

PAQL - Priax Asset Query Language

Manual rápido para entendimento da sintaxe da linguagem PAQL, usada para consultar ICs na CMDB do Priax.

- [PAQL Quick Reference](#)

PAQL Quick Reference

PAQL (Priax Asset Query Language) Quick Reference

Conteúdo

- [1\) O que é a PAQL e para que ela serve](#)
- [2\) Conceitos: CI Types e Related CI Types](#)
- [3\) Estrutura geral de uma PAQL](#)
- [4\) Blocos da PAQL](#)
- [5\) Exemplos completos](#)
- [6\) Checklist rápido](#)

1) O que é a PAQL e para que ela serve

PAQL (Priax Asset Query Language) é a linguagem de consulta usada no Priax para **pesquisar ativos (CIs)** na CMDB, filtrando por tipos de IC, atributos do próprio IC, pelas relações do IC e, quando necessário, **pesquisar por características de ICs relacionados** (por vínculos/relacionamentos) ao IC que se deseja encontrar/listar.

Com a PAQL você consegue:

- Selecionar **quais tipos de ativos** (citype) você quer listar (ex.: *Servers, Web Application*).
- Filtrar ativos por **atributos/campos** (ex.: Nome, Fabricante, Tipo de criação).
- Filtrar por relações que o ativo possui.
- Restringir a pesquisa a ativos **relacionados** por um tipo de relação (ex.: *Service Owned, Composes*) com algum outro IC (relatedcitype).
- Definir quais campos devem voltar no resultado (**returns**) e como nomeá-los (**alias**).
- Ordenar o resultado (**orders**).

A PAQL é composta por **blocos**, e cada bloco tem um papel específico na filtragem dos ativos. Você monta uma expressão juntando blocos em sequência.

2) Conceitos: CI Types e Related CI Types

CI Types (`citype`)

Um **CI Type** (ou `citype`) é o **tipo do ativo principal** que você quer buscar.

Exemplos comuns: `Servers`, `Team`, `Web Application`, `Relational Databases`.

Quando você usa apenas `citype`, a consulta busca ativos **somente desse tipo**, sem obrigar a existência de relacionamento com outro ativo.

Related CI Types (`relatedcitype`)

Um **Related CI Type** (ou `relatedcitype`) é o **tipo do ativo relacionado** ao seu ativo principal (o “outro lado” da relação).

“Quero achar ativos do tipo **A** que estejam relacionados a ativos do tipo **B** (e possivelmente filtrar o B também).”

3) Estrutura geral de uma PAQL

Uma PAQL é um texto com um ou mais blocos:

```
ictypefilters{...}
relsfilters{...}
cifilters{...}
relatedcifilters{...}
returns{...}
orders{...}
```

Regras gerais importantes

- Blocos são escritos como: `nomeDoBloco{ ... }`
- Dentro de alguns blocos, você pode combinar condições usando a palavra `and`.

