

# Manual de Criação de Configuradores para Algoritmos do CSGGrid

TecGraf/PUC-Rio  
*csggrid@tecgraf.puc-rio.br*

## Resumo

O objetivo deste documento é registrar em um único local todas as informações necessárias para criar os arquivos de configuração de algoritmos executáveis no CSGGrid. A partir de um configurador xml de um algoritmo do repositório, o cliente CSGGrid cria dinamicamente uma janela para entrada dos parâmetros para execução desse algoritmo. O arquivo xml com esse configurador é obrigatório e deve ser cadastrado no repositório de algoritmos de acordo com as diferentes versões disponíveis. Este documento está organizado da seguinte forma: na Seção 1 fazemos uma breve introdução aos conceitos principais de uma descrição xml. Na Seção 2 apresentamos os elementos para definição de algoritmos e seus parâmetros de entrada e saída. Na Seção 3 mostramos como definir expressões booleanas e algumas funções pré-definidas que podem ser usadas nessas expressões. Na Seção 4 apresentamos ações que podem ser aplicadas aos parâmetros dos algoritmos e que alteram a exibição da interface gráfica associada ao algoritmo. Por fim, na Seção 5, apresentamos alguns exemplos de configuração de algoritmos, partindo de um algoritmo simples e mostrando uma construção passo a passo. O Apêndice A possui a lista dos tipos de arquivos que podem ser usados nos elementos xml usados para definir parâmetros de entrada e saída desses tipos.

# Sumário

<b>1</b>	<b>Introdução ao XML</b>	<b>6</b>
1.1	O que é XML? . . . . .	6
1.2	Estrutura de um XML . . . . .	6
1.3	Declaração XML . . . . .	7
1.4	Comentários . . . . .	8
1.5	Elementos . . . . .	8
1.6	Elemento Raiz . . . . .	9
1.7	Elementos Filhos . . . . .	9
1.8	Atributos . . . . .	9
<b>2</b>	<b>Algoritmos e Parâmetros</b>	<b>11</b>
2.1	Algoritmos . . . . .	11
2.1.1	<algoritmo> . . . . .	11
2.2	Grupo de Parâmetros . . . . .	12
2.2.1	<grupo> . . . . .	12
2.2.2	<grupo_de_entrada> . . . . .	12
2.2.3	<grupo_de_saida> . . . . .	12
2.3	Parâmetros . . . . .	13
2.3.1	<booleano> . . . . .	13
2.3.2	<enumeracao> . . . . .	14

2.3.3	<texto>	15
2.3.4	<inteiro>	15
2.3.5	<real>	16
2.3.6	<arquivo_de_entrada>	16
2.3.7	<arquivo_de_saida>	17
2.3.8	<tabela>	17
2.3.9	<lista_de_texto>	18
2.3.10	<lista_de_reais>	18
2.3.11	<lista_de_inteiros>	19

### **3 Expressões** 20

3.1	Funções de Tabelas	20
3.2	Funções de Listas	20
3.3	Funções Matemáticas	20
3.4	<expressao>	22

### **4 Ações** 23

4.1	Habilitar/Desabilitar elemento	23
4.1.1	<habilitar>	23
4.1.2	<desabilitar>	24
4.2	Exibir/Ocultar elemento	24
4.2.1	<ocultar>	24

4.3	Exibir/Ocultar item de enumeração . . . . .	25
4.3.1	<ocultar_item> . . . . .	25
4.3.2	<exibir_item> . . . . .	25
4.4	Trocar Máximo/Mínimo . . . . .	26
4.4.1	<trocar_maximo> . . . . .	26
4.4.2	<trocar_minimo> . . . . .	26
4.5	Trocar Tipo de Arquivo . . . . .	27
4.5.1	<trocar_tipo_de_arquivo> . . . . .	27
4.6	Carregar Parâmetros . . . . .	27
4.6.1	<carregar_parametros> . . . . .	27
<b>5</b>	<b>Exemplos</b>	<b>29</b>
5.1	Exemplo Básico . . . . .	29
5.2	Exemplo com Vários Elementos . . . . .	30
5.2.1	Passo 1: Todo configurador precisa ter no mínimo o elemento-raiz algo- ritmo . . . . .	32
5.2.2	Passo 2: Inserindo o grupo de saída . . . . .	33
5.2.3	Passo 3: Inserindo o grupo de entrada . . . . .	33
5.2.4	Passo 4: Criando o grupo de seleção . . . . .	35
5.2.5	Passo 5: Inserindo uma expressão para validar campos relacionados . . .	36
5.2.6	Passo 6: Inserindo três campos de inteiros . . . . .	38
5.2.7	Passo 7: Inserindo booleano que controlará exibição dos inteiros . . . . .	39



5.2.8	Passo 8: Inserindo ações nos itens de enumeração . . . . .	41
5.3	Simulador MultiEnv . . . . .	43
<b>A</b>	<b>Tipos de Arquivos</b>	<b>47</b>

# 1 Introdução ao XML

No CSGGrid, cada algoritmo é composto de um arquivo que descreve o algoritmo, um arquivo que contém a definição de interface e os arquivos binários para cada plataforma de execução. Essa seção descreve como criar os arquivos de definição de interface, chamados configuradores e representados por um arquivo config.xml.

## 1.1 O que é XML?

XML, eXtensible Markup Language, é uma linguagem para descrição de dados.

## 1.2 Estrutura de um XML

Um documento XML é composto por elementos de dados. A sintaxe básica para um elemento XML é mostrada a seguir:

```
<nome_do_elemento>Texto</nome_do_elemento>
```

É possível colocar elementos dentro de outros elementos de forma a representar informação relacionada. O exemplo a seguir mostra como agrupar o nome e endereço para um empregado no elemento:

```
<empregado>  
  <nome>Maria Aparecida</nome>  
  <endereço>Rua Terra, 13</endereço>  
</empregado>
```

Os outros itens que compõem um XML, além de elementos, são:

- Declaração XML
- Comentários

- Elemento raiz
- Elementos filhos
- Elementos vazios
- Atributos

Um XML exemplo seria:

```
<?xml version="1.0" encoding="ISO-8859-1">
<!-- Empregados da Empresa -->
<empregados>
  <empregado>
    <nome idade="26">Maria Aparecida</nome>
    <endereco>Rua Terra, 13</endereco>
    <estagiario/>
  </empregado>
</empregados>
```

Todas as construções existentes no exemplo acima serão explicadas a seguir.

### 1.3 Declaração XML

A declaração XML no começo de um documento XML é a forma de dizer a um processador que o tipo do conteúdo do documento é XML. A sintaxe da declaração XML é a seguinte:

```
<?xml version="1.0" encoding="encoding"?>
```

O atributo **version** indica a versão de XML que está sendo usada no documento.

O atributo **encoding** especifica o conjunto de caracteres em uso no documento. Alguns dos valores mais comuns para o atributo **encoding** são:

- **ISO-8859-1**: Alfabeto latino 1. Usado para a maioria das linguagens do oeste europeu. Similar ao ASCII de 8 bits. Permite o uso de acentuação.

- **UTF-8:** Unicode de 8 bits.
- **UTF-16:** Unicode de 16 bits usado para conjuntos de caracteres internacionais.

## 1.4 Comentários

Comentários podem ser colocados em qualquer lugar dentro de um documento XML, exceto dentro de uma tag. Comentários são cercados por `<!--` e `-->` e podem abranger várias linhas. A restrição é que não podemos colocar `--` dentro de um comentário. Comentários aninhados também não são permitidos. Exemplo de comentário aninhado que **NÃO SÃO PERMITIDOS**:

```
<!-- Comentário externo <!-- comentário aninhado --> -->
```

## 1.5 Elementos

Como dito anteriormente, elementos são os itens básicos de um XML. São de dois tipos: aqueles que possuem conteúdo e aqueles que são vazios. Quando o elemento possui conteúdo, o texto deve estar entre a tag de abertura e tag de fechamento, como a seguir:

```
<dados>texto do dados</dados>
```

Elementos vazios são aqueles que não possuem texto e não possuem elementos filhos dentro dele. Se você precisa de um elemento que não precisa de nenhum texto ou filho associado, uma forma de representá-lo é:

```
<elementovazio></elementovazio>
```

Ou a forma abreviada:

```
<elementovazio/>
```



## 1.6 Elemento Raiz

O elemento raiz contém todos os outros elementos e atributos no documento. Pode haver um único elemento raiz no documento XML. Todos os outros elementos devem ser definidos dentro do elemento raiz.

