**Sistema de Consultório Médico**

**Autores: Tarcísio Andrade e Thais Silva**

**Visão geral do sistema**

É proposto o desenvolvimento de um sistema de controle de consultório medico, que visa informatizar as funções de cadastro de pacientes e funcionários, dentistas por especialidades, orçamento por cliente, permitir o agendamento de pacientes e emissão de relatórios de serviços prestados e de atendimento. O objetivo do sistema é agilizar o processo de atendimento e garantir maior segurança, ao mesmo tempo em que possibilita um melhor controle das informações por parte da gerência. Deverá ser gerado relatórios de acesso.

**Descrição do Produto**

Medsys é um software para utilização em consultórios; O sistema irá conter um cadastro de pacientes, funcionários, médicos, serviços, agendamento de consultas e emissão de relatórios.

**Especificação do Projeto**

O projeto deve estar de acordo com os seguintes documentos de especificação:

- Documento de requisitos

- Documento de casos de uso

**Visão geral do sistema**

O sistema deve seguir os seguintes passos:

1. Funcionário (atendente) cadastra o paciente no sistema, solicitando os seus dados pessoais (Nome, RG, CPF, endereço, etc.)

2. Funcionário (administrador do sistema) cadastra novo funcionário e médicos para o consultório.

3. Atendente agendará para o paciente previamente cadastrado, uma consulta encaminhando-o a um determinado médico, sendo emitida a receita e o valor da consulta.

4. Funcionário irá fazer o cadastro dos serviços prestados como receita emitida pelo médico após a consulta e outros serviços relacionados ao consultório.

5. Funcionário emitirá relatórios de pacientes, médicos e funcionários cadastrados.

**Requisitos Funcionais e Não Funcionais**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **F1 Cadastro de Paciente** | | **Oculto ( )** | | | |
|  | **Descrição:** O sistema deve cadastrar o paciente, médico e funcionários informando o nome, CPF, rg, data de nascimento, endereço, e a data do cadastro | | | | | |
|  | ***Requisitos Não Funcionais*** | | | | | |
|  | **Nome** | **Restrição** | | **Categoria** | **Desejável** | **Permanente** |
|  | NF1.1 Controle de Acesso | A função só pode ser acessada por usuário com perfil de operador ou superior. | | Segurança | **( )** | **(x)** |
|  | NF1.5 Janela única | Todas as funções relacionadas a cadastro devem ser efetuadas em uma única janela | | Interface | **(x)** | **(x)** |
|  | **F2 Agendar Consulta** | | **Oculto ( )** | | | |
|  | **Descrição:** O sistema deve registrar ao agendamento de consulta, indicando o paciente e o medico, bem como a data do atendimento e valor previsto para pagamento da consulta. | | | | | |
|  | ***Requisitos Não Funcionais*** | | | | | |
|  | **Nome** | **Restrição** | | **Categoria** | **Desejável** | **Permanente** |
|  | NF1.1 Controle de Acesso | A função só pode ser acessada por usuário com perfil de operador ou superior. | | Segurança | **( )** | **(x)** |
|  | NF1.5 Janela única | Todas as funções relacionadas a agendamento devem ser efetuadas em uma única janela | | Interface | **(x)** | **(x)** |
|  | **F3 Cadastro de Médico** | | **Oculto ( )** | | | |
|  | **Descrição:** O sistema deve cadastrar o medico informando o nome, CPF, rg, CRM, data de nascimento, endereço, e a data do cadastro | | | | | |
|  | ***Requisitos Não Funcionais*** | | | | | |
|  | **Nome** | **Restrição** | | **Categoria** | **Desejável** | **Permanente** |
|  | NF1.1 Controle de Acesso | A função só pode ser acessada por usuário com perfil de operador ou superior. | | Segurança | **( )** | **(x)** |
|  | NF1.5 Janela única | Todas as funções relacionadas a cadastro devem ser efetuadas em uma única janela | | Interface | **(x)** | **(x)** |
|  | **F4 Registrar atendimento** | | **Oculto ( )** | | | |
|  | **Descrição:** O sistema deve registrar o atendimento de pacientes, indicando o código do funcionário que fez o atendimento e a matricula do paciente, bem como a data do atendimento. | | | | | |
|  | ***Requisitos Não Funcionais*** | | | | | |
|  | **Nome** | **Restrição** | | **Categoria** | **Desejável** | **Permanente** |
|  | NF1.1 Controle de Acesso | A função só pode ser acessada por usuário com perfil de operador ou superior. | | Segurança | **( )** | **(x)** |
|  | NF1.5 Janela única | Todas as funções relacionadas a registro devem ser efetuadas em uma única janela | | Interface | **(x)** | **(x)** |
|  | **F5 Cadastro de serviços** | | **Oculto ( )** | | | |
|  | **Descrição:** O sistema deve cadastrar os serviços informando o valor do serviço e a descrição | | | | | |
|  | ***Requisitos Não Funcionais*** | | | | | |
|  | **Nome** | **Restrição** | | **Categoria** | **Desejável** | **Permanente** |
|  | NF1.1 Controle de Acesso | A função só pode ser acessada por usuário com perfil de operador ou superior. | | Segurança | **( )** | **(x)** |
|  | NF1.5 Janela única | Todas as funções relacionadas a cadastro devem ser efetuadas em uma única janela | | Interface | **(x)** | **(x)** |
|  | **F6 Cadastro de Funcionário** | | **Oculto ( )** | | | |
|  | **Descrição:** O sistema deve registrar o atendimento de pacientes, indicando o código do funcionário que fez o atendimento e a matricula do paciente, bem como a data do atendimento. | | | | | |
|  | ***Requisitos Não Funcionais*** | | | | | |
|  | **Nome** | **Restrição** | | **Categoria** | **Desejável** | **Permanente** |
|  | NF1.1 Controle de Acesso | A função só pode ser acessada por usuário com perfil de operador ou superior. | | Segurança | **( )** | **(x)** |
|  | NF1.5 Janela única | Todas as funções relacionadas a cadastro devem ser efetuadas em uma única janela | | Interface | **(x)** | **(x)** |

