

## ESTRUTURA E PROGRAMAÇÃO DO PARSEMAT (*PARSEME TOOLBOX*)

**autor:** Pauxy Gentil-Nunes

**e-mail:** [pauxy@uol.com.br](mailto:pauxy@uol.com.br)

**orientadora:** Prof<sup>a</sup>. Dr<sup>a</sup>. Carole Gubernikoff

### **A fatura na formação do compositor**

Fatura, na acepção do Houaiss, significa “*modo característico de cada autor compor sua obra de arte*”. Muitas vezes usa-se esta palavra para se referir à maneira como a partitura está organizada em termos de escrita, ou em termos de distribuição de elementos. Esta distribuição é um dos fatores que contribuem para a construção do que se chama de ‘textura’ (de acordo com Berry<sup>1</sup>). A palavra ‘fatura’ é, assim, restrita ao aspecto poético e distribucional da construção das texturas musicais.

A formação acadêmica do compositor inclui uma série de métodos de treinamento de fatura. Esses métodos são resultado de tradições, muitas vezes seculares e orais, e que tendem a se constituir em verdadeiros objetos de culto por parte de professores e de profissionais. O caráter sagrado se impõe pela delimitação, através de regras rígidas e algumas vezes justificadas apenas pela própria força da tradição, de sentidos de competência e de poder.

A solidez e a força destes métodos contrasta de forma berrante com a prática múltipla e ambígua de audição e produção musical contemporânea (contemporânea entendida aqui em sentido literal: práticas do mundo globalizado). O multiculturalismo exige dos ouvintes e dos produtores musicais um conhecimento mais extenso. Músicas de etnias e de gêneros diversos convivem em nosso espaço subjetivo. Diferentes funções são propostas para “A” música, ou seja, fenômenos diversos que, no entanto, recebem da academia sempre o mesmo nome.

Esse descompasso cria, para a universidade, um desafio adicional. Diante da convivência dos gêneros, o mundo acadêmico oscila entre abrigar um número cada vez maior de estudos sobre práticas específicas, tentando delimitar sua área de atuação de maneira artificial, ou seja, entender aquelas práticas como *corpora* fixos, rígidos no tempo, ou por outro lado admitir a interação complexa entre os gêneros e buscar uma

---

<sup>1</sup> Berry, Wallace. *Structural functions in music*. New York: Dover, 1987, p. 184-190

maneira de lidar com hiper-gêneros que se formam na medida em que as práticas inevitavelmente dialogam.

Um exemplo da tentativa de abrigar, em um mesmo ambiente, práticas diversas de organização da fatura (estratégias de composição) é a convivência da tradição do contraponto, dentro da tradição de Fux<sup>2</sup>, com as chamadas ‘técnicas mecânicas em bloco’ (como traduzido para o português por Guest<sup>3</sup>), que são constituídas dentro da escola de arranjo de jazz e pop de Berklee. Ambas as práticas referem-se ao mesmo assunto, ou seja, a questão distribucional dos ataques e durações<sup>4</sup>. No entanto, elas são vistas como assuntos tão diferentes e incompatíveis, que é comum (com exceção notória daqueles músicos que buscam um conhecimento mais amplo por conta própria) encontrar um compositor de música de concerto que não consiga ou não saiba trabalhar com blocos sonoros, e vice-versa, um arranjador de MPB, por exemplo, que não saiba construir uma polifonia. Não só as práticas excluem-se mutuamente como criam sistemas de valores, nos quais uma prática despreza a outra, baseadas unicamente na força da tradição e nos significados de vida que elas articulam.

O estudo das partições surge exatamente da observação daquilo que é comum a essas práticas, dentro de um espírito modelizador. Pretende-se, assim, que possa ser um espaço de convergência de técnicas e de construção de ferramentas de controle para o compositor que está imerso em um ambiente multicultural.

Há uma semelhança flagrante entre o espaço de fase particional<sup>5</sup> e a tabela de conjuntos de Forte<sup>6</sup>, no sentido de se constituírem ambos em catalogações exaustivas de um determinado universo conceitual. A teoria dos conjuntos também tem sido uma tentativa de superar antigas polarizações (como tão desgastada discussão entre o tonal e o atonal) através de um procedimento de síntese de práticas cuja divisão não fazia mais sentido no momento de sua formulação.

