BNF DA LINGUAGEM DE PROGRAMAÇÃO NOPL

```
<program>               ::= FBE <id> <fbe_body> END_FBE

<fbe_body>              ::= <include_block>
                          | <include_block> <fbe_body>
                          | <properties_block>
                          | <properties_block> <fbe_body>
                          | <instance>
                          | <instance> <fbe_body>
                          | <attribute>
                          | <attribute> <fbe_body>
                          | <vector_instance>
                          | <vector_instance> <fbe_body>
                          | <vector_attribute>
                          | <vector_attribute> >fbe_body>
                          | <method>
                          | <method> <fbe_body>
                          | <rule>
                          | <rule> <fbe_body>
                          | <formation_rule>
                          | <formation_rule> <fbe_body>
                          | <main_block>
                          | <main_block> <fbe_body>

<include_block>         ::= INCLUDES <target> END_INCLUDES

<properties_block>      ::= PROPERTIES <properties> END_PROPERTIES

<properties>            ::= <property>
                          | <property> <properties>

<property>              ::= TARGET <target>
                          | STRATEGY <strategy>

<target>                ::= CODE_GENERATION_EXAMPLE
                          | NAMESPACES
                          | FRAMEWORK_CPP_2_0
                          | FRAMEWORK_CPP_3_0

<strategy>              ::= NO_ONE
                          | BREADTH
                          | DEPTH
                          | PRIORITY

<instance>              ::= <visibility> <id> <id>

<attribute>             ::= <visibility> <type> <id> ASSIGN <factor>

<vector_instance>       ::= <visibility> <id> LB INTEGER_VALUE RB <id>

<vector_attribute>      ::= <visibility> <type> LB INTEGER_VALUE RB <id>

<vector_attribute>      ::= <visibility> <type> LB INTEGER_VALUE RB <id>
                               ASSIGN LC <array_factor> RC

<method>                ::= <visibility> METHOD <id> <method_body>
END_METHOD

<method_body>           ::= <code_blocks>
```

```
                          |  <params_body> <attribution>
                          |  <params_body> <code_blocks>
                          |  <attribution>

<params_body>            ::= PARAMS <params> END_PARAMS

<params>                 ::= <param>
                          |  <param> <params>

<param>                  ::= <type> <id>

<code_blocks>            ::= <code_block>
                          |  <code_block> <code_blocks>

<code_block>             ::= CODE <target> END_CODE

<attribution>            ::= ATTRIBUTION <element> ASSIGN <factor>
END_ATTRIBUTION

<rule>                   ::= RULE <id> <rule_properties_block> <condition>
                                <action> END_RULE
                          |  RULE <id> <condition> <action> END_RULE

<formation_rule>         ::= FORMATION_RULE <id> <rule_indexes>
                                <rule_properties_block> <condition>
                                <action> END_FORMATION_RULE
                          |  FORMATION_RULE <id> <rule_indexes> <condition>
                                <action> END_FORMATION_RULE

<rule_indexes>           ::= <rule_index>
                          |  <rule_index> <rule_indexes>

<rule_index>             ::= INDEX <id> FROM INTEGER_VALUE TO INTEGER_VALUE

<rule_properties_block> ::= PROPERTIES <rule_properties> END_PROPERTIES

<rule_properties>        ::= <rule_property>
                          |  <rule_property> <rule_properties>

<rule_property>          ::= PRIORITY INTEGER_VALUE
                          |  KEEPER <boolean>

<condition>              ::= CONDITION <subconditions> END_CONDITION
                          |  CONDITION <id> <subconditions> END_CONDITION
                          |  CONDITION <premises> END_CONDITION
                          |  CONDITION <id> <premises> END_CONDITION

<subconditions>          ::= <subcondition>
                          |  <subcondition> <conjunction> <subconditions>

<subcondition>           ::= SUBCONDITION <premises> END_SUBCONDITION
                          |  SUBCONDITION <id> <premises> END_SUBCONDITION

