



Relatório Trabalho I

Desenho e construção de uma base de dados

Introdução

O objetivo deste trabalho foi criar uma base de dados em MySQL e fazer uma interface para o utilizador, em java, para consulta e edição da mesma.

Deste modo criou-se uma base de dados sobre o espaço, com informação acerca de planetas, constelações, etc., que contém também informações acerca de fotos das classes referidas bem como dos telescópios que estão na origem das fotos.

Criação do Modelo

ESCOLHA DAS ENTIDADES

As entidades escolhidas foram: Planetas, Sistemas Planetários, Constelações, Galáxias, Grupos de Galáxias, Clusters de Galáxias, Telescópios e Fotos (surgiram posteriormente novas entidades provenientes das relações n:m).

ESCOLHA DOS ATRIBUTOS

Escolheram-se os atributos tendo por base a sua relevância e utilidade para descrever as diferentes classes, bem como a facilidade em encontrar informação acerca destas.

ESCOLHA DAS RELAÇÕES - DIAGRAMA ER

As relações escolhidas tiveram por base o organograma do espaço:

- Um planeta tem de ter obrigatoriamente um sistema planetário, e um sistema planetário (leia-se estrela), pode não ter nenhum planeta mas também pode ter vários.
- Uma constelação tem de ser obrigatoriamente formada por vários sistemas planetários (leia-se estrelas) mas um sistema planetário (leia-se estrela) pode não ser um elemento de uma constelação.
- Um sistema planetário tem obrigatoriamente de pertencer a uma galáxia e uma galáxia pode ter vários sistemas planetários.
- Uma galáxia tem obrigatoriamente de pertencer a um grupo de galáxias e um grupo de galáxias pode ter várias galáxias.
- Um grupo de galáxias tem obrigatoriamente de pertencer a um cluster de galáxias e um cluster de galáxias pode ter vários grupos de galáxias.
- Um telescópio pode tirar nenhuma ou várias fotos e uma foto tem obrigatoriamente de mostrar 1 ou mais objetos da mesma classe.

Nota: Um sistema planetário é caracterizado pela sua estrela. Deste modo para facilitar considera-se também um sistema planetário mesmo que este não contenha quaisquer planetas.



O diagrama ER da base de dados criada encontra-se na figura 1.

