

## UM GERENCIADOR PARA ATRIBUTOS ESPACIAIS DE OBJETOS GEOGRÁFICOS

Casanova, M.A. (1); Hemerly, A.S. (1); Duarte, M.P.C. (1); Mediano, M.R. (1)  
Câmara, G. (2); Freitas, U.M. (2)

### RESUMO

Uma arquitetura para sistemas de gerência de bancos de dados geográficos é inicialmente delineada. A arquitetura introduz subsistemas especializados para cada tipo de atributo - convencional, espacial e pictorial - e torna natural a ligação de um banco de dados geográfico com um banco de dados convencional, entre outras características. Em seguida, a arquitetura do subsistema responsável pelos atributos espaciais é discutida em mais detalhe, enfatizando o problema de processamento de consultas espaciais. A arquitetura proposta para este subsistema inclui um esquema de *buffers* em dois níveis que otimiza o processamento das operações espaciais.

### ABSTRACT

An architecture for geographic database management systems is first delineated. The architecture introduces subsystems specialized for each type of attribute - conventional, spatial and pictorial - and facilitates connecting a geographic database with a conventional database, among other features. Then, an architecture for the subsystem responsible for the treatment of spatial attributes is discussed in more detail, emphasizing the problem of processing spatial queries. The architecture proposed for this subsystem includes a dual buffer schema that optimizes the processing of spatial operations.

---

(1) Centro Científico Rio - IBM Brasil

(2) Divisão de Processamento de Imagens - INPE

## 1. INTRODUÇÃO

Neste trabalho inicialmente propomos uma arquitetura para sistemas de gerência de bancos de dados geográficos que, além de facilitar a implementação, oferece novas oportunidades de otimização e torna natural a ligação entre um banco de dados geográfico com um banco de dados convencional. Em seguida, apresentamos uma arquitetura para o subsistema responsável pelo armazenamento e processamento dos atributos espaciais. Em ambos os casos, a ênfase será sobre processamento de consultas espaciais.

As arquiteturas propostas permitem diversas formas de implementação e têm como objetivo a sua possível utilização no SPRING [2], um sistema para processamento de informações georeferenciadas, desenvolvido pela Divisão de Processamento de Imagens do INPE.

A ênfase em processamento de consultas espaciais sobre bancos de dados geográficos justifica-se por dois motivos. Por um lado, este problema é substancialmente mais difícil do que o caso convencional pois a capacidade de filtragem dos métodos de acesso espacial é comparativamente mais restrita e o custo computacional dos operadores espaciais, muito mais elevado. Por outro lado, o processamento é facilitado pelo fato dos usuários tipicamente trabalharem sobre regiões geográficas delimitadas, mas não necessariamente predefinidas, e sobre temas com uma forte correlação, como hidrografia e desflorestamento em um banco de dados geográfico sobre a região amazônica.

Este trabalho está dividido da seguinte forma. A seção 2 apresenta os conceitos básicos sobre bancos de dados geográficos e sobre processamento de consultas espaciais. As seções 3 e 4 descrevem as arquiteturas propostas para sistemas de gerência de bancos de dados geográficos e para um subsistema responsável pelos atributos espaciais. Finalmente, a seção 5 contém as conclusões.

## 2. PRELIMINARES

### 2.1 Bancos de Dados Geográficos

*Bancos de dados geográficos (BDGs)*, os repositórios de dados de sistemas de informação geográfica, são caracterizados por:

- um grande número de objetos, provenientes de fontes diferentes tais como mapas, imagens obtidas por satélites, fotografias aéreas, medições de campo, etc...
- objetos cujas geometrias podem ser codificadas das mais diversas formas e que freqüentemente ocupam muito espaço de armazenamento;
- relacionamentos complexos entre os objetos; e
- evolução no tempo [1].

Ooi [9] classifica os atributos de um objeto armazenado em um BDG em três tipos. Os *atributos convencionais* descrevem as propriedades convencionais do objeto, tais como o nome e população de uma cidade. Sua definição utiliza um dos tipos alfanuméricos usuais.

