



## Paginação de Consultas obtidas a partir de um método GET

Para paginar os registros obtidos a partir de uma consulta, devem ser utilizados os parâmetros “início” e “quantidade” na requisição. Por exemplo:

<https://%dominio%.flow.gp/metadados.api/api/v1/Pessoas?inicio=1&quantidade=5>

## Ordenação de Consultas obtidas a partir de um método GET

Para ordenar os registros obtidos a partir de uma consulta, deve ser utilizado o parâmetro “ordem” indicando o campo referente a ordenação, caso a ordenação desejada seja uma combinação de campos os mesmos deverão ser separados por vírgula na requisição.

O tipo padrão das ordenações será ascendente, mas caso o usuário queira utilizar ordenação descendente deverá acrescentar também o parâmetro “ordemtipo” indicando “1” para descendente e “0” para ascendente.

Exemplo:

<https://%dominio%.flow.gp/metadados.api/api/v1/Pessoas?ordem=ativo,nome&ordemtipo=1>

## Parâmetros de filtro para Consultas obtidas a partir de um método GET

As consultas poderão utilizar diversos parâmetros distintos de filtragem que irão variar conforme o tipo do dado do campo utilizado:

- Se o campo for uma string poderão ser utilizadas as funções:
  - Pesquisa por um conteúdo exato sem função: deve ser identificado o campo testado após a requisição do método e utilizado apenas o texto a ser buscado sem aspas, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?nome=Camila Cabral`;
  - Pesquisa por um conteúdo exato com função: deve ser identificado o campo testado após a requisição do método e o texto deve ser inserido na função “\_equals()”, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?nome=_equals(Camila Cabral)`;
  - Pesquisa por um conteúdo diferente de: deve ser utilizada a função “\_notEquals()” passando entre os parêntesis o conteúdo, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?nome=_notEquals(Camila Cabral)`;
  - Pesquisa por um conteúdo aproximado: deve ser utilizada a função “\_like()” passando entre os parêntesis o conteúdo buscado, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?nome=_like(Cabral)`.
- Se o campo for um valor inteiro, currency ou enumeradores poderão ser utilizadas as funções:
  - Pesquisa exata por um conteúdo sem e com função tal qual o exemplo para campo string;
  - Pesquisa por conteúdo diferente de tal qual o exemplo para campo string;
  - Pesquisa por um conteúdo maior do que: deve ser identificado o campo testado após a requisição do método e utilizada a função “\_greater()” passando o valor entre os parêntesis, por exemplo:  
`...flow.gp/metadados.api/api/v1/ResultadosDeFaturamento?valor=_greater(1000)`;
  - Pesquisa por um conteúdo menor do que: deve ser utilizada a função “\_less()” passando o valor entre os parêntesis, por exemplo:  
`...flow.gp/metadados.api/api/v1/ResultadosDeFaturamento?valor=_less(1000)`.
- Se o campo for um datetime poderão ser utilizadas as funções:

- Pesquisa exata por uma data sem e com função tal qual o exemplo para campo string;
- Pesquisa por uma data diferente de tal qual o exemplo para campo string;
- Pesquisa por uma data maior do que: deve ser identificado o campo testado após a requisição do método e utilizada a função “\_greater()” passando uma data entre os parêntesis, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?dataNascimento=_greater(2010-10-25);`
- Pesquisa por uma data maior ou igual: deve ser identificado o campo testado após a requisição do método e utilizada a função “\_equalOrGreater()” passando uma data entre os parêntesis, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?dataNascimento=_equalOrGreater(2010-10-25);`
- Pesquisa por uma data menor do que: deve ser utilizada a função “\_less()” passando uma data entre os parêntesis, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?dataNascimento=_less(1940-12-31).`
- Pesquisa por uma data menor ou igual: deve ser utilizada a função “\_equalOrLess()” passando uma data entre os parêntesis, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?dataNascimento=_equalOrLess(1940-12-31).`
- Pesquisa por uma data entre um período informado: deve ser utilizada a função “\_between()” passando a primeira e a segunda data do período de filtro separando elas por vírgula e entre os parêntesis, por exemplo:  
`...flow.gp/metadados.api/api/v1/Pessoas?dataNascimento=_between(1940-12-01, 1940-12-31).`

### Estrutura de Resposta:

Os validadores de negócio e erros não tratados devem retornar num modelo padrão de json de resposta.