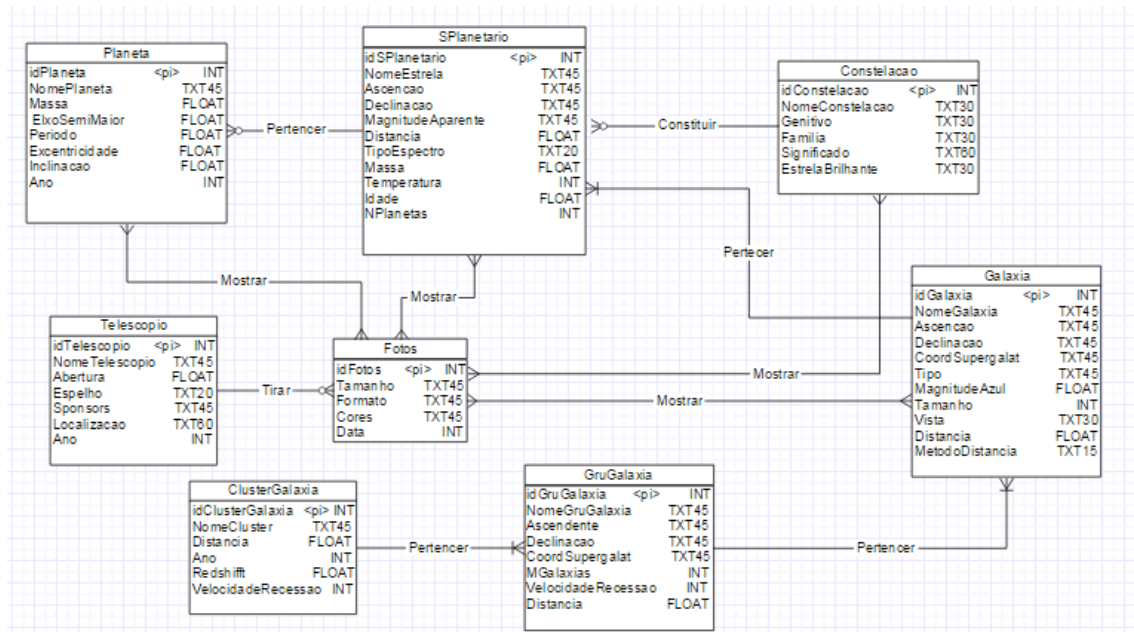


Figura 1 Diagrama ER da base de dados "basedosespaco".

Posteriormente passou-se do diagrama ER para o modelo físico. Este encontra-se na figura 2 e pode-se evidenciar a presença de mais 4 entidades relacionais resultantes das relações n:m entre as entidades fotos-planetas, fotos-splanetários, fotos-galaxias e fotos-constelações. As outras relações não demonstram necessidade de criação de mais nenhuma tabela.

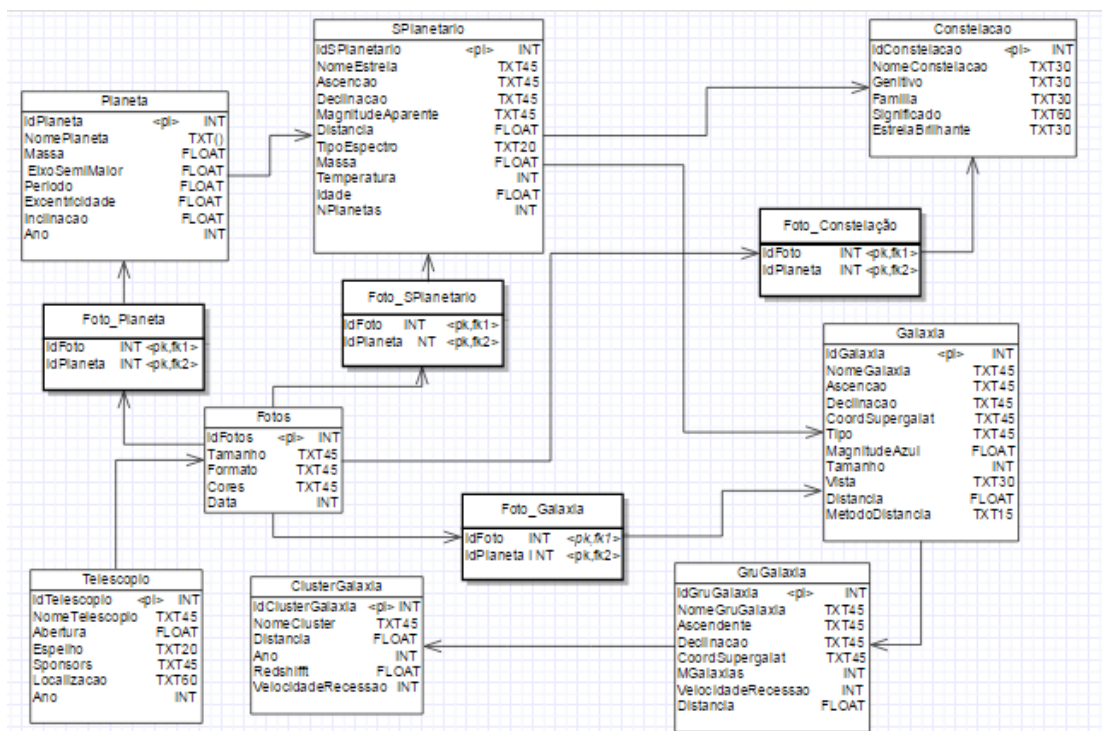


Figura 2 Diagrama físico da base de dados "basedosespaco".



