

3

Arquitetura MVC baseada em modelos

3.1.

Visão geral

Na arquitetura proposta os componentes de Controle e Visão, da arquitetura tradicional do padrão de projetos MVC, incorporam inteligência semântica. Esta inteligência é representada pelo conjunto de informações dos modelos de domínio, interface e usuário persistido no banco de dados, conforme Figura 4.

A incorporação desta inteligência semântica aos componentes de Controle e Visão amplia as suas funcionalidades, visto que, além das funções básicas, o Controle passa a deter o poder de personalização e adaptação junto ao sistema. Além disso, a Visão pode apresentar as informações segundo uma semântica.

Esta semântica é representada pela informação de *threshold* contida no modelo de interface. A Visão consulta esta informação para saber qual a forma de exibir as informações recuperadas da base de dados.

De acordo com as características das informações, assumimos que as distintas bases de dados das classes de informações, se referem a termos com conceituação diferenciada, portanto, não havendo necessidade de resolução de problemas de interoperabilidade semântica.

As instâncias associadas a conceitos dos modelos utilizados estão armazenadas em banco de dados. Esta decisão torna a arquitetura mais versátil. Já que permite que sistemas legados usufruam das informações contidas nestes bancos sem que para isto seja necessária a associação com nenhum novo modelo.

Nas seções posteriores cada um dos elementos que compõe a arquitetura será detalhado.

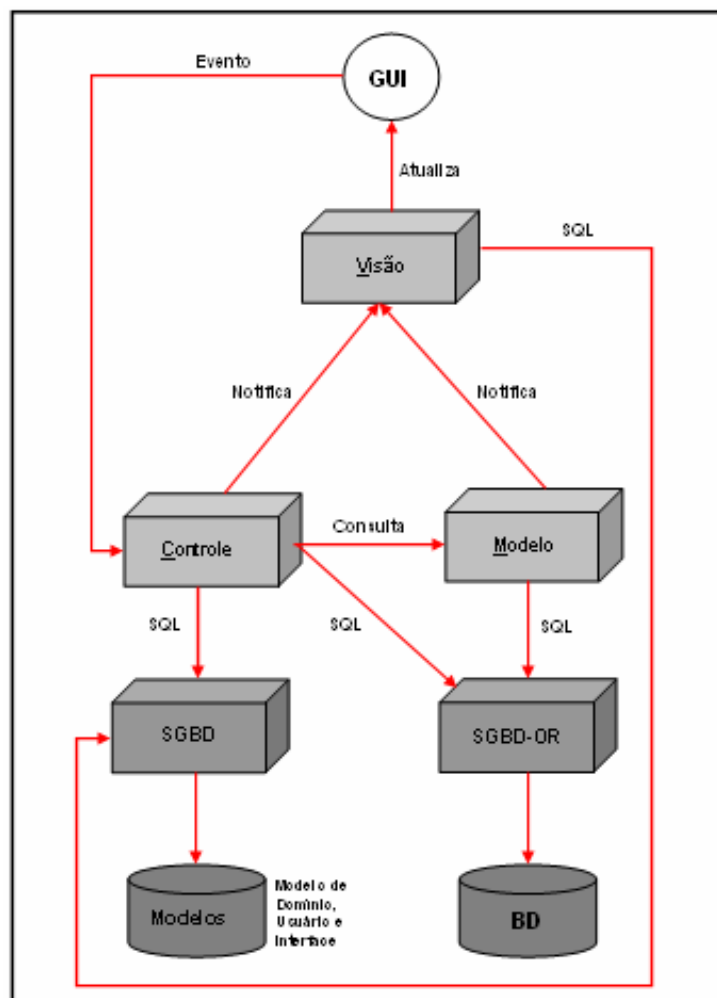


Figura 4: Arquitetura MVC baseada em modelos

3.2. Modelos

A utilização dos modelos descritos na arquitetura ocorrerá em tempo de execução, pois eles serão interpretados durante a execução do sistema, de forma a dispormos de um sistema de informação baseado em modelos. Desta maneira, se as regras de negócio forem modificadas, não serão necessárias alterações dos códigos, somente dos modelos específicos. Assim, se novos conceitos forem adicionados aos modelos, o sistema deverá adaptar-se a eles, sem necessidade de alteração do mesmo. Nas seções posteriores cada um dos modelos que compõe a arquitetura será detalhado.