<premises>               ::= <premise>
                          |  <premise> <conjunction> <premises>

<premise>                ::= PREMISE <expression> END_PREMISE
                          |  PREMISE <id> <expression> END_PREMISE
                          |  PREMISE IMPERTINENT <expression> END_PREMISE
                          |  PREMISE IMPERTINENT <id> <expression>
END_PREMISE
```

```
120
121 <expression>            ::= <factor> <symbol> <factor>
122
123 <symbol>                ::= EQ
124                           |  NE
125                           |  LT
126                           |  GT
127                           |  LE
128                           |  GE
129
130 <action>                ::= ACTION <execution> <instigations> END_ACTION
131                           |  ACTION <id> <execution> <instigations>
132 END_ACTION
133                           |  ACTION <instigations> END_ACTION
134                           |  ACTION <id> <instigations> END_ACTION
135
136 <instigations>           ::= <instigation>
137                           |  <instigation> <instigations>
138
139 <instigation>            ::= INSTIGATION <execution> <calls> END_INSTIGATION
140                           |  INSTIGATION <id> <execution> <calls>
141 END_INSTIGATION
142                           |  INSTIGATION <calls> END_INSTIGATION
143                           |  INSTIGATION <id> <calls> END_INSTIGATION
144
145 <execution>             ::= SEQUENTIAL
146                           |  PARALLEL
147
148 <calls>                 ::= <call>
149                           |  <call> <calls>
150
151 <call>                  ::= CALL <elementcall>
152
153 <elementcall>           ::= <id> POINT <id> LP RP
154                           |  THIS POINT <id> LP RP
155                           |  <id> POINT <id> LP <arguments> RP
156                           |  THIS POINT <id> LP <arguments> RP
157
158 <arguments>             ::= <argument>
159                           |  <argument> COMMA <arguments>
160
161 <argument>              ::= <factor>
162
163 <type>                  ::= <basictype>
164                           |  <supertype>
165
166 <supertype>             ::= <basictype> LT INTEGER_VALUE GT
167
168 <basictype>             ::= BOOLEAN
169                           |  INTEGER
170                           |  DOUBLE
171                           |  STRING
172                           |  CHAR
173
174 <visibility>            ::= PRIVATE
175                           |  PUBLIC
176
177 <conjunction>           ::= AND
178                           |  OR
179
180 <array_factor>          ::= <array_factor> COMMA <factor>
```

```
181                          |  <factor>
182
183  <factor>                ::= <element>
184                          |  <boolean>
185                          |  INTEGER_VALUE
186                          |  DOUBLE_VALUE
187                          |  CHAR_VALUE
188                          |  STRING_VALUE
189
190  <boolean>               ::= TRUE
191                          |  FALSE
192
193  <element>               ::= <ID_or_VectElem> POINT <ID_or_VectElem>
194                          |  THIS POINT <ID_or_VectElem>
195                          |  <ID_or_VectElem>
196                          |  <ID_or_VectElem> POINT <VectElem>
197                          |  <VectElem> POINT <ID_or_VectElem>
198                          |  <VectElem> POINT <VectElem>
199                          |  THIS POINT <VectElem>
200
201  <ID_or_VectElem>        ::= <id> LB INTEGER_VALUE RB
202                          |  <id>
203
204  <VectElem>              ::= <id> LB <vector_operation> RB
205                          |  <id> LB <id> RB
206
207  <id>                    ::= ID
208
209  <vector_operation>      ::= <id> <operation> INTEGER_VALUE
210                          |  INTEGER_VALUE <operation> <id>
211
212  <operation>             ::= PLUS
213                          |  MINUS
214
215  <main_block>            ::= MAIN <main_attributions> END_MAIN
216                          |  MAIN END_MAIN
217
218  <main_attributions>     ::= <main_attribution>
219                          |  <main_attribution> <main_attributions>
220
221  <main_attribution>      ::= <element> ASSIGN <factor>
```

Código 1: BNF da linguagem de programação NOPL
Fonte: Ronszcka, 2019