## 1.7 Elementos Filhos

Dados em um documento XML são hierárquicos. Elementos podem conter outros elementos que são chamados de elementos filhos. No exemplo a seguir o elemento nome é um elemento filho do elemento empregado:

```
<empregado>
  <nome>Maria Aparecida</nome>
</empregado>
```

## 1.8 Atributos

Utilizam-se atributos para fornecer informações adicionais sobre um elemento. Os valores dos atributos devem ser sempre cercados por aspas (simples ou duplas). Alguns exemplos:

- A informação do tipo do telefone é armazenada no atributo tipo do elemento telefone:

```
<telefone tipo="celular">95644545</telefone>
```

- Se o valor do atributo contiver aspas duplas, utilize aspas simples:

```
< pessoa nome='Jonas "Cenoura" da Silva' />
```

- Se o valor do atributo contiver aspas simples utilize aspas duplas:

```
<pessoa nome="Jonas 'Cenoura' da Silva" />
<empregado sexo="masculino">
  <nome>Fernando</nome>
</empregado>
```

O configurador de algoritmos utiliza uma extensão dos atributos para permitir atributos multivalorados. Assim como qualquer valor de atributo, o conjunto de valores deve estar entre aspas, mas os valores são separados por vírgula. Exemplo:

```
<item_de_enumeracao dependentes="dep1,dep2,mais1,outro">
```

## 2 Algoritmos e Parâmetros

Atenção: os atributos marcados com \* são obrigatórios.

Os tipos dos atributos podem ser os seguintes:

- **string**: é uma cadeia de quaisquer caracteres.  
Ex: "Rodrigo", "abcD1234", "01234".
- **booleano**: é um valor verdadeiro ou falso.  
Os valores aceitos são os seguintes (não importa se usam maiúsculas ou minúsculas):  
Para verdadeiro: "sim", "verdadeiro", "v", "s", "true"  
Para falso: "não", "nao", "falso", "n", "f", "false"
- **inteiro**: é um valor inteiro. Ex: "0", "10", "-130".
- **real**: é um valor real. Ex: "0.21", "10.10".
- **lista**: é um conjunto de valores de um dos tipos acima, separados por vírgula. Exemplos:  
Lista de Inteiros: "1, 2, 10, 100"  
Lista de Reais: "10.1, 20.6"  
Lista de Strings: "INITCHR, Rodrigo, 1245abcd"

### 2.1 Algoritmos

#### 2.1.1 <algoritmo>

O elemento algoritmo é o elemento raiz dos configuradores de simuladores. Todos os outros elementos devem estar dentro do elemento algoritmo. Os atributos deste elemento são:

**\*comando** (*string*): define o nome do binário e, possivelmente, seus parâmetros.

**shell** (*string*): define em qual shell o algoritmo será executado.

**tipo\_de\_execucao** (*simples, multipla, paralela*): define se o simulador poderá ser executado em diversas máquinas simultaneamente ou se usará um cluster.

**capturar\_codigo\_de\_saida** (*booleano*): define se o binário sabe informar, através do seu código de retorno, se ele foi executado com sucesso (0) ou erro (diferente de 0).

**fornecer\_identificador** (*booleano*): inclui o parâmetro `cmdId=<identificador único para o comando>` nos parâmetros enviados para a execução.

## 2.2 Grupo de Parâmetros

### 2.2.1 <grupo>

**\*rotulo** (*string*): define o rótulo que será exibido na tela, para representar o grupo.

Existem alguns elementos já criados baseados no elemento <grupo>. O rótulo desses elementos já é criado automaticamente.

### 2.2.2 <grupo\_de\_entrada>

É equivalente ao elemento <grupo rotulo="Parâmetros de entrada">. O exemplo a seguir produz o *widget* da Figura 1.

```
<grupo_de_entrada>
  <arquivo_de_entrada nome="entrada"
    rotulo="Arquivo de entrada"
    dica="Informe o arquivo (.p3d) de entrada."
    tipo="P3D"/>
</grupo_de_entrada>
```

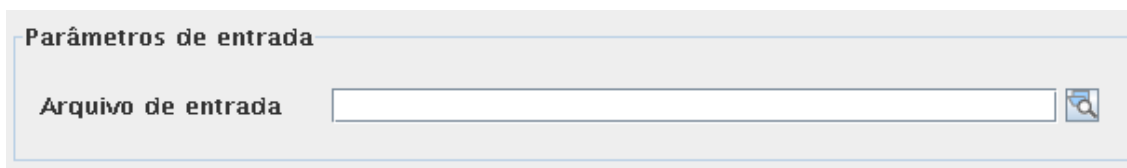
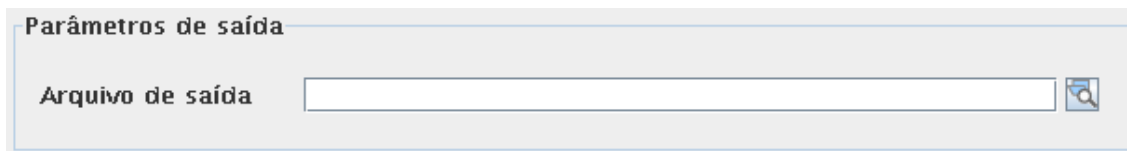


Figura 1: Exemplo de uso do elemento <grupo\_de\_entrada>

### 2.2.3 <grupo\_de\_saida>

É equivalente ao elemento <grupo rotulo="Parâmetros de saída">. O exemplo a seguir produz o *widget* da Figura 2.

```
<grupo_de_saida>  
  <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"  
    dica="Informe o arquivo de saída." tipo="ZIP"  
    forcar_extensao="sim"/>  
</grupo_de_saida>
```



Parâmetros de saída

Arquivo de saída

Figura 2: Exemplo de uso do elemento <grupo\_de\_saida>

## 2.3 Parâmetros

Todos os parâmetros simples possuem os seguintes atributos:

**padrão** (*depende do tipo de parâmetro*): define o valor que será utilizado como padrão pelo parâmetro.

**\*rotulo** (*string*): define o nome do parâmetro, que será mostrado ao usuário.

**\*nome** (*string*): define o nome real do parâmetro. Esse nome pode ser usado para se fazer referência a esse atributo em outros tags no configurador.

**opcional** (*booleano*): define se um parâmetro será obrigatório ou opcional.

**\*dica** (*string*): define a dica que será apresentada ao usuário quando este aponta o mouse em cima do parâmetro.

### 2.3.1 <booleano>

Os valores utilizados por padrão para falso e verdadeiro são, respectivamente, 0 e 1.

Caso seja necessário passar outros valores, basta informá-los nos atributos abaixo:

**verdadeiro** (*string*): define o novo valor para verdadeiro.

**falso** (*string*): define o novo valor para falso.

O exemplo a seguir produz o *widget* da Figura 3.

```
<booleano nome="poten" rotulo="Executar o Poten?"  
dica="Marque se deseja executar o Poten."  
padrao="sim" verdadeiro="SIM" falso="NAO"/>
```

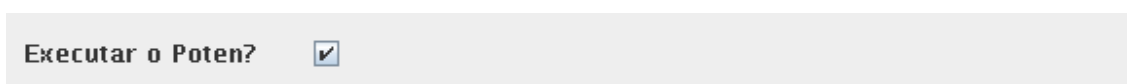


Figura 3: Exemplo de uso do elemento `<booleano>`

### 2.3.2 `<enumeracao>`

Um parâmetro de `enumeracao` é composto por um ou mais itens do tipo `<item_de_enumeracao>`. O tipo `<item_de_enumeracao>` possui os seguintes atributos:

**dependentes** (*lista de strings*): define uma lista de nomes de parâmetros que serão exibidos quando o item é selecionado.