3.2.1.

Modelo de domínio

Em uma aplicação baseado em modelos, o modelo de domínio é usado como uma espinha dorsal conceitual, estruturando a informação do domínio fornecida na aplicação (Stojanovic et al., 2003).

O modelo de domínio, baseado na arquitetura proposta, representa características das classes de informação do banco de dados e das classes de usuários. O Controle as utilizará 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, o modelo de domínio representa o 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, e pode até mesmo não existir. Um grau forte indica que uma consulta Q sobre C deve ser transformada também em uma consulta sobre C e C', sempre que um usuário da classe U executar Q. Neste caso o sistema irá executar as consultas Q sobre C e C sobre C', disponibilizando o resultado final para o usuário. Um grau fraco indica, por exemplo, que um menu envolvendo uma consulta Q sobre C deve também incluir uma consulta relacionada sobre C', para usuários da classe U. Neste caso o sistema irá executar a consulta Q sobre C e oferecerá, por exemplo, um menu onde o usuário poderá optar por executar a consulta Q sobre C'. O sistema não oferecerá nada quando não houver proximidade semântica entre as classes de informação. Estas transformações ficarão mais claras no exemplo dado na seção 4.3.3.

3.2.2.

Modelo de interface

O modelo de interface fornecerá ao sistema as informações necessárias sobre os tipos de interfaces que deverão ser utilizadas ao longo da aplicação, como por exemplo, nas situações de seleção de opções e na representação das informações obtidas nas consultas tanto das bases de dados, quanto do modelo de domínio.

Nele também serão obtidas informações a respeito dos elementos que compõem as interfaces, como por exemplo, as *widgets* associadas aos atributos utilizados nas interfaces de filtro de uma classe de informação (CI), bem como a sua ordem de apresentação, dentre outras.

3.2.3.

Modelo de usuário

O modelo do usuário pode ser interpretado como sendo o modelo do domínio visto por um particular usuário, como se fosse uma visão do banco de dados. Sua função é representar diversas características do usuário como conhecimento, preferências, interesses, tarefas e objetivos (Kuruc, 2005).

Nos bancos de dados existem inúmeras relações e atributos referentes às CI's modeladas no contexto de um determinado domínio, sendo que, na maioria das vezes, as consultas feitas nos sistemas de informações tradicionais, isto é, sistemas não baseados em um modelo de usuário, retornam para a maior parte dos seus usuários respostas que não lhe são relevantes ou que possuem mais informações do que ele realmente necessita para realizar a sua tarefa.

Devido a isto, a modelagem dos conceitos no modelo do domínio deve levar em consideração os interesses dos grupos de usuários existentes. Desta forma, respostas mais apropriadas às consultas submetidas ao sistema poderão ser fornecidas.

3.3.

Descrição da arquitetura MVC baseada em modelos

3.3.1.

Modelo

O Modelo não é afetado pelo uso dos modelos, visto que o seu papel dentro da arquitetura MVC é o de gerenciamento do comportamento e dos dados do domínio da aplicação, respondendo as solicitações por informações sobre seu estado (geralmente pela Visão) e respondendo as instruções para mudança de estado (geralmente pelo Controle).

O uso de *caches* dinâmicos e o emprego da técnica de multi-resolução, são os diferenciais adicionados a este componente na arquitetura proposta. O primeiro

para o armazenamento dos identificadores das instâncias das CI's recuperadas da base de dados. O segundo para representação de algumas instâncias das CI's do domínio compostas de linhas poligonais. Na seção 4.3.1 as funcionalidades deste componente serão detalhadas.