Código SQL para gerar base de dados código

Após se ter definido o conceito, e os diagramas ER e físico procedeu-se à geração do código SQL para gerar a base de dados:

- Criação do esquema;
- Criação das tabelas e respetivos atributos;
- Inserção das instâncias nas tabelas (teve-se o cuidado de se ter mais que 50 instâncias na maioria das tabelas das entidades);

Interface Java

O interface é constituído inicialmente por um painel para o utilizador fazer login. Se este for válido (tem apenas 3 tentativas para tentar) é disponibilizado ao utilizador uma janela para consultar e editar parcialmente a base de dados (inserção e eliminação de instâncias).

A classe “Main” constrói a interface para login, que dependendo da validade do login chama outra classe “Interface” com o interface para consulta e edição da base de dados. Esta última por sua vez chama as classes “ResultSetTable” e “UpdateQuery” para executeQuery() e executeUpdate() respectivamente. “ResultSetTable” executa queries do tipo SELECT e “UpdateQuery” executa queries do tipo INSERT e DELETE.

O utilizador dispõe de 5 separadores para trabalhar em 5 das entidades da base de dados. Após a escolha da entidade em que se quer trabalhar, o utilizador pode escolher se quer consultar, inserir ou eliminar. Se escolher consultar, o resultado da consulta é uma tabela que aparece numa janela pop-up. O utilizador pode fechá-la e executar outras query's. Quando fechar a janela do interface, “GBD Space 1.0”, a conexão à base de dados é terminada automaticamente.