É com essa disposição que se propõe a construção de um programa de computador que perfaça as diversas tarefas de análise já definidas em trabalhos

---

<sup>2</sup> Fux, Johann Joseph & Mann, Alfred (ed.). *Gradus ad parnassum*. New York: Norton, 1725/1971.

<sup>3</sup> Guest, Ian. *Arranjo*. Rio de Janeiro: Lumiar, 1996, p. 69 e seguintes.

<sup>4</sup> Ver Gentil-Nunes, Pauxy. Parsemas e o método de Fux. In Barros e Barcellos (eds.). *Pesquisa e música*. Rio de Janeiro: CBM, 2006, p. 38.

<sup>5</sup> Ver Gentil-Nunes, Pauxy. Partições e música: uma pequena resenha. In *Anais dos Congressos da ANPPOM*, 2005.

<sup>6</sup> Forte, Allen. *The structure of atonal music*. New Haven: Yale University, 1973, p.179-181

anteriores sobre a teoria das partições<sup>7</sup>. Busca-se a agilização do processo de análise, visando a construção de taxonomias, passo imprescindível para o entendimento em larga escala de procedimentos composicionais já usados em uma determinada prática.

Um outro aspecto importante do programa é o explicitamento dos procedimentos de análise das partições. As linhas de programação das diversas funções do programa constituem uma documentação do próprio pensamento matemático e musical que gera os resultados analíticos e podem, assim, servir de referência para que outros pesquisadores possam entender, questionar, alterar, ou mesmo refutar os princípios em jogo.

Um terceiro objetivo é proporcionar motivação para o trabalho do compositor, como já foi realizado em algumas obras<sup>8</sup>, e assistência para a investigação, por parte do mesmo, de sua própria linguagem.

---

<sup>7</sup> Gentil-Nunes, Pauxy e Carvalho, Alexandre. *Densidade e linearidade na configuração de texturas musicais*. In Anais do 4º. Colóquio de Pesquisa do Programa de Pós-Graduação. Rio de Janeiro: UFRJ, 2004, p. 40-49.

<sup>8</sup> Alexandre Carvalho, *Rondó*, para quarteto de madeiras (2004); Paulo Dantas, *Aparência*, para percussão e piano (2004); e Pauxy Gentil-Nunes, *Ermo*, para flauta e difusão (2006).

## Programação do Parsemat

Um período extenso foi dedicado à escolha da ferramenta adequada à programação. Algumas alternativas foram abordadas, e um estudo realizado em cada linguagem, com mais ou menos sucesso. O C++ e o Java foram as primeiras tentativas, e apesar do poder de programação destas linguagens, a leitura de arquivos MIDI através delas se mostrou possível, porém complicada. Uma segunda tentativa foi feita com MAX/MSP, que tem uma leitura mais direta do arquivo MIDI, mas que (apesar de conter o módulo Jitter, que é uma ferramenta gráfica), não produz satisfatoriamente os gráficos que constituem o objetivo desejado.

Foi no MATLAB que encontramos a ferramenta ideal para a construção do *Parsemat*. Além de apresentar uma linguagem de fácil programação, ainda contava com uma ferramenta decisiva no processo, a *MIDI Toolbox* (Eerola & Toiviainen 2004), que se constitui em uma caixa de ferramentas para ler e manipular arquivos MIDI, de forma simples e produtiva. Várias de suas funções são baseadas em trabalhos recentes de importante pesquisadores em música (Lerdahl, Narmour, Krumhansl, Repp, entre outros).