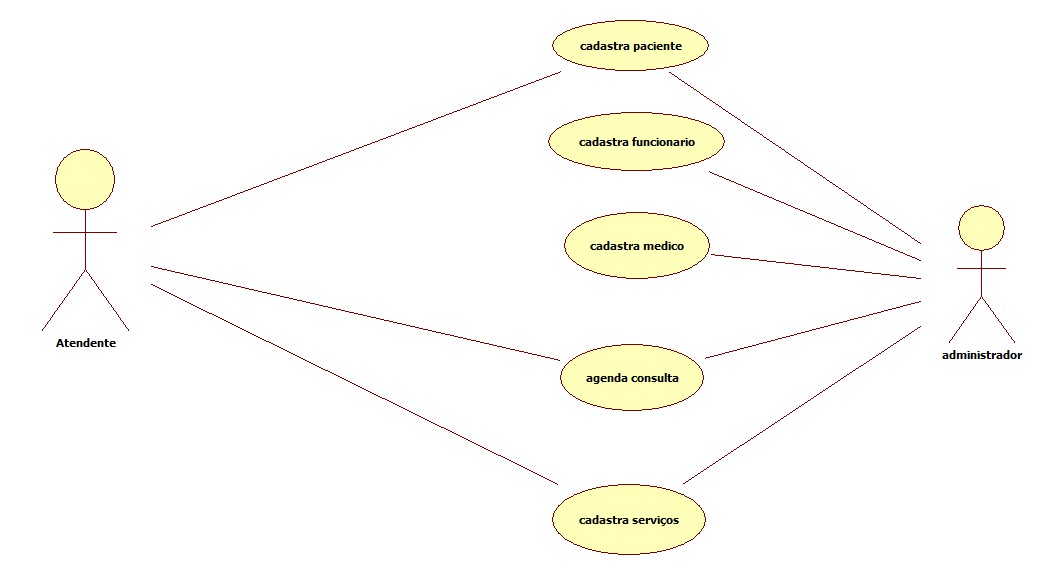
**Requisitos Suplementares**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| S1 Tipo de Interface | As interfaces do sistema devem ser implementadas como formulários acessíveis em linguagem java e em um browser html. | Interface | **( )** | **( x )** |
| S2 Armazenamento de dados | A camada de persistência deve ser implementada em java DB. | Persistência | **( )** | **( x )** |
| S3 Perfis de usuário | Os perfis de usuário para acesso ao sistema são:  3. Administrador - pode efetuar todas as operações.  2. Operador - pode efetuar as operações de cadastro, agendamento e pagamento . | Segurança | **( )** | **( x )** |
| ... | ... | ... | ... | ... |