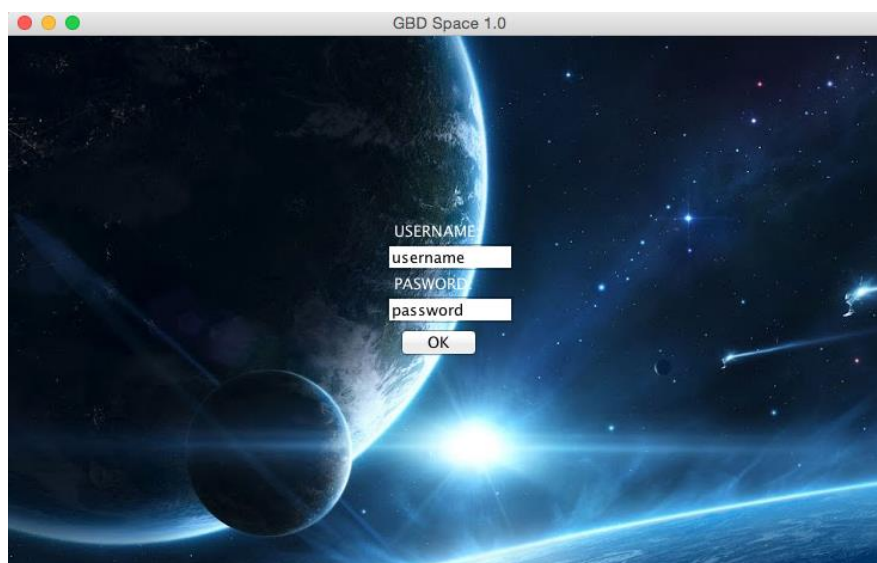


Figura 3 Interface para fazer login na base de dados

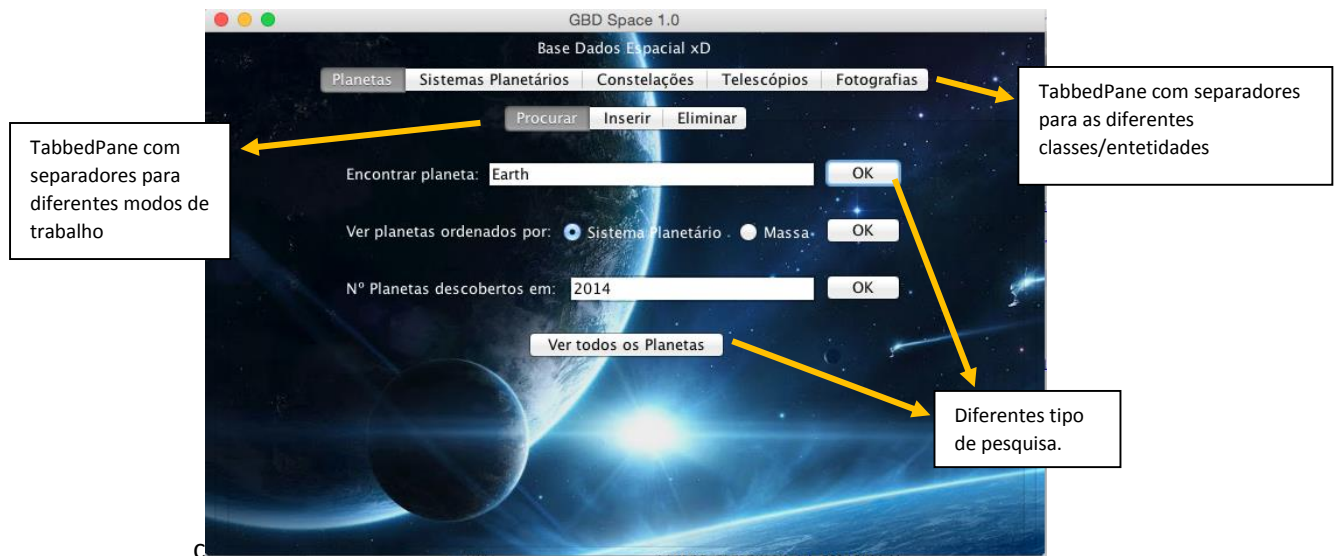


Figura 4 Interface para gestão da base de dados. Encontra-se no separador de procura de planetas.

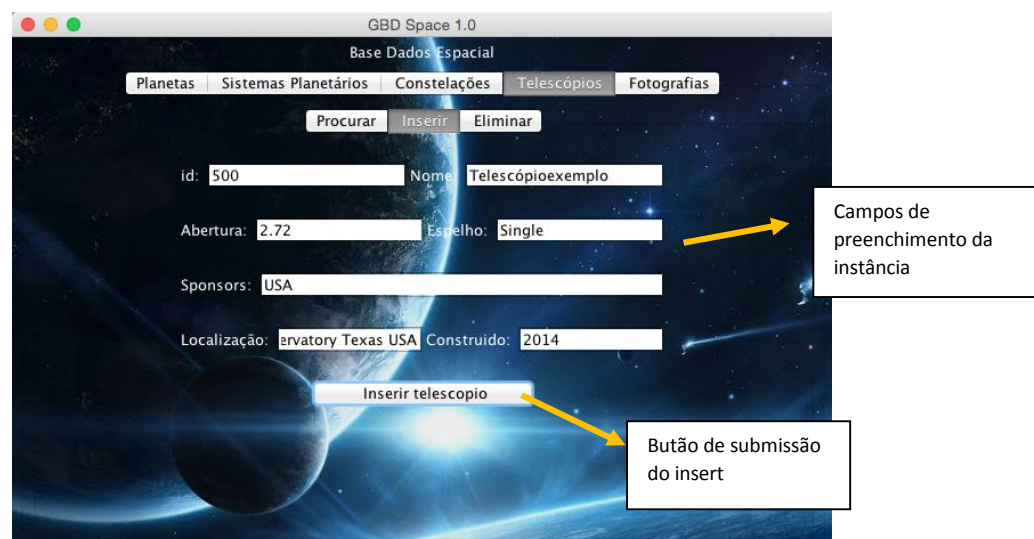


Figura 5 Interface para gestão de base de dados. Encontra-se no separador inserir telescópio.

