

Arquivo populate json

O arquivo **populate.json** (Endereço: `src/main/java/app/populate.json`) possui vários registros das entidades da estrutura de segurança do Cronapp, como `User`, `Securable` e `Role`. Seu objetivo é popular o Banco de dados do projeto com uma base de dados mínima. Assim, um projeto recém-criado no Cronapp pode ser imediatamente executado e sua aplicação será funcional, um usuário Administrador já estará disponível para logar (usuário e senha: "admin") nas aplicações web e mobile.

Não é necessário manipular esse arquivo e não recomendamos que o faça, seu conteúdo é editável de forma low-code através da ferramenta Permissão de segurança. O relacionamento entre as classes e a estrutura de segurança do Cronapp são apresentados na documentação [Permissão de segurança](#).

Essa documentação apresenta a estrutura do arquivo `populate.json` apenas para conhecimento, mas não incentivamos a sua manipulação. Como se trata de um conteúdo high-code, a manipulação do código é de responsabilidade do usuário.

Tenha cuidado ao adicionar publicamente seu projeto em sistemas de controle de versões, como GitHub, bitbucket, GitLab e outros, pois os dados informados no arquivo ficarão expostos. 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](#)).

```
1  {} populate.json x
2  {
3  {
4    "_class": "app.entity.User",
5    "id": "3048F43E-0E62-4F4A-8A63-3F22D7AD4611",
6    "name": "admin",
7    "login": "admin",
8    "password": "$2a$10$LjzpM1Q3VoAtG2dTCCabNuW0/amVPjL3Iyvy12Dj7NP.HngtzSewu"
9  },
10 {
11   "_class": "app.entity.Role",
12   "id": "Administrators",
13   "user": { "id": "3048F43E-0E62-4F4A-8A63-3F22D7AD4611" }
14 }
```

Figura 1 - Arquivo populate.json

Campo password da tabela User

O valor do campo `password` da entidade `User` no arquivo `populate.json` já deve estar criptografado. Caso queira alterar esse valor, sugerimos que acesse a página de usuários da aplicação, insira um usuário e copie o hash gerado no campo `password` na tabela `User` do Banco de dados para alimentar o arquivo `populate.json`.

Estrutura

Nesse exemplo vamos mostrar como alimentar as entidades criadas no [Diagrama de dados](#) e como ocorre as interações de relacionamentos no arquivo `populate.json`.

Necessário:

- Criar a [camada de persistência](#) das classes.
- Para alimentar o campo `id`, sugerimos utilizar algum site para gerar UUID para os identificadores.

Entidade

O primeiro exemplo a ser utilizado será da classe "Anime" que não possui relacionamentos (Figura 2). Essa classe possui quatro campos:

- **id:** tipo TEXTO.

Nessa página

- Estrutura
 - Entidade
 - Relacionamentos 1-1 e 1-N
 - Duplo relacionamento 1 - N
 - Relacionamento N-M
 - Auto relacionamento
 - Arquivo final

- **titulo:** tipo TEXTO.
- **episodio:** tipo INTEIRO.
- **completo:** tipo LÓGICO.

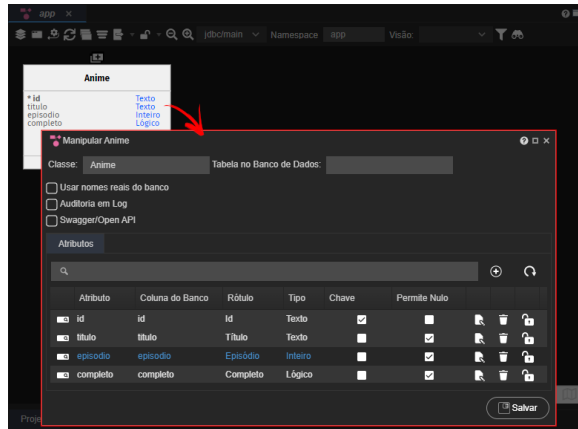


Figura 2 - Classe sem relacionamento

Para inserir dados nessa classe através do arquivo, você precisa especificar a classe e todos os seus campos nos atributos do objeto JSON. Para indicar a classe, é necessário informar a chave "`_class`" e o seu valor será o "caminho" (Figura 2.2) onde encontra a classe ("`<namespace>.<pacote>.<nomedaclass>`" - sem a extensão `.java`). No exemplo da figura 2.1, ele ficará do seguinte modo: "`_class : "app.entity.Anime"`".