Já os *atributos espaciais* contêm uma descrição geo-referenciada da geometria dos objetos, tal como a fronteira política de uma cidade. São definidos sobre um dos *tipos de dados espaciais geo-referenciados* especiais, equipados com *operadores espaciais*, e que usualmente são construídos a partir dos tipos primitivos 'ponto', 'linha' e 'polígono', com a restrição de que todas as coordenadas devem ser com respeito a um sistema de coordenadas geográficas. Os valores dos atributos espaciais de um objeto serão chamados de *geometrias* do objeto.

Finalmente, *atributos pictóricos* contêm uma descrição pictórica do objeto, tal como uma foto aérea de uma cidade. O seu tipo é um dos *tipos de dados de varredura geo-referenciada*, que são usualmente construídos a partir do tipo 'varredura' (ou 'pixmap'), com a restrição de que todas as células da varredura estão associadas a uma coordenada em um sistema de coordenadas geográficas. Os valores dos atributos pictoriais de um objeto serão chamados de *imagens* do objeto.

## 2.2 Processamento de Consultas Espaciais

Linguagens de consulta para BDGs (ver, por exemplo, [5,9,10,13]) naturalmente são mais complexas do que aquelas projetadas para bancos de dados convencionais pois incluem a definição de operadores para os três tipos de atributos mencionados na seção 2.1. Além deste ponto, devem oferecer facilidades para guiar a apresentação gráfica dos resultados das consultas.

No resto deste trabalho, ignoraremos aspectos temporais e a questão de apresentação gráfica. Além disto, daremos maior ênfase ao processamento dos atributos espaciais, já que o tratamento dos atributos convencionais está ao alcance da tecnologia dos sistemas de bancos de dados tradicionais, e a utilização de atributos pictoriais em consultas é muito limitada.

O processamento de consultas sobre BDGs segue essencialmente os mesmos passos que o caso convencional. Porém, o processamento de tais consultas é substancialmente mais difícil do que o caso convencional pois os operadores espaciais possuem uma semântica muitas vezes complexa e a representação das geometrias dos objetos geográficos é usualmente bastante longa. Assim, a capacidade de filtragem dos métodos de acesso espacial é comparativamente mais limitada [9] e o custo computacional dos operadores espaciais é bem mais elevado. Em conjunto, isto significa recuperar mais objetos do que o necessário, na maior parte dos casos, e executar um pós-processamento pesado para determinar quais objetos realmente satisfazem a qualificação da consulta. Por exemplo, enquanto que operações de comparação entre cadeias de caracteres podem ser implementadas quase diretamente por vários métodos de acesso, operações topológicas sobre polígonos (interseção, etc...) só o são em casos muito simples, como retângulos. De fato, no caso de BDGs, métodos de acesso espacial freqüentemente trabalham apenas sobre aproximações para polígonos irregulares com muitos pontos, como "bounding boxes" [9].

Podemos, no entanto, identificar certas peculiaridades na utilização de BDGs que facilitam o processamento de consultas, se bem exploradas. Em primeiro lugar, os usuários de BDGs tipicamente começam uma sessão de trabalho especificando uma região geográfica de interesse, não necessariamente predefinida, para em seguida seleccionar e manipular objetos geográficos dentro da região. Esta suposição

chamaremos de *correlacionamento geográfico*. Além disto, ao formularem suas consultas, os usuários utilizam certas classes de objetos freqüentemente em conjunto, como é o caso de rios e desflorestamento em um BDG sobre a região amazônica. Designaremos esta suposição por *correlacionamento temático*.

### 3. UMA ARQUITETURA PARA SGBDGS

#### 3.1 Arquitetura

Resumidamente, um *sistema de gerência de bancos de dados geográficos (SGBDG)* possui três funções básicas: armazenamento de objetos geográficos; processamento de consultas geográficas; e gerência de transações.

O projeto do subsistema de armazenamento deve levar em conta que os valores de atributos espaciais e pictóricos podem ser longos mas, entretanto, os objetos são relativamente estáticos e, para certas aplicações, os objetos exibem alto correlacionamento. As características do processamento de consultas em bancos de dados geográficos já foram comentadas na seção 2.2. Já a gerência de transações requer transações longas e, em muitos casos, suporte a versões. De fato, a situação em bancos de dados geográficos é muito semelhante à de aplicações de CAD, o que significa que as soluções dadas à gerência de transações nesta última área podem ser adaptadas à primeira.

