

GODBases – Módulo de Banco de Dados Orientado a Objetos

Lucy Choque Mansilla
lucyacm@ime.usp.br

Silvia Scheunemann Silva
silviass@ime.usp.br

December 5, 2014

1 GODBases

O módulo de banco de dados orientado a objetos (GODBases) foi elaborado com o intuito de compor o Projeto GOD e cumprir os requisitos da disciplina de Programação Orientada a Objetos. Tem por objetivo fornecer um banco de dados capaz de armazenar informações sobre o projeto GOD, possibilitando operações de inserção, remoção e busca. Seu desenvolvimento foi realizado no Squeak Smalltalk, através do banco de dados orientado a objetos, Magma.

1.1 Magma

Magma é uma biblioteca de banco de dados orientado a objetos disponível no Squeak, no modo monousuário (single-user) e também multiusuário (arquitetura cliente-servidor). Os pacotes e métodos disponíveis podem ser obtidos através do download na página:

- <http://map.squeak.org/packagesbyname>.

1.1.1 Instalação do magma

Nesta seção são descritos os requisitos para instalação e instruções principais para trabalhar com a biblioteca.

Para o uso do magma, é recomendável a instalação de uma máquina virtual (Virtual Machine - VM) para lidar com problemas de sistema operacional na parte de persistência de informações (uso de imagens). De uma forma geral, a VM proporciona um ganho significativo de performance.

1.1.2 Passos de Instalação

Um modo de instalar é usando o "SqueakMap Catalog" que está em "Apps" na barra de menu do Squeak (ver Fig. 1).

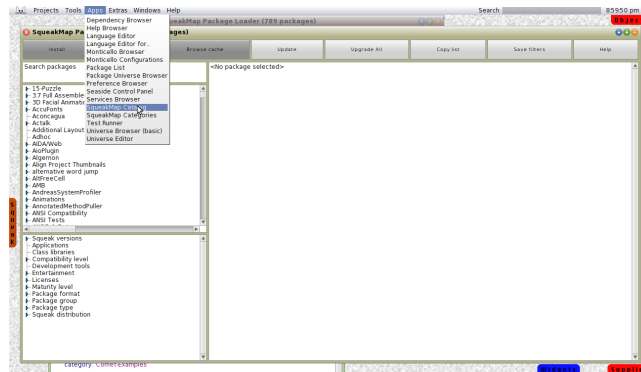


Figure 1: Instruções para instalação do magma.

Logo, seguir os seguintes passos:

- No SqueakMap clique com o botão direito do mouse no quadro esquerdo superior e garantir que todas as opções estejam desmarcadas. Isso permitirá visualizar os pacotes do Magma (Ver Fig. 2).

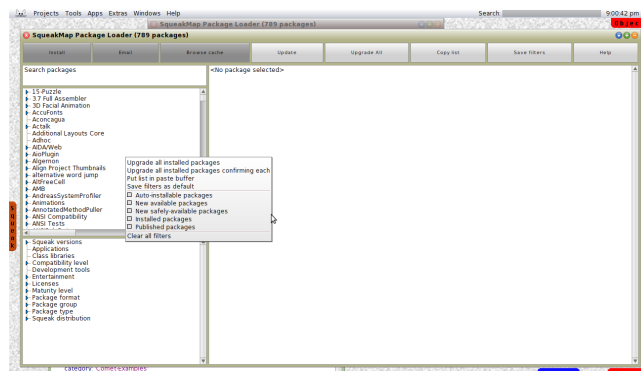


Figure 2: Instruções passo 2.

- Clique com o botão direito do mouse no pacote "client" (versão 1.4) do Magma e escolher a opção "install".(Ver Fig. 3)
- Repetir o ponto 2 para instalar os pacotes "server" e o "test" do Magma. (Ver Fig. 4)

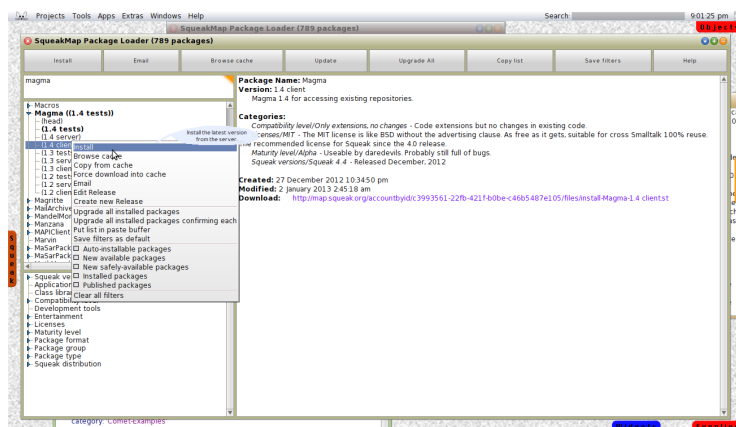


Figure 3: Instruções passo 3.