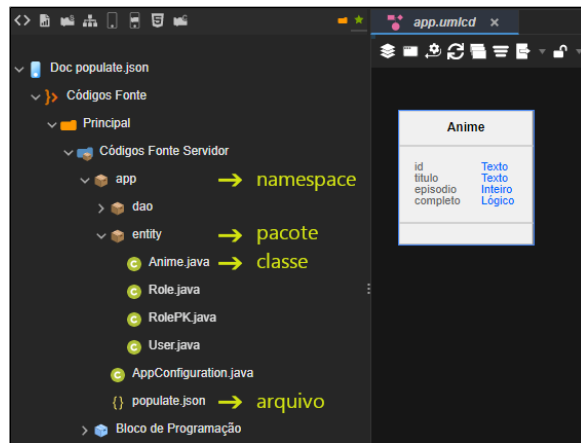


Figura 2.1 - Caminho: namespace.pacote.classe

Os demais atributos do objeto devem ser incluídos da mesma forma, através do par "`chave : "valor"`". Além disso, os valores dos atributos da classe precisam estar em conformidade com os tipos definidos, seguindo o padrão dos atributos JSON, ou seja, String possui aspas duplas, já Números e Boleano não (veja mais detalhes em [JSON](#)).

Resultado final para a estrutura básica

```
{
  "_class" : "app.entity.Anime",
  "id": "7A0DCC99-668B-4E28-8EE0-90D607CDAB5D",
  "titulo": "Naruto",
  "episodio": 700,
  "completo": true
}
```

Duas coisas importantes (padrão JSON):

- Como estamos enviando um array de objetos, o par de colchetes é usado uma única vez no arquivo, ele vai englobar todos os objetos.
- Cada objeto e cada atributos devem ser separados por vírgulas.

Relacionamentos 1-1 e 1-N

Tanto no relacionamento 1-1 quanto 1-N, a ordem como são apresentados os objetos no arquivo `populate.json` tem fundamental importância, o objeto que possui a *chave primária* do relacionamento deve ser inserido primeiro e só depois o objeto que possui a *chave estrangeira* relacionada. Por exemplo, na figura 2.2 a tabela **Cliente** carrega a chave estrangeira da tabela **Carro**, dessa forma, no arquivo JSON, o objeto **Carro** precisa ser inserido primeiro que o objeto **Cliente**, evitando que seja inserido no banco uma chave estrangeira sem que exista ainda a chave primária da outra tabela.

Para saber mais sobre a tabela de relacionamentos 1-1 e 1-N no Cronapp, consulte a página: [Relacionamento entre classes no CRUD](#)

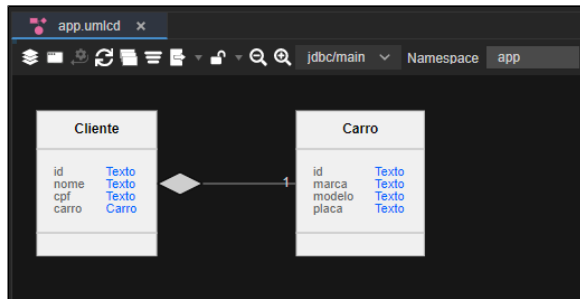


Figura 2.2 - Relacionamento 1-1

Caso a ordem seja invertida, durante a execução do projeto será exibida no console uma exceção parecida com a do bloco abaixo:

Mensagem exibida em um banco H2 ao alimentar os relacionamentos na ordem invertida

Internal Exception: org.h2.jdbc.JdbcSQLException: Constraint "FK_CLIENTE_fk_carro" already exists;
SQL statement: ALTER TABLE "CLIENTE" ADD CONSTRAINT "FK_CLIENTE_fk_carro" FOREIGN
KEY (fk_carro) REFERENCES "CARRO" (id) [90045-197]

A inserção da chave estrangeira no objeto é bem simples, basta indica o nome da chave primária do outro objeto e o seu valor entre chaves. Na figura 2.2, a chave estrangeira recebe o nome da classe, então ele ficará do seguinte modo:

Sintaxe:

```
"_class": "classe",
"chave primária da classe": "valor da chave primária",
"classe relacionada": {"chave primária da classe relacionada": "valor da chave primária"}
```