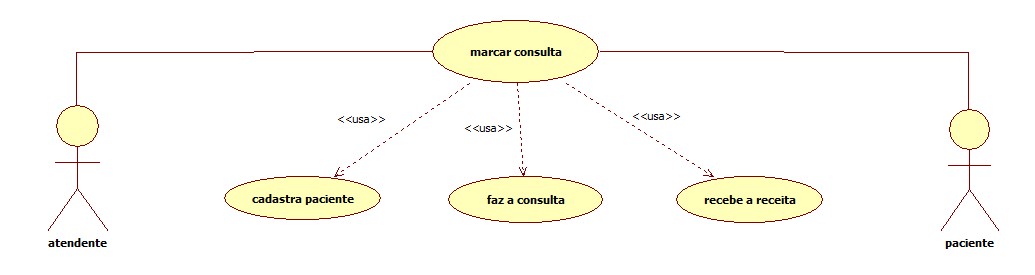
**Casos de Uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Atores** | **Descrição** | **Referências Cruzadas** |
| Cadastrar  Paciente | paciente,  Secretária | O paciente se identifica e solicita agendamento de uma consulta. O funcionário faz o cadastro desse paciente no sistema. | F1 |
| Cadastrar um funcionário | administrador | O administrador do sistema cadastra os dados de um novo funcionário. | F3 |
| Cadastrar um médico | administrador | O administrador do sistema cadastra os dados de um novo médico. | F6 |
| Agenda Consulta | paciente,  Secretária | O paciente solicita agendamento de uma consulta. A atendente agenda a consulta. | F2 |
| Cadastro de serviços | Atendente, | O serviço (emissão de receita e outros relacionados com o consultório) e cadastrado no sistema pelo funcionário. | F5 |

