

Análise e Concepção de Software  
Mestrado em Engenharia Informática  
Universidade do Minho

Projecto Prático de Engenharia Web - 2ª Fase

**Placar de Ideias Online**  
Modelo de Entidade e Relacionamento

Pedro Maia PG25323 (*teamleader*)  
Miguel Carvalho PG25287  
Mário Meireles A51743  
Ana Zita Sousa PG24161

17 de Janeiro de 2014

# Conteúdo

1	Introdução	3
2	Entidades	4
3	Atributos	5
4	Relações	7
5	Modelo de Entidade e Relacionamento	8
6	Código SQL	9
7	Conclusão	13

# 1 Introdução

A segunda entrega do trabalho consiste na representação do modelo de dados da nossa aplicação através do modelo de entidade e relacionamento.

O processo de desenho do modelo de dados começa a partir de um núcleo inicial, constituído pelos conceitos principais da aplicação, que vai sendo progressivamente refinado.

Assim deve começar-se por adicionar as entidades principais, especificando as propriedades de cada uma; quando já tivermos as entidades principais definidas, devemos estabelecer relações entre elas, criando associações semânticas; em seguida, devemos criar entidades de especialização para categorizar as entidades principais; por fim, devemos adicionar entidades para os utilizadores e para os grupos de utilizadores, que serão ligadas às entidades principais no sentido de expressar preferências dos utilizadores ou dos grupos, bem como conteúdos pessoais.

Como podemos verificar, o modelo de entidade e relacionamento deve portanto ser estruturado em torno das entidades principais, através de esquemas de acesso, de relações e de personalização.

O dicionário de dados, elaborado na primeira fase do projecto, representa um papel fundamental pois é através dele que obtemos toda a informação necessária para este processo.

Assim, começamos por criar as entidades principais tendo por base a definição dos dados. As propriedades das entidades terão obviamente de ser valores atômicos, sendo que propriedades mais complexas terão de ser transformadas em componentes, que são representados como novas entidades ligadas à entidade principal que lhes diz respeito.

As relações entre as entidades são obtidas a partir das relações já enumeradas na definição dos dados no dicionário de dados. No caso dos esquemas de acesso, baseamo-nos nos use cases e nos site views para analisar as diferentes formas necessárias de categorização das entidades. Os esquemas de personalização baseiam-se nos grupos de utilizadores e nas site views a que eles têm acesso.

## 2 Entidades

O dicionário de dados facilitou imenso a identificação das entidades do modelo de entidade e relacionamento. Todos os objectos identificados no dicionário foram convertidos para entidades, sendo que foram ainda surgindo novas entidades como resultado do refinamento das propriedades desses objectos (por exemplo, as *tags*).

Segue-se uma tabela que descreve as entidades do sistema.

Entidades	
Nome	Descrição
Ideia	Entidade que representa o contributo de um cidadão registado na plataforma para a sua região.
Grupo Orçamental	Entidade que representa um conjunto, limitado por um valor mínimo e um valor máximo, em que se insere uma ideia.
Categoria	Entidade que representa o tema ou área a que uma ideia pertence.
Voto	Entidade que representa a manifestação de opinião em relação a uma ideia.
Tag	Entidade que identifica o conteúdo de uma ideia.
Utilizador	Entidade que representa todos os utilizadores registados.
Grupo	Entidade que representa um grupo de utilizadores.
Module	Entidade que representa a site view associada a um grupo.

Figura 1: Descrição das entidades.

### 3 Atributos

Como no decorrer desta fase foram surgindo novas entidades, foi necessário redefinir também algumas entidades já existentes do sistema. Como consequência achamos importante apresentar novamente, neste relatório, as propriedades de cada entidade, bem como o tipo de dados de cada uma.