A decisão de trabalhar com arquivos MIDI também foi um passo importante. O MIDI é uma linguagem desvalorizada no meio da pesquisa musical. É vista como uma alternativa pobre, que reduz o discurso musical a poucos elementos, e que despreza aspectos musicais tímbricos, como os próprios autores da *MIDI Toolbox* reconhecem<sup>9</sup>. No entanto, como a aplicação da teoria das partições estava focada neste momento nas DADs<sup>10</sup>, o MIDI se apresentou como o tipo de informação ideal para a construção do programa. Uma série de autores já se debruçou sobre a questão do MIDI como ferramenta válida de pesquisa musical e nas técnicas de superação de suas desvantagens.<sup>11</sup>

---

<sup>9</sup> Eerola, T. & Toiviainen, P. *MIDI toolbox: MATLAB tools for music research*. Jyväskylä: University of Jyväskylä, 2004, p. 6.

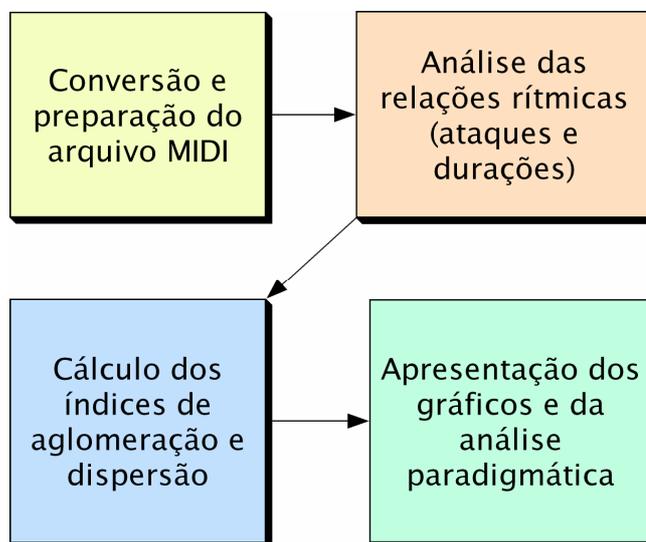
<sup>10</sup> Gentil-Nunes, Pauxy, op. cit. (2006), p. 38.

<sup>11</sup> Ver E. F. Clarke and N. Cook (Eds): *Empirical Musicology: Aims, Methods, Prospects*. Oxford: Oxford University Press, 2004, p. 77-102; Cook, N. (1995). "The conductor and the theorist: Furtwängler, Schenker and the first movement of Beethoven's Ninth Symphony". In J. Rink (ed.), *The Practice of Performance*. Cambridge: Cambridge University Press, p. 105-125; Repp, B. H. (1998). "A microcosm of musical expression: 1. Quantitative analysis of pianists timing in the initial measures of Chopin's Etude in E major". *Journal of the Acoustical Society of America* 104: 1085-1100; Bowen, J. A. (1996). "Tempo, duration and flexibility: Techniques in the analysis of performance". *Journal of Musicological Research* 16: 111-156; Martin, S. (2002). "The case of compensating rubato". *Journal of the Royal Musical Association* 127: 95-129.

## Estrutura do Parsemat

O Parsemat divide-se em quatro módulos (fig. 1):

- 1) Módulo de conversão e preparação do arquivo MIDI.
- 2) Módulo de análise rítmica das DADs.
- 3) Módulo de cálculo dos índices (  $a, d$  )<sup>12</sup>
- 4) Módulo de apresentação dos gráficos e da análise paradigmática.



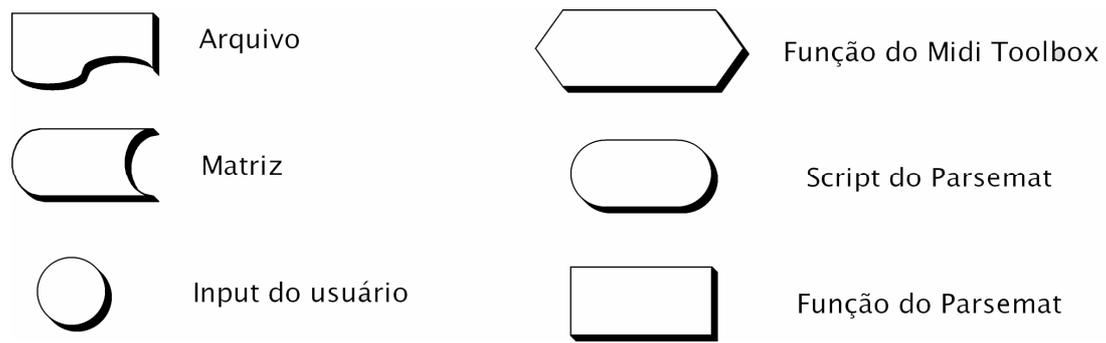
**Figura 1 - Parsemat: Módulos**

Apresentaremos em seguida cada módulo, de forma resumida, não sendo o objetivo, neste momento, dar uma visão detalhada de cada função. Deixaremos essa tarefa para um trabalho próximo, mais extenso.