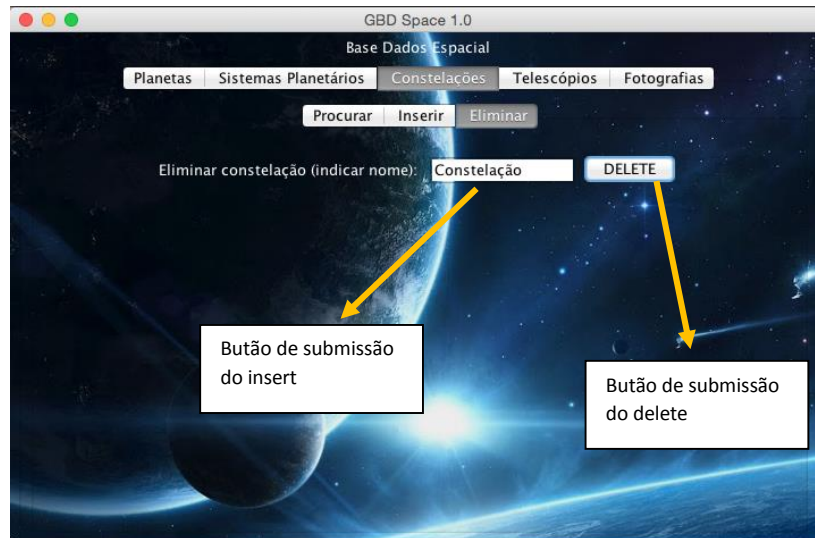


Figura 6 Interface para gestão de base de dados. Encontra-se no separador delete constelação.

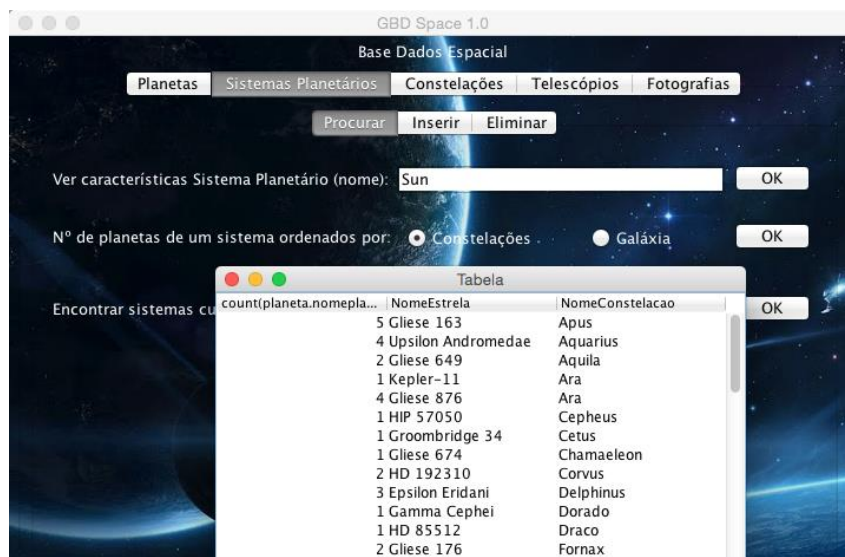


Figura 7 Interface para gestão de base de dados. Plano de destaque para o resultado de uma query: N° de planetas por sistema planetário ordenados pelo nome da constelação a que a estrela pertence (`SELECT count(planeta.nomeplaneta), splanetario.nomeestrela, constelacao.nomeconstelacao FROM planeta,splanetario,constelacao WHERE (splanetario.idplanetario=planeta.splanetario_idplanetario and splanetario.constelacao_idconstelacao=constelacao.idconstelacao) GROUP BY splanetario.idplanetario ORDER BY constelacao_idconstelacao`)

QUERY'S

Como se pode ver nas figuras anteriores, nos painéis de consulta a interface disponibiliza uma série de query's ao utilizador. Deste modo o utilizador apenas pode fazer consultas usando query's já predefinidas. Tentou-se escolher query's que disponibilizem o máximo de informação que achamos ser interessante do ponto de vista do utilizador.