Resultado final do relacionamento 1-1

```
{
  "_class": "app.entity.Carro",
  "id": "3AA6CAF2-ACB3-4A45-9C53-03FD2FE7A393",
  "marca": "Lexus",
  "modelo": "LFA",
  "placa": "DEB7I94"
},
{
  "_class": "app.entity.Cliente",
  "id": "AF498375-FCC6-4197-8B04-C7F3EC7E8D2C",
  "nome": "Deborah",
  "cpf": "60694536032",
  "carro": { "id": "3AA6CAF2-ACB3-4A45-9C53-03FD2FE7A393" }
}
```

Duplo relacionamento 1 - N

A figura abaixo mostra como fica os objetos do arquivo populate com o mesmo objeto se relacionando duas vezes com outra classe. Nesse caso, a classe `Negocio` só guarda as chaves estrangeiras da classe `Pessoa`.

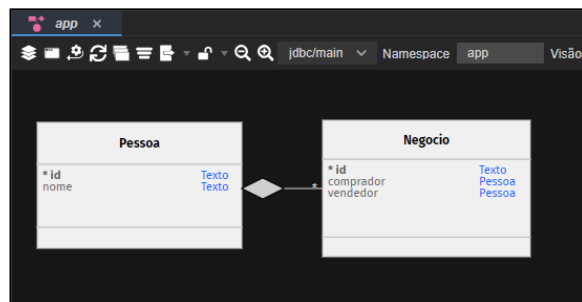


Figura 2.3 - Duplo relacionamento 1-N

Sintaxe:

```
"_class": "classe",
"chave primária da classe": "valor da chave primária",
"classe relacionada 1": {"chave primária da classe relacionada 1":
"valor da chave primária 1"},
"classe relacionada 1": {"chave primária da classe relacionada 1":
"valor da chave primária 2"}
```

Autorrelacionamento N-M

```
{
  "_class": "app.entity.Pessoa",
  "id": "67C9F4D1-2231-4904-94DD-CDE0E888A17D",
  "nome": "Mayumi"
},
{
  "_class": "app.entity.Pessoa",
  "id": "12033AAB-DC75-4E85-9994-80A74824FEDA",
  "nome": "Deborah"
},
{
  "_class": "app.entity.Pessoa",
  "id": "DB1A914C-32B6-4453-B773-83764D773F5B",
  "nome": "Rodrigo"
},
{
  "_class": "app.entity.Negocio",
  "id": "43A654F2-3736-448D-B645-8233FE2C066B",
  "comprador": { "id": "67C9F4D1-2231-4904-94DD-CDE0E888A17D" },
  "vendedor": { "id": "DB1A914C-32B6-4453-B773-83764D773F5B" }
},
{
  "_class": "app.entity.Negocio",
  "id": "D7AE6195-F31E-4AF2-AC10-16A798165AA1",
  "comprador": { "id": "12033AAB-DC75-4E85-9994-80A74824FEDA" },
  "vendedor": { "id": "DB1A914C-32B6-4453-B773-83764D773F5B" }
},
{
  "_class": "app.entity.Negocio",
  "id": "802254BF-F64A-4185-B05A-4373ECE31014",
  "comprador": { "id": "12033AAB-DC75-4E85-9994-80A74824FEDA" },
  "vendedor": { "id": "67C9F4D1-2231-4904-94DD-CDE0E888A17D" }
}
```

Relacionamento N-M

Da mesma forma que o relacionamento 1-1 ou 1-N, os objetos que não possuem a chave estrangeira precisam ser adicionados primeiros para depois adicionar o objeto que a carrega - nesse caso, será o objeto referente a classe de associação. Por exemplo, para a figura 2.4, precisam ser adicionados no JSON os objetos referentes as classes `Ator` e `Filme` para depois adicionar o objeto referente a classe `AtorFilme`.

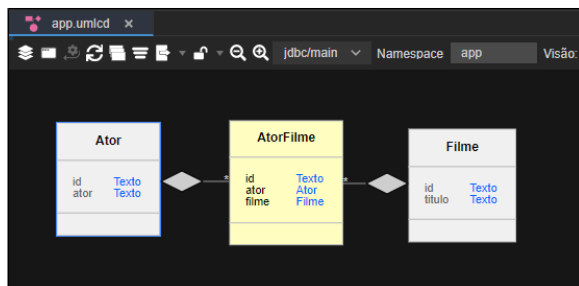


Figura 2.4 - Relacionamento N-M

A forma de inserir a chave estrangeira também é a mesma que no relacionamento 1-1 ou 1-N, mas nesse caso as chaves se encontram na classe `AtorFilme` - então ela ficará do seguinte modo:

Sintaxe:

```

"_class": "classe associativa",
"chave primária da classe": "valor da chave primária",
"classe relacionada 1": {"chave primária da classe relacionada 1":
"valor da chave primária 1"},
"classe relacionada 2": {"chave primária da classe relacionada 2":
"valor da chave primária 2"}

```

Resultado final para relacionamento n-m

```

{
  "_class": "app.entity.Ator",
  "id": "66130FE8-F8DC-4887-B815-A35E42EC0BB4",
  "ator": "Orlando Bloom"
},
{
  "_class": "app.entity.Filme",
  "id": "1C17F46F-F176-4E90-BABF-52AD3B3E4AEA",
  "titulo": "Piratas do Caribe"
},
{
  "_class": "app.entity.AtorFilme",
  "id": "9522FDA2-65E2-4E41-811C-22F2CBC818D7",
  "ator": {"id": "66130FE8-F8DC-4887-B815-A35E42EC0BB4"},
  "filme": {"id": "1C17F46F-F176-4E90-BABF-52AD3B3E4AEA"}
},
{
  "_class": "app.entity.Filme",
  "id": "E5AE7B60-2EE9-4025-9021-5755510AFE21",
  "titulo": "O Senhor dos Anéis"
},
{
  "_class": "app.entity.AtorFilme",
  "id": "9522FDA2-65E2-4E41-811C-22F2CBC818D7",
  "ator": {"id": "66130FE8-F8DC-4887-B815-A35E42EC0BB4"},
  "filme": {"id": "E5AE7B60-2EE9-4025-9021-5755510AFE21"}
}

```

Auto relacionamento

Os objetos com auto relacionamentos devem ser iniciados com um objeto que não possua relacionamento ou que se relacione com ele mesmo, nesse caso, a inserção do atributo de *chave estrangeira* precisa ser inserida por último no objeto.

Para saber mais sobre a tabela de auto relacionamento no Cronapp, consulte a página: [Relacionamento entre classes no CRUD](#).

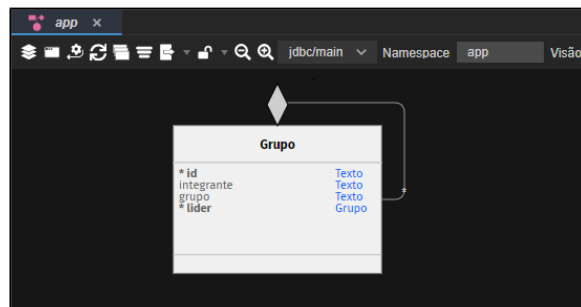


Figura 2.5 - Classes com auto relacionamento

Sintaxe:

```

    "_class": "classe",
    "chave primária da classe": "valor da chave primária",
    "classe relacionada": {"chave primária da classe": "valor da chave primária"}

```

O bloco abaixo mostra como fica os objetos do arquivo JSON para inserção de uma classe com auto relacionamento 1-N, os dados são para o exemplo da Figura 2.5 (classe Grupo).

Auto relacionamento 1-N

```

{
  "_class": "app.entity.Grupo",
  "id": "48631515-F532-4674-BF30-7C0A3D92882C",
  "integrante": "Kim Taeyeon",
  "grupo": "SNSD",
  "lider": { "id": "48631515-F532-4674-BF30-7C0A3D92882C" }
},
{
  "_class": "app.entity.Grupo",
  "id": "F7F1D934-EF30-4FF2-B6EB-631A0E75DADE",
  "integrante": "Jessica Jung",
  "grupo": "SNSD",
  "lider": { "id": "48631515-F532-4674-BF30-7C0A3D92882C" }
},
{
  "_class": "app.entity.Grupo",
  "id": "EAF777F3-3980-4340-A683-91875813D653",
  "integrante": "Tiffany Young",
  "grupo": "SNSD",
  "lider": { "id": "48631515-F532-4674-BF30-7C0A3D92882C" }
}

```

Arquivo final

No bloco abaixo exibimos todos os objetos apresentados nesse documento. Não exibimos os objetos das entidades da estrutura de segurança do Cronapp.