3.3.2.

Visão

A Visão é responsável pela funcionalidade de apresentação na GUI das interfaces de seleção da aplicação, das interfaces de filtro das CI's e dos desenhos das instâncias da CI recuperadas pelo Modelo. Estes desenhos podem ser as coordenadas das instâncias ou um símbolo que as represente.

O uso dos modelos pela Visão é necessário nesta última função, pois ela necessita antes obter, do modelo de interface, o valor do *threshold*. Este valor representa a semântica que a Visão utiliza para determinar qual o tipo de ação mais apropriada para representar as informações recuperadas, em conjunto com a escala atual do mapa e o perfil do usuário. Na seção 4.3.2 as funcionalidades deste componente serão mais detalhadas.

3.3.3.

Controle

O Controle é o elemento principal de toda a arquitetura proposta, executando o maior conjunto de tarefas. Cabe a ele fazer a interface com todo o conjunto de componentes da arquitetura, desde o *canvas* até o SGBD-OR.

Dentre as tarefas a serem desempenhadas pelo Controle estão a identificação e classificação do elemento selecionado no *canvas*. Cabe também a ele a recuperação de várias informações dos modelos. Estas dizem respeito ao conjunto de CI's a que o sistema dará suporte e o grau de relacionamento semântico entre elas, se forte ou fraco, dependendo do perfil do usuário, bem como o tipo de representação geométrica da CI, se ponto, linha ou poligonal e a relação de ações que podem ser executadas sobre cada uma das CI's, além de gerar a interface para representar a escolha destas ações. O Controle obtém também a lista de consultas de filtro associadas a cada uma das CI's, juntamente com os seus atributos, sua ordem de aparição na interface e as *widgtes* associadas. Além disso, este

componente recupera o código SQL que ele próprio executa para o preenchimento dos atributos das consultas de filtro. Como também obtém os *templates* das consultas espaciais entre as CI's, que depois de instanciados são executados pelo Modelo junto à base de dados. Na seção 4.3.3 as funcionalidades deste componente serão mais detalhadas.

Através de pseudocódigos, apresentados abaixo nas tabelas 1 e 2, são representadas algumas tarefas desempenhadas pelo Controle. A primeira delas é a de preenchimento de instâncias de um atributo de uma consulta de filtro de uma CI e a segunda uma consulta espacial entre duas CI's. Ambas utilizam *templates* de código SQL para esta exemplificação.

```

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

```

Tabela 1: Pseudocódigo de obtenção de instâncias de um atributo

```

/*
    Obtém as instâncias da CI Y que atendam as restrições selecionadas e
    possuam alguma relação topológica com a(s) instância(s) da CI X.
*/
Begin {ID da CI X, Nome do Código da CI X, Nome da Tabela X,
      Nome da Tabela Y, restrições da CI Y}
    Obtém a conexão com o Banco de Dados
    Begin { executa consulta espacial }
        Monta a consulta espacial

        Ex. SELECT /*+ ORDERED */ y.CODIGO_CI_Y
            FROM  NOME_TABELA_X x,
                  NOME_TABELA_Y y
            WHERE x.CODIGO_CI_X = ID_X
            AND mdsys.sdo_relate(x.gdo_geometry,
                                y.gdo_geometry, 'mask=ANYINTERACT
                                querytype=window') = 'TRUE'

        Monta as cláusulas de filtro da CI Y e acrescenta à consulta espacial
        Executa a consulta completa (espacial + relacional)
        Preenche estrutura auxiliar com os resultados obtidos
        Devolve as instâncias da CI Y que atenderam todas as condições
    End { executa consulta espacial }
    Testa erros de SQL
    Fecha conexão com o Banco de Dados
End {}

```

Tabela 2: Pseudocódigo de uma consulta espacial

3.4.