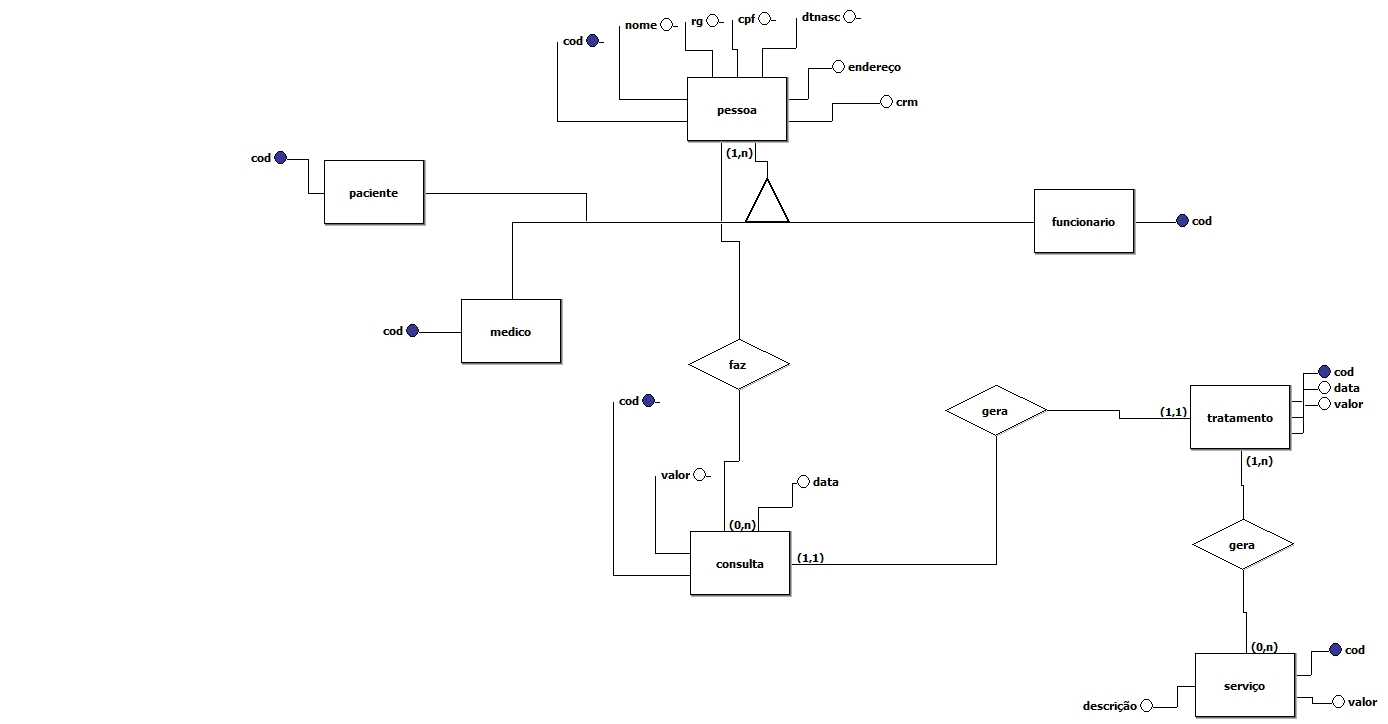
**Diagrama de Casos de Uso**

****

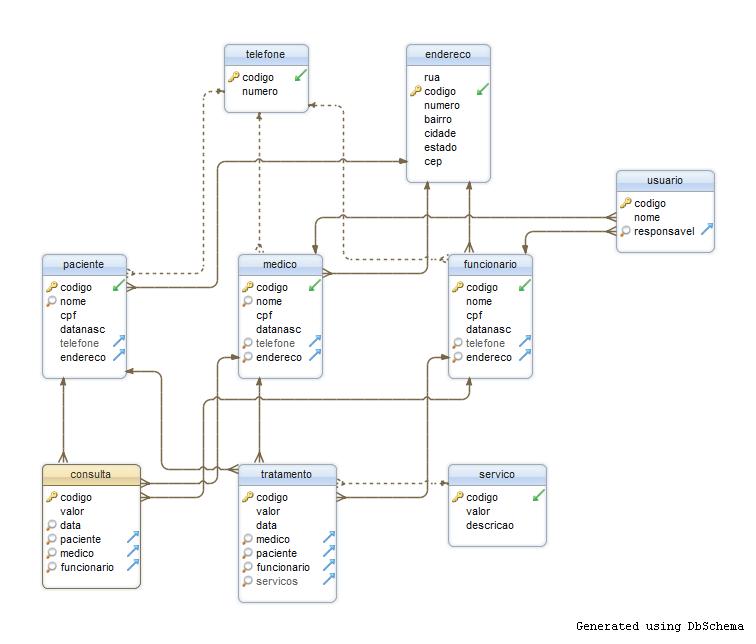
|  |
| --- |
| **Caso de Uso: Marcar consulta** |
| **Fluxo Principal**  1. O paciente se identifica e quer marcar uma consulta.  2. O funcionário faz seu cadastro no sistema.  3. O funcionário indica qual horário deverá ser sua consulta.  4. O sistema informa a data da consulta.  5. O paciente realiza a consulta:  6. O medico conclui a consulta e emite a receita. |



