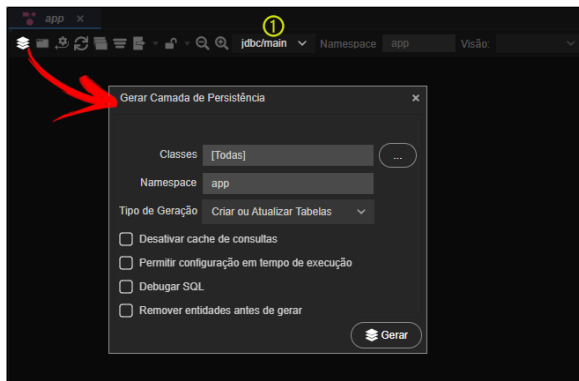


Figura 1 - Gerar camada de persistência

- **Classes:** permite selecionar quais as classes do [Diagrama de dados](#) terão os arquivos das camadas **DAO e Entity** criados ou modificados.
- **Namespace:** nome do pacote que irá manter os arquivos das camadas **DAO e Entity**. Informe outro valor para gerar um novo pacote.
- **Tipo de Geração:** define como serão tratadas as tabelas do banco de dados que possuem vínculo com as classes das camadas **DAO e Entity** (Acesse a documentação oficial do [EclipseLink](#) para detalhes sobre os campos abaixo). A cada nova inicialização do sistema, o banco de dados será afetado com a opção selecionada.
 - **Criar ou Atualizar Tabelas:** gera as tabelas do banco de dados, caso elas não existam, ou inclui novas colunas de tabelas existentes. É importante destacar que essa opção não apaga os dados ou realiza alterações como renomear, excluir ou alterar o tipo de dados das colunas existentes.
 - **Apagar e Recriar Tabelas:** apaga todas as tabelas do banco de dados, juntamente com seus dados, que estão vinculadas às classes do projeto. Em seguida, recria essas tabelas com as alterações feitas no [Diagrama de dados](#).

Cuidado ao utilizar essa opção vinculando o Diagrama de dados ao Banco de dados de produção.

- **Utilizar Tabelas Existentes:** não realiza alterações nas tabelas do banco de dados.
- **Desativar cache de consultas:** desmarque para permitir que o sistema utilize cache de consulta, evitando várias requisições ao banco de dados com o mesmo resultado.
- **Permitir configuração em tempo de execução:** opção usada em conjunto com o evento "Para configurar conexão". Esse campo possibilita que o banco de dados do sistema seja alterado em tempo de execução a partir de uma lógica pré-definida.
- **Debugar SQL:** ao selecionar essa opção, toda declaração feita ao banco de dados será exibida no log da aplicação (Console do depurador).
- **Remover entidades antes de gerar:** essa caixa de seleção só será exibida se existir alguma classe nas camadas DAO e entity não relacionadas com o Diagrama de dados. Ao marcar, as classes que não possuem vínculo com as camadas DAO e entity serão removidas.
- **Botão Gerar:** executa a funcionalidade. Os arquivos do projeto serão modificados, mas as alterações no Banco de dados só ocorrerão a partir da próxima inicialização do sistema.

Nesta página

- [Gerar camada de persistência](#)
 - [Menu de contexto](#)
 - [Seleção das classes](#)
 - [Janela de conflitos](#)
 - [Classes inexistentes](#)
 - [Camadas DAO e Entity](#)
 - [Debugar SQL](#)

Conteúdo complementar

- [Diagrama](#)

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: [Diagrama de dados](#)

As opções selecionadas no campo **Classes** afetarão os arquivos Java das camadas **DAO e Entity**, esses arquivos possuem detalhes sobre a estrutura da classe e seus atributos, definindo inclusive, como serão construídas as tabelas do banco de dados. Já o campo **Tipo de Geração** é responsável por definir se o banco de dados, vinculado ao **Namespace** selecionado, será modificado ou não a partir da próxima reinicialização do sistema. Assim, se as opções "Criar ou Atualizar Tabelas" ou "Apagar e Recriar Tabelas" estiverem selecionadas, o sistema irá verificar as alterações nas camadas **DAO e Entity** e modificar o Banco de dados na próxima inicialização do sistema.

Menu de contexto

Utilize a opção **Criar camada de dados para a entidade** (Figura 2) do menu de contexto de uma classe do Diagrama de dados para abrir a janela **Gerar camada de persistência** (Figura 1) contendo apenas a classe selecionada no campo **Classes**.