**\*id** (*string*): define o identificador do item.

**\*rotulo** (*string*): define o nome do item, que será mostrado ao usuário.

O exemplo a seguir produz o *widget* da Figura 4.

```
<enumeracao nome="solver" rotulo="Solver"  
dica="Selecione o solver que será utilizado" padrao="skyline">  
  <item_de_enumeracao id="skyline" rotulo="SKYLINE"  
    valor="SKYLINE"/>  
  <item_de_enumeracao id="snll" rotulo="SNLL" valor="SNLL"/>  
  <item_de_enumeracao id="jacobi" rotulo="JACOBI" valor="JACOBI"/>  
  <item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>  
</enumeracao>
```

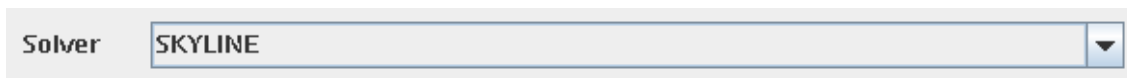


Figura 4: Exemplo de uso do elemento <enumeracao>

### 2.3.3 <texto>

**maximo\_de\_caracteres** (*inteiro*): quantidade máxima de caracteres que o usuário poderá digitar na caixa de texto.

O exemplo a seguir produz o *widget* da Figura 5.

```
<texto nome="CARTAO" rotulo="Código de cartão"  
dica="Informe o código de cartão." padrao="VELF"  
maximo_de_caracteres="5"/>
```



Figura 5: Exemplo de uso do elemento <texto>

### 2.3.4 <inteiro>

**maximo** (*inteiro*): define o valor máximo que será aceito pelo parâmetro.

**minimo** (*inteiro*): define o valor mínimo que será aceito pelo parâmetro.

O exemplo a seguir produz o *widget* da Figura 6.

```
<inteiro nome="CDP_INC" rotulo="Incremento de CMP"  
dica="Informe o incremento de CMP." padrao="1" minimo="1"/>
```



Figura 6: Exemplo de uso do elemento <inteiro>

### 2.3.5 <real>

**maximo** (*real*): define o valor máximo que será aceito pelo parâmetro.

**minimo** (*real*): define o valor mínimo que será aceito pelo parâmetro.

**incluir\_maximo** (*booleano*): define se o valor definido no atributo máximo será aceito no intervalo de valores.

**incluir\_minimo** (*booleano*): define se o valor definido no atributo mínimo será aceito no intervalo de valores.

O exemplo a seguir produz o *widget* da Figura 7.

```
<real nome="TINI" rotulo="Tempo inicial (ms)" dica="Informe o  
CMP inicial." padrao="0.0" minimo="0.0"/>
```



Tempo inicial (ms)

Figura 7: Exemplo de uso do elemento <real>

### 2.3.6 <arquivo\_de\_entrada>

**tipo** (*string*): define o tipo do arquivo a ser utilizado. O Apêndice A contém uma tabela com todos os tipos de arquivos suportados.

**selecao\_multipla** (*booleano*): define se será possível selecionar múltiplos arquivos de entrada.

**diretorio** (*booleano*): define se será possível selecionar diretórios como arquivos de entrada.

O exemplo a seguir produz o *widget* da Figura 8.

```
<arquivo_de_entrada nome="ARQUIVO_ENTRADA" rotulo="Arquivo de  
entrada" dica="Informe o caminho para o arquivo de entrada."  
tipo="TEXT"/>
```





Figura 8: Exemplo de uso do elemento <arquivo\_de\_entrada>

### 2.3.7 <arquivo\_de\_saida>

**tipo** (*string*): define o tipo do arquivo a ser utilizado. O Apêndice A contém uma tabela com todos os tipos de arquivos suportados.

**forçar\_extensao** (*booleano*): Indica se o arquivo deve criar uma extensão se ela não for informada.

O exemplo a seguir produz o *widget* da Figura 9.

```
<arquivo_de_saida nome="ARQUIVO_SAIDA" rotulo="Arquivo de  
saída" dica="Informe o caminho para o arquivo de saída"  
tipo="TEXT"/>
```

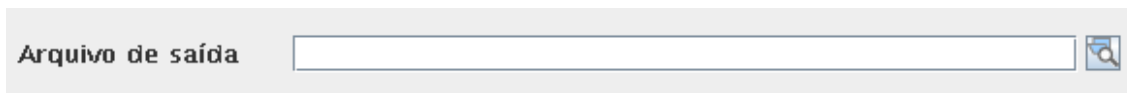


Figura 9: Exemplo de uso do elemento <arquivo\_de\_saida>

### 2.3.8 <tabela>

O elemento <tabela> é usado para criar uma estrutura tabular. Pode ter o seguinte atributo:

**delimitar\_tabela** (*booleano*): define se serão inseridos delimitadores na tabela.

Tabelas são compostas por uma ou mais colunas. As colunas podem ser:

- <coluna\_de\_texto>
- <coluna\_de\_reais>

- <coluna\_de\_inteiros>

As colunas possuem os seguintes atributos:

**opcional** (*booleano*): define se uma coluna pode ser deixada em branco.

**\*rotulo** (*string*): define o nome que será mostrado na interface como sendo o nome da coluna.

O exemplo a seguir produz o *widget* da Figura 10.

```
<tabela nome="LARG_FREQ" rotulo="Larguras e freqüências (Hz)
de interesse" dica="Informe as larguras e freqüências de
interesse.">
  <coluna_de_reais rotulo="Larguras" minimo="0.0"
    incluir_minimo="nao" maximo="20.0"/>
  <coluna_de_reais rotulo="Freqüências" minimo="0.0"
    incluir_minimo="nao" maximo="500.0"/>
</tabela>
```



Larguras	Freqüências

Adicionar  
Remover  
Remover todos

Figura 10: Exemplo de uso do elemento <tabela>

### 2.3.9 <lista\_de\_texto>

### 2.3.10 <lista\_de\_reais>

O exemplo a seguir produz o *widget* da Figura 11.

```
<lista_de_reais nome="FREQ" rotulo="Freqüências de definição
das faixas (Hz)" dica="Informe as freqüências de definição
das faixas." minimo="0"/>
```



Frequências de definição das faixas (Hz)

Adicionar

Editar...

Remover

Figura 11: Exemplo de uso do elemento <lista\_de\_reais>

### 2.3.11 <lista\_de\_inteiros>

### 3 Expressões

<![CDATA[EXPRESSÃO\_BOOLEANA]]>

Uma expressão do tipo:

*valor1* <= *valor2*

Deve ser escrita da seguinte forma:

<![CDATA[valor1 <= valor2]]>

Os valores passados para uma função podem ser literais (Ex: 10, 10.2, Rodrigo) ou nomes de atributos. É possível ainda chamar funções dentro de uma expressão. A seguir apresentamos as funções pré-definidas.

#### 3.1 Funções de Tabelas

**table\_min (nomeDaTabela, numeroDaColuna)** : retorna o elemento de menor valor na coluna de uma tabela.

**table\_max (nomeDaTabela, numeroDaColuna)** : retorna o elemento de maior valor na coluna de uma tabela.

#### 3.2 Funções de Listas

**list\_min (nomeDaLista)** : retorna o elemento de menor valor numa lista.

**list\_max (nomeDaLista)** : retorna o elemento de maior valor numa lista.

#### 3.3 Funções Matemáticas

As funções matemáticas podem ser aplicadas em valores inteiros e em valores reais.