Usaremos aqui a seguinte legenda (fig. 2):

---

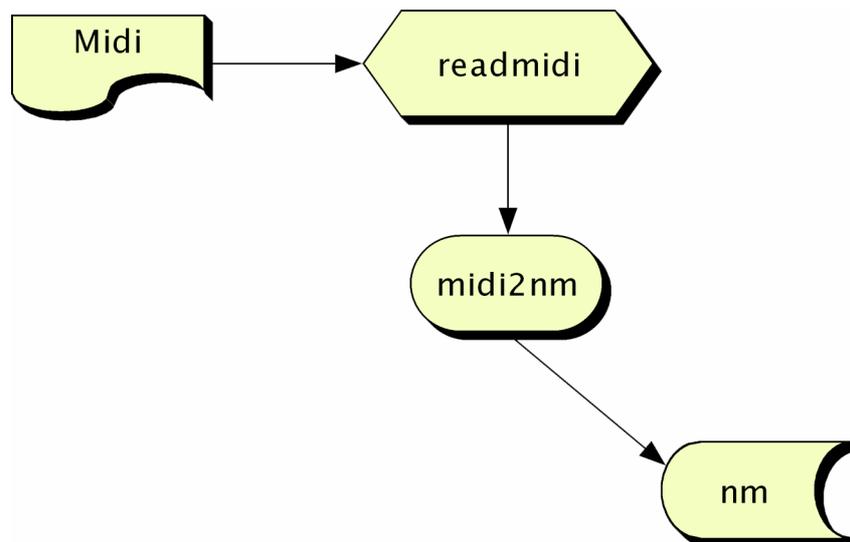
<sup>12</sup> Gentil-Nunes, Pauxy, op. cit. (2005)



**Figura 2 - Parsemat: legenda de modelização**

### Módulo de conversão

O módulo de conversão basicamente recebe o arquivo MIDI e o processa através da função *readmidi*, da *MIDI Toolbox*, para transformá-lo em uma matriz (que é o elemento básico de programação do MATLAB). O processo é controlado por um *script* do Parsemat Toolbox (*midi2nm*), que contém o input e a aplicação da função *readmidi* (fig. 3).



**Figura 3 - Parsemat: módulo de conversão**

### Módulo de análise das DADs (análise rítmica)

Este módulo define a aplicação da teoria das partições e pretende-se que ele possa ser eventualmente substituído por módulos diversos, em aplicações da teoria das partições em outros procedimentos da composição musical<sup>13</sup>. Neste caso específico, estamos analisando as DADs (disposição de ataques e durações), e portanto a análise é eminentemente rítmica<sup>14</sup>(fig. 4).

Três funções do *MIDI Toolbox* são usadas: *length*, *onset* e *dur*. Estas funções estão embutidas nas funções do *Parsemat Toolbox* *ataques*, *pfinal*, *parsemas* e *junções*. A função *parsemas* é, de longe, a mais complexa e importante, pois constitui a análise que define a aplicação em si. Basicamente, ela compara as diversas partes envolvidas no discurso musical, encontrando semelhanças e dessemelhanças rítmicas, agrupando-as em “componentes reais”<sup>15</sup> e montando uma tabela de componentes associados a cada ponto de tempo.