**Modelo Conceitual**

****

**Diagrama de Físico**

****

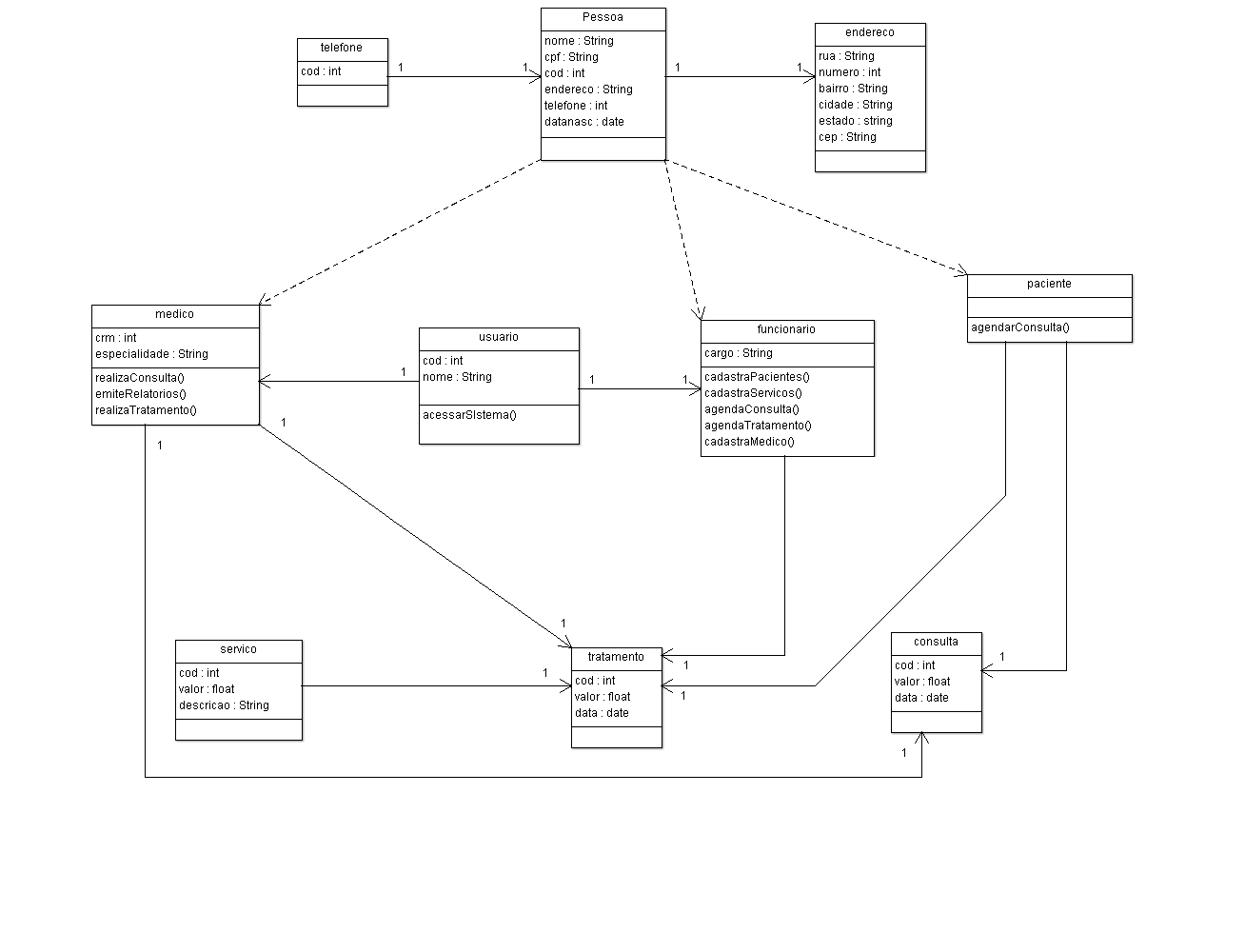
**Estudo da normalização do BD**

• O banco se encontra na primeira forma normal, pois todo o cadastro de pessoas (pacientes, médicos e funcionários), endereço e telefone não estão na mesma tabela, não contendo grupo repetido e campos multivalorados.

• o banco se encontra na segunda forma normal, pois, atende os requisitos da primeira forma e todos os atributos não chave foram identificados e criadas novas entidades com eles.

• o banco se encontra na terceira forma normal, pois atende a primeira e segunda forma normal e nenhum atributo não chave depende funcionalmente de nenhum outro atributo não-chave.

**Diagrama de Classes**

****

**Operações de Manutenção**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Conceito** | **I** | **A** | **E** | **C** | **Observação** | **Ref. Cruzadas** |
| Paciente | x | x | x | x | Só é possível excluir se não houver consultas associadas | F1 |
| Secretaria | x | x | x | x |  | F6 |
| Medico | x | x | x | x | Só é possível excluir se não houver empréstimos associados | F3 |
| Agendamento |  |  | x | x | A inclusão de agendamento só pode acontecer através do caso de uso “agendar consulta”.  é possível alterar uma consulta. | F1,F2 |

**Consultas**

|  |  |
| --- | --- |
| **Nome** | **Referências Cruzadas** |
| Consultas agendadas | F2,F1 |
| Paciente cadastrados | F1 |
| Funcionários cadastrados | F6 |
| medico cadastrados | F3 |
| Cadastra serviços | F5 |
|  |  |

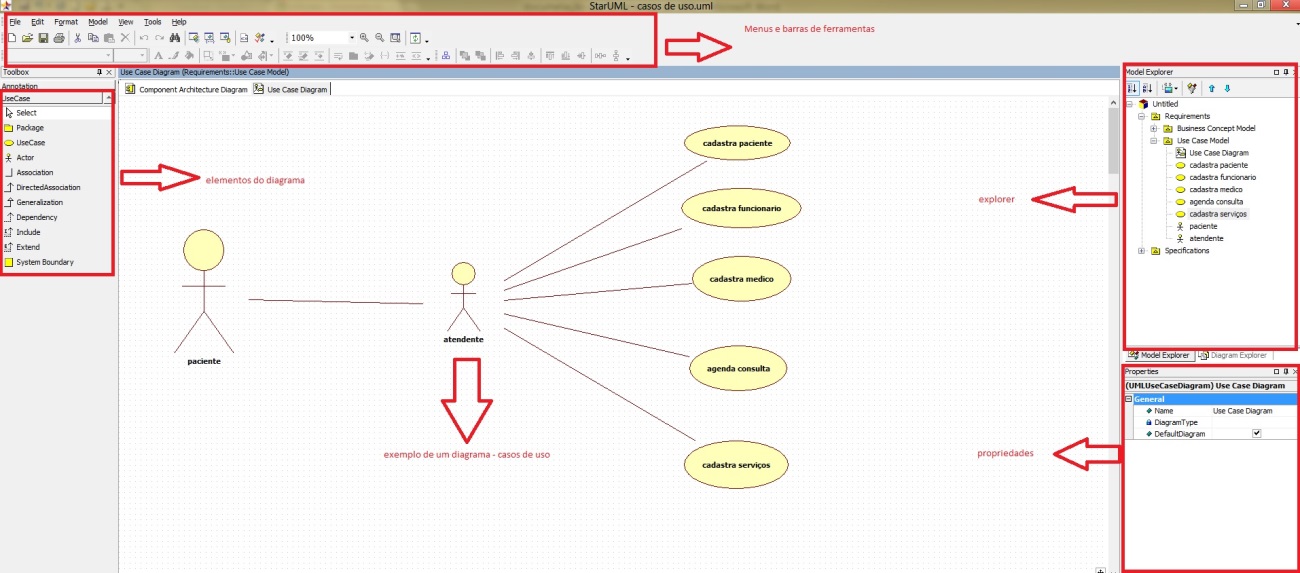
**Ciclos Iterativos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ciclo** | **Casos de Uso** | **Manutenção de Informações** | **Consultas** | **Observações** | **Esforço estimado** |
| 1 | Cadastra dentista,paciente,serviço (30) | - | - | Neste ciclo é implantado a persistência de dados | 30 horas |
| 2 | Agenda consulta (10) | - | - | Implementar mecanismo de persistência (50 horas) | 10 horas |
| 3 | Registra atendimento (10) |  | - | - | 10 horas |