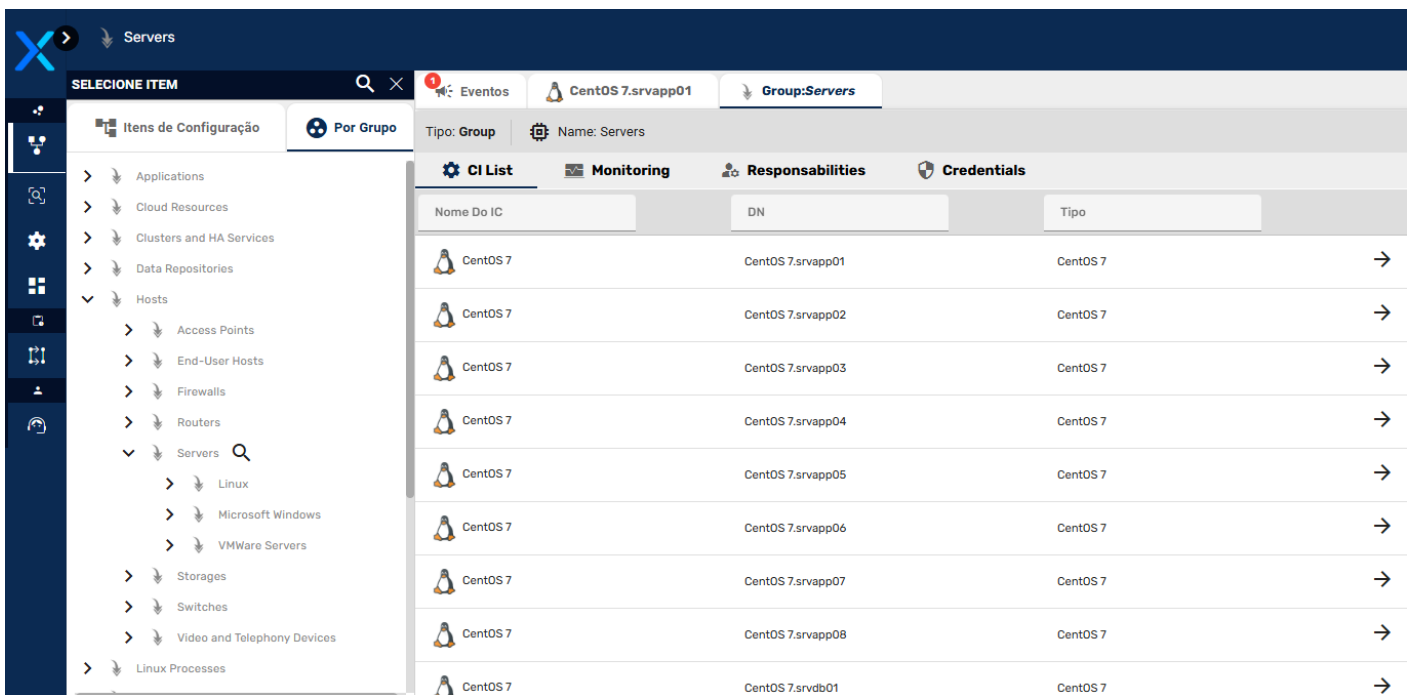
- Listas são escritas como: ["Valor 1", "Valor 2"]
- Campos/atributos são referenciados como strings entre aspas no formato: "Grupo\Campo"

4) Blocos da PAQL

4.1) `ictypefilters{...}` — Tipos de ativos (principal e relacionado)

Para identificar o tipo do IC que se deseja filtrar podemos usar o nome dos grupos ou dos tipos de IC.

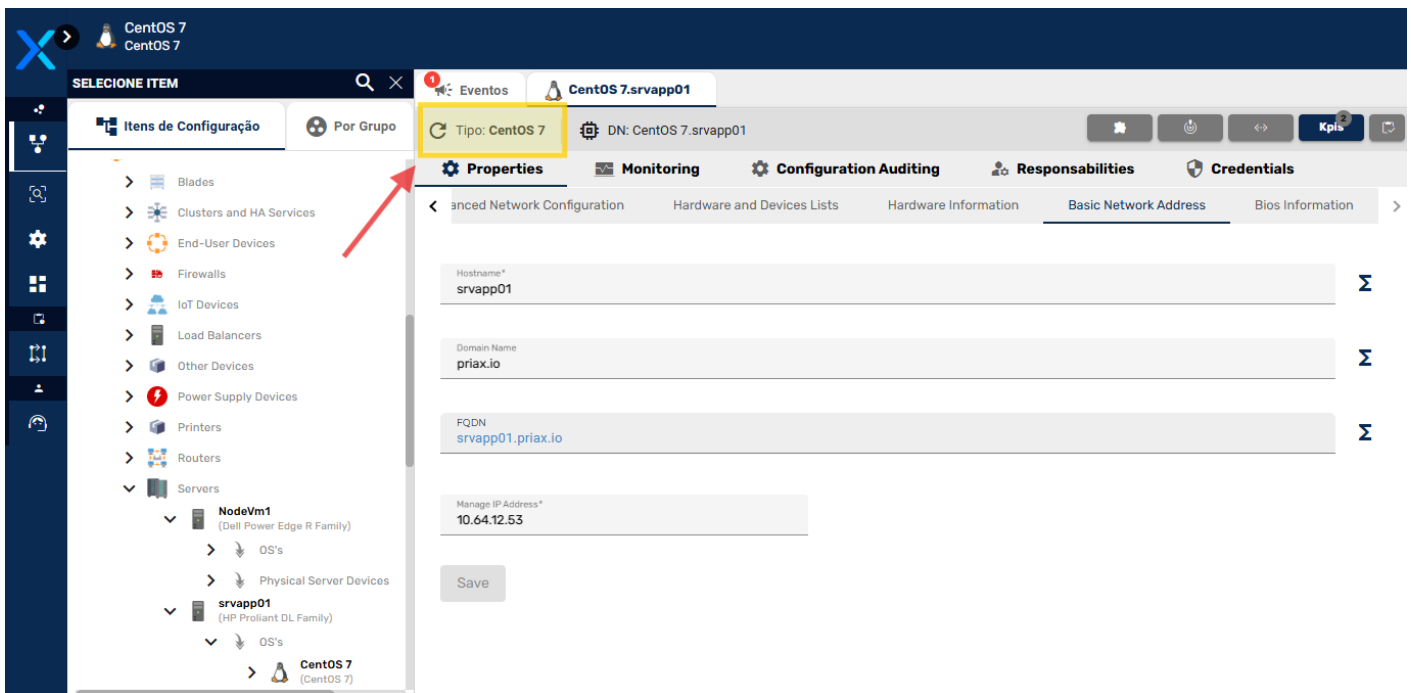
Para identificar o nome de um grupo de ICs navegue para a aba "Por Grupo" da árvore de IC e identifique o nome de um grupo na árvore:



The screenshot shows the 'Servers' management interface. On the left, there is a navigation tree with 'Servers' expanded. The main area displays a table of servers under the 'Group: Servers' filter. The table has columns for 'Nome Do IC', 'DN', and 'Tipo'. All servers listed are of type 'CentOS 7'.

Nome Do IC	DN	Tipo
CentOS 7	CentOS 7.srvapp01	CentOS 7
CentOS 7	CentOS 7.srvapp02	CentOS 7
CentOS 7	CentOS 7.srvapp03	CentOS 7
CentOS 7	CentOS 7.srvapp04	CentOS 7
CentOS 7	CentOS 7.srvapp05	CentOS 7
CentOS 7	CentOS 7.srvapp06	CentOS 7
CentOS 7	CentOS 7.srvapp07	CentOS 7
CentOS 7	CentOS 7.srvapp08	CentOS 7
CentOS 7	CentOS 7.srvdb01	CentOS 7

Nome de um tipo específico de IC:



Define o tipo do ativo principal (`citype`) e, opcionalmente, o tipo do ativo relacionado (`relatedcitype`).

Sintaxe

```
ictypefilters{citype in ["Tipo A","Tipo B"]}
```

Ou com ativo relacionado:

```
ictypefilters{citype in ["Tipo A","Tipo B"] and relatedcitype in ["Tipo C"]}
```

Exemplos

```
ictypefilters{citype in ["Web Application"]}
```

```
ictypefilters{citype in ["Team"] and relatedcitype in ["Web Application"]}
```

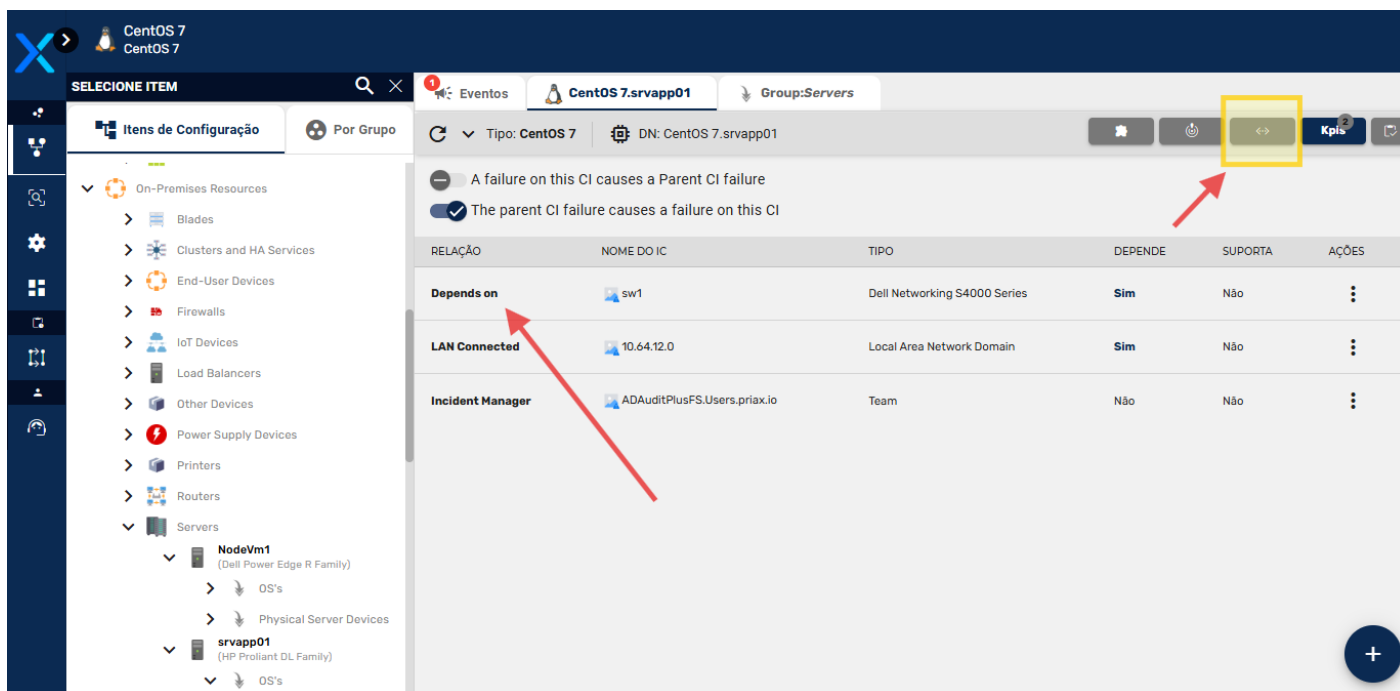
“ Dica: se você informar `relatedcitype` você também poderá usar `relsfilters{...}` e/ou `relatedcifilters{...}` .