Populate.JSON

```

[
  {
    "_class": "app.entity.Anime",
    "id": "7A0DCC99-668B-4E28-8EE0-90D607CDAB5D",
    "titulo": "Naruto",
    "episodio": 700,
    "completo": true
  },
  {
    "_class": "app.entity.Carro",
    "id": "3AA6CAF2-ACB3-4A45-9C53-03FD2FE7A393",
    "marca": "Lexus",
    "modelo": "LFA",
    "placa": "DEB7I94"
  },
  {
    "_class": "app.entity.Cliente",
    "id": "AF498375-FCC6-4197-8B04-C7F3EC7E8D2C",
    "nome": "Deborah",
    "cpf": "60694536032",
    "carro": { "id": "3AA6CAF2-ACB3-4A45-9C53-03FD2FE7A393" }
  },
  {

```

```

    "_class": "app.entity.Ator",
    "id": "66130FE8-F8DC-4887-B815-A35E42EC0BB4",
    "ator": "Orlando Bloom"
  },
  {
    "_class": "app.entity.Filme",
    "id": "1C17F46F-F176-4E90-BABF-52AD3B3E4AEA",
    "titulo": "Piratas do Caribe"
  },
  {
    "_class": "app.entity.AtorFilme",
    "id": "9522FDA2-65E2-4E41-811C-22F2CBC818D7",
    "ator": { "id": "66130FE8-F8DC-4887-B815-A35E42EC0BB4" },
    "filme": { "id": "1C17F46F-F176-4E90-BABF-52AD3B3E4AEA" }
  },
  {
    "_class": "app.entity.Filme",
    "id": "E5AE7B60-2EE9-4025-9021-5755510AFE21",
    "titulo": "O senhor dos anéis"
  },
  {
    "_class": "app.entity.AtorFilme",
    "id": "9522FDA2-65E2-4E41-811C-22F2CBC818D7",
    "ator": { "id": "66130FE8-F8DC-4887-B815-A35E42EC0BB4" },
    "filme": { "id": "E5AE7B60-2EE9-4025-9021-5755510AFE21" }
  },
  {
    "_class": "app.entity.Grupo",
    "id": "48631515-F532-4674-BF30-7C0A3D92882C",
    "integrante": "Kim Taeyeon",
    "grupo": "SNSD",
    "lider": { "id": "48631515-F532-4674-BF30-7C0A3D92882C" }
  },
  {
    "_class": "app.entity.Grupo",
    "id": "F7F1D934-EF30-4FF2-B6EB-631A0E75DADE",
    "integrante": "Jessica Jung",
    "grupo": "SNSD",
    "lider": { "id": "48631515-F532-4674-BF30-7C0A3D92882C" }
  },
  {
    "_class": "app.entity.Grupo",
    "id": "EAF777F3-3980-4340-A683-91875813D653",
    "integrante": "Tiffany Young",
    "grupo": "SNSD",
    "lider": { "id": "48631515-F532-4674-BF30-7C0A3D92882C" }
  },
  {
    "_class": "app.entity.Pessoa",
    "id": "67C9F4D1-2231-4904-94DD-CDE0E888A17D",
    "nome": "Mayumi"
  },
  {
    "_class": "app.entity.Pessoa",
    "id": "12033AAB-DC75-4E85-9994-80A74824FEDA",
    "nome": "Deborah"
  },
  {
    "_class": "app.entity.Pessoa",
    "id": "DB1A914C-32B6-4453-B773-83764D773F5B",
    "nome": "Rodrigo"
  },
  {
    "_class": "app.entity.Negocio",
    "id": "43A654F2-3736-448D-B645-8233FE2C066B",
    "comprador": { "id": "67C9F4D1-2231-4904-94DD-CDE0E888A17D" },
    "vendedor": { "id": "DB1A914C-32B6-4453-B773-83764D773F5B" }
  },
  {
    "_class": "app.entity.Negocio",
    "id": "D7AE6195-F31E-4AF2-AC10-16A798165AA1",

```



```
    "comprador": {"id": "12033AAB-DC75-4E85-9994-80A74824FEDA"},
    "vendedor": {"id": "DB1A914C-32B6-4453-B773-83764D773F5B"}
  },
  {
    "_class": "app.entity.Negocio",
    "id": "802254BF-F64A-4185-B05A-4373ECE31014",
    "comprador": {"id": "12033AAB-DC75-4E85-9994-80A74824FEDA"},
    "vendedor": {"id": "67C9F4D1-2231-4904-94DD-CDE0E888A17D"}
  }
]
```