Algumas das query's utilizadas, e respectivos código SQL, foram:

- Encontrar um planeta: `"SELECT * FROM planeta WHERE nomeplaneta= 'Earth' "`;
- Ver planetas ordenados por sistemas planetários: `" Select planeta.*, splanetario.nomeestrela FROM planeta, splanetario where (splanetario_idplanetario=idsplanetario) ORDER BY splanetario.nomeestrela "`;
- Contar planetas descobertos em determinado ano: `"SELECT count(idplaneta) FROM planeta WHERE ano=2014";`
- Eliminar um planeta: `DELETE FROM planeta WHERE nomeplaneta=input;`
- Encontrar um sistema planetário: `"SELECT * FROM splanetario WHERE nomeestrela='Sun'";`
- Encontrar sistema planetário a uma distância do nosso inferior a x: `"SELECT * FROM splanetario WHERE distancia<x";`
- Ver a tabela toda dos sistemas planetários: `"SELECT * FROM splanetario";`
- Contar as constelações visíveis no hemisfério Sul: `"SELECT count(NomeConstelacao), HS FROM constelacao GROUP BY HS";`
- Ver as constelações visíveis no hemisfério norte: `"SELECT NomeConstelacao,hn FROM constelacao";`
- Mostrar todos os atributos dos telescópios que têm uma abertura dentro de um determinado intervalo. `" SELECT * from telescopio WHERE (abertura > minimo AND abertura < máximo)";`
- Inserir um telescópio: `"INSERT INTO telescopio VALUES (atributos)";`
- Procurar todas as fotos com as características escolhidas pelo utilizador: `" SELECT * FROM fotos WHERE (atributo_1=característica_1 and atributo_i=característica_i)";`

Conclusão

Conclui-se que os objectivos do trabalho prático foram cumpridos na medida em que se conseguiu desenhar e criar uma base de dados e elaborar um interface para o utilizador que permita a consulta e edição (inserção e eliminação de instâncias) da mesma.

Anexos

Embora se tenha pensado colocar em anexo os programas java utilizados na construção do interface, devido a sua extensão, tornou-se impossível. Deste modo apresentamos a sua estrutura nas secções seguintes.



Estrutura da classe Interface

```
public class Interface extends JFrame {  
    // Declaração das variáveis necessárias  
    public Interface() { // Construtor  
  
        // Chamar classes UpdateQuerye ResultSetTableModel  
        // Construção do interface  
    }  
    private class thehandler implements ActionListener  
    {  
        public void actionPerformed(ActionEvent event){  
            // método para ouvir os diferentes botoes, e tentar enviar as  
            query's, bem como receber resposta. Se a query for do tipo  
            executeQuery() é a chamada a classe ResultSetTable e é devolvida uma  
            tabela correspondente à query pedida sendo impressa numa nova janela.  
            Se a query for do tipo executeUpdate() é chamada a classe inserts que  
            envia a query á base de dados.  
        }  
    }  
}
```

Estrutura da classe Main

```
public class Main extends JFrame {  
    // Declaração das variáveis necessárias  
    public Main() // Construtor  
    {  
        // Construção do interface de login  
  
        private class ouvido implements ActionListener{  
            public void actionPerformed(ActionEvent event){  
  
                // Ouvir os campos do login, utilizador e respectiva  
                password. Se forem válidos é chamado um objecto do tipo  
                Interface  
  
                new Interface (USERNAME, PASSWORD); // o objecto aceita  
                como argumento o username e password válidos. Deste modo para  
                tornar o programa executável basta mudar estas strings no Main  
                sendo depois passados pelo construtor do Interface à classe  
                Interface.  
            }  
        }  
        public static void main( String args[] )  
        {  
            new Main(); // Criar e chamar um objecto de tipo Main  
        }  
    }  
}
```