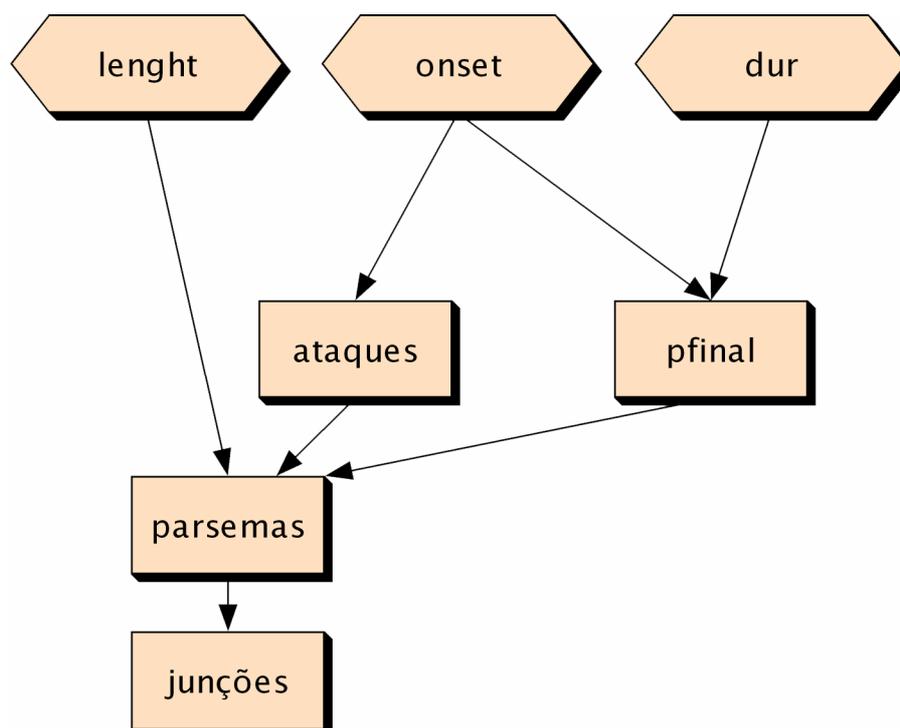


Figura 4 - Parsemat: Módulo de análise rítmica

<sup>13</sup> Aventados em Gentil-Nunes, Pauxy, id.

<sup>14</sup> De acordo com Berry (op. cit.); ver também Gentil-Nunes, Pauxy (op. cit., 2006).

<sup>15</sup> Berry, op. cit., p. 194-186

### Módulo de cálculo dos índices ( $a, d$ )

Consideramos que o módulo de cálculo seja o mais importante para o trabalho da teoria das partições, uma vez que ele constitui o que de fato se apresenta como algo novo na aplicação da teoria das partições à música<sup>16</sup>; ou seja, a apresentação da relação entre as relações de aglomeração e de dispersão, que permite relacionar as partições de forma significativa.

Este módulo é todo programado com funções autóctones e é uma representação das equações apresentadas no trabalho já citado<sup>17</sup>. Basicamente, ele recebe a tabela da análise rítmica e calcula, a cada ponto de tempo, e usando estas mesmas equações, as relações binárias totais, relativas a cada densidade-número; e as divide entre relações de aglomeração e de dispersão (fig. 5).

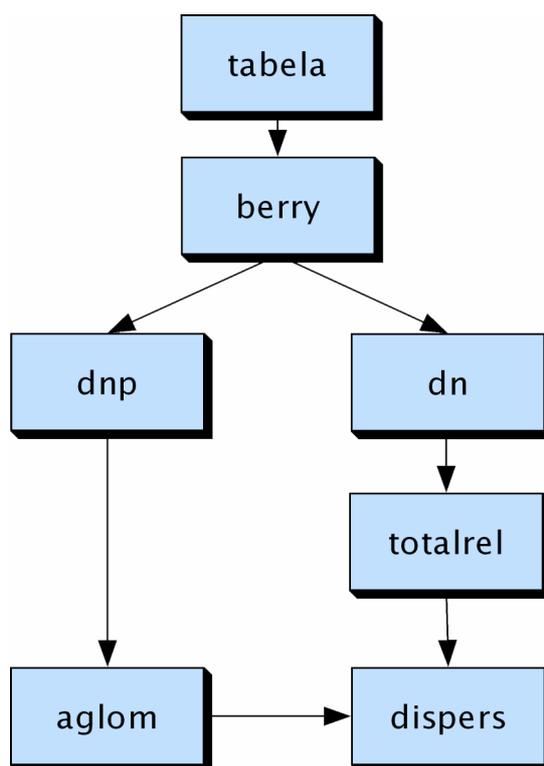


Figura 5 - Parsemat: módulo de cálculo

<sup>16</sup> ver Gentil-Nunes, Pauly e Carvalho, Alexandre, op. cit.