Exemplos de tarefas utilizando a arquitetura proposta

De forma a exemplificar algumas idéias propostas neste trabalho, será apresentada uma seqüência de figuras representando telas de seleção de atributos de filtro, seleção de ações e a exibição de instâncias em escalas diferentes, de classes de informação. Na última figura, aproveitaremos para destacar alguns elementos que formam o *canvas*.

Na Figura 5 é apresentado um exemplo de interface de filtro de uma CI, no caso Bloco. Na parte superior desta interface aparece uma lista de tipos de consulta que podem ser escolhidas. A existência de mais de um tipo de consulta está relacionada diretamente com o perfil do usuário. Dependendo do perfil, uma

ou mais consultas podem ser disponibilizadas. Para cada CI sempre haverá um tipo de consulta de filtro padrão.

Na Figura 6 é apresentado um exemplo de interface de seleção de ações associadas a uma CI, no caso Bacia. A relação de ações e CI's associadas a algumas destas ações, também estão relacionadas diretamente com o perfil do usuário. Dependendo da CI, o conjunto de ações e CI relacionadas podem variar. Pois nem todas as CI's possuem relacionamento semântico com todas as outras CI's do domínio, além disso algumas ações só fazem sentido para alguma delas.

Na Figura 7 é apresentado um exemplo de interface de exibição do resultado de uma consulta. No caso, são instâncias da CI de Sísmica 2D de aquisição que possuem algum relacionamento topológico (vide seção 1.1) com a Bacia de Campos. Na realidade este resultado foi obtido após serem realizados dois tipos de consulta, uma relacional e outra espacial. Primeiramente os atributos escolhidos pelo usuário na interface de filtro da CI de Sísmica 2D de aquisição tiveram que ser atendidos na consulta relacional. Depois disso é que o sistema executará a consulta espacial entre a poligonal que representa a instância da Bacia de Campos contra todas as linhas que representam as instâncias do resultado obtido anteriormente.

Na Figura 8 é apresentado o mesmo resultado que foi exibido na figura anterior só que em uma escala menor para termos mais detalhes, utilizando técnicas de multi-resolução para algumas das CI's exibidas.

Na Figura 9 é apresentado um exemplo de interface da exibição de instâncias de várias CI's, no caso Bacia, Bloco, Sísmica 2D de aquisição e Poço, resultado de várias consultas entre elas. Esta relação de CI's que tiveram suas instâncias apresentadas faz parte de um domínio associado com o perfil do usuário. Somente as CI's relevantes para a realização das tarefas do usuário são disponibilizadas. Além disso, alguns elementos que aparecem no *canvas* são destacados como: a barra de menu principal, a régua de escala, a legenda de cores das CI's e a área de ajuda rápida para o usuário, onde do lado direito aparecem o meridiano central e as coordenadas do *mouse* e do lado esquerdo o tipo da CI selecionada e o valor da sua instância.

