



Universidade do Minho

Mestrado em Engenharia Informática – 1º Ano

Análise e Conceção de Software

Engenharia de Aplicações Web

Modelo de Dados

Elementos do Grupo:

Luís André Mota Araújo - PG19835

Rafael Eduardo de Carvalho Pereira - PG19753 (Gestor)

Rui Alexandre Afonso Pereira - PG20201

Braga, novembro de 2011

Índice

Introdução	3
Metodologia	4
Identificação das Entidades	4
Identificação dos Atributos	5
Identificação dos Relacionamentos	6
Construção do Modelo de Dados	7
Conclusão	8
Anexos.....	9
Anexo 1 Modelo conceptual Peter Chen.....	9
Anexo 2 Modelo conceptual ER (WebRatio)	9
Anexo 3 Código SQL	11

Introdução

Construir aplicações Web implica muitas vezes a modelação e implementação da camada de dados, a não ser que a mesma seja estática ou utilize o sistema de ficheiros para serializar dados de que necessita.

Neste relatório procuramos explicar o processo usado para identificar a informação que terá que ser serializada pela aplicação Web em desenvolvimento, identificando as duas entidades, os atributos das entidades e os relacionamentos entre as entidades.

O facto de ter sido realizada uma análise de requisitos de forma exaustiva facilitou muito esta fase, pois o dicionário de dados desenvolvidos já continha algumas entidades, atributos e relações evidenciadas.

Assim, nesta fase, modelamos conceptualmente, usando duas notações diferentes, os dados que a aplicação Web utilizará.

Metodologia

Uma abordagem bem estruturada é essencial para poupar tempo de desenvolvimento e obter melhores resultados.

Neste caso aproveitar o trabalho realizado na fase de análise de requisitos foi crucial e era algo lógico, pois a verdade é que no dicionário de dados desenvolvido estavam identificadas as entidades, os atributos e os relacionamentos, embora não explicitamente.

Assim, a metodologia seguida foi a partir dos objetos, atributos e relacionamentos detetados na fase anterior, transformando num modelo conceptual usando a notação Peter Chen, uma linguagem gráfica especializada na modelação *ER*. O modelo conceptual foi posteriormente inserido no *Web Ratio*, algo exigido no módulo em causa.

Identificação das Entidades

Partindo do Dicionário de Dados existem entidades que são identificadas automaticamente, como Utilizador, Evento, Comentário, uma vez que foram os objetos detetados.

Existem algumas entidades que não são detetadas com tanta facilidade, pois as mesmas estão implícitas em relações, como é o caso do Evento Favorito, ou dizem respeito ao facto de ser requisito devido a ser uma aplicação Web, como é o caso da entidade Login que, como é óbvio, auxiliará a autenticação.

Na tabela introduzida abaixo são descritas as entidades do sistema individualmente.

Entidades	
Nome	Descrição
Evento	Entidade principal do sistema que representa todas as atividades do mundo físico que podem ser divulgadas no sistema.
Utilizador	Entidade que representa todos os utilizadores registados do sistema.
Comentário	Entidade que representa todas as opiniões que os utilizadores podem deixar registadas relativamente aos eventos divulgados no sistema.
Newsletter	Entidade que representa as edições de Email's semanais em que podem ser divulgados eventos presentes no sistema, mediante pré-pagamento. Esta é a entidade principal do modelo de negócio.
Tipo	A entidade tipo é auxiliar permitindo identificar os eventos com um nome mais simples, tornando as pesquisas em potência mais eficientes.
Login	Entidade que representa a informação que o sistema precisa de guardar na autenticação de um utilizador registado.

Identificação dos Atributos

A identificação dos atributos foi bastante simples, pois estes estavam parcialmente identificados no dicionário de dados. No entanto, a este nível, não faltavam apenas os atributos, que advêm do facto de se tratar do desenvolvimento de uma aplicação Web, também foram identificados alguns requisitos que derivam das exigências do modelo relacional, como são os casos das chaves primárias

