

# Desenvolvimento de Sistemas de Informação Geográfica Cooperativos seguindo o Padrão Modelo-Visão-Controle

Leonardo M. Couto<sup>1,2</sup>, Marco A. Casanova<sup>1</sup>, Daniel Schwabe<sup>1</sup>

<sup>1</sup>Departamento de Informática – PUC-Rio  
Rua Marquês de São Vicente, 225 – 22.453-900 – Rio de Janeiro – RJ – Brasil

<sup>2</sup>Tecnologia da Informação de E&P – Petrobras  
Av. República do Chile, 65 – 20031-912 – Rio de Janeiro – RJ – Brasil  
[matriciano@petrobras.com.br](mailto:matriciano@petrobras.com.br), {casanova, dschwabe}@inf.puc-rio.br

**Abstract.** *This paper introduces a strategy to develop geographic information systems, based on a variant of the Model-View-Controller (MVC) design pattern. The strategy adopts ontologies to model user classes and to capture additional characteristics of the information classes. The MVC components are modified to implement cooperativeness techniques, which interpret the ontologies. The paper also describes an example application designed according to the proposed strategy.*

**Resumo.** *Este trabalho apresenta uma estratégia para desenvolver sistemas de informação geográfica, seguindo uma variante do padrão de projeto Modelo-Visão-Controle (MVC). A estratégia adota ontologias para modelar classes de usuário e capturar características adicionais das classes de informação da aplicação. A variante do padrão MVC proposta altera os componentes para implementar técnicas de cooperatividade, interpretando as ontologias definidas. O trabalho descreve ainda um fragmento de uma aplicação projetada segundo a estratégia proposta.*

## 1. Introdução

Este trabalho apresenta uma estratégia para desenvolver sistemas de informação geográfica, seguindo uma modificação do padrão de projeto Modelo-Visão-Controle (MVC) (Krasner & Pope, 1998). A estratégia adota ontologias, transformadas em componentes ativos do sistema de informação (Guarino, 1998; Uschold & Gruninger, 1996; Fonseca & Egenhofer, 1999), para modelar classes de usuário e capturar características adicionais das classes de informação da aplicação. A variante do padrão MVC proposta altera os componentes para implementar técnicas de cooperatividade (Michelis et al. 1997), interpretando as ontologias definidas.

Uma aplicação projetada segundo a estratégia apresentará o seguinte comportamento. A aplicação apresenta interfaces para seleção de tarefas (consultas, relatórios, etc.) personalizadas, baseadas nos interesses do usuário, seus requisitos e papéis. Após o usuário escolher uma tarefa, a aplicação identifica a existência de alguma informação associada àquela que foi solicitada e que seja relevante ao usuário. Caso



O banco de dados contém os dados operacionais da aplicação, da forma usual.

A ontologia de uma aplicação modela características das classes de informação do banco de dados e das classes de usuários, que o Controle utiliza para tomar decisões sobre: (1) a forma de organizar os diálogos que a Visão apresentará ao usuário; (2) a forma de preparar as consultas que o Modelo executará contra o banco de dados.

Em particular, a ontologia modela um meta-relacionamento entre as classes de informação indicando o grau de proximidade semântica entre elas, para cada classe de usuários. O meta-relacionamento pode indicar um grau *forte* ou *fraco* de proximidade entre duas classes C e C', para cada classe U de usuários. Um grau forte indica que uma consulta Q sobre C deve ser transformada em uma consulta sobre C e C', sempre que um usuário da classe U executar Q. Um grau fraco indica que um menu envolvendo uma consulta Q sobre C deve também incluir uma consulta relacionada sobre C', para usuários da classe U. Estas transformações ficarão mais claras na discussão sobre o Controle.

O Modelo executa as consultas instanciadas pelo Controle, levando em consideração a escala atual do mapa, de forma a minimizar o volume de dados acessado no banco de dados.

A Visão é responsável por apresentar, na GUI, os menus de seleção de tarefas e de filtro das classes de informação, além de obter do SGO n o tipo de interface mais apropriada para representar as informações retornadas pela consulta ao banco de dados feita pelo Modelo, segundo a escala atual do mapa.