<sup>17</sup> Id.

## Módulo de apresentação

Espera-se a apresentação dos resultados em duas formas: em um gráfico dinâmico, onde os índices de aglomeração e dispersão estão plotados no tempo, e em um gráfico de fase, onde um inventário das partições usadas no trecho é apresentado, com suas relações explicitadas<sup>18</sup>.

Através da intervenção do usuário, define-se a opção entre um e outro ou pelos dois. Na configuração atual, isso é feito através de botões dentro da janela do MATLAB. Para o espaço de fase, foi feito um acréscimo opcional de apresentação de uma análise paradigmática dos resultados, para que se possa entender melhor a relação entre as partições utilizadas (fig. 6).

As funções envolvidas neste módulo também são autóctones e podem ser aproveitadas, caso se faça a aplicação da teoria das partições a outros parâmetros, com o cuidado de se aproveitar a informação de *pfinal* e *junções*.

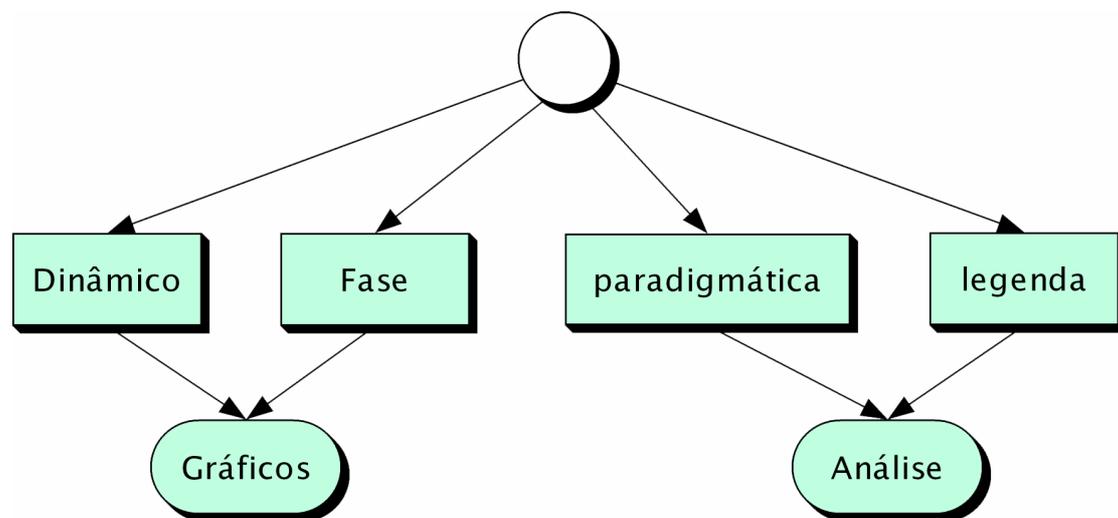


Figura 6 - Parsemat: módulo de apresentação

## Conclusão

---

<sup>18</sup> Id.

No momento, os módulos estão perfeitamente conectados (fig. 7) e o programa já funciona. É capaz de apresentar resultados interessantes, principalmente ao analisar peças de maior duração, para as quais o esforço analítico manual seria imenso.