Neste trabalho propomos uma arquitetura genérica, mostrada na Figura 1 no final do trabalho, para SGBDGS. Brevemente, a arquitetura decompõe o SGBDG em um *gerenciador global de objetos, GGO*, que coordena as atividades de três gerenciadores especializados, um para cada tipo de atributos: o *gerenciador de atributos convencionais, GAC*, o *gerenciador de atributos espaciais, GAE*, e o *gerenciador de atributos pictoriais, GAP*. No que se segue, trataremos apenas do problema de processamento de consultas.

Dada uma consulta sobre o banco de dados geográfico, o papel do GGO consiste em criar um plano de execução para a consulta e coordenar a sua execução. Cada plano consiste da execução, em paralelo ou seqüencialmente, de: subconsultas

convencionais pelo GAC; subconsultas espaciais pelo GAE; e subconsultas pictoriais pelo GAP; e combinações de resultados intermediários pelo GGO.

Para armazenamento das geometrias dos objetos, o GAE deverá utilizar uma estrutura de dados tal que a correlação temática e geográfica induza contigüidade física em memória secundária, condição essencial para se obter um bom desempenho em sistemas de bancos de dados. O projeto do GAE deverá atingir este objetivo de forma transparente para o usuário, ou seja, o usuário não deve ser forçado a definir de antemão áreas de trabalho para guiar o armazenamento físico das geometrias. Assim, a organização conceitual mantém-se independente da organização física, outra característica importante de uma boa arquitetura de sistemas de bancos de dados.

Discutiremos uma arquitetura para o GAE em mais detalhe na seção 4.

### **3.2 Alternativas de Implementação para o SGBDG**

Esta seção apresenta uma avaliação preliminar de quatro alternativas de implementação para o SGBDG sobre sistemas de gerência de bancos de dados (SGBDs) convencionais, ou sistemas de armazenamento já disponíveis.

#### *Implementação sobre SGBD com Campos Longos*

A implementação mais simples utiliza um SGBD convencional apenas com suporte para campos longos. Neste caso, todos os três gerenciadores do SGBDG podem ser implementados como aplicações do SGBD. No caso do GAE, o SGBD ficará a cargo apenas do armazenamento das geometrias e das estruturas necessárias à implementação dos métodos de acesso espacial. A implementação do GAP segue uma estratégia semelhante, exceto que as imagens poderão ser armazenadas externamente ao SGBD, dado o seu tamanho, agindo o SGBD quase que apenas como um catálogo.

Esta alternativa exige que o GGO implemente um processador de consultas completo, incluindo analisador sintático, otimizador e interpretador do plano de execução. A construção deste componente pode ser simplificada escolhendo-se como linguagem de consulta do SGBDG uma extensão da linguagem aceita pelo SGBD e

supondo-se que uma consulta só poderá ter subconsultas aninhadas, se estas forem homogêneas.

Neste contexto, dizemos que uma subconsulta é *homogênea* se ela for sobre atributos do mesmo tipo - convencionais, espaciais ou pictoriais - dos quais herda o tipo. Assim, o otimizador poderá ser trivializado de tal forma que, dada uma consulta submetida pelo usuário, sempre gere um plano de execução para a consulta consistindo da execução inicial de subconsultas homogêneas independentes entre si, seguida da combinação dos resultados intermediários destas subconsultas. A suposição acima garante que é possível criar as subconsultas necessárias de forma simples. Cada subconsulta será otimizada e processada em separado pelo gerenciador apropriado: as subconsultas convencionais pelo próprio SGBD; as espaciais pelo GAE; e as pictoriais pelo GAP.

Em particular, note que, se a consulta submetida pelo usuário especificar uma área geográfica dentro da qual deverão estar todos os objetos de interesse, todas as subconsultas espaciais geradas para o seu processamento serão sobre tal região, mas não necessariamente sobre os mesmos tipos de objetos. Portanto, o GAE poderá otimizá-las processando em sequência todas as cláusulas de todas as subconsultas espaciais sobre os mesmos tipos de objetos. Desta forma, o GAE minimiza o acesso às páginas de memória secundária contendo os objetos de um dado tipo.

Esta implementação exige ainda que se tome cuidado com a duplicação dos mecanismos de *buffers*, situação que ocorrerá caso o SGBD mantenha os dados em *buffers* antes de transferi-los para o GAE ou para o GAP, que por sua vez também mantém *buffers* para os dados sob seu controle. Esta duplicação é particularmente crítica pois geometrias e imagens são estruturas longas.