Por exemplo, em aplicações de petróleo, quando a Visão apresentar informações sobre poços de uma determinada área, dependendo da quantidade de poços recuperados e da escala atual do mapa, a Visão poderá exibir todo o conjunto de informações ou simplesmente exibir um símbolo que represente todo o conjunto recuperado, e fornecer ao usuário a opção de detalhar o conjunto.

O Controle desempenha as seguintes tarefas: identificação e classificação do elemento selecionado no *canvas*; recuperação, através do SGO n, das informações a respeito do conjunto de classes de informação da aplicação e do grau de relacionamento entre elas; o tipo de representação geográfica; a relação de ações que podem ser executadas sobre cada uma das classes de informação; o tipo de interface mais apropriada para representar a escolha destas ações.

O Controle obtém ainda a lista de consultas associadas a cada uma das classes de informação, juntamente com os seus parâmetros, seus tipos básicos, as *widgtes* associadas, o código SQL que será executado para o preenchimento destes parâmetros e os *templates* das consultas espaciais, que depois de instanciados serão executados pelo Modelo junto ao banco de dados. Estas informações são necessárias para que o Controle instancie dinamicamente as interfaces de seleção de tarefas e as de consulta de filtro, de acordo com as preferências do usuário.

Para executar suas funções, o Controle acessa tanto a ontologia quanto o banco de dados. O acesso ao banco é necessário pois as interfaces de filtro associadas às consultas das classes de informação, normalmente, possuem parâmetros com valores

tirados de um domínio enumerável, mantido no banco, por uma questão de desempenho e compatibilidade com os sistemas legados.

Por exemplo, o parâmetro *classificação*, que normalmente é utilizado para filtrar a classe de informação *Poços*, possui cinco instâncias: “especial”, “estratigráfico”, “extensão”, “injeção” e “jazida mais profunda”.

Os tipos de consultas de filtro de uma classe de informação e seus parâmetros devem sempre estar associados às preferências de um usuário ou de um grupo.

Além dos meta-relacionamentos entre as classes de informação, a ontologia indica quando uma classe de informação possui informação geográfica e os *templates* das consultas espaciais entre ela e as demais classes de informação. Estes serão instanciados pelo Controle segundo a política de recomendação do sistema para um usuário ou a um grupo.

Por exemplo, suponha que o usuário solicite dados de poços de uma instância da classe de informação de bacia. Suponha ainda que a classe de informação *Poços* possui forte relacionamento com a classe de informação *Blocos*, segundo o perfil deste usuário. O Controle instanciará então o *template* da consulta espacial entre poço e bloco para que, além dos poços, os blocos que possuem algum relacionamento topológico com os poços solicitados sejam também recuperados.

Os *templates* das consultas espaciais instanciados, juntamente com o código SQL contendo as cláusulas de restrições geradas pela escolha das instâncias dos parâmetros na consulta de filtro da classe de informação selecionada, determinam o código da consulta final que o Controle enviará ao Modelo.

### **3. Exemplo de uma aplicação desenvolvida segundo o padrão proposto**

Esta seção ilustra brevemente um fragmento de aplicação projetado segundo o padrão proposto (todas as figuras e tabelas entram-se ao final do texto).

A Tabela 1 descreve a seqüência de passos e a Figura 2 o diagrama de seqüência de uma tarefa genérica.

A Tabela 2 apresenta o pseudo-código de uma tarefa de preenchimento de instâncias de parâmetro de uma consulta de filtro de uma classe de informação. Já a Tabela 3 apresenta uma consulta espacial entre duas classes de informação. Ambas tem como exemplo um *template* do código SQL apropriado para cada tarefa.