Não é possível alterar o **Tipo de Geração** apenas para uma única classe selecionada. Veja mais detalhes sobre os campos **Classes** e **Tipo de Geração** no tópico [Gerar camada de persistência](#).

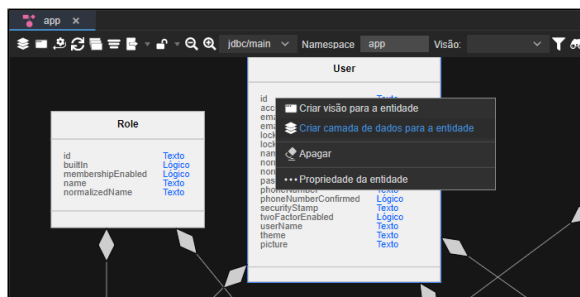


Figura 2 - Gerar camada de persistência somente para a classe selecionada

Seleção das classes

Utilize o botão "..." do campo **Classes** para selecionar quais as classes Java das camadas **DAO e Entity** serão afetadas ao **Gerar** a camada de persistência.

Por padrão, todas as classes existentes no diagrama são selecionadas no momento de gerar a persistência, mas isso pode ser alterado. Ao clicar no botão "..." (seta da Figura 3) que fica ao lado do campo **Classes**, será aberta a janela para selecionar as classes. Além disso, é possível filtrar as classes selecionando uma visão (destaque 1 da Figura 3) ou através do campo de pesquisa (2 da Figura 3).

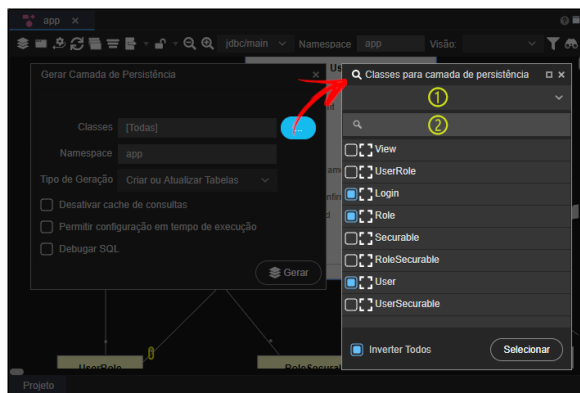


Figura 3 - Janela de seleção das classes para gerar persistência

- **Filtro Visão:** caixa de seleção para filtrar as classes do Diagrama vinculadas a visão selecionada.
- **Pesquisa:** campo de busca.
- **Opção Inverter todos:** alterna o status do **checkbox** de todas as classes da janela.

Janela de conflitos

Ao gerar a camada de persistência pela segunda vez, uma janela de conflito de conteúdo será exibida, permitindo selecionar quais arquivos serão sobrescritos. Com um duplo clique sobre o arquivo, é possível visualizar quais partes do código ocorreu o conflito.

A janela de conflito só ocorre quando a IDE estiver no [Modo Avançado](#).

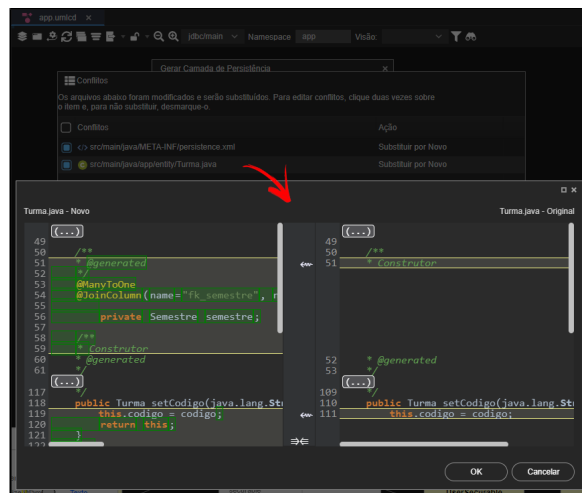


Figura 4 - Janela de conflitos

Alteração de arquivos High-code

O conteúdo High-code, como as classes Java das camadas **entity** e **DAO**, além dos arquivos persistence.xml, SpringBootApplication.java e outros, são mantidos e atualizados pelas ferramentas Low-code do Cronapp. Dessa forma, não recomendamos modificar esses arquivos manualmente, pois, ao utilizar recursos como Gerar camada de persistência, os dados inseridos manualmente nesses arquivos serão modificados ou excluídos.