Atributos			
Nome das entidades	Nome dos atributos	Tipo de dados	Descrição
Utilizador	Id	Integer	Identificador único do utilizador
	Username	String	Nome através do qual o utilizador é identificado
	Password	String	Chave de acesso à plataforma, encriptada, do utilizador
	Email	String	Endereço electrónico do utilizador
	DataNascimento	Date	Data em que o utilizador nasceu
	Morada	String	Local de residência do utilizador
	Foto	String	Fotografia identificativa do utilizador
	IP	String	Endereço de IP através do qual o utilizador acede ao sistema
	DataCriação	Date	Data da criação do utilizador
	DataUltimoAcesso	Date	Data da última vez que o utilizador esteve online no sistema
	DataUltimaEdicao	Date	Data da última edição de feita sobre o perfil do utilizador
Ideia	Id	Integer	Identificador único da ideia
	Titulo	String	Ideia sintetizada
	Descrição	String	Texto com a ideia detalhada, onde a explique e defina
	DataInserção	Date	Data da criação da ideia
	DataEdição	Date	Data da última edição feita sobre alguma propriedade da ideia
	Rating	Integer	Número total de votos
	Quantia	Integer	Valor estimado necessário para concretização da ideia

Figura 2: Atributos das entidades do sistema.

Atributos			
Nome das entidades	Nome dos atributos	Tipo de dados	Descrição
Categoria	Id	Integer	Identificador único da categoria
	Título	String	Breve designação da categoria
	Descrição	String	Explicação detalhada sobre a categoria
	DataInserção	Date	Data de criação da categoria
	DataEdição	Date	Data da última edição feita sobre alguma propriedade da categoria
Grupo Orçamental	Id	Integer	Identificador único do grupo orçamental
	Título	String	Nome do grupo orçamental
	Descrição	String	Texto com detalhes sobre o grupo, onde explique as regras do mesmo
	DataInserção	Date	Data da criação do grupo orçamental
	DataEdição	Date	Data da última edição feita sobre alguma propriedade do grupo
Voto	Id	Integer	Identificador único do voto
Tag	Id	Integer	Identificador único da tag
	Nome	String	Designação da etiqueta
Grupo	Id	Integer	Identificador único do grupo de utilizadores
	NomeGrupo	String	Designação do grupo de utilizadores
Module	Id	Integer	Identificador único do módulo
	ModuleID	String	Identificador da site view
	ModuleName	String	Designação da site view

Figura 3: Atributos das entidades do sistema (continuação).

## 4 Relações

Com o objectivo de clarificar as relações entre as entidades do sistema, apresenta-se em baixo uma tabela que pretende sintetizar estas relações.

Entidade	Multiplicidade	Semântica	Multiplicidade	Entidade
Utilizador	1	<i>pertence a</i>	1	Grupo de Utilizadores
Utilizador	1	<i>vota em</i>	N	Grupo Orçamental
Utilizador	1	<i>tem</i>	N	Ideia
Utilizador	1	<i>faz</i>	N	Voto
Grupo de Utilizadores	1	<i>tem</i>	N	Utilizadores
Grupo Orçamental	1	<i>tem</i>	N	Ideia
Ideia	1	<i>pertence a</i>	1	Grupo Orçamental
Ideia	1	<i>tem</i>	N	Tag
Ideia	1	<i>tem</i>	N	Voto
Ideia	1	<i>pertence a</i>	1	Categoria
Tag	1	<i>tem</i>	N	Ideia
Categoria	1	<i>tem</i>	N	Ideia

Figura 4: Relações entre as entidades do sistema.