A Figura 3 apresenta um exemplo de interface de consulta de filtro da classe de informação *Poço*, para profissionais de acompanhamento geológico. A Figura 4 mostra uma consulta que considera o grau de meta-relacionamento entre classes de informação de acordo com o perfil de uma classe de usuários. Neste caso, o usuário solicitou poços da Bacia de Campos e, através da consulta ao perfil da classe de usuários, o Controle identificou um meta-relacionamento forte entre as classe de informação *Poço* e *Bloco* para esta classe de usuário, e modificou a consulta para recuperar os poços que atendem à consulta, juntamente com os blocos que possuem algum relacionamento topológico com os poços já recuperados.

Por fim, a Figura 5 e a Figura 6 apresentam exemplos de menus gerados pelo Controle ao interpretar os meta-relacionamentos entre as classes de informação, em presença de modelos do usuário.

#### **4. Conclusões**

Este trabalho apresentou uma variante do padrão de projeto Modelo-Visão-Controle (MVC) para desenvolver sistemas de informação geográfica, adotando técnicas de cooperatividade (Michelis et al. 1997) governadas por ontologias.

A ontologia de uma aplicação modela características das classes de informação do banco de dados e das classes de usuários, que o Controle utiliza para tomar decisões sobre a forma de organizar os diálogos que a Visão apresentará ao usuário, e a forma de preparar as consultas que o Modelo executará contra o banco de dados.

O trabalho ilustrou ainda as principais características do modelo proposto através de um fragmento de um sistema real (Aquino, Matriciano e Nunes 2001), re-escrito de acordo com o novo modelo.

#### **Referências Selecionadas**

AQUINO, F.; MATRINO, L.; NUNES M. Gestão das Informações Sísmicas. Anais do VIII Simpósio de Geofísica da Petrobras, Vitória, 2001.

FONSECA, F.T.; EGENHOFER, M.G. Ontology-Driven Geographic Information Systems. In C. Bauzer Medeiros (ed.), Proceedings of the 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO. November 1999.

Guarino, N. Formal Ontology and Information System. In: FOIS 1998, Proceedings of FOIS'98, Trento, Italy, 6-8 jun. 1998. Amsterdam, IOS Press, p. 3-15. Disponível em: <<http://www.loa-cnr.it/Papers/FOIS98.pdf>>. Acesso em: 30 maio 2005.

KRASNER, G. E.; POPE, S. T. A cookbook for using the Model-View-Controller user interface paradigm in Smalltalk-80. Journal of Object-Oriented Programming, v. 1, n.3, p. 26-49, 1998.

MICHELIS, G. D.; MYLOPOULOS, J.; PAPAZOGLU, M. Cooperative Information Systems: A Manifesto. In. Academic Press, 1997. p. 28-37.

USCHOLD, M.; GRÜNINGER, M. Towards a Methodology for Building Ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95. 1995.

**Tabela 1: Descrição da tarefa.**

**Tarefa:** Visualizar todos os poços em perfuração dos blocos de produção de uma bacia sedimentar.

**Ator:** Geólogo da UN-BC.

**Perfil do Usuário:** Técnico da área de acompanhamento geológico, responsável pelas informações dos poços em perfuração de blocos de produção na bacia de Campos.

**Contexto de visualização atual:** O *canvas* já apresenta o mapa do Brasil, bem como todas as suas bacias sedimentares.

**Seqüência da navegação da tarefa:**

1. O usuário navega na aplicação e clica o *mouse* sobre o *canvas*, gerando o **Evento 1**.
2. O Controle obtém as coordenadas do mouse e executa uma função para classificar e identificar, se for o caso, o que foi selecionado.

*Ex.* Usuário seleciona a bacia de Campos

3. O Controle, de posse da classe de informação selecionada, executa uma consulta ao Gerenciador de Ontologias, para obter a lista das ações relacionadas com ela, bem como a interface de seleção apropriada, a ser apresentada.

*Ex.* Ações: Exibir, Ocultar, Visualizar Documentos, etc. Interface: menu *Drop-Down*.

4. O Controle notifica a Visão, repassando a interface de seleção composta pela lista das ações relacionadas com a classe de informação selecionada no Evento 1.
5. A Visão atualiza a GUI.
6. O usuário clica o *mouse* sobre a interface gerada no passo anterior, gerando o **Evento 2**.
7. O Controle obtém as coordenadas do mouse para classificar a opção selecionada e identificar a ação e a classe de informação sobre a qual a ação irá incidir.