Classes inexistentes

Quando uma classe gerada pelo diagrama for excluída somente do diagrama, ao gerar novamente a camada de persistência, uma mensagem aparecerá informando que a classe ainda existe no pacote (Namespace) e dará a opção do usuário excluí-la do pacote automaticamente (destaque da Figura 5). A opção **Remover entidades antes de gerar** (destaque 1 da figura 5) só será exibida nesse tipo de situação.

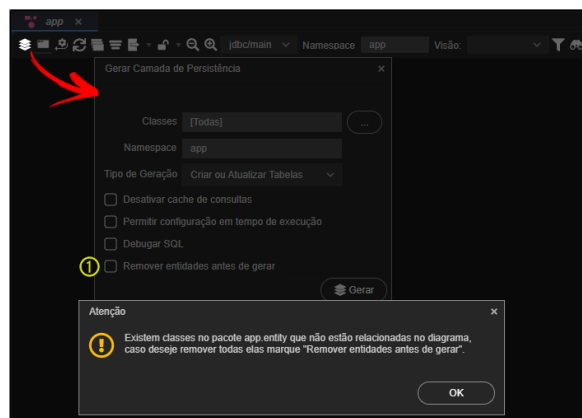


Figura 5 - Remover entidades antes de gerar camada

Por segurança, após selecionar a opção **Remover entidades antes de gerar** (destaque 1 da figura 5.1) e clicar no botão **Gerar**, uma janela será exibida para confirmar a ação.

Observação

Apagar a entidade do diagrama utilizando as opções **Apagar e recriar tabelas** e **Remover entidades antes de gerar** não fará com que a tabela da entidade também seja deletada do banco de dados. Para fazer isso o usuário terá que removê-la manualmente do banco de dados.

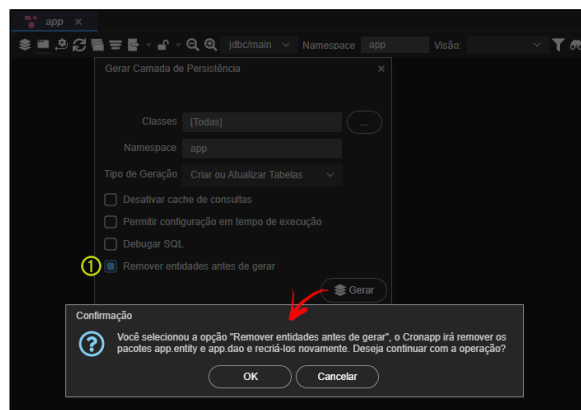


Figura 5.1 - Mensagem de confirmação para a exclusão das classes Java

Camadas DAO e Entity

Após a execução da ferramenta de [camada-persistencia](#), os arquivos das camadas **DAO** e **Entity** são modificados para refletir as alterações feitas nas janelas de configuração das classes do [Diagrama de dados](#).

Estes arquivos só serão exibidos após habilitar o [Modo Avançado](#).