**StarUML - UML/MDA Plataform**

Star UML (STAR UML, 2010) é uma ferramenta CASE de código aberto (opensource) e está sob a licença GPL (General Public License). Ela dá suporte à modelagem de sistemas utilizando os diagramas da UML2 e também à MDA, com definições de transformações para algumas plataformas específicas. É permitida também a importação/exportação de modelos utilizando o formato XMI.

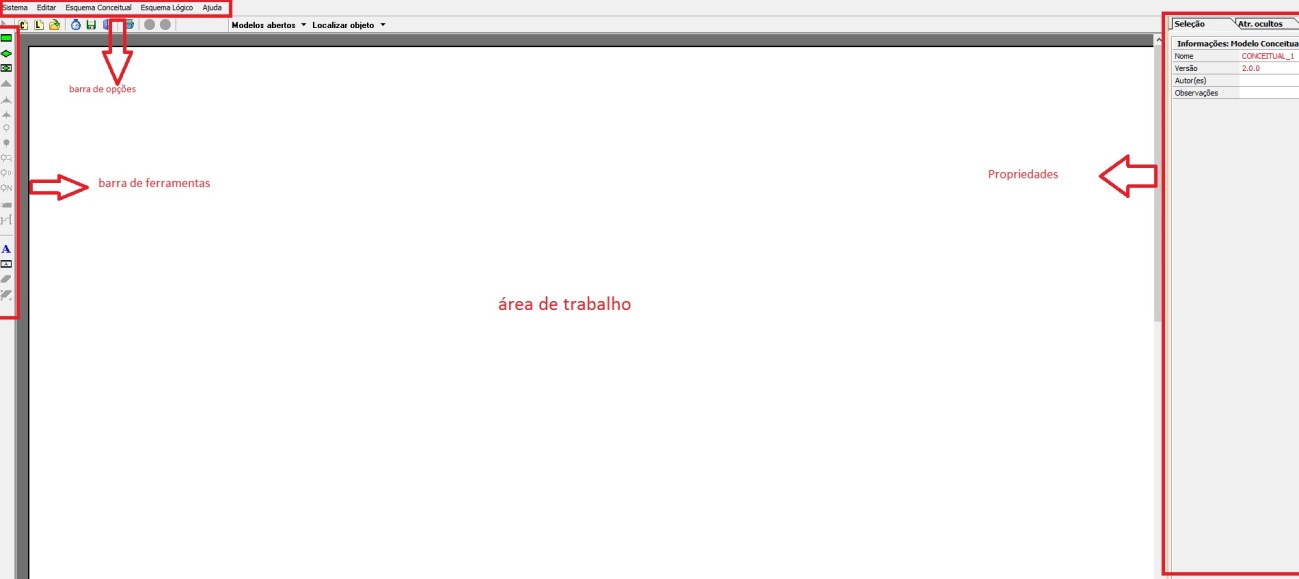
A especificação de perfis UML na ferramenta é feita de forma diferente das outras ferramentas analisadas. Não há uma forma visual de se implementar perfis. É necessário escrever o código do perfil em um documento XML, salvar o arquivo com a extensão.PRF e colocá-lo em um dos diretórios de instalação da ferramenta.



**Br Modelo**

Ferramenta freeware voltada para ensino de modelagem em banco de dados relacional com base na metodologia defendida por Carlos A. Heuser no livro "Projeto de Bando de Dados" (capa a baixo).

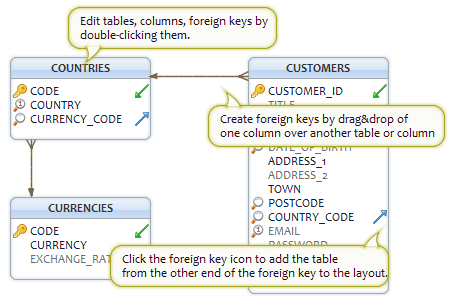
Esta ferramenta foi desenvolvida por Carlos Henrique Cândido sob a orientação do Prof. Dr. Ronaldo dos Santos Mello (UFSC), como trabalho de conclusão do cusro de pós-graduação em banco de dados (UNVAG - MT e UFSC).