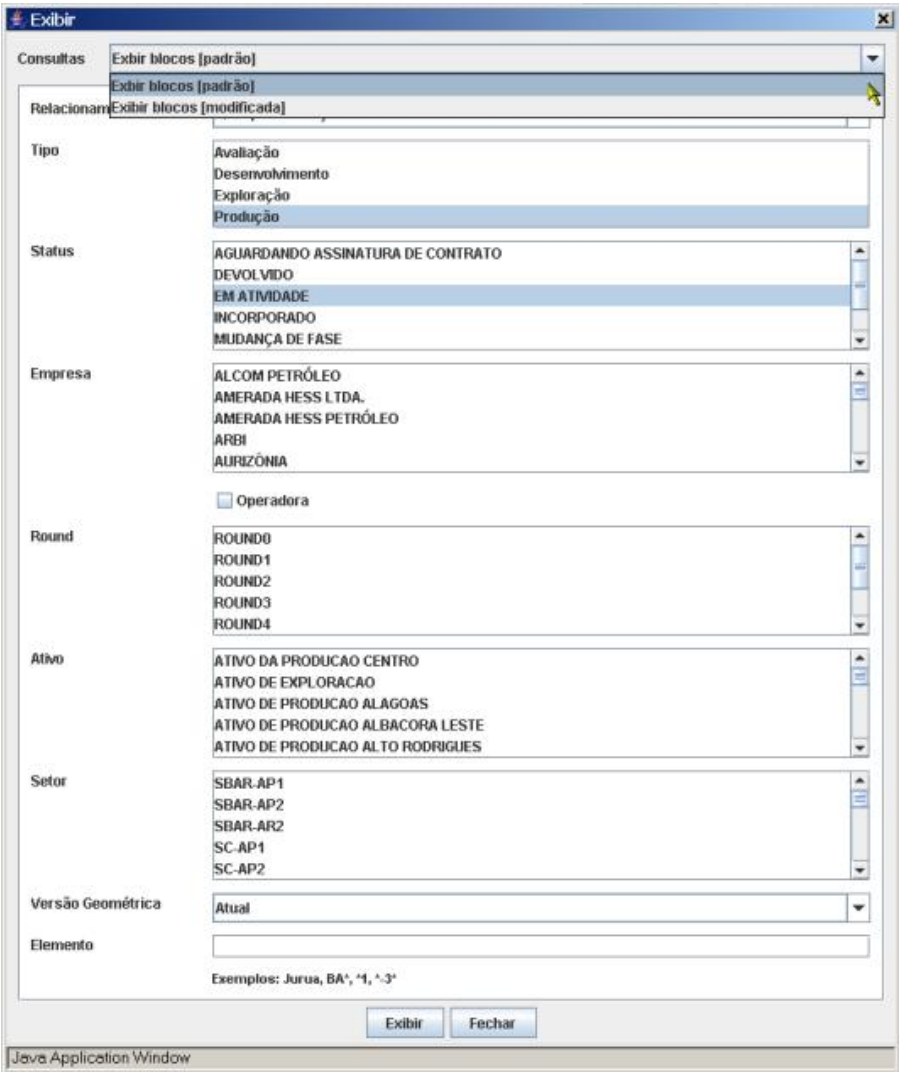


Figura 5: Interface de filtro de uma classe de informação (Bloco)