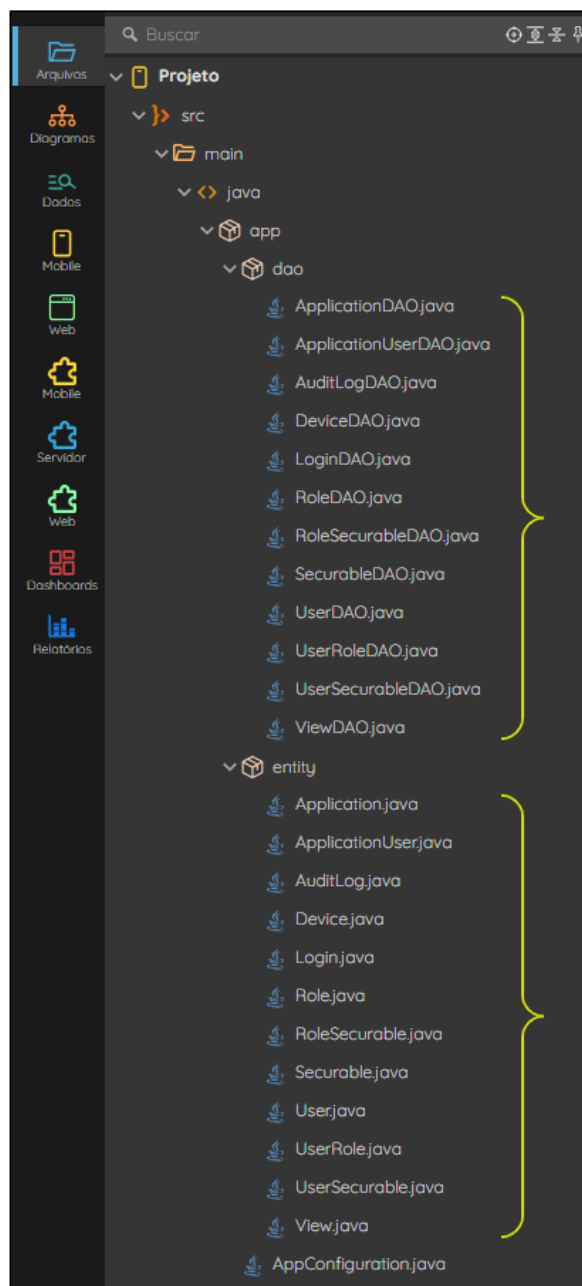


Figura 6 - Arquivos das camadas DAO e Entity

Debugar SQL

O recurso **Debugar SQL** exibe no Console do depurador as declarações realizadas pela aplicação, permitindo que você acompanhe e verifique as operações de banco de dados em tempo real. Isso auxilia no processo de depuração e solução de problemas relacionados ao acesso e manipulação de dados no banco de dados. Esse recurso irá funcionar com o projeto sendo executado pelo Cronapp, em modo debug, e em produção.

Vamos utilizar um exemplo para melhor ilustrar esse recurso. Na figura 7, estamos inserindo dados de cadastro em uma classe chamada Pessoa. Ao clicar no botão que realiza a inserção desses dados no banco de dados, a função SQL `INSERT INTO` será executada. Ao habilitar o recurso de depuração de SQL, sempre que essa instrução for executada, ela será exibida no console do depurador.

Pessoa - Inserindo

Name
Fábio

Idade
30

Figura 7 - Inserindo dados de cadastro de na classe Pessoa

O formato exibido no console será a declaração SQL que está sendo enviada para o banco de dados (destaque da figura 7.1). A declaração será exibida na primeira linha, cada parâmetro será representado pelo caractere "?", enquanto na segunda linha, bind, exibirá um array com os dados informados, representando cada parâmetro.

```
letMethodAllowed AS a2, ti.getMethodAllowed AS a3, ti.headMethodAllowed AS a4, ti.optionsMethodAllowed AS a5, ti.patMethodAllowed AS a6, ti.pattern AS a7, ti.postMethodAllowed AS a8, ti.putMethodAllowed AS a9, ti.traceMethodAllow  
bind => [Authenticated, 21e55de-b688-477a-89cb-7ff32073e312, 21e55de-b688-477a-89cb-7ff32073e312, 100, 0]  
[EL Fine]: sql: 2023-06-07 17:49:46.898--ServerSession(148757621)--Connection(123072564)--SELECT t1.id AS a3, t1.de  
letMethodAllowed AS a2, ti.getMethodAllowed AS a3, ti.headMethodAllowed AS a4, ti.optionsMethodAllowed AS a5, ti.pat  
tchMethodAllowed AS a6, ti.pattern AS a7, ti.postMethodAllowed AS a8, ti.putMethodAllowed AS a9, ti.traceMethodAllow  
bind => [Administrators, 21e55de-b688-477a-89cb-7ff32073e312, 21e55de-b688-477a-89cb-7ff32073e312, 100, 0]  
[EL Fine]: sql: 2023-06-07 17:49:47.013--ClientSession(2127282243)--Connection(2084487959)--INSERT INTO "Pessoa" (i  
bind => [77D04805-5837-4AFF-ADEB-53C9EE22FC66, 30, Fábio]  
[EL Fine]: sql: 2023-06-07 17:49:47.017--ClientSession(2127282243)--Connection(2084487959)--SELECT id, idade, name F  
bind => [77D04805-5837-4AFF-ADEB-53C9EE22FC66]  
[EL Fine]: sql: 2023-06-07 17:53:49.796--ServerSession(148757621)--Connection(349070648)--SELECT t0.normalized_name  
AS a1 FROM "ROLE" t0, "USER" t2, "USER_ROLE" t1 WHERE (((t2.normalized user name = ?) AND ((t0.application id IS NULL  
y
```

Figura 7.1 - declarações exibidas no Console do depurador

A declaração de inserção pode ser conferida no trecho de código abaixo.

Declaração SQL

```
[EL Fine]: sql: 2023-06-07 17:49:47.017--ClientSession(2127282243)--  
Connection(2084487959)--INSERT INTO "Pessoa" (id, idade, nome) VALUES (?,  
?, ?)  
bind => [77D04805-5837-4AFF-ADEB-53C9EE22FC66, 30, Fábio]
```