O modelo padrão de json de resposta deve exibir um “status\_processamento” que pode ser:

- 1 Solicitação processada com Sucesso
- 2 Solicitação processada com Erros
- 3 Sucesso com Erros

No caso de “Sucesso” os registros deverão ser retornados dentro do nodo “registros” com a estrutura de json apropriada para cada entidade, enquanto no caso de “Erro” o nodo de “registros” deverá retornar conteúdo nulo enquanto o nodo de “erros” deverá listar o erro identificado o “codigoDoErro”, a “mensagemDoErro” e o “registroComErro” para identificar o registro que originou o problema. Caso a API esteja realizando um POST ou PUT de uma coleção de registros, poderá ocorrer o caso de “Sucesso com Erro” aonde ambos os nodos de “registros” e “erros” terão conteúdo. (Ex.: Admissões Coletivas, Rescisões Coletivas, etc).

### Resposta com Sucesso:

```
{
  "retorno": {
    "status_processamento": 1,
    "status": "Sucesso",
    "total": 76,
    "registros": [
      {
        "idDaEmpresa": 5107,
        "codigo": "00100009",
        "descricao": "Indústria de Plásticos Limeira",
        "razaoSocial": "Indústria de Plásticos Limeira Ltda",
```

```

        "ativo": 1,
        "tipoDeInscricaoEmpregador": 1,
        "inscricaoEmpregador": "02017619000134",
        "naturezaJuridica": "206-2 - Sociedade Empresária Limitada",
        "idDaNaturezaJuridica": 97
    },
    { ... }
],
"erros": []
}
}

```

Resposta com Erro:

```

{
  "retorno": {
    "status_processamento": 2,
    "status": "Erro",
    "total": 0,
    "registros": [],
    "erros": [
      {
        "codigoErro": 9,
        "descricaoErro": "Registro não localizado",
        "registroErro": null,
        "mensagem": null
      }
    ]
  }
}

```

Os Códigos de Erro poderão ser:

- 0 token não informado
- 1 token inválido ou não encontrado
- 2 Formato de JSON inválido ou com erros
- 3 Erro no processamento do JSON
- 4 Parâmetro não informado
- 5 A Consulta não retornou registros
- 6 A Consulta esgotou o tempo de retorno do BD
- 7 Erro do validador de registro único.
- 8 Erros de Validação
- 9 Registro não localizado

**Exemplo de requisição GET combinando parâmetros de filtro, ordem e paginação**

[.flow.gp/metadados.api/api/v1/Pessoas?dataNascimento=\\_less\(1950-12-31\)&ordem=nome&ordemtipo=1&inicio=1&quantidade=5](https://flow.gp/metadados.api/api/v1/Pessoas?dataNascimento=_less(1950-12-31)&ordem=nome&ordemtipo=1&inicio=1&quantidade=5)