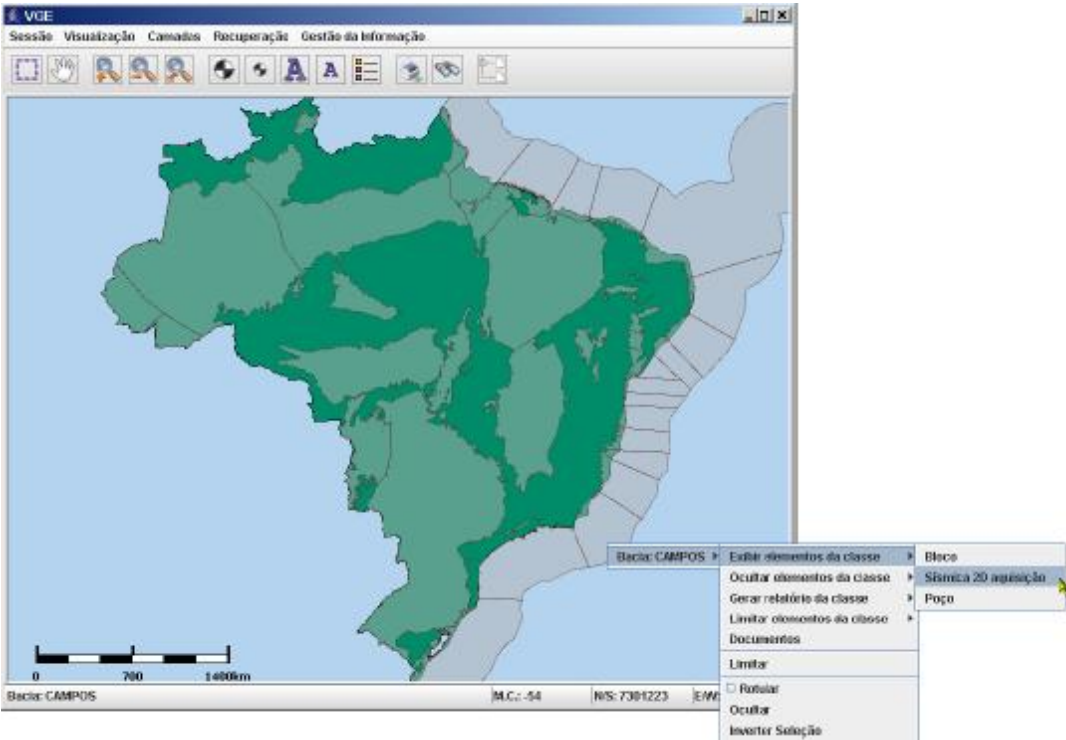


Figura 6: Menu de ações de uma CI associada ao perfil do usuário

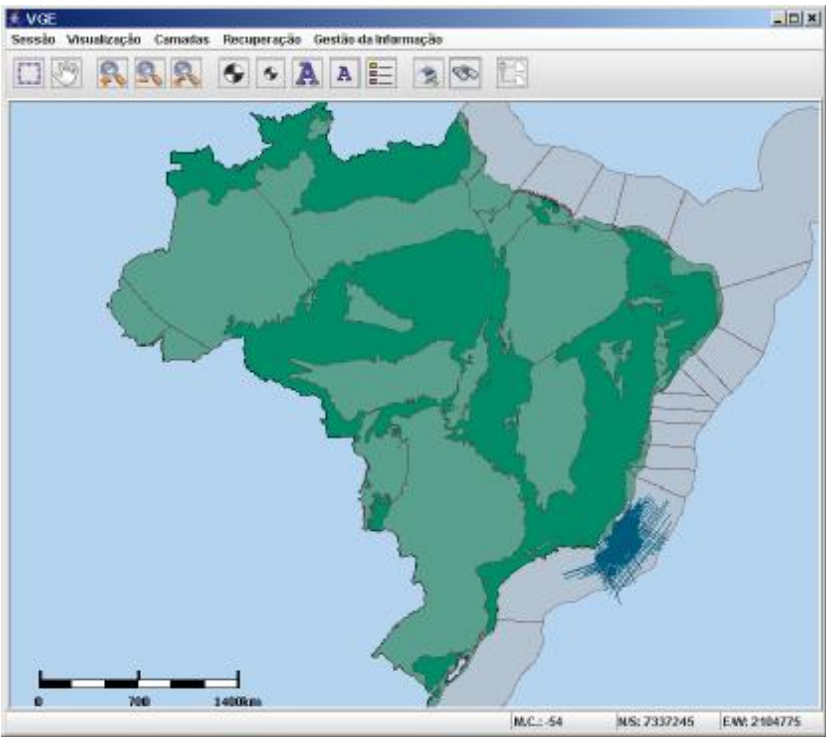


Figura 7: Sísmicas 2D de aquisição da Bacia de Campos na escala 1:15.000.000