4.2) `relsfilters{...}` — Quais relacionamentos considerar (entrada e saída)

Restringe por quais tipos de relacionamento os ativos devem estar ligados.

- `outrels in [...]` (relações “de saída”)
- `inrels in [...]` (relações “de entrada”)

Para identificar o nome de um relacionamento veja os relacionamentos dos ICs na tela do Priax:



Neste caso temos três tipos de relação:

- Depends On
- Lan Connected
- Incident Manager

Sintaxe

```
relsfilters{inrels in ["Nome da Relação"]}
```

```
relsfilters{outrels in ["Nome da Relação"]}
```

```
relsfilters{outrels in ["Relação A"] and inrels in ["Relação B"]}
```

Exemplos

```
relsfilters{inrels in ["Service Owner"]}
```

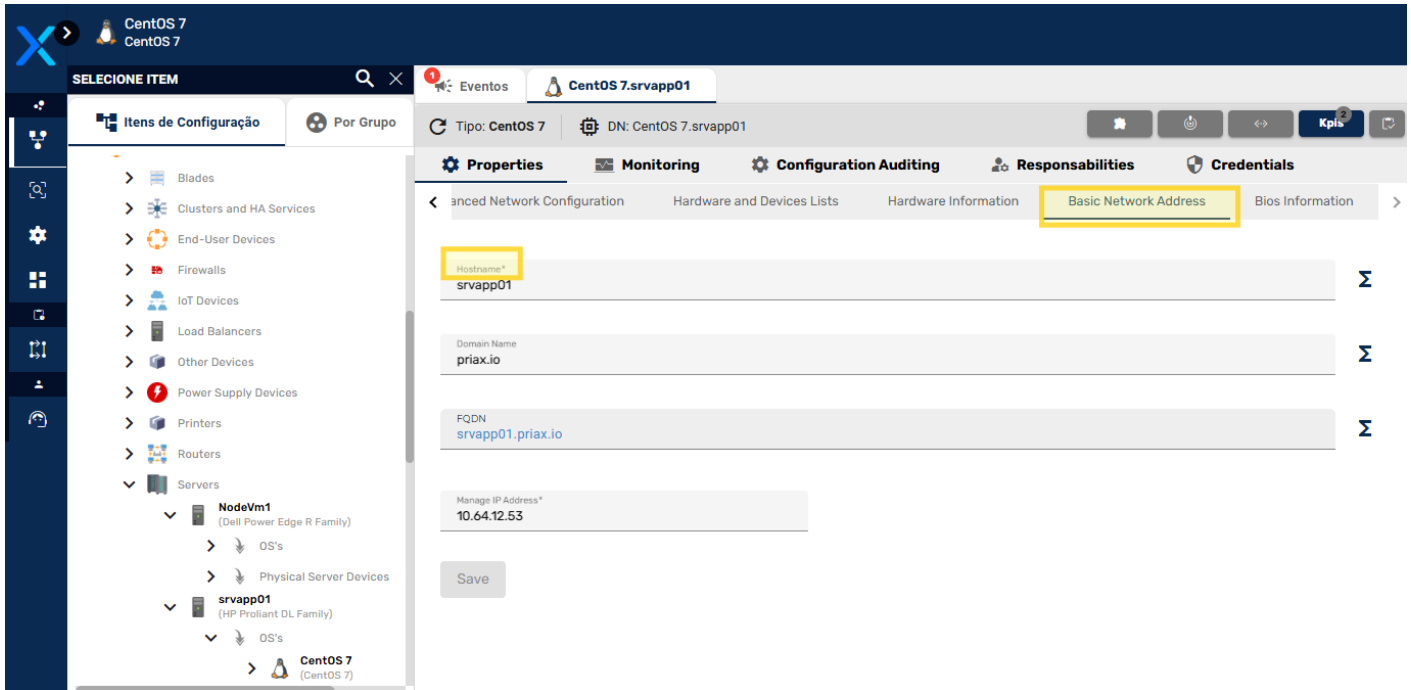
```
relsfilters{outrels in ["Service Owned"] and inrels in ["Service Owner"]}
```

```
relsfilters{outrels in ["Is Composed"] and inrels in ["Composes"]}
```