## 5 Modelo de Entidade e Relacionamento

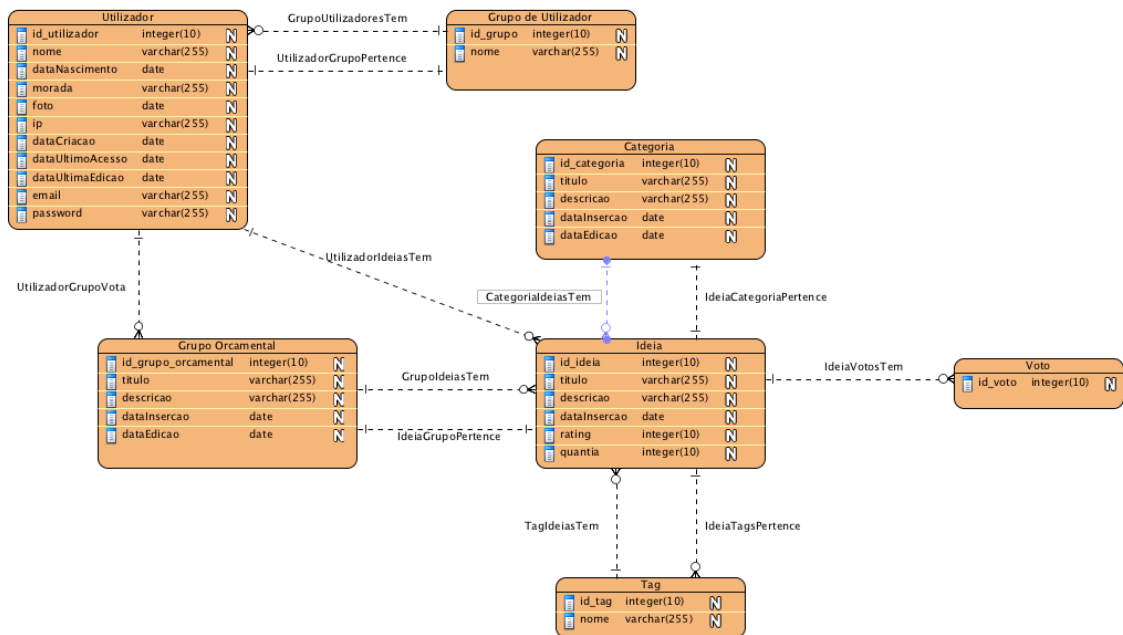


Figura 5: Protótipo do Modelo ER em Visual Paradigm, com as relações explicitamente descritas.

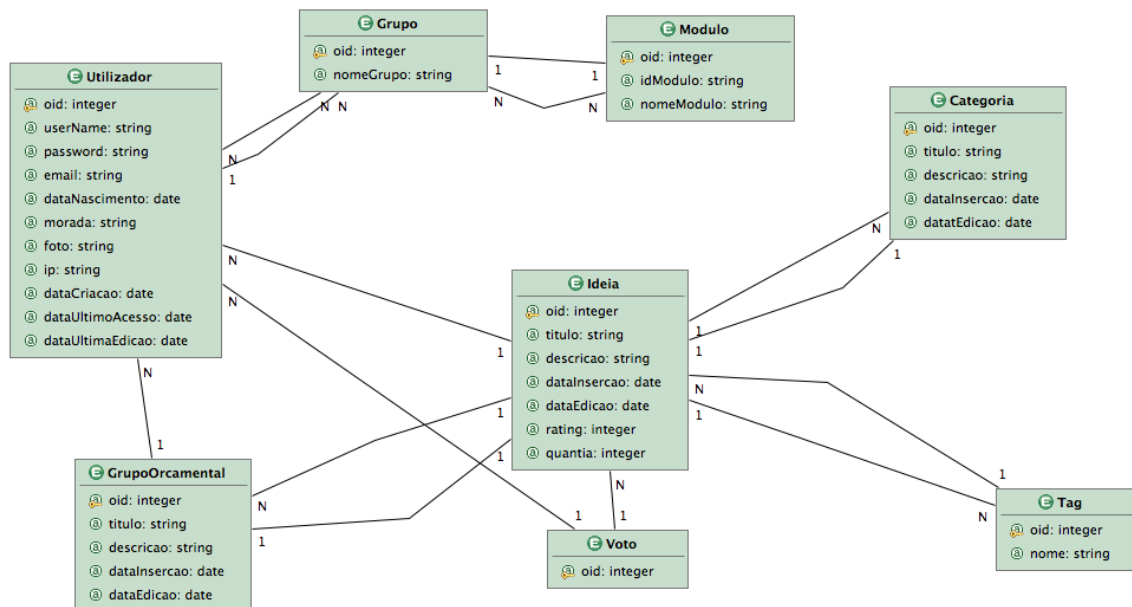


Figura 6: Modelo ER final no WebRatio.