**abs (real) ou abs (inteiro)** : retorna o valor absoluto de um número

**acos (real)** : retorna o arco-co-seno de um ângulo, no intervalo 0.0 a PI.

**asin (real)** : retorna o arco-seno de um ângulo, no intervalo -PI/2 a PI/2.

**atan (real)** : retorna o arco-tangente de um ângulo, no intervalo -PI/2 a PI/2.

**ceil (real)** : retorna o menor valor inteiro que seja maior do que o valor real passado.

**cos (real)** : retorna o co-seno de um ângulo.

**exp (real)** : retorna o número de *Euler* elevado a potência passada como parâmetro.

**floor (real)** : retorna o maior valor inteiro que seja menor do que o valor real passado.

**log (real)** : retorna o logaritmo real (base *e*) de um número.

**max (real, real) / max (inteiro, inteiro)** : retorna o maior dentre os dois valores passados como parâmetro.

**min (real, real) / min (inteiro, inteiro)** : retorna o menor dentre os dois valores passados como parâmetro.

**pow (real, real)** : retorna o valor do primeiro parâmetro elevado a potência do valor do segundo parâmetro.

**rint (real)** : retorna o valor inteiro mais próximo do valor real passado.

**sin (real)** : retorna o seno do ângulo passado como parâmetro.

**sqrt (real)** : retorna a raiz quadrada do valor passado como parâmetro.

**tan (real)** : retorna a tangente de um ângulo.

Exemplo de expressão com chamada de função:

```
<![CDATA[LARGJAN >= (1000/table_min(NOME_DA_TABELA,  
NUMERO_DA_COLUNA) ) ] ]>
```

### 3.4 <expressao>

**\*erro** (*string*): define a mensagem de erro que será apresentada ao usuário caso a expressão seja falsa.

Exemplo:

```
<expressao erro="O inline inicial deve ser menor ou igual ao  
inline final.">  
<![CDATA[INLINE_INICIAL <= INLINE_FINAL]]>  
</expressao>
```

## 4 Ações

### 4.1 Habilitar/Desabilitar elemento

#### 4.1.1 <habilitar>

O elemento <habilitar> determina a condição para que um parâmetro esteja habilitado na interface. É um elemento raiz, composto de um ou mais elementos filhos:

- <parametro>
- <condicao>

O elemento <parametro> possui os seguintes atributos:

**nome** (*string*): define o nome do parâmetro que será habilitado.

O elemento <condicao> possui os seguintes atributos:

**parametro** (*string*): define o nome do parâmetro que indicará se os demais parâmetros serão habilitados.

**valor** (*string*): define o valor que o parâmetro definido na condição deve ter para que os demais parâmetros sejam habilitados.

O exemplo a seguir habilita o parâmetro *listInt* quando a valor do parâmetro *checkBool* for verdadeiro.

```
<habilitar>
  <parametro nome="listInt"/>
  <condicao parametro="checkBool" valor="verdadeiro"/>
</habilitar>
```

#### 4.1.2 <desabilitar>

O elemento <desabilitar> determina a condição para que um parâmetro esteja desabilitado na interface. É um elemento raiz, composto de um ou mais elementos filhos:

- <parametro>
- <condicao>

O elemento <parametro> possui os seguintes atributos:

**nome** (*string*): define o nome do parâmetro que será desabilitado.

O elemento <condicao> possui os seguintes atributos:

**parametro** (*string*): define o nome do parâmetro que indicará se os demais parâmetros serão desabilitados.

**valor** (*string*): define o valor que o parâmetro definido na condição deve ter para que os demais parâmetros sejam desabilitados.

### 4.2 Exibir/Ocultar elemento

#### 4.2.1 <ocultar>

O elemento <ocultar> determina a condição para que um parâmetro não seja exibido na interface. Possui o seguinte atributo:

**parametro** (*string*): define o nome do parâmetro que será ocultado.

O elemento <condicao>, filho de <ocultar>, tem como atributos:

**parametro** (*string*): define o nome do parâmetro que indicará se os demais parâmetros serão ocultados.



**valor** (*string*): define o valor que o parâmetro definido na condição deve ter para que os demais parâmetros sejam ocultados.

## 4.3 Exibir/Ocultar item de enumeração

### 4.3.1 <ocultar\_item>

O elemento <ocultar\_item> determina a condição para que um item de um parâmetro de enumeração não seja exibido na interface. Possui o seguinte atributo:

**item** (*string*): define o identificador do item que será ocultado.

**parametro** (*string*): define o nome do parâmetro cujo item será ocultado.

O elemento <condicao>, filho de <ocultar>, tem como atributos:

**parametro** (*string*): define o nome do parâmetro que indicará se o item será ocultado.

**valor** (*string*): define o valor que o parâmetro definido na condição deve ter para que o item seja ocultado.

### 4.3.2 <exibir\_item>

O elemento <exibir\_item> determina a condição para que um item de um parâmetro de enumeração seja exibido na interface. Possui os seguintes atributos:

**item** (*string*): define o identificador do item que será exibido.

**parametro** (*string*): define o nome do parâmetro cujo item será exibido.

O elemento <condicao>, filho de <ocultar>, tem como atributos:

**parametro** (*string*): define o nome do parâmetro que indicará se o item será exibido.

**valor** (*string*): define o valor que o parâmetro definido na condição deve ter para que o item seja exibido.

## 4.4 Trocar Máximo/Mínimo

### 4.4.1 <trocar\_maximo>

O elemento <trocar\_maximo> substitui o valor máximo aceito por um determinado parâmetro do tipo real. Possui os seguintes atributos:

**parametro** (*string*): define o nome do parâmetro cujo valor máximo será alterado

**novo\_maximo** (*real*): permite informar o novo valor que será aceito como máximo pelo parâmetro.

**incluir\_novo\_maximo** (*booleano*): permite cadastrar o novo valor informado como máximo no intervalo de valores aceitos.

O exemplo a seguir atribui 100.0 como valor máximo aceito pelo parâmetro LIMITE\_INFERIOR.

```
<trocar_maximo parametro="LIMITE_INFERIOR" novo_maximo="100.0"/>
```

### 4.4.2 <trocar\_minimo>

O elemento <trocar\_minimo> substitui o valor mínimo aceito por um determinado parâmetro do tipo real. Possui os seguintes atributos:

**parametro** (*string*): define o nome do parâmetro cujo valor mínimo será alterado

**novo\_minimo** (*real*): permite informar o novo valor que será aceito como mínimo pelo parâmetro.

**incluir\_novo\_minimo** (*booleano*): permite cadastrar o novo valor informado como mínimo no intervalo de valores aceitos.

O exemplo a seguir atribui 1.0 como valor mínimo aceito pelo parâmetro LIMITE\_SUPERIOR.

```
<trocar_minimo parametro="LIMITE_SUPERIOR" novo_minimo="1.0"/>
```

## 4.5 Trocar Tipo de Arquivo

### 4.5.1 <trocar\_tipo\_de\_arquivo>

O elemento <trocar\_tipo\_de\_arquivo> substitui o tipo de arquivo de um parâmetro pelo novo tipo especificado. Possui os seguintes atributos:

**parametro** (*string*): define o nome do parâmetro cujo tipo de arquivo será trocado.

**novo\_tipo** (*string*): define o novo tipo do arquivo. Se não for especificado, o tipo do arquivo é retirado. O Apêndice A contém uma tabela com todos os tipos de arquivos suportados.

O exemplo a seguir faz com que o parâmetro ARQUIVO\_SAIDA passe a aceitar apenas arquivos do tipo SEGY.

```
<trocar_tipo_de_arquivo parametro="ARQUIVO_SAIDA" novo_tipo="SEGY"/>
```

## 4.6 Carregar Parâmetros

O carregador de parâmetros dispara outro simulador para preencher os valores dos elementos da interface descrita.

### 4.6.1 <carregar\_parametros>

O elemento <carregar\_parametros> Possui os seguintes atributos:

**algoritmo** (*string*): define o nome do simulador que será executado para preencher os elementos da interface.

**versao** (*string*): define a versão do simulador que será executado para preencher os elementos da interface.

**saida** (*string*): define o nome do arquivo de saída que o simulador irá gerar com os dados a serem preenchidos.

Tem como filho, o elemento <parametro\_de\_entrada> com os seguintes atributos:

**nome** (*string*): define o nome do parâmetro que será enviado ao simulador carregador de parâmetros.

Exemplo:

```
<carregar_parametros algoritmo="ExtractENV" versao="1.0.0" saida="ARQ_SAI">  
  <parametro_de_entrada nome="arquivos_entrada"/>  
</carregar_parametros>
```

## 5 Exemplos

### 5.1 Exemplo Básico

O seguinte XML produz o widget da Figura 12. Quando clicamos no parâmetro checkBool, o parâmetro listInt é habilitado.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Algoritmo Teste -->
<algoritmo comando="env">
  <grupo rotulo="Grupo do TesteEnv">
    <booleano nome="checkBool"
      rotulo="Deseja exibir uma lista de inteiros?"
      dica="Esta é a dica" padrao="falso"/>
    <lista_de_inteiros nome="listInt"
      rotulo="Uma lista de inteiros" dica="Dica da lista"/>
  </grupo>
  <habilitar>
    <parametro nome="listInt"/>
    <condicao parametro="checkBool" valor="verdadeiro"/>
  </habilitar>
</algoritmo>
```



Figura 12: Janela resultado do configurador xml do Exemplo Básico

## 5.2 Exemplo com Vários Elementos

O seguinte XML produz o widget da Figura 13.