*Ex.* Ação: Exibir. Classe de Informação: Poço.

8. O Controle executa uma consulta ao Gerenciador de Ontologias para obter a lista das classes de informação relacionadas com a classe de informação selecionada no Evento 2, os graus de relacionamentos, o tipo de representação geográfica das classes de informação, a lista de consultas de filtro da classe de informação selecionada, a interface de seleção apropriada para apresentar esta lista, bem como os *templates* das consultas espaciais entre a classe de informação selecionada e as classes de informação com forte relacionamento com ela, quando este relacionamento for topológico.

*Ex.* classes de informação relacionadas: Bloco (alto, polígono), Interpretação (baixo, polígono), Sísmica 2D (baixo, linha) e Sísmica 3D/4D (alto, polígono). Consultas de filtro: Acompanhamento Geológico, Laboratório, Perfuração e Produção. Interface: menu *Drop-Down*. *Templates*: Bacia + Poço, Poço + Bloco, Poço + Sísmica 3D/4D.

9. O Controle notifica a Visão, repassando a interface de seleção composta pela lista das consultas de filtro da classe de informação selecionada no Evento 2.
10. A Visão atualiza a GUI.
11. O usuário clica o *mouse* sobre a interface gerada no passo anterior, gerando o **Evento 3**.
12. O Controle obtém as coordenadas do mouse para classificar a opção selecionada e identificar a consulta de filtro a ser realizada.

*Ex.* Consulta: Acompanhamento Geológico.

13. O Controle executa uma consulta ao Gerenciador de Ontologias para obter os parâmetros da consulta, seus tipos básicos e *widgets* associadas, as consultas SQL para o seu preenchimento, no caso de parâmetros com valores variados.

*Ex.* Parâmetros: Classificação (caracter, lista, SQL<sub>1</sub>), Status (caracter, lista, SQL<sub>2</sub>).

14. O Controle executa as consultas SQL, para a obtenção das instâncias dos parâmetros, junto à

base de dados.

15. A base de dados retorna para o Controle os valores das instâncias associadas aos parâmetros da consulta selecionada para filtrar a classe de informação selecionada.

*Ex.* Instâncias dos parâmetros Classificação e Status.

16. O Controle gera a interface de filtro da classe de informação e notifica a Visão da mudança de estado.
17. A Visão atualiza a GUI com a interface de filtro da classe de informação selecionada no Evento 2.
18. O usuário seleciona, ou não, parâmetros na interface de filtro da classe de informação, gerando o **Evento 4**.

*Ex.* Usuário seleciona o parâmetro: “status”, com o valor: “em perfuração”.

19. O Controle obtém os valores dos parâmetros selecionados para filtrar a classe de informação.
20. O Controle obtém do Gerenciador de Ontologias os valores *default* dos parâmetros de filtro das classes de informação que também serão apresentadas com a classe de informação selecionada no Evento 2.

*Ex.* Classe de Informação: Bloco. Tipo: Produção. Status: em atividade. Versão Geométrica: atual.

21. O Controle de posse dos parâmetros selecionados juntamente com os *templates* das consultas espaciais SQL obtidos no passo 8, instancia as consultas enviando-as para o Modelo juntamente com a escala do mapa.
22. O Modelo executa as consultas SQL junto à base de dados.
23. A base de dados retorna o conjunto de coordenadas de mundo que representam as classes de informação que possuem um forte relacionamento com a classe de informação selecionada no Evento 2 e que possam ser representados na escala atual do mapa. Notificando a Visão da sua mudança de estado.
24. A Visão executa uma função para transformar as coordenadas de mundo, das classes de informação mencionadas no passo anterior, em coordenadas de tela. De posse destas informações e da escala atual do mapa, a Visão executa uma consulta ao Gerenciador de Ontologias para obter o tipo de interface apropriada para apresentá-la.