Atributos			
Nome das Entidades	Nome dos Atributos	Tipo de Dados	Descrição
Evento	<u>NR</u>	Inteiro	Identificador único por cada Evento
	Título	String	Frase sugestiva relativa a um Evento
	Descrição	String	Texto que descreve o conteúdo de um Evento
	Latitude	Float	Longitude geográfica do Evento
	Longitude	Float	Latitude geográfica do Evento
	DataInício	DateTime	Início do Evento
	DataFim	DateTime	Final do Evento
Utilizador	<u>NR</u>	Inteiro	Identificador único por utilizador
	Nome	String	Nome pelo qual o utilizador gosta de ser identificado
	Email	String	Endereço de correio eletrónico
	Hpassword	String	Password encriptada
	Latitude	Float	Latitude predefinida
	Longitude	Float	Longitude predefinida
	Foto	String	Endereço de uma fotografia no servidor
Comentário	<u>NR</u>	Integer	Identificador do Comentário no sistema
	Título	String	Título atribuído pelo utilizador ao comentário
	Texto	String	Corpo do comentário do autor
	DateTime	DateTime	Data e hora em que o comentário entrou no sistema
Newsletter	<u>NR</u>	Integer	Identificador único de uma Newsletter
	Título	String	Título sugestivo da Newsletter
	DateTimeEnvio	DateTime	Tuplo data-hora de envio da newsletter
Tipo	<u>NR</u>	Integer	Identificador único por cada tipo de evento existente
	Nome	String	Frase ou palavra que cataloga eventos
Login	<u>NR</u>	Integer	Identificador de uma autenticação no sistema
	StringSessão	String	String de sessão que identifica o login e o utilizador
	DateTimeLogin	DateTime	Data e hora da autenticação
	DateTimeExpira	DateTime	Data e hora em que a autenticação expira no caso de haver cookie
	Cookie	Boolean	Booleano que indica se utilizador pretende manter a sessão iniciada

Foram identificados e descritos os atributos, sendo o seu domínio parcialmente definido a partir dos tipos de dados, que ficaram desde já identificados. Os atributos estão na tabela acima organizados por entidade, estando a chave primária de cada entidade identificada com o sublinhado. A explicação adicional dos atributos segue na última coluna da tabela.

Os tipos de dados definidos foram cruciais para a construção do Modelo de Dados, daí terem sido utilizados os tipos acima. Se fosse para criar uma base de dados, os atributos teriam sempre a eles associados os seus tamanhos.

Identificação dos Relacionamentos

Depois de identificadas as entidades e os atributos as relações foram algo simples de identificar, uma vez que as mesmas foram descritas no dicionário de dados e já tínhamos inclusive introduzido alguma semântica para as mesmas.

Foi decidido documentar os relacionamentos referindo para cada relação as duas entidades envolvidas, as multiplicidades da relação junto de cada entidade e a semântica da relação, tornando o relacionamento compreensível e legível. Posteriormente os relacionamentos foram alvo de uma descrição textual adicional, estando estes referenciados a partir do seu ID.

Relações					
ID	Entidade	Multiplicidade	Semântica	Multiplicidade	Entidade
1	Utilizador	1	cria	N	Evento
2	Utilizador	N	marca como favoritos	M	Evento
3	Utilizador	1	escreve	N	Comentário
4	Utilizador	1	tem	N	Login
5	Evento	1	tem	N	Comentário
6	Tipo	1	cataloga	N	Evento
7	Newsletter	1	tem	N	Evento

Descrição das Relações	
ID	Descrição
1	Um utilizador registado pode criar muitos eventos.
2	Um utilizador pode marcar como favoritos os eventos do sistema.
3	Um utilizador pode escrever comentários acerca dos eventos do sistema.
4	Um utilizador pode fazer várias autenticações.
5	Um utilizador tem comentários.
6	Um evento tem um tipo que o resume.
7	As newsletters publicitam vários eventos.

Construção do Modelo de Dados

A construção do Modelo de Dados no *WebRatio* foi muito simples, pois toda a informação estava já devidamente identificada, sendo por isso simples utilização da interface gráfica da aplicação.

A partir deste modelo de dados será possível criar classes *Java* que permitirão interagir com as entidades do projeto, realizando várias operações de forma simples, sem ser necessário escrever qualquer linha de código. É inclusive possível estabelecer uma ligação com um sistema de gestão de base de dados, criando a camada de persistência de forma automática.

Nota: A imagem do modelo de dados desenvolvido segue no anexo 2, não tendo sido aqui inserida para manter o relatório o mais breve possível.