```
<algoritmo comando="exemplo">
  <grupo_de_entrada>
    <arquivo_de_entrada nome="diretorio" rotulo="Diretório"
      dica="Informe o diretório que contém os arquivos de entrada."
      categoria="diretorio"/>
    <enumeracao nome="solver" rotulo="Solver"
      dica="Selecione o solver que será utilizado"
      padrao="skyline">
      <item_de_enumeracao id="skyline" rotulo="SKYLINE"
        valor="SKYLINE"/>
      <item_de_enumeracao id="snll" rotulo="SNLL" valor="SNLL"/>
      <item_de_enumeracao id="jacobi" rotulo="JACOBI"
        valor="JACOBI"/>
    </enumeracao>
  </grupo_de_entrada>
</algoritmo>
```

```
<item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>
</enumeracao>
</grupo_de_entrada>
<grupo rotulo="Parâmetros de Seleção">
  <real nome="TMPINI" rotulo="Tempo inicial (ms)"
    dica="Informe o tempo inicial." minimo="0.0"
    incluir_minimo="sim"/>
  <real nome="TMPFIM" rotulo="Tempo final (ms)"
    dica="Informe o tempo final." minimo="0.0"
    incluir_minimo="nao"/>
  <booleano nome="SELTRACO" rotulo="Selecionar Traços?"
    dica="Marque se desejar marcar traços." padrao="sim"/>
  <inteiro nome="INICHVSECD" rotulo="Chave secundária inicial"
    dica="Informe a chave secundária inicial." padrao="1"
    minimo="1"/>
  <inteiro nome="FIMCHVSECD" rotulo="Chave secundária final"
    dica="Informe a chave secundária final." padrao="999"
    minimo="1"/>
  <inteiro nome="INCCHVSECD" rotulo="Incremento da chave
    secundária" dica="Informe incremento da chave secundária"
    padrao="1" minimo="1"/>
</grupo>
<grupo_de_saida>
  <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"
    dica="Informe o arquivo de saída." tipo="TXT"/>
</grupo_de_saida>
<expressao erro="O tempo inicial tem que ser menor ou igual ao tempo final"
  <![CDATA[TMPINI <= TMPFIM]]>
</expressao>
<ocultar>
  <parametro nome="INICHVSECD"/>
  <parametro nome="FIMCHVSECD"/>
  <parametro nome="INCCHVSECD"/>
  <condição parametro="SELTRACO" valor="nao"/>
</ocultar>
<ocultar_item parametro="solver" item="jacobi">
  <condicao parametro="secsolver" valor="snll"/>
</ocultar_item>
</algoritmo>
```

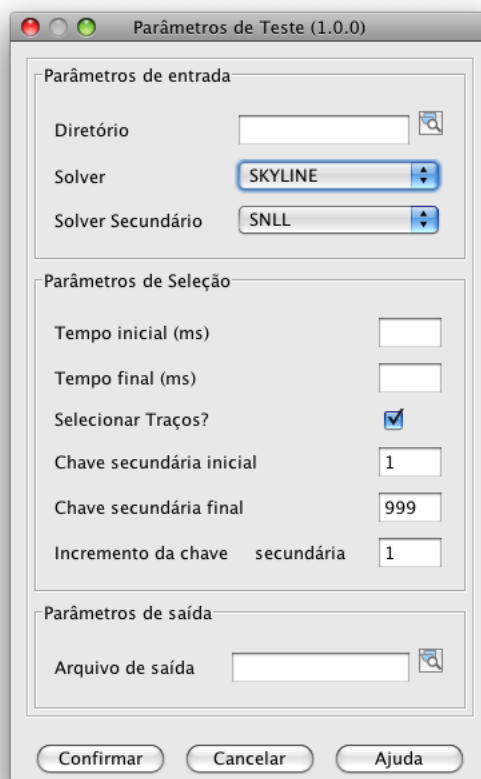


Figura 13: Janela resultado do configurador xml do Exemplo com Vários Elementos

### 5.2.1 Passo 1: Todo configurador precisa ter no mínimo o elemento-raiz algoritmo

```
<algoritmo comando="exemplo">  
</algoritmo>
```



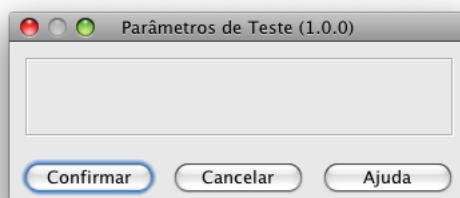


Figura 14: Passo 1

### 5.2.2 Passo 2: Inserindo o grupo de saída

```
<algoritmo comando="exemplo">  
  <grupo_de_saida>  
    <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"  
      dica="Informe o arquivo de saída." tipo="TXT"/>  
  </grupo_de_saida>  
</algoritmo>
```

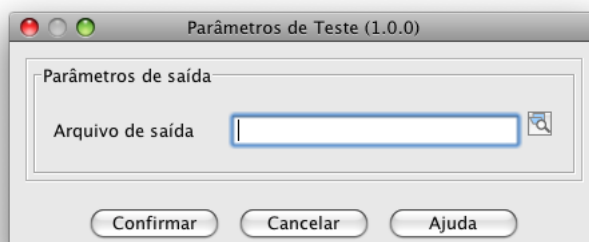


Figura 15: Passo 2

### 5.2.3 Passo 3: Inserindo o grupo de entrada

É importante notar que a ordem em que os elementos aparecem na tela é a ordem em que eles são inseridos no arquivo de configuração. Portanto, como queremos que o grupo\_de\_entrada apareça antes do grupo\_de\_saída, devemos inseri-lo antes no arquivo também.

```
<algoritmo comando="exemplo">
  <grupo_de_entrada>
    <arquivo_de_entrada nome="diretorio" rotulo="Diretório"
      dica="Informe o diretório que contém os arquivos de entrada."
      categoria="diretorio"/>
    <enumeracao nome="solver" rotulo="Solver"
      dica="Selecione o solver que será utilizado"
      padrao="skyline">
      <item_de_enumeracao id="skyline" rotulo="SKYLINE"
        valor="SKYLINE"/>
      <item_de_enumeracao id="snll" rotulo="SNLL" valor="SNLL"/>
      <item_de_enumeracao id="jacobi" rotulo="JACOBI"
        valor="JACOBI"/>
      <item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>
    </enumeracao>
  </grupo_de_entrada>
  <grupo_de_saida>
    <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"
      dica="Informe o arquivo de saída." tipo="TXT"/>
  </grupo_de_saida>
</algoritmo>
```

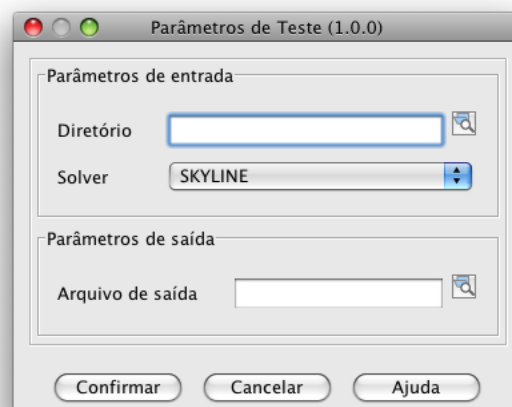


Figura 16: Passo 3

#### 5.2.4 Passo 4: Criando o grupo de seleção