No caso de uma arquitetura centralizada, deve-se escolher então um SGBD que passe para o programa de aplicação um ponteiro para o conteúdo de campos longos residindo nos seus *buffers*, em lugar do valor do próprio campo. No caso de uma arquitetura cliente-servidor, o GAE (ou GAP) residirá no cliente. Portanto, é interessante manter *buffers* do lado do cliente, geridos pelo GAE (ou GAP), e *buffers* do lado do servidor, como parte do SGBD. A questão de duplicação de *buffers* em arquiteturas cliente-servidor é discutida em detalhe em [4].

### *Implementação sobre SGBD com Campos Longos e TADs*

A segunda alternativa baseia-se em um SGBD convencional com suporte para campos longos e tipos abstratos de dados (TADs). Neste caso, a implementação dos tipos de dados espaciais (ou pictoriais), embora do ponto de vista funcional continue a fazer parte do GAE (ou GAP), será feita através dos mecanismos para criação de TADs oferecidos pelo SGBD.

A vantagem desta alternativa está em que o GGO pode ser completamente eliminado pois todo o processamento de consultas ficará a cargo do SGBD, inclusive o processo de montagem dos resultados intermediários. Porém, o otimizador de consultas do SGBD continuará a desconhecer a existência dos métodos de acesso particulares, implementados pelo GAE (ou GAP).

No caso específico do GAE, uma estrutura especial de armazenamento para as geometrias dos objetos continuará a existir para reduzir o acesso à memória secundária. Isto significa dizer que a implementação dos tipos de dados espaciais como TADs do SGBD fará uso das técnicas especiais de armazenamento e acesso implementadas pelo GAE. Além disto, no caso de uma arquitetura cliente-servidor, é imperativo que o SGBD permita a implementação de TADs do lado do cliente, onde reside o GAE, o que não é regra unânime em SGBDs relacionais.

### *Implementação sobre SGBD Estendível*

A adoção de um SGBD estendível [6,8,11,12] permite implementar toda a funcionalidade do SGBDG como extensões do SGBD, incluindo todos os tipos de dados espaciais e pictoriais, o processo de otimização e execução de consultas, e os métodos de acesso e gerência de *buffers* peculiares ao SGBDG. As restrições a este enfoque são a complexidade do processo de criar extensões profundas como estas e o fato do código tornar-se totalmente dependente do SGBD estendível escolhido.

### *Implementação sobre Subsistemas de Armazenamento*

Esta alternativa, a de mais difícil implementação, parte de um ou mais subsistemas de armazenamento que ofereçam objetos longos e serviços de controle de concorrência e de integridade física (*recovery*) (como por exemplo o WiSS [3]).

Exige portanto uma implementação de um processador de consultas completo, a nível do GGO e de cada subsistema de armazenamento.

#### 4. PROPOSTA DE UM GERENCIADOR DE ATRIBUTOS ESPACIAIS

Nesta seção, discutiremos uma arquitetura para um gerenciador de atributos espaciais (GAE) que adota uma estratégia cliente-servidor e baseia-se em um SGBD dotado apenas de campos longos. Supomos que a linguagem de consulta espacial seja declarativa de alto nível.

Em linhas gerais, a arquitetura do GAE tenta minimizar o custo computacional dos operadores espaciais fazendo com que sejam implementados por rotinas acessando estruturas de dados em memória principal. A arquitetura decompõe o GAE nos seguintes componentes (ver figura 2 no final do trabalho):

- CA catálogo
- AS analisador sintático
- OT otimizador de consultas espaciais
- EX executor de planos de acesso
- MA métodos de acesso espacial
- BG *buffer* de geometrias
- OE operadores espaciais
- BP *buffer* de páginas
- GA gerente de armazenamento

O catálogo armazena a descrição de cada banco de dados geográfico e não é propriamente específico do GAE. O analisador sintático complementa o trabalho do GGO, verificando a correção de uma consulta espacial contra o esquema do banco de dados corrente e traduz a consulta da forma externa para uma forma interna. O otimizador cria um *plano de acesso* otimizado para a consulta, utilizando estatísticas sobre o banco mantidas no catálogo. Por sua vez, o executor de planos de acesso, como o nome indica, controla a execução do plano e constrói o resultado da consulta.