*Ex.* Interface: todas as coordenadas.

25. A Visão atualiza a GUI e possibilita que as informações provenientes dos relacionamentos fracos com a classe de informação selecionada no Evento 2, possam ser selecionadas pelo usuário, caso assim ele desejar.

**Tabela 2: Pseudo-código de obtenção de instâncias de um parâmetro.**

```
/*  
  Obtém as instâncias de um Parâmetro de uma consulta de filtro de uma classe de  
  informação.  
*/  
Begin {}  
  Obtém a conexão com o Banco de Dados  
  Begin{ executa consulta }  
    Monta a consulta  
    Ex. SELECT DISTINCT NOME_PARAMETRO_DE_FILTRO_classe de informação  
          FROM   NOME_TABELA_classe de informação  
          WHERE  NOME_PARAMETRO_DE_FILTRO_classe de informação IS NOT NULL  
    Executa a consulta  
    Preenche estrutura auxiliar com os resultados obtidos  
    Ordena o resultado  
    Devolve as instâncias do Parâmetro  
  End { executa consulta }  
  Testa erros de SQL  
  Fecha conexão com o Banco de Dados  
End {}
```

**Tabela 3: Pseudo-código de uma consulta espacial.**

```
/*  
  Obtém as instâncias da classe de informação Y que atendam as restrições selecionadas e  
  possua alguma relação topológica com o classe de informação X.  
*/  
Begin {ID do classe de informação X, Nome do Código do classe de informação X,  
        Nome da Tabela X, Nome da Tabela Y, restrições do classe de informação Y}  
  
  Obtém a conexão com o Banco de Dados  
Begin { executa consulta espacial }  
  Monta a consulta espacial  
  Ex. SELECT /*+ ORDERED */ y.CODIGO_classe de informação_Y  
        FROM NOME_TABELA_X x,  
             NOME_TABELA_Y y  
        WHERE x.CODIGO_classe de informação_X = ID_X  
        AND mdsys.sdo_relate(y.gdo_geometry,  
                             x.gdo_geometry, 'mask=ANYINTERACT  
                             querytype=window') = 'TRUE'  
  
  Monta as cláusulas de filtro da classe de informação Y e acrescenta à consulta espacial  
  Executa a consulta completa (espacial + relacional)  
  Preenche estrutura auxiliar com os resultados obtidos  
  Devolve as instâncias da classe de informação Y que atenderam todas as condições  
End { executa consulta espacial }  
  
  Testa erros de SQL  
  Fecha conexão com o Banco de Dados  
End {}
```