```
<algoritmo comando="exemplo">
  <grupo_de_entrada>
    <arquivo_de_entrada nome="diretorio" rotulo="Diretório"
      dica="Informe o diretório que contém os arquivos de entrada."
      categoria="diretorio"/>
    <enumeracao nome="solver" rotulo="Solver"
      dica="Selecione o solver que será utilizado"
      padrao="skyline">
      <item_de_enumeracao id="skyline" rotulo="SKYLINE"
        valor="SKYLINE"/>
      <item_de_enumeracao id="snll" rotulo="SNLL" valor="SNLL"/>
      <item_de_enumeracao id="jacobi" rotulo="JACOBI"
        valor="JACOBI"/>
      <item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>
    </enumeracao>
  </grupo_de_entrada>
  <grupo rotulo="Parâmetros de Seleção">
    <real nome="TMPINI" rotulo="Tempo inicial (ms)"
      dica="Informe o tempo inicial." minimo="0.0"
      incluir_minimo="sim"/>
    <real nome="TMPFIM" rotulo="Tempo final (ms)"
      dica="Informe o tempo final." minimo="0.0"
      incluir_minimo="nao"/>
  </grupo>
  <grupo_de_saida>
    <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"
      dica="Informe o arquivo de saída." tipo="TXT"/>
  </grupo_de_saida>
</algoritmo>
```

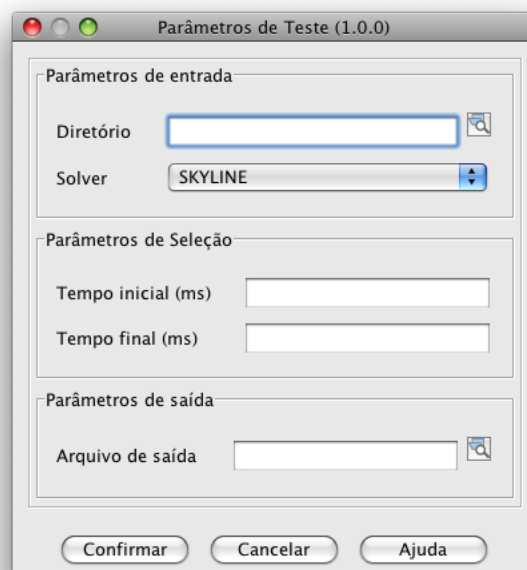


Figura 17: Passo 4

### 5.2.5 Passo 5: Inserindo uma expressão para validar campos relacionados

```
<algoritmo comando="exemplo">
  <grupo_de_entrada>
    <arquivo_de_entrada nome="diretorio" rotulo="Diretório"
      dica="Informe o diretório que contém os arquivos de entrada."
      categoria="diretorio"/>
    <enumeracao nome="solver" rotulo="Solver"
      dica="Selecione o solver que será utilizado"
      padrao="skyline">
      <item_de_enumeracao id="skyline" rotulo="SKYLINE"
        valor="SKYLINE"/>
      <item_de_enumeracao id="snll" rotulo="SNLL" valor="SNLL"/>
      <item_de_enumeracao id="jacobi" rotulo="JACOBI"
        valor="JACOBI"/>
      <item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>
    </enumeracao>
  </grupo_de_entrada>
```

```
<grupo rotulo="Parâmetros de Seleção">
  <real nome="TMPINI" rotulo="Tempo inicial (ms)"
    dica="Informe o tempo inicial." minimo="0.0"
    incluir_minimo="sim"/>
  <real nome="TMPFIM" rotulo="Tempo final (ms)"
    dica="Informe o tempo final." minimo="0.0"
    incluir_minimo="nao"/>
</grupo>
<grupo_de_saida>
  <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"
    dica="Informe o arquivo de saída." tipo="TXT"/>
</grupo_de_saida>
<expressao erro="O tempo inicial tem que ser menor ou igual ao tempo final."
  <![CDATA[TMPINI <= TMPFIM]]>
</expressao>
</algoritmo>
```

Quando a expressão é avaliada como falsa, a mensagem passada no atributo erro é exibida na tela.

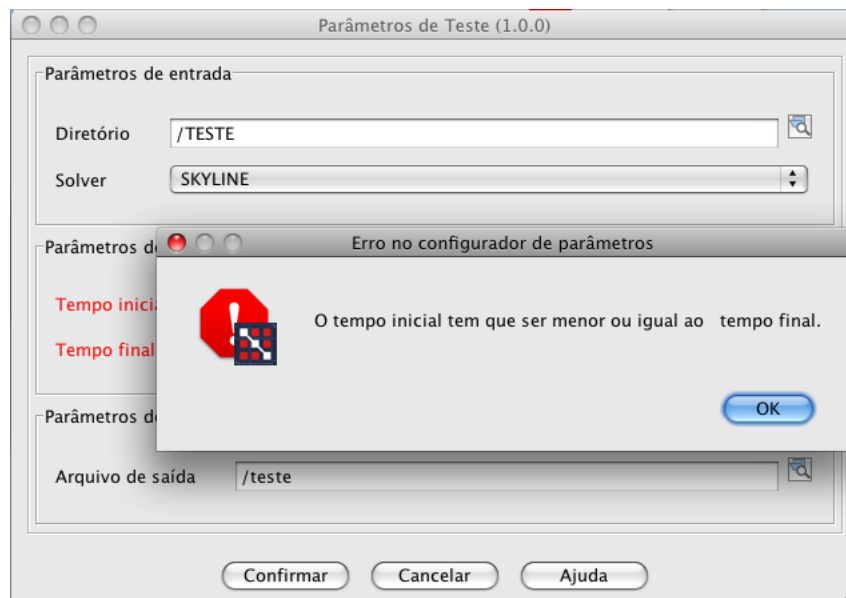


Figura 18: Passo 5

### 5.2.6 Passo 6: Inserindo três campos de inteiros

```
<algoritmo comando="exemplo">
  <grupo_de_entrada>
    <arquivo_de_entrada nome="diretorio" rotulo="Diretório"
      dica="Informe o diretório que contém os arquivos de entrada."
      categoria="diretorio"/>
    <enumeracao nome="solver" rotulo="Solver"
      dica="Selecione o solver que será utilizado"
      padrao="skyline">
      <item_de_enumeracao id="skyline" rotulo="SKYLINE"
        valor="SKYLINE"/>
      <item_de_enumeracao id="snll" rotulo="SNLL" valor="SNLL"/>
      <item_de_enumeracao id="jacobi" rotulo="JACOBI"
        valor="JACOBI"/>
      <item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>
    </enumeracao>
  </grupo_de_entrada>
  <grupo rotulo="Parâmetros de Seleção">
    <real nome="TMPINI" rotulo="Tempo inicial (ms)"
      dica="Informe o tempo inicial." minimo="0.0"
      incluir_minimo="sim"/>
    <real nome="TMPFIM" rotulo="Tempo final (ms)"
      dica="Informe o tempo final." minimo="0.0"
      incluir_minimo="nao"/>
    <inteiro nome="INICHVSECD" rotulo="Chave secundária inicial"
      dica="Informe a chave secundária inicial." padrao="1"
      minimo="1"/>
    <inteiro nome="FIMCHVSECD" rotulo="Chave secundária final"
      dica="Informe a chave secundária final." padrao="999"
      minimo="1"/>
    <inteiro nome="INCCHVSECD" rotulo="Incremento da chave
      secundária" dica="Informe incremento da chave secundária"
      padrao="1" minimo="1"/>
  </grupo>
  <grupo_de_saida>
    <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"
      dica="Informe o arquivo de saída." tipo="TXT"/>
  </grupo_de_saida>
  <expressao erro="O tempo inicial tem que ser menor ou igual ao tempo final.">
```