Um plano de acesso incluirá *operações de varreduras* de conjuntos de geometrias, possivelmente através de métodos de acesso espacial, *operações espaciais*, e

*operações auxiliares* para gerência de resultados intermediários e para criação do resultado final. Uma operação de varredura consiste no acesso seqüencial a um conjunto de geometrias de acordo com uma dada ordem (por exemplo, de norte para sul e de leste para oeste; ou de acordo com a ordem em que as geometrias estão armazenados). Uma varredura poderá especificar que as geometrias do conjunto poderão ser reutilizadas. Uma varredura poderá ainda ser executada em *pipeline* com operações espaciais ou, ainda, aninhadamente com outras operações de varredura.

Os métodos de acesso espacial indexam as geometrias mantidas em memória secundária pelo gerente de armazenamento e implementam certos tipos de varredura. O conjunto de operadores espaciais, como o nome indica, é um conjunto de rotinas implementando os operadores espaciais oferecidos pela linguagem de consulta. Estas rotinas operam sobre listas de geometrias materializadas pelas varreduras dos geometrias e mantidas no *buffer* de geometrias.

O gerente de armazenamento e o *buffer* de páginas são os componentes mais simples de implementar diretamente sobre um SGBD convencional (que supomos adotar uma arquitetura cliente-servidor). O *buffer de páginas* mantém em memória principal as páginas contendo as geometrias, ou parte delas, dos objetos requisitados e as páginas contendo as estruturas auxiliares compondo os métodos de acesso espacial necessárias às varreduras. O *buffer* de páginas é implementado por dois processos, um executado no servidor e outro no cliente. Por fim, o gerente de armazenamento mantém em memória secundária as geometrias dos objetos armazenados no banco, bem como as estruturas de dados dos métodos de acesso espaciais.

Completaremos esta seção com alguns comentários sobre os problemas a serem resolvidos na implementação de um GAE seguindo esta arquitetura. Inicialmente, a implementação do otimizador exige a definição de métodos para estimar os custos das operações espaciais e de heurísticas para simplificar a criação de planos de execução otimizados. Exige ainda a criação de sub-planos padronizados para controlar a execução das operações espaciais. A implementação do executor de planos de acesso depende naturalmente da definição precisa das interfaces com os métodos de acesso espacial e com o *buffer* de geometrias. O elenco de métodos de acesso espacial a ser oferecido deverá se basear em um estudo detalhado dos tipos de geometrias a serem oferecidos e dos tipos de consultas mais frequentes que se

estima o sistema irá processar. Por fim, a escolha da estrutura do *buffer* de geometrias depende fundamentalmente da própria estratégia de implementação das operações espaciais.

## 5. CONCLUSÕES

Neste trabalho delineamos inicialmente uma arquitetura que decompõe um sistema de gerência de objetos geográficos em subsistemas especializados para tratamento de cada tipo de atributo exibido pelos objetos geográficos. Esta arquitetura facilita a implementação do sistema, oferece novas oportunidades de otimização e torna natural a ligação de um banco de dados geográfico com um banco de dados convencional (através do gerenciador de atributos convencionais). Por outro lado, o gerenciador global deverá ser cuidadosamente implementado para garantir o processamento otimizado de consultas e a atomicidade das transações, e para evitar duplicação de funções com os gerenciadores especializados.

Em seguida, apresentamos uma arquitetura para o gerenciador de atributos espaciais, no que tange processamento de consultas. A arquitetura proposta, além dos componentes tradicionais, inclui um esquema duplo de *buffers*: o *buffer de geometrias* materializa as geometrias necessárias ao processamento das operações espaciais; já o *buffer de páginas* contém imagens de páginas físicas das quais são extraídas as geometrias. Este esquema permite otimizar a implementação das operações espaciais e, conseqüentemente, o processamento de consultas espaciais.

## REFERÊNCIAS

- [1] Barrera, R.; Frank, A.; Al-Taha, K. Temporal relations in geographic information systems: a workshop at the University of Maine. In: SIGMOD RECORD 20(3), p.85-91 (1991).
- [2] Câmara, G.; Souza, R.C.; Freitas, U.M.; Casanova, M.A. SPRING: Processamento de Imagens e Dados Georeferenciados. In: Anais do SIBGRAPH V, Águas de Lindóia, SP p.233-242 (1992).