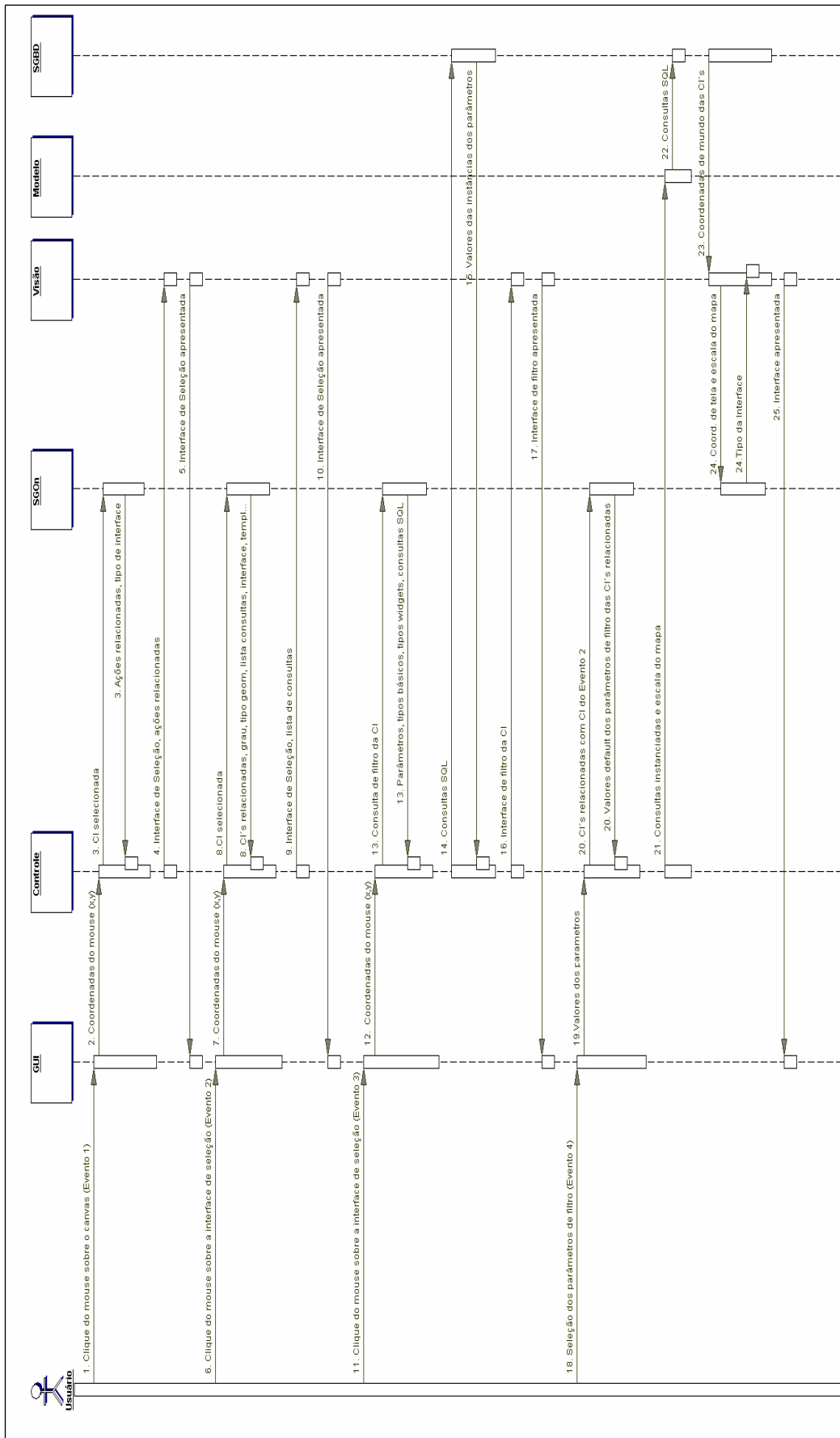


Figura 2: Diagrama de Seqüência de uma tarefa genérica.

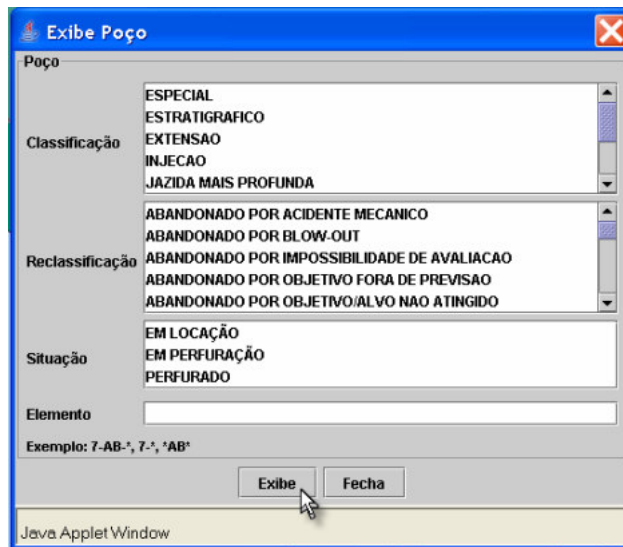


Figura 3: Exemplo de interface de filtro de uma classe de informação (Poço).