Resultado:

```

{
  "retorno": {

```

```
"status_processamento": 1,
"status": "Sucesso",
"total": 32,
"registros": [
  {
    "id": 8376,
    "nome": "Victor e Léo",
    "nomeReduzido": "Victor Léo",
    "trabalhadorAposentado": 0,
    "dataAposentadoria": null,
    "deficienciaFisicaMatora": 0,
    "deficienciaVisual": 0,
    "deficienciaAuditiva": 0,
    "deficienciaIntelectual": 0,
    "deficienciaMental": 0,
    "reabilitado": 0,
    "observacoesDeficiencia": null,
    "dataNascimento": "1943-09-09T01:00:00",
    "sexo": 2,
    "cpf": "02348976005",
    "pis": "49014496332",
    "identidade": "111119431",
    "tipoIdentidade": 1,
    "dataIdentidade": null,
    "carteiraProfissional": null,
    "serieCarteiraProfissional": null,
    "dataCarteiraProfissional": null,
    "dataCadastroPIS": null,
    "numeroCartaoSUS": null,
    "orgaoEmissorIdentidade": null,
    "ativo": 1
  },
  {
    "id": 5147,
    "nome": "Roberto Gomes Bolaños",
    "nomeReduzido": "Roberto Bolaños",
    "trabalhadorAposentado": 0,
    "dataAposentadoria": null,
    "deficienciaFisicaMatora": 0,
    "deficienciaVisual": 0,
    "deficienciaAuditiva": 0,
    "deficienciaIntelectual": 0,
    "deficienciaMental": 0,
    "reabilitado": 0,
    "observacoesDeficiencia": null,
    "dataNascimento": "1940-04-10T00:00:00",
    "sexo": 2,
    "cpf": "17651917006",
    "pis": "13762315182",
    "identidade": "4449874511471",
    "tipoIdentidade": 1,
    "dataIdentidade": "1974-01-01T00:00:00",
    "carteiraProfissional": "47778551",
    "serieCarteiraProfissional": "111",
    "dataCarteiraProfissional": "1974-01-01T00:00:00",
    "dataCadastroPIS": "1974-01-01T00:00:00",
    "numeroCartaoSUS": null,
    "orgaoEmissorIdentidade": "SSP",
  }
]
```

```
    "ativo": 1
  },
  {
    "id": 9119,
    "nome": "PESSOA TESTE DE FATURAMENTO 1",
    "nomeReduzido": "PESSOA 1",
    "trabalhadorAposentado": 0,
    "dataAposentadoria": null,
    "deficienciaFisicaMatora": 0,
    "deficienciaVisual": 0,
    "deficienciaAuditiva": 0,
    "deficienciaIntelectual": 0,
    "deficienciaMental": 0,
    "reabilitado": 0,
    "observacoesDeficiencia": null,
    "dataNascimento": "1944-04-04T00:00:00",
    "sexo": 1,
    "cpf": "52362387062",
    "pis": "86725466429",
    "identidade": null,
    "tipoIdentidade": null,
    "dataIdentidade": null,
    "carteiraProfissional": null,
    "serieCarteiraProfissional": null,
    "dataCarteiraProfissional": null,
    "dataCadastroPIS": null,
    "numeroCartaoSUS": null,
    "orgaoEmissorIdentidade": null,
    "ativo": 1
  },
  {
    "id": 9133,
    "nome": "Pessoa Teste 1",
    "nomeReduzido": " ",
    "trabalhadorAposentado": 0,
    "dataAposentadoria": null,
    "deficienciaFisicaMatora": 0,
    "deficienciaVisual": 0,
    "deficienciaAuditiva": 0,
    "deficienciaIntelectual": 0,
    "deficienciaMental": 0,
    "reabilitado": 0,
    "observacoesDeficiencia": "",
    "dataNascimento": "1900-01-01T00:00:00",
    "sexo": 2,
    "cpf": "13820806032",
    "pis": null,
    "identidade": null,
    "tipoIdentidade": 1,
    "dataIdentidade": null,
    "carteiraProfissional": null,
    "serieCarteiraProfissional": null,
    "dataCarteiraProfissional": null,
    "dataCadastroPIS": null,
    "numeroCartaoSUS": null,
    "orgaoEmissorIdentidade": null,
    "ativo": 1
  },
}
```

```
{
  {
    "id": 8508,
    "nome": "Nair Regina Rocha",
    "nomeReduzido": "Nair Rocha",
    "trabalhadorAposentado": null,
    "dataAposentadoria": null,
    "deficienciaFisicaMatora": 0,
    "deficienciaVisual": 0,
    "deficienciaAuditiva": 0,
    "deficienciaIntelectual": 0,
    "deficienciaMental": 0,
    "reabilitado": 0,
    "observacoesDeficiencia": null,
    "dataNascimento": "1946-02-27T00:00:00",
    "sexo": 1,
    "cpf": "97505551558",
    "pis": "44795578042",
    "identidade": "431314172",
    "tipoIdentidade": 1,
    "dataIdentidade": null,
    "carteiraProfissional": null,
    "serieCarteiraProfissional": null,
    "dataCarteiraProfissional": null,
    "dataCadastroPIS": null,
    "numeroCartaoSUS": null,
    "orgaoEmissorIdentidade": null,
    "ativo": 1
  }
},
"erros": []
}
}
```