- [3] Chou, H.T.; DeWitt, D.J.; Katz, R.H.; Klug, A.C. Design and Implementation of the Wisconsin Storage System. In: *Software - Practice and Experience* 15(10), p.942-962 (1985).
- [4] DeWitt, D.J.; Maier, D.; Fattersack, P.; Velez, F. A study of three alternative workstation-server architectures for object oriented database systems. In: *Proc. 16th International Conference on Very Large Data Base, Brisbane, Australia*, p.107-121 (1990).
- [5] Egenhofer, M.J. *Spatial Query Languages*. PhD thesis, University of Maine, Orono, ME, USA (1989).
- [6] Gutting, R.H. Gral: An extensible relational database system for geometric applications. In: *Proc. 15th International Conference on Very Large Data Bases, Amsterdam, The Netherlands*, p.33-44 (1989).
- [8] Haas, L; Cody, W.F. Exploiting extensible DBMS in integrated geographic information systems. In: *Advances in Spatial Databases*, O. Gunther and H.J. Schek (eds), *Lecture Notes in Computer Science*, Vol. 525, Springer-Verlag, Berlin (1991).
- [9] Ooi, B.C. *Efficient Query Processing in Geographic Information Systems*, *Lecture Notes in Computer Science*, Vol. 471, Springer-Verlag, Berlin (1990).
- [10] Roussopoulos, N.; Faloutsos, R.; Sellis, T. An efficient pictorial database system for PSQL. In: *IEEE Transaction and Software Engineering* 14(5), P.639-650 (1988).
- [11] Scholl, M.; Voisard, A. Object-oriented database systems for geographic applications: an experiment with O2. In: *Building an Object-Oriented Database System: the Story of O2*, F. Bancilhon, C. Delobel e P. Kanellakis (eds), Morgan and Kaufmann Pub. (1991).
- [12] Vijilbrief, T.; van Oosterom, P. The {GEO++} system: an extensible {GIS}. In: *Proc. 5th International Symposium on Spatial Data Handling, Charleston, SC, USA*, p.40-50 (1992).