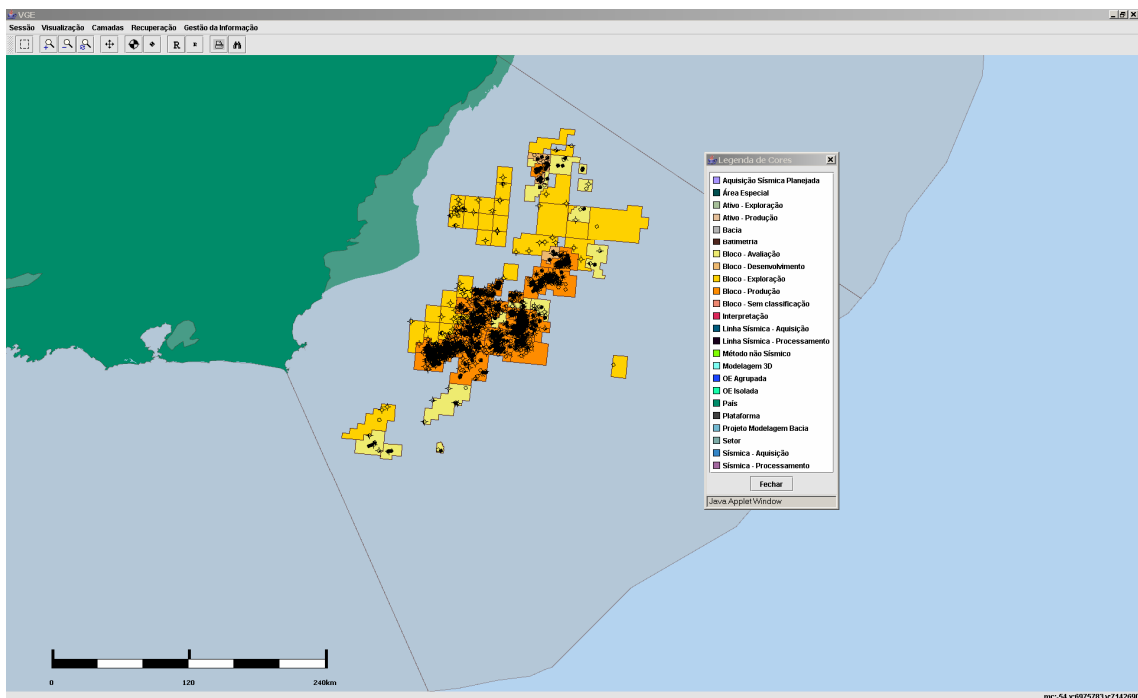


Figura 4: Todos os Blocos com Poços perfurados na Bacia de Campos.