**DB Squema**

Esquema é representado em layouts diagramada. Editar tabelas ou colunas diretamente no layout, com um duplo-clique sobre elas. Criar novas chaves estrangeiras por drag & drop de uma coluna da tabela.

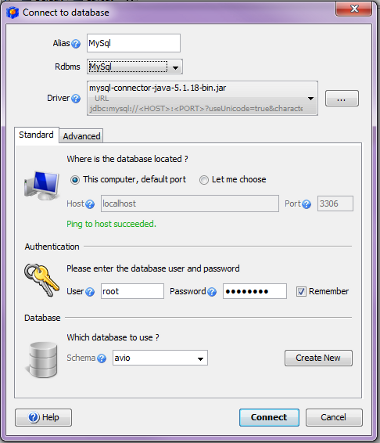
Obter uma melhor compreensão do esquema através da criação de layouts múltiplos, cada um focado em uma parte específica do esquema. Os layouts e da estrutura do esquema serão salvos no arquivo de projeto dbschema



Novos projetos podem ser iniciados por engenharia reversa do esquema do banco de dados. A estrutura do esquema será salvo arquivo de projeto e pode ser reaberta mais tarde sem se conectar ao banco de dados.

De um projeto criado, você pode se conectar a outro banco de dados e sincronizar os esquemas. Como você sincronizar um esquema de desenvolvimento com o esquema em um banco de dados de produção.

Dbschema usa drivers JDBC para se conectar a qualquer banco de dados relacional.



Um extremamente simples e eficiente explorador de dados relacional que oferece vistas simultâneas de dados de várias tabelas e recursos inúmeros filtros, dados os mecanismos de classificação, e muito mais.

Os dados podem ser exploradas de várias tabelas em mesmo tempo, com base nas chaves estrangeiras entre tabelas. Virtual chaves estrangeiras pode ser criado. Novos registros podem ser inseridos, os registros existentes podem ser editados. Leia mais

• Tamanho ilimitado de árvore de navegação

• Filtrar dados de tabela

• Inserir dados na tabela de procura

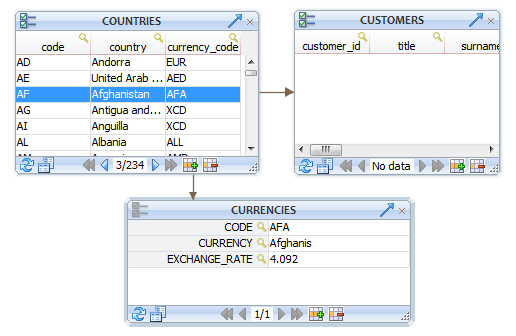
• Salve navegar árvore no arquivo de projeto e reabri-lo mais tarde

• Uso de memória otimizada

• Ver Blob ou Clob dados do banco de dados

• Maneira fácil de dados de busca no banco de dados

• Impressão



Graficamente compor consultas SQL. Você pode fazer uso de cláusulas de junção diferentes, como junção externa esquerda , use onde filtros ou gerar grupo por consultas. As tabelas podem ser adicionados por arrastar e soltar ou pressionando o ícone de chave estrangeira.

• Editar consultas complexas usando o mouse

• Selecione as colunas queriy

• Dinamicamente adicionar novas tabelas para a consulta

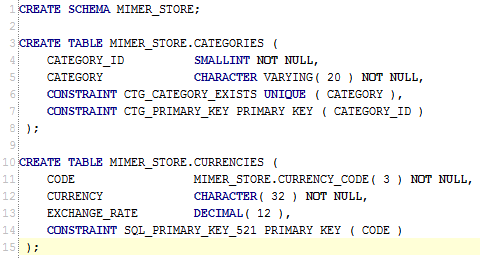
• Use Group By condições para a consulta

• Definir consulta onde as condições

• Executar a consulta

• Visualize a consulta SQL

Em dbschema você pode gerar os scripts de esquema e de exportação óptima os dados para o script. Usando o script gerado, você pode recriar o esquema em qualquer banco de dados.



**ArgoUML**

ArgoUML é uma aplicação open source que usa UML para modelar o desenho de software de computador. A aplicação corre na maior parte das plataformas uma vez que é implementada em Java. Está distribuída sob a licença BSD. Providencia suporte para quase todos os tipos de diagrama da UML padrão e inclui suporte cognitivo.

A Software Development Magazine realiza uma premiação anual entre ferramentas populares de desenvolvimento de software em várias categorias. Em 2003 o ArgoUML foi um dos finalistas na categoria "Ferramentas de Design e Análises". Ele recebeu um prêmio de revelação, derrotando várias ferramentas comerciais.