```
<![CDATA[TMPINI <= TMPFIM]]>
</expressao>
</algoritmo>
```

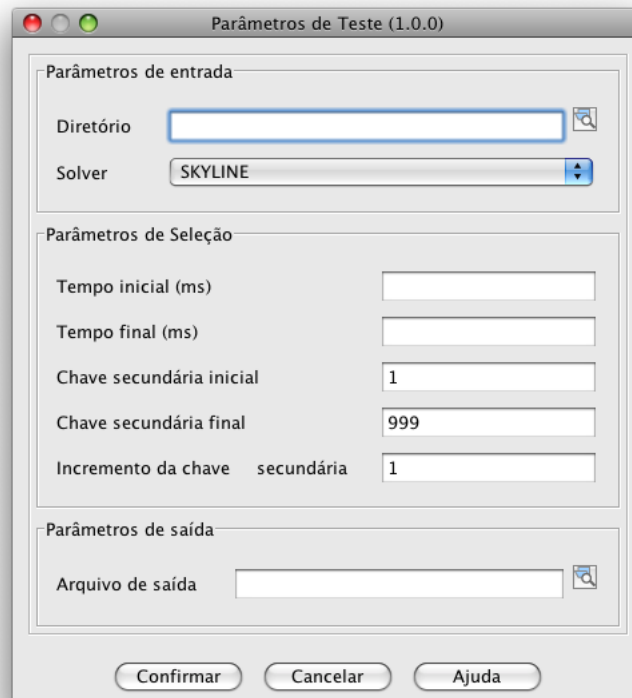


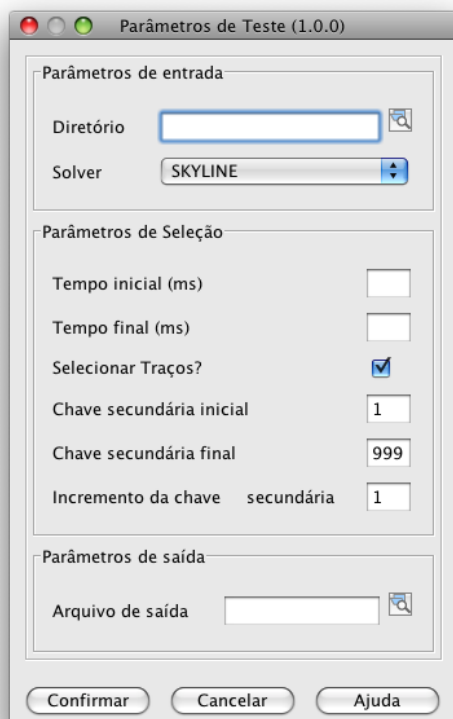
Figura 19: Passo 6

### 5.2.7 Passo 7: Inserindo booleano que controlará exibição dos inteiros

```
<algoritmo comando="exemplo">
  <grupo_de_entrada>
    <arquivo_de_entrada nome="diretorio" rotulo="Diretório"
      dica="Informe o diretório que contém os arquivos de entrada."
      categoria="diretorio"/>
    <enumeracao nome="solver" rotulo="Solver"
      dica="Selecione o solver que será utilizado"
      padrao="skyline">
```

```
<item_de_enumeracao id="skyline" rotulo="SKYLINE"
  valor="SKYLINE"/>
<item_de_enumeracao id="snll" rotulo="SNLL" valor="SNLL"/>
<item_de_enumeracao id="jacobi" rotulo="JACOBI"
  valor="JACOBI"/>
<item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>
</enumeracao>
</grupo_de_entrada>
<grupo rotulo="Parâmetros de Seleção">
  <real nome="TMPINI" rotulo="Tempo inicial (ms)"
    dica="Informe o tempo inicial." minimo="0.0"
    incluir_minimo="sim"/>
  <real nome="TMPFIM" rotulo="Tempo final (ms)"
    dica="Informe o tempo final." minimo="0.0"
    incluir_minimo="nao"/>
  <booleano nome="SELTRACO" rotulo="Selecionar Traços?"
    dica="Marque se desejar marcar traços." padrao="sim"/>
  <inteiro nome="INICHVSECD" rotulo="Chave secundária inicial"
    dica="Informe a chave secundária inicial." padrao="1"
    minimo="1"/>
  <inteiro nome="FIMCHVSECD" rotulo="Chave secundária final"
    dica="Informe a chave secundária final." padrao="999"
    minimo="1"/>
  <inteiro nome="INCCHVSECD" rotulo="Incremento da chave
    secundária" dica="Informe incremento da chave secundária"
    padrao="1" minimo="1"/>
</grupo>
<grupo_de_saida>
  <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"
    dica="Informe o arquivo de saída." tipo="TXT"/>
</grupo_de_saida>
<expressao erro="O tempo inicial tem que ser menor ou igual ao tempo final."
  <![CDATA[TMPINI <= TMPFIM]]>
</expressao>
<ocultar>
  <parametro nome="INICHVSECD"/>
  <parametro nome="FIMCHVSECD"/>
  <parametro nome="INCCHVSECD"/>
  <condição parametro="SELTRACO" valor="nao"/>
</ocultar>
</algoritmo>
```





Parâmetros de Teste (1.0.0)

Parâmetros de entrada

Diretório

Solver SKYLINE

Parâmetros de Seleção

Tempo inicial (ms)

Tempo final (ms)

Selecionar Traços? ☒

Chave secundária inicial 1

Chave secundária final 999

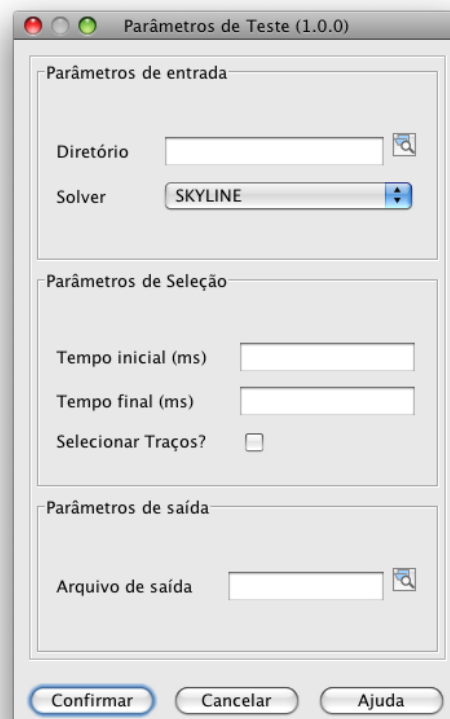
Incremento da chave secundária 1

Parâmetros de saída

Arquivo de saída

Confirmar Cancelar Ajuda

Figura 20: Passo 7 (opção habilitada)



Parâmetros de Teste (1.0.0)

Parâmetros de entrada

Diretório

Solver SKYLINE

Parâmetros de Seleção

Tempo inicial (ms)

Tempo final (ms)

Selecionar Traços? ☐

Parâmetros de saída

Arquivo de saída

Confirmar Cancelar Ajuda

Figura 21: Passo 7 (opção desabilitada)

## 5.2.8 Passo 8: Inserindo ações nos itens de enumeração

```
<algoritmo comando="exemplo">
  <grupo_de_entrada>
    <arquivo_de_entrada nome="diretorio" rotulo="Diretório"
      dica="Informe o diretório que contém os arquivos de entrada."
      categoria="diretorio"/>
    <enumeracao nome="solver" rotulo="Solver"
      dica="Selecione o solver que será utilizado"
      padrao="skyline">
      <item_de_enumerao id="skyline" rotulo="SKYLINE"
        valor="SKYLINE"/>
      <item_de_enumerao id="snll" rotulo="SNLL" valor="SNLL"/>
      <item_de_enumerao id="jacobi" rotulo="JACOBI"
        valor="JACOBI"/>
    </enumeracao>
  </grupo_de_entrada>
</algoritmo>
```

```
<item_de_enumeracao id="snmf" rotulo="SNMF" valor="SNMF"/>
</enumeracao>
</grupo_de_entrada>
<grupo rotulo="Parâmetros de Seleção">
  <real nome="TMPINI" rotulo="Tempo inicial (ms)"
    dica="Informe o tempo inicial." minimo="0.0"
    incluir_minimo="sim"/>
  <real nome="TMPFIM" rotulo="Tempo final (ms)"
    dica="Informe o tempo final." minimo="0.0"
    incluir_minimo="nao"/>
  <booleano nome="SELTRACO" rotulo="Selecionar Traços?"
    dica="Marque se desejar marcar traços." padrao="sim"/>
  <inteiro nome="INICHVSECD" rotulo="Chave secundária inicial"
    dica="Informe a chave secundária inicial." padrao="1"
    minimo="1"/>
  <inteiro nome="FIMCHVSECD" rotulo="Chave secundária final"
    dica="Informe a chave secundária final." padrao="999"
    minimo="1"/>
  <inteiro nome="INCCHVSECD" rotulo="Incremento da chave
    secundária" dica="Informe incremento da chave secundária"
    padrao="1" minimo="1"/>
</grupo>
<grupo_de_saida>
  <arquivo_de_saida nome="SAIDA" rotulo="Arquivo de saída"
    dica="Informe o arquivo de saída." tipo="TXT"/>
</grupo_de_saida>
<expressao erro="O tempo inicial tem que ser menor ou igual ao tempo final"
  <![CDATA[TMPINI <= TMPFIM]]>
</expressao>
<ocultar>
  <parametro nome="INICHVSECD"/>
  <parametro nome="FIMCHVSECD"/>
  <parametro nome="INCCHVSECD"/>
  <condição parametro="SELTRACO" valor="nao"/>
</ocultar>
<ocultar_item parametro="solver" item="jacobi">
  <condicao parametro="secsolver" valor="snll"/>
</ocultar_item>
</algoritmo>
```

O comando executado pelo exemplo acima terá o formato:

```
./exemplo diretorio=valor solver=valor secsolver=valor TMPINI=valor TMPFIM=valor  
SELTRACO=valor INICHVSECD=valor FIMCHVSECD=valor INCCHVSECD=valor  
SAIDA=valor
```

### 5.3 Simulador MultiEnv

O simulador **MultiEnv** utiliza o simulador **ExtractEnv** para preencher duas enumerações (linhas e variáveis) com base no arquivo fornecido pelo parâmetro *textarquivos\_entrada*.

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!-- Algoritmo MultiENV -->  
<algoritmo comando="multienv.lua" shell="lua5.1"  
  fornecer_identificador="sim">  
  <grupo rotulo="Seleção de Arquivos">  
    <carregar_parametros algoritmo="ExtractENV" versao="1.0.0"  
      saida="ARQ_SAI">  
      <parametro_de_entrada nome="arquivos_entrada"/>  
    </carregar_parametros>  
    <arquivo_de_entrada nome="arquivos_entrada"  
      rotulo="Diretório:"  
      dica="Informe o diretório que contém os arquivos de entrada."  
      selecao_multipla="sim" categoria="ambos" tipo="ENV"/>  
    <booleano nome="estatica" rotulo="Análise Estática:"  
      dica="Marque se deseja gerar a planilha a partir dos arquivos da análise.">  
    </booleano>  
    <booleano nome="dinamica" rotulo="Análise Dinâmica:"  
      dica="Marque se deseja gerar a planilha a partir dos arquivos da análise.">  
    </booleano>  
  </grupo>  
  <grupo_de_entrada>  
    <enumeracao nome="linhas" rotulo="Nomes das Linhas:"  
      selecao_multipla="sim" dica="Informe os nomes das linhas"/>  
    <enumeracao nome="variaveis" rotulo="Nomes das Variáveis:"  
      selecao_multipla="sim" dica="Informe o nomes das variáveis"/>  
    <enumeracao nome="opcoes_pesquisa"  
      rotulo="Opções de Pesquisa:"  
      dica="Selecione a opção de pesquisa" padrao="extreme">
```

```
<item_de_enumeracao id="extreme" rotulo="Extreme"
  valor="extreme"/>
<item_de_enumeracao id="segment" rotulo="Segment"
  valor="segment"/>
<item_de_enumeracao id="position" rotulo="Position"
  valor="position"/>
</enumeracao>
<real nome="posicao_linha" rotulo="Posição na Linha:"
  dica="Informe a posição na linha quando a opção de pesquisa for Positi
<real nome="posicao_inicial" rotulo="Posição Inicial:"
  dica="Informe a posição inicial quando a opção de pesquisa for Segment
<real nome="posicao_final" rotulo="Posição Final:"
  dica="Informe a posição inicial quando a opção de pesquisa for Segment
<enumeracao nome="opcoes_resultado"
  rotulo="Opções de Resultado:"
  dica="Selecione a opção de resultado" padrao="por_posicao">
  <item_de_enumeracao id="por_elemento" rotulo="Por Elemento"
    valor="element"/>
  <item_de_enumeracao id="por_posicao" rotulo="Por Posição"
    valor="position"/>
</enumeracao>
</grupo_de_entrada>
<grupo_de_saida>
  <arquivo_de_saida nome="saida" rotulo="Arquivo de Saída"
    dica="Informe o arquivo de saída" tipo="XLS"
    forcar_extensao="sim"/>
</grupo_de_saida> da análise estática" padrao="sim"/>
<habilitar>
  <parametro nome="posicao_inicial"/>
  <parametro nome="posicao_final"/>
  <condicao parametro="opcoes_pesquisa" valor="segment"/>
</habilitar>
<habilitar>
  <parametro nome="posicao_linha"/>
  <condicao parametro="opcoes_pesquisa" valor="position"/>
</habilitar>
<expressao erro="Pelo menos uma das análises deve ser selecionada.">
  <![CDATA[( estatica || dinamica )]]>
</expressao>
</algoritmo>
```

Parâmetros de MultiENV (1.0.0)

**Seleção de Arquivos**

Diretório: /TESTE

Análise Estática: ☒

Análise Dinâmica: ☐

Carregar Parâmetros

**Parâmetros de entrada**

Nomes das Linhas:

Disponíveis

Adicionar >

< Remover

Adicionar Todos >>

<< Remover Todos

Selecionados

Nomes das Variáveis:

Disponíveis

Adicionar >

< Remover

Adicionar Todos >>

<< Remover Todos

Selecionados

Opções de Pesquisa: Extreme

Posição na Linha:

Posição Inicial:

Posição Final:

Opções de Resultado: Por Posição

**Parâmetros de saída**

Arquivo de Saída

Confirmar Cancelar Ajuda

Figura 22: Janela resultado do configurador xml do Simulador MultiEnv

O simulador **ExtractEnv** gera um arquivo, cujo nome é dado pelo parâmetro *ARQ\_SAI*, com o conteúdo as enumerações, também no formato XML que é descrito neste texto.

O XML do configurador do **ExtractEnv** é o seguinte.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Algoritmo ExtractENV -->
<algoritmo comando="extractenv.lua" shell="lua5.1">
  <grupo_de_entrada>
    <arquivo_de_entrada rotulo="Diretório:"
      dica="Informe o diretório que contém os arquivos de entrada."
      selecao_multipla="sim" categoria="ambos" tipo="ENV"/>
  </grupo_de_entrada>
  <grupo_de_saida>
    <arquivo_de_saida nome="ARQ_SAI" rotulo="Arquivo de Saída"
      dica="Informe o arquivo de saída"/>
  </grupo_de_saida>
</algoritmo>
```

A saída da execução do simulador **ExtractEnv** é a seguinte:

```
<algoritmo>
  <parametro nome="linhas">
    <item_de_enumeracao id="19EG" rotulo="19EG" valor="19EG"/>
    <item_de_enumeracao id="19EG" rotulo="19EG" valor="18EG"/>
  </parametro>
  <parametro nome="variaveis">
    <item_de_enumeracao id="X" rotulo="X" valor="X"/>
    <item_de_enumeracao id="Y" rotulo="Y" valor="Y"/>
    <item_de_enumeracao id="Z" rotulo="Z" valor="Z"/>
    <item_de_enumeracao id="RX" rotulo="RX" valor="RX"/>
  </parametro>
</algoritmo>
```

## A Tipos de Arquivos

<b>Tipo</b>	<b>Formato</b>
2IF	Arquivo do UFLEX
ALGPAP	Arquivo de Parâmetros de Simuladores
COMPRESS	Arquivo Compress/UNIX
DAT	Arquivo de Dados
DOC	Arquivo Microsoft Word Document
DYN	Arquivo de Dados DYN do Dynasim
ENV	Arquivo de Envoltórias do Anflex
FLX	Fluxo de Algoritmos
GIF	Arquivo GIF
HIS	Arquivo de Séries Temporais do Anflex
HTML	Arquivo HTML
INC	Arquivo INC
JPEG	Arquivo JPEG
LOG	Arquivo de Log
MG	Arquivo MG
MTL	Modelo Geométrico
NF	Arquivo Neutro
P3B	Arquivo Binário do Pos3D
P3D	Arquivo de Dados do TPN
PARAMETERS	Parâmetros de Simulador
PDF	Arquivo PDF
POS	Arquivo de Pós-Processamento
SST	Arquivo do Sstab
TEXT	Arquivo Texto
TND	Arquivo de Dados TND do Dynasim
UNKNOWN	Formato Desconhecido
WAN	Arquivo de Dados do Wamit
XLS	Arquivo Excel
ZIP	Arquivo ZIP