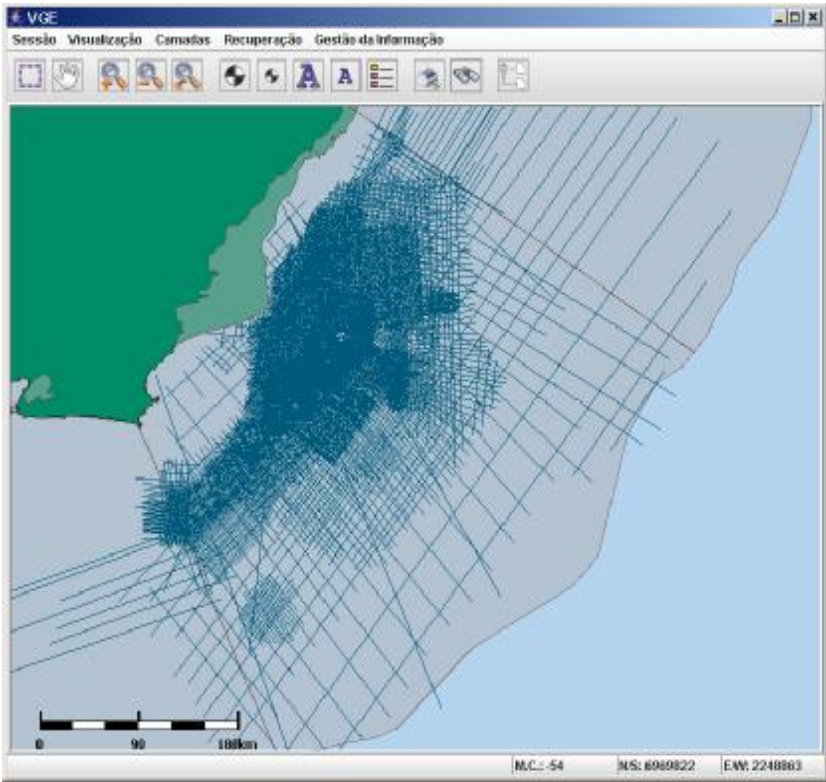


Figura 8: Sísmicas 2D de aquisição da Bacia de Campos na escala 1:2.000.000

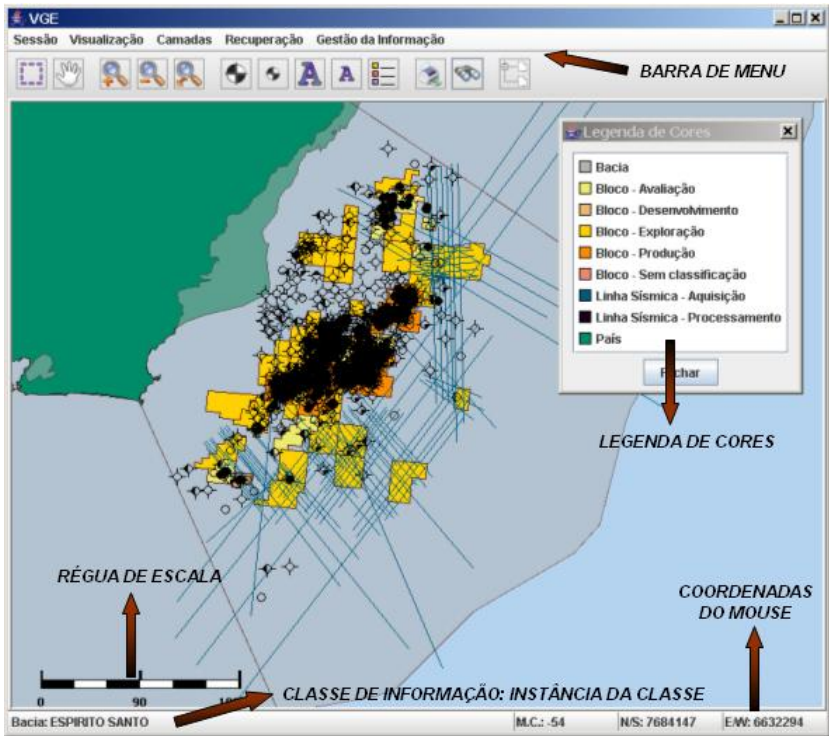


Figura 9: Instâncias de CI's associadas ao perfil do usuário