**Características**

Novidades na versão 0.20:

• UML 1.4 - Características de extensibilidade aprimoradas do UML 1.4,

• Diagramas de Sequências

• Compatibilidade com AndroMDA,

• Muitas das funções agora suportam seleção múltipla de modelamentos,

• Arrastar e Soltar da árvore de diretório ao diagrama ou dentro da própria árvore.

• Interface de módulos extensíveis,

• Suporte a internacionalização para Inglês, Alemão, Francês, Espanhol e Russo,

• Restrições OCL para Classes,

•Suporte de geração de código: Java, PHP, Python, C++ e Csharp (C#),

• Engenharia reversa,

• Diagrama de classes com layout automático,

• Geração de arquivos PNG, GIF, JPG, SVG e EPS a partir de diagramas,

• Suporte a comentários para múltiplos elementos,

• Todos os diagramas UML 1.4 são suportados.

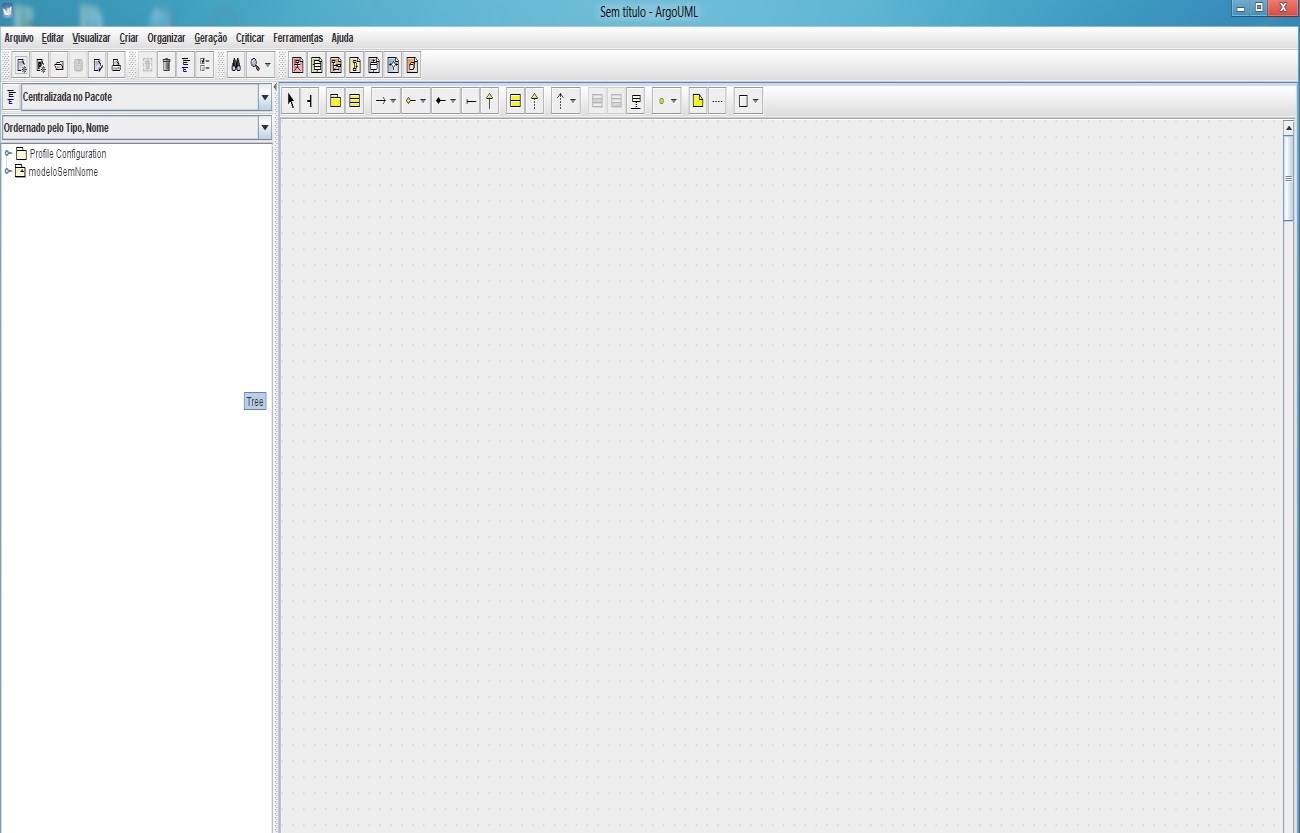
**Desvantagens**

• Incompatibilidade entre versões,

• Importação/Exportação para Java,

• Não há auto chamadas em diagramas de sequências,

•Ausência de undo (desfazer a última ação).



ArgoUML – área de trabalho