## 6 Código SQL

O WebRatio possibilita a geração de código SQL através do modelo ER, o que facilita imenso o processo da criação da base de dados em si. Assim sendo, escolhemos o tipo de base de dados que iremos adoptar (MySQL) e o WebRatio gerou o respectivo código SQL.

```
-- Grupo [Group]
create table 'grupo' (
  'oid' integer not null,
  'nomegrupo' varchar(255),
  primary key ('oid')
);

-- Modulo [Module]
create table 'modulo' (
  'oid' integer not null,
  'idmodulo' varchar(255),
  'nomemodulo' varchar(255),
  primary key ('oid')
);

-- Utilizador [User]
create table 'utilizador' (
  'oid' integer not null,
  'username' varchar(255),
  'password' varchar(255),
  'email' varchar(255),
  'datanascimento' date,
  'morada' varchar(255),
  'foto' varchar(255),
  'ip' varchar(255),
  'datacriacao' date,
  'dataultimoacesso' date,
  'dataultimaedicao' date,
  primary key ('oid')
);

-- Tag [ent1]
create table 'tag' (
  'oid' integer not null,
  'nome' varchar(255),
  primary key ('oid')
);

-- Voto [ent2]
create table 'voto' (
  'oid' integer not null,
  primary key ('oid')
);

-- GrupoOrcamental [ent4]
create table 'grupoorcamental' (
```

```

        'oid' integer not null,
        'titulo' varchar(255),
        'descricao' varchar(255),
        'datainsercao' date,
        'dataedicao' date,
        primary key ('oid')
    );

-- Ideia [ent5]
create table 'ideia' (
    'oid' integer not null,
    'titulo' varchar(255),
    'descricao' varchar(255),
    'datainsercao' date,
    'rating' integer,
    'quantia' integer,
    'dataedicao' date,
    primary key ('oid')
);

-- Categoria [ent6]
create table 'categoria' (
    'oid' integer not null,
    'titulo' varchar(255),
    'descricao' varchar(255),
    'datainsercao' date,
    'datatedicao' date,
    primary key ('oid')
);

-- Group_DefaultModule [Group2DefaultModule_DefaultModule2Group]
alter table 'modulo' add column 'grupo_oid' integer;
alter table 'modulo' add index fk_modulo_grupo ('grupo_oid'), add constraint
fk_modulo_grupo foreign key ('grupo_oid') references 'grupo' ('oid');

-- Group_Module [Group2Module_Module2Group]
create table 'group_module' (
    'grupo_oid' integer not null,
    'modulo_oid' integer not null,
    primary key ('grupo_oid', 'modulo_oid')
);
alter table 'group_module' add index fk_group_module_grupo ('grupo_oid'),
add constraint fk_group_module_grupo foreign key ('grupo_oid')
references 'grupo' ('oid');
alter table 'group_module' add index fk_group_module_modulo ('modulo_oid'),
add constraint fk_group_module_modulo foreign key ('modulo_oid')
references 'modulo' ('oid');

-- User_DefaultGroup [User2DefaultGroup_DefaultGroup2User]
alter table 'utilizador' add column 'grupo_oid' integer;
alter table 'utilizador' add index fk_utilizador_grupo ('grupo_oid'),
add constraint fk_utilizador_grupo foreign key ('grupo_oid')
references 'grupo' ('oid');

-- User_Group [User2Group_Group2User]
create table 'user_group' (