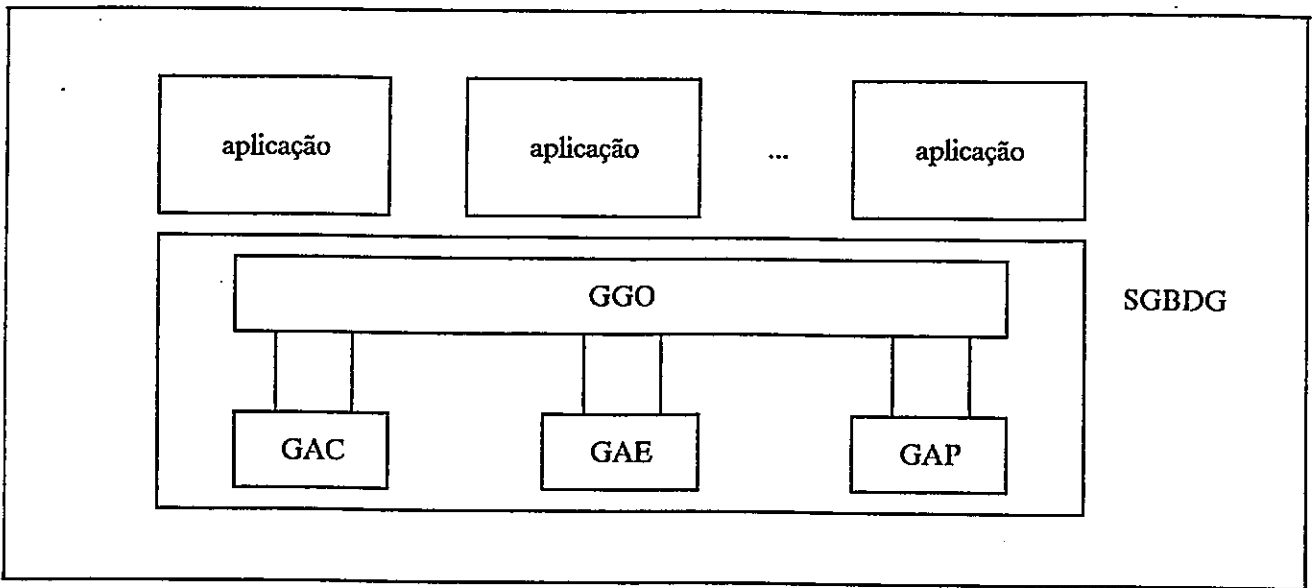


Figura 1: Arquitetura do SGBDG

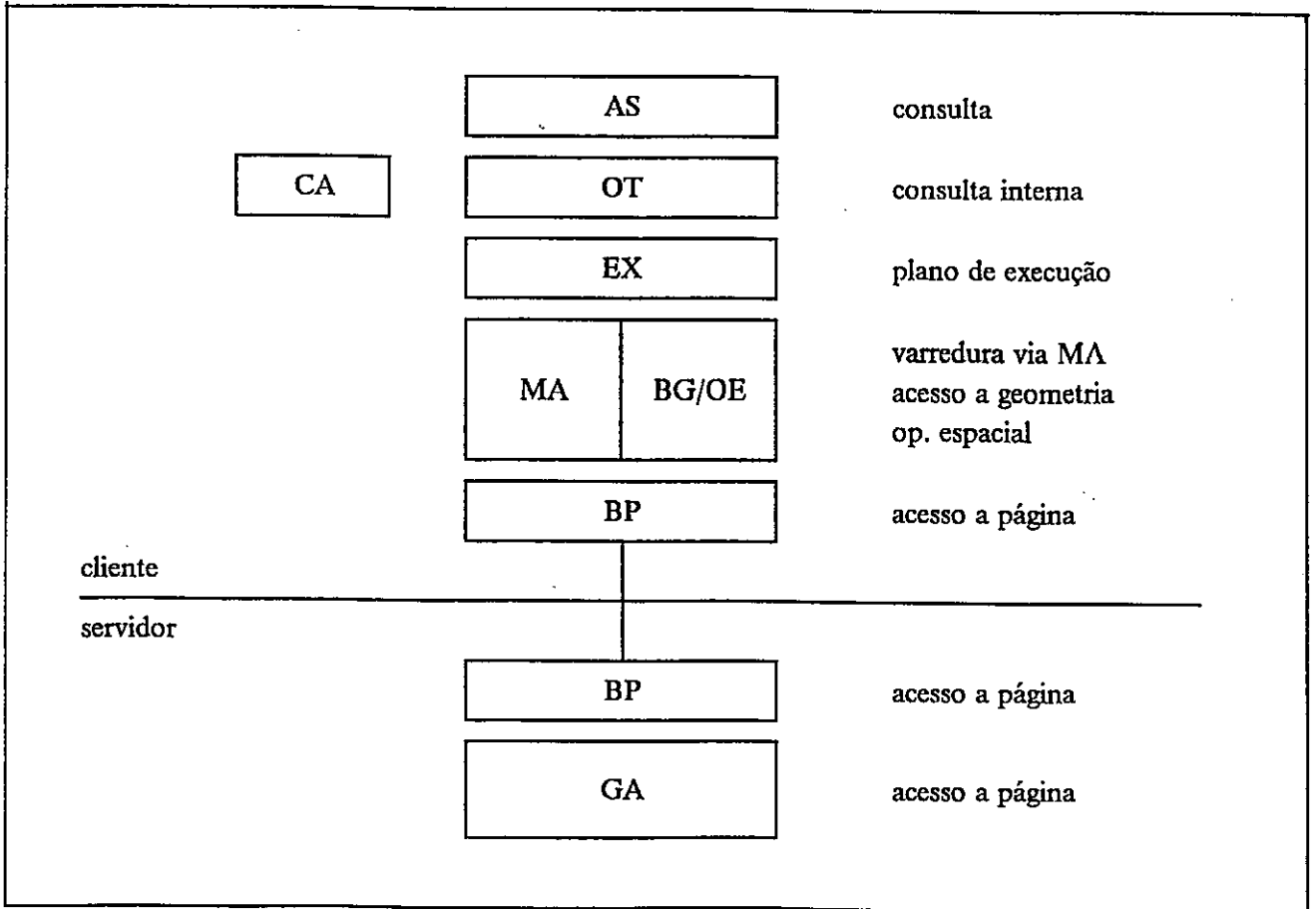


Figura 2: Arquitetura do GAE