Conclusão

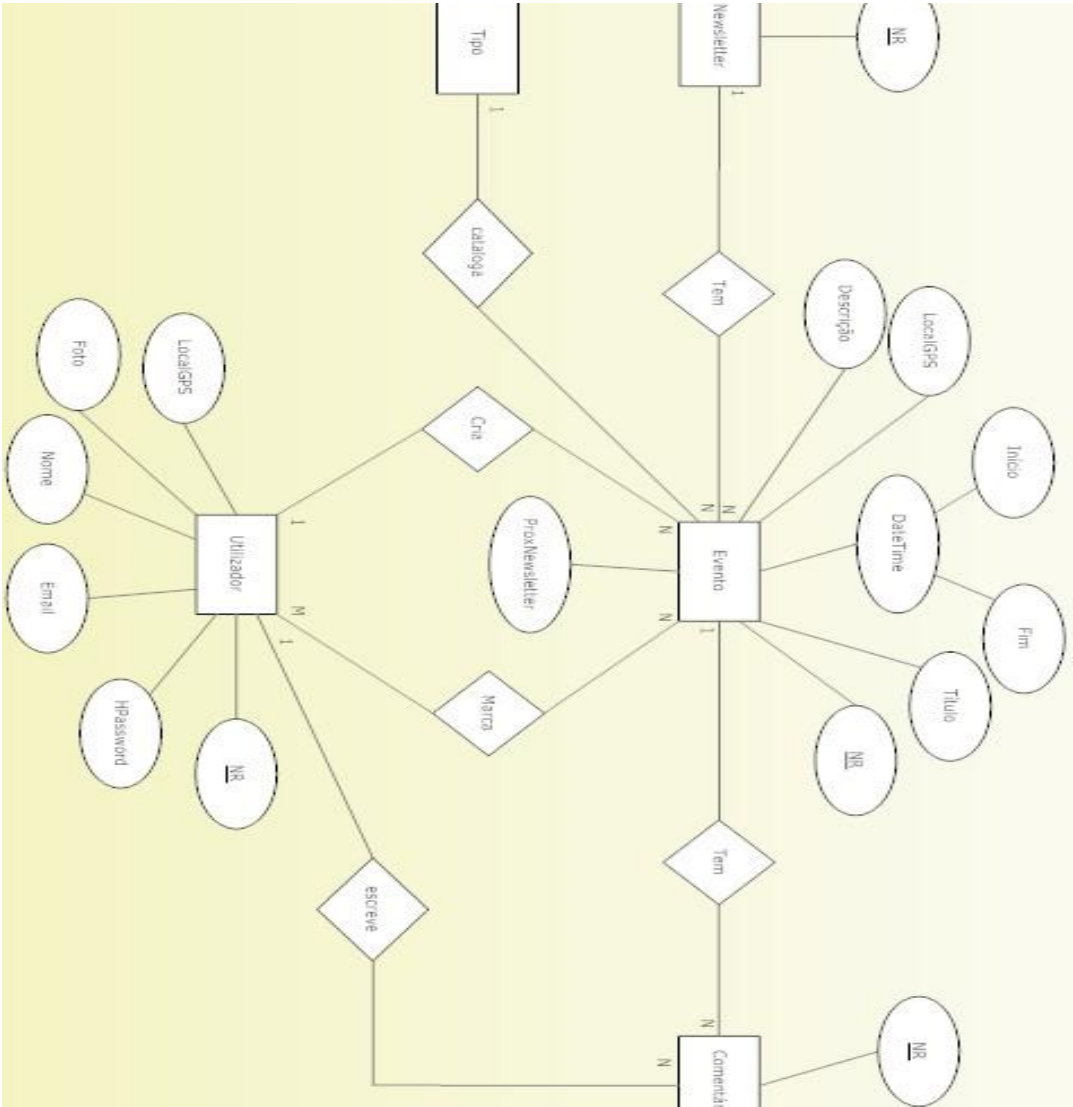
Ao longo da realização desta etapa do projeto, utilizando o dicionário de dados, dicionário que foi especificado na fase de requisitos do projeto, identificar as entidades, os atributos e os relacionamentos entre as entidades.

Posteriormente, utilizando ferramentas como o Web Ratio, foi desenhado o Modelo de Dados. Com o uso do *WebRatio* foi possível gerar o código SQL que nos permitirá, posteriormente, construir as tabelas relacionais da nossa base de dados. Tal é necessário porque terá que se desenvolver o projeto selecionado em *PHP* e *MySQL*, conforme foi acordado com os docentes, ora o *WebRatio* é um *software* que gera uma aplicação em *Java*, pelo que é melhor desenvolver paralelamente o projeto em *PHP*.

A construção do Modelo de Dados e a documentação do mesmo foram atividades muito interessantes, principalmente porque este trabalho de modelação e especificação permitiram-nos realmente caminhar na implementação da aplicação, uma vez que foram utilizadas ferramentas que a partir do modelo geram realmente partes do produto. Tal é algo que se enquadra num módulo de Engenharia Web e faz com que a atividade de modelar não pareça um desperdício.

Anexos

Anexo 1 Modelo conceptual Peter Chen



Anexo 2 Modelo de Dados (WebRatio)

Anexo 3 Código SQL

Depois de desenvolver o Modelo de Dados, foi gerada a base de dados. Uma vez que foi acordado desenvolver a aplicação usando linguagem PHP e o SGBD MySQL, geramos o código SQL de geração da base de dados compatível com o MySQL.

```
/*=====*/
/* Database name: EVENTOS */
/* DBMS name: MySQL 5.0 */
/* Created on: 21-11-2011 18:19:22 */
/*=====*/
```

drop database if exists EVENTOS;

```
/*=====*/
/* Database: EVENTOS */
/*=====*/
create