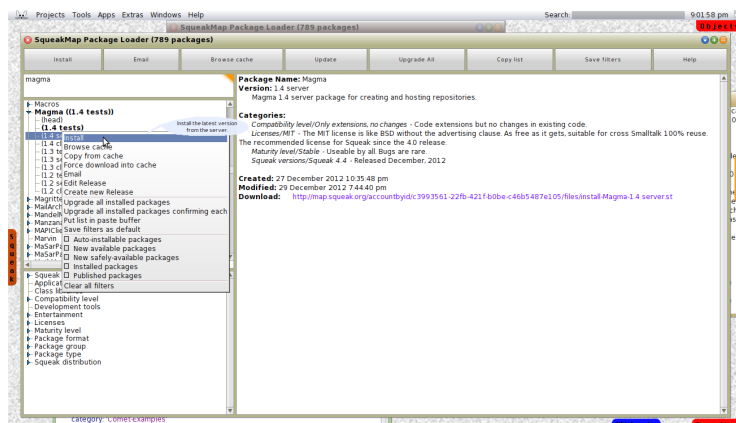


Figure 4: Instruções passo 4.

1.2 Classes do GODBases

O pacote GODBases contém os principais métodos do banco de dados orientado a objetos do projeto GOD. A seguir fazemos uma pequena descrição sobre cada uma de suas classes.

1.2.1 GDBServerController

A classe GDBControllerServer permite fazer transações diferentes no servidor do banco de dados, bem como criar e eliminar um repositório.

- Para criar um repositório padrão, execute a seguinte linha no workspace:
GDBServerController createRepository.
- Para criar um repositório específico, execute:
GDBServerController createRepository: 'nomeDoRepositorio'.
- Para inicializar o servidor com um repositório padrão, execute:
GDBServerController startServer.

- Para inicializar o servidor com um repositório específico, execute:
GDBServerController startServer:'nomeDoRepositorio'.
- Para saber se o servidor foi iniciado, execute: GDBServerController is-Started.
esse comando retorna 'true' se o servidor estiver inicializado.

Uma vez inicializado o servidor, salve como uma imagem-servidor. Assim, para fechar o servidor, só precisa fechar a imagem salva, e quando você abri-lo novamente, o servidor já estará funcionando.

- Uma outra opção para fechar o servidor, é executar:
GDBServerController stopServer.
- Para apagar um repositório específico, execute:
GDBServerController deleteRepository:'nomeDoRepositorio'

1.2.2 GDBClientController

A classe GDBClientController permite fazer a conexão de um usuário/ cliente com o servidor do banco de dados. Primeiro é necessário inicializar o servidor em uma imagem separada, como indicado anteriormente, em seguida, executar as seguintes instruções:

- Para conectar um cliente com o banco de dados, execute o seguinte:
GDBClientController initializeSession

Uma vez inicializada a sessão do usuário é possível fazer qualquer transação no banco de dados, ou seja, é possível fazer uso dos métodos contidos nas classes: GDBDatabase, GDBData e GDBConference.

- Para atualizar qualquer alteração de outro cliente, antes de qualquer transação execute:
GDBClientController refresh.
- Para alterar a sessão do cliente, execute o seguinte:
GDBClientController refresh: 'nomeDeOutroCliente'.

Uma vez que não se precise de fazer nenhuma transação a mais no banco de dados, só temos que desconectar a sessão do cliente, para isso execute o seguinte: GDBClientController release.

1.2.3 GDBDatabase

A classe GDBDatabase permite criar o banco de dados. Nosso banco de dados é representado por um dicionário que armazena duas coleções, uma coleção do tipo MagmaCollection para armazenar os objetos GODData e uma outra do tipo OrderedCollection para armazenar uma coleção que contém objetos GC-Conference.

- Para criar o banco de dados, execute:
GDBDatabase createDatabase

- Se você quiser criar apenas o banco de dados para armazenar objetos GODData, execute:
GDBData createGODDataCollection.
- Se você quiser criar apenas o banco de dados para armazenar a coleção que vai conter objetos GCConference, execute:
GDBConference createConferenceCollection.

1.2.4 GDBData

A classe GDBData permite fazer transações diferentes no banco de dados para objetos GODData.

- Para inserir um novo objeto GODData no banco de dados, execute:
GDBData add: meuGODData.
- Para remover um objeto GODData do banco de dados, execute:
GDBData remove: meuGODData.
- Para atualizar um objeto GODData por um outro objeto GODData, execute:
GDBData update: meuAntigoGODData with: meuNovoGODData.
- Se você quiser apagar todos os objetos GODData armazenados no banco de dados, execute:
GDBData resetData.
- Para buscar um objeto GODData usando parte do nome do autor, execute:
GDBData searchAuthor: 'parteDoNomeDoAutor'.
- Para buscar um objeto GODData usando parte do nome do título, execute:
GDBData searchTitle: 'parteDoNomeDoTitulo'.
- Para buscar um objeto GODData usando um bloco, execute:
GDBData searchFor: umBloco.
onde um bloco deve ter a forma:
[: objeto|objetoatributoDoObjetoGODData = valorDeComparacaoParaABusca].
- Para buscar um objeto GODData usando seu ID, execute:
GDBData searchById: umIdDoTipoInteiro.
- Para fazer uma busca exata pelo nome do autor, execute:
GDBData exactSearchByAuthor: 'nomeExatoDoAutor'.
- Para fazer uma busca exata pelo título, execute:
GDBData exactSearchByTitle: 'nomeExatoDoTitulo'.

1.2.5 GDBConference

A classe GDBConference permite fazer diferentes transações no banco de dados para objetos GCConference.

- Para inserir uma nova coleção que armazena objetos `GCConference`, execute:
`GDBCConference saveConferences: minhaColeçãoGCConference.`
- Para recuperar essa coleção armazenada no banco de dados, execute:
`GDBCConference loadConferences.`
- Se você quiser apagar essa coleção armazenada no banco de dados, execute:
`GDBCConference resetConferenceData.`