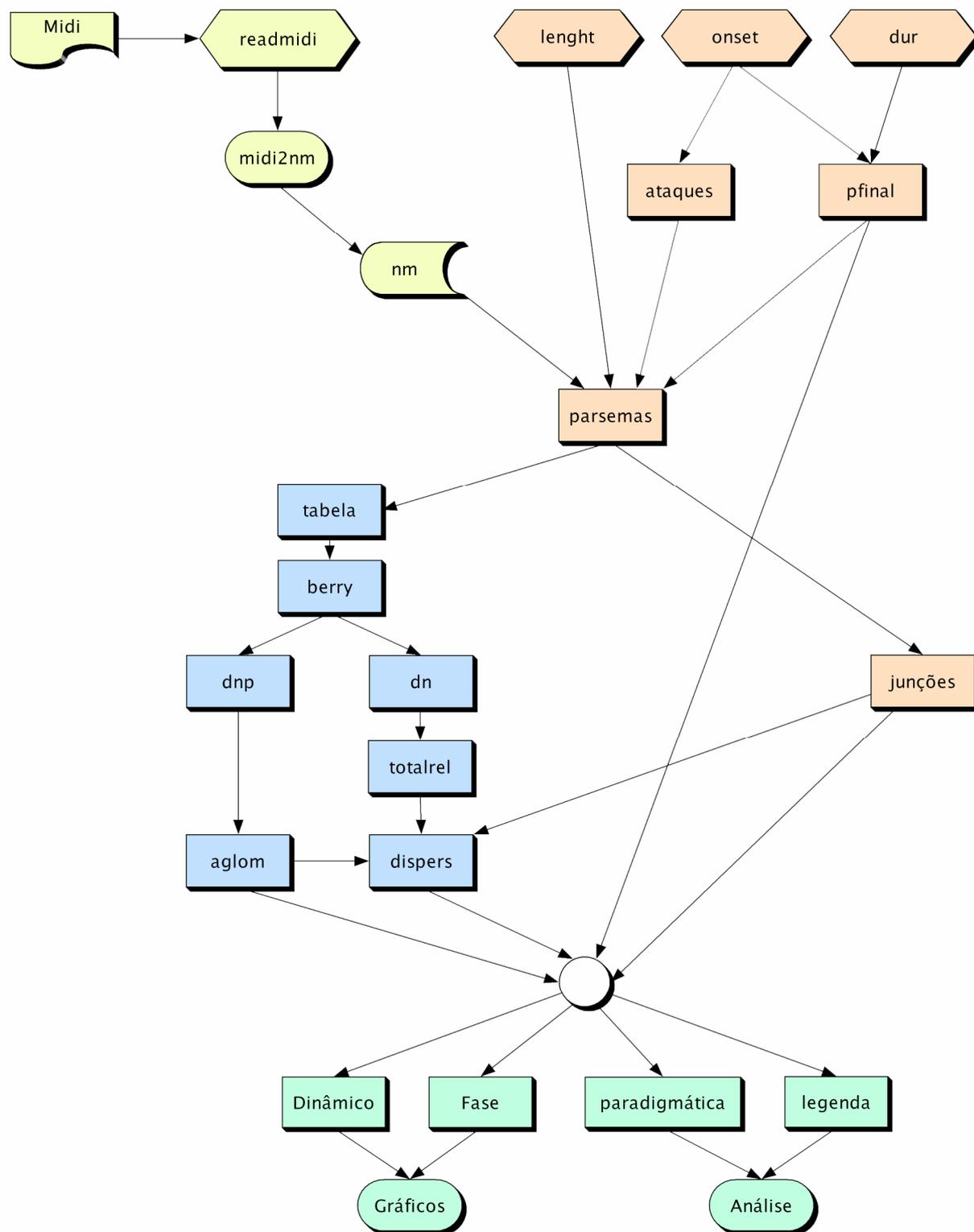


Figura 7 - Parsemat: Fluxograma

O planejamento atual em relação ao futuro do programa inclui os seguintes itens:

- 1) Aperfeiçoamento da apresentação da análise paradigmática, apresentação que no momento ainda é precária (uma questão puramente de interface, mas complicada);
- 2) Construção de um módulo de análise em tempo real de performance MIDI;
- 3) Construção de um módulo de produção, em que o computador possa gerar música fractal ou aleatoriamente a partir de um determinado esquema prévio de partições no espaço de fase;
- 4) Programação de um módulo de cálculo em MAX/MSP, com finalidades de processamentos diversos de performance;
- 5) Transporte (*deploy*) do programa para fora do ambiente MATLAB, como um aplicativo independente (Windows, MAC, Linux).