```

```

        'utilizador_oid' integer not null,
        'grupo_oid' integer not null,
        primary key ('utilizador_oid', 'grupo_oid')
    );
alter table 'user_group' add index fk_user_group_utilizador ('utilizador_oid'),
add constraint fk_user_group_utilizador foreign key ('utilizador_oid')
references 'utilizador' ('oid');
alter table 'user_group' add index fk_user_group_grupo ('grupo_oid'),
add constraint fk_user_group_grupo foreign key ('grupo_oid')
references 'grupo' ('oid');

-- Voto_Ideia [rel1]
alter table 'voto' add column 'ideia_oid' integer;
alter table 'voto' add index fk_voto_ideia ('ideia_oid'),
add constraint fk_voto_ideia foreign key ('ideia_oid')
references 'ideia' ('oid');

-- Utilizador_Ideia [rel10]
alter table 'ideia' add column 'utilizador_oid' integer;
alter table 'ideia' add index fk_ideia_utilizador ('utilizador_oid'),
add constraint fk_ideia_utilizador foreign key ('utilizador_oid')
references 'utilizador' ('oid');

-- Tag_Ideia_2 [rel11]
alter table 'ideia' add column 'tag_oid' integer;
alter table 'ideia' add index fk_ideia_tag ('tag_oid'),
add constraint fk_ideia_tag foreign key ('tag_oid')
references 'tag' ('oid');

-- Tag_Ideia [rel2]
alter table 'tag' add column 'ideia_oid' integer;
alter table 'tag' add index fk_tag_ideia ('ideia_oid'),
add constraint fk_tag_ideia foreign key ('ideia_oid')
references 'ideia' ('oid');

-- Categoria_Ideia [rel3]
alter table 'ideia' add column 'categoria_oid' integer;
alter table 'ideia' add index fk_ideia_categoria ('categoria_oid'),
add constraint fk_ideia_categoria foreign key ('categoria_oid')
references 'categoria' ('oid');

-- Utilizador_Voto [rel4]
alter table 'voto' add column 'utilizador_oid' integer;
alter table 'voto' add index fk_voto_utilizador ('utilizador_oid'),
add constraint fk_voto_utilizador foreign key ('utilizador_oid')
references 'utilizador' ('oid');

-- GrupoOrcamental_Utilizador [rel6]
alter table 'grupoorcamental' add column 'utilizador_oid' integer;
alter table 'grupoorcamental' add index fk_grupoorcamental_utilizador
('utilizador_oid'),
add constraint fk_grupoorcamental_utilizador foreign key ('utilizador_oid')
references 'utilizador' ('oid');

```

```
-- GrupoOrcamental_Ideia [rel7]
alter table 'ideia' add column 'grupoorcamental_oid' integer;
alter table 'ideia' add index fk_ideia_grupoorcamental ('grupoorcamental_oid'),
add constraint fk_ideia_grupoorcamental foreign key ('grupoorcamental_oid')
references 'grupoorcamental' ('oid');

-- GrupoOrcamental_Ideia_2 [rel8]
alter table 'ideia' add column 'grupoorcamental_oid_2' integer;
alter table 'ideia' add index fk_ideia_grupoorcamental_2
('grupoorcamental_oid_2'),
add constraint fk_ideia_grupoorcamental_2 foreign key ('grupoorcamental_oid_2')
references 'grupoorcamental' ('oid');

-- Categoria_Ideia_2 [rel9]
alter table 'ideia' add column 'categoria_oid_2' integer;
alter table 'ideia' add index fk_ideia_categoria_2 ('categoria_oid_2'),
add constraint fk_ideia_categoria_2 foreign key ('categoria_oid_2')
references 'categoria' ('oid');
```

## 7 Conclusão

Reconhecemos a importância do documento de requisitos elaborado na primeira fase, uma vez que se revelou extremamente útil na concepção do Modelo de Entidade e Relacionamento.

A ferramenta WebRatio permitiu-nos obter fácil e rapidamente o código SQL do modelo que idealizamos.

Fazemos um balanço positivo desta fase do trabalho, visto que aplicamos conhecimentos das aulas e começamos a dar os primeiros passos na modelação da nossa aplicação no WebRatio.