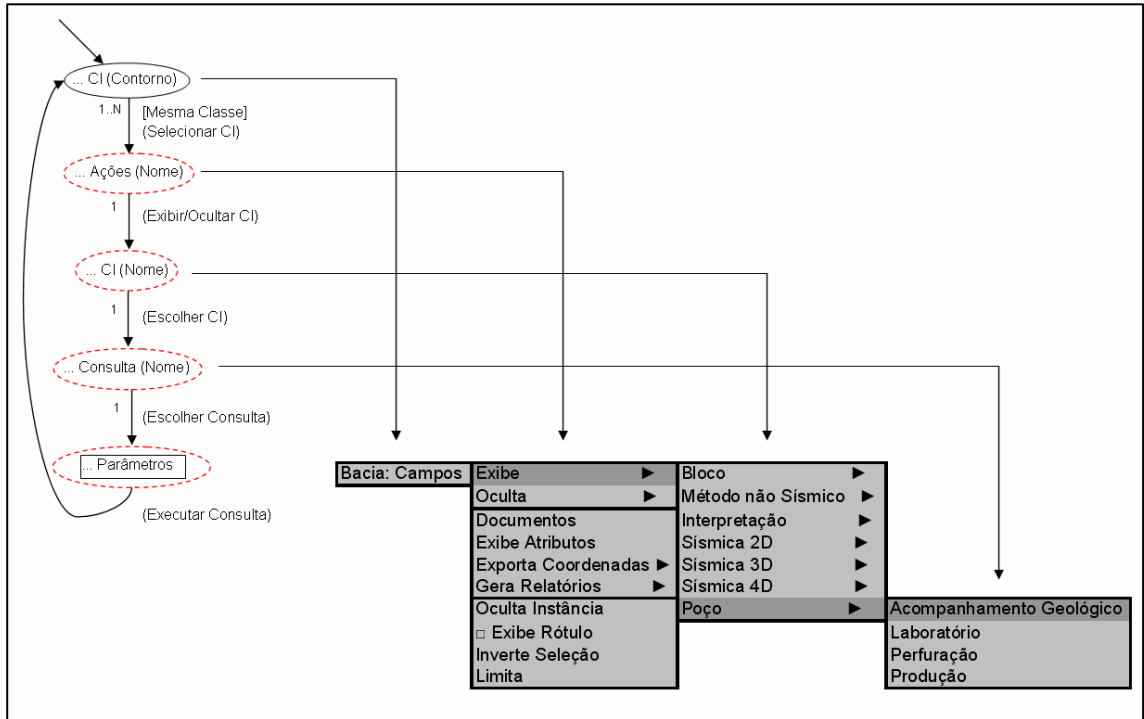


Figura 5: Exemplo da navegação pelos menus dinâmicos.

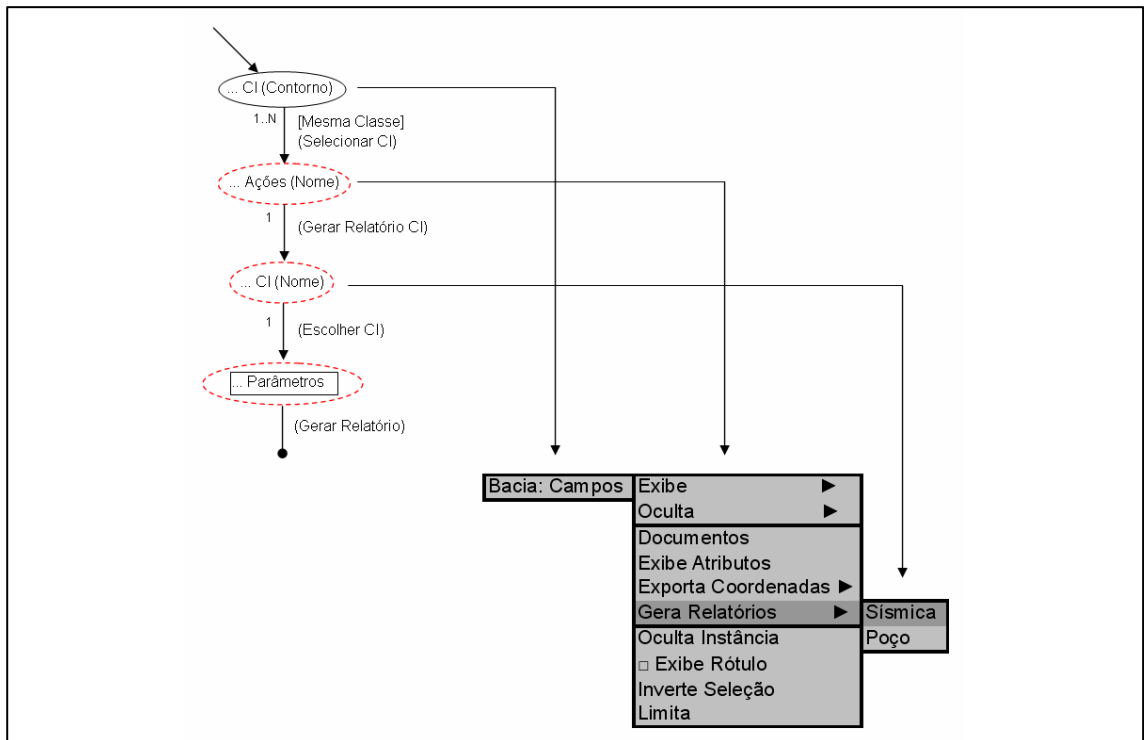


Figura 6: Exemplo de navegação.