4.3) `cifilters{...}` — Filtros por atributos do ativo principal (A)

Filtra o ativo principal por campos/atributos.

Para definir o nome do atributo, veja na tela de edição do IC no Priax:



Neste caso de exemplo ao editar o Sistema Operacional de um servidor, temos a aba "Basic Network Address" e o atributo "Hostname". Devemos usar portanto para filtrar por esse atributo "Basic Network Address\Hostname".

Sintaxe

```
"Grupo\Campo" = "Valor"
```

Múltiplas condições:

```
cifilters{"Grupo\Campo 1" = "Valor 1" and "Grupo\Campo 2" = "Valor 2"}
```

Exemplos

```
cifilters{"General\Name" = "Windows 2016"}
```

```
cifilters{"General\Create Type" = "Manual"}
```

E por fim, para filtrar especificamente pelo servidor da imagem de exemplo usaríamos:

```
cifilters{"Basic Network Address\Hostname" = "srvapp01"}
```

4.4) `relatedcifilters{...}` — Filtros por atributos do ativo relacionado (B)

Filtra o ativo relacionado por campos/atributos.

Sintaxe

```
relatedcifilters{"Grupo\Campo" = "Valor"}
```

Exemplos

```
relatedcifilters{"General\Name" = "CRM"}
```

```
relatedcifilters{"General\Name" = "C:"}
```

4.5) `returns{...}` — Quais campos retornar (colunas) e aliases

Define quais atributos serão devolvidos no resultado.

Sintaxe

```
returns{"Grupo\Campo" as Alias, "Grupo\Campo 2" as Alias2, "Grupo\Campo 3"}
```

Exemplos

```
returns{"General\Name" as Name}
```

```
returns{
  "Basic Network Addresses\Manage IP Addresses" as IP,
  "General\Name" as Name,
  "Basic Network Addresses\Hostname" as Hostname
}
```

“ Boa prática: use `as` para nomes curtos (ex.: `Name`, `Hostname`, `IP`).

4.6) `orders{...}` — Ordenação

Define a ordenação dos resultados, normalmente pelo **alias** definido em `returns`.

Sintaxe

```
orders{Alias}
```

Exemplos

```
orders{Name,Hostname}
```

```
orders{Hostname}
```

5) Exemplos completos (prontos para uso)

Exemplo A — Web Application (sem relacionamentos)

```
ictypefilters{citytype in ["Web Application"]}  
returns{"General\Name" as Name}  
orders{Name}
```

Exemplo B — Team relacionado a Web Application (filtrando a aplicação)

```
ictypefilters{citytype in ["Team"] and relatedcitytype in ["Web Application"]}  
relsfilters{outrels in ["Service Owned"] and inrels in ["Service Owner"]}  
relatedcifilters{"General\Name" = "CRM"}  
returns{"General\Name" as Name}  
orders{Name}
```

Exemplo C — Server relacionado a Windows Disk Volume (filtrando ambos)

```
ictypefilters{citype in ["Servers"] and relatedcitype in ["Windows Disk Volume"]}
relsfilters{outrels in ["Is Composed"] and inrels in ["Composes"]}
cifilters{"General\Name" = "Windows 2016"}
relatedcifilters{"General\Name" = "C:"}
returns{
  "Basic Network Addresses\Manage IP Addresses" as IP,
  "General\Name" as Name,
  "Basic Network Addresses\Hostname" as Hostname
}
orders{Hostname}
```

6) Checklist rápido para montar sua PAQL

1. Escolha o tipo principal: `ictypefilters{citype in ["..."]}`
2. Se existir um “lado relacionado”, adicione:
 - `relatedcitype in ["..."]`
 - `relsfilters{...}`
 - `relatedcifilters{...}` (se precisar filtrar o relacionado)
3. Filtre o principal com `cifilters{...}` (se necessário)
4. Defina colunas com `returns{...}`
5. Ordene com `orders{...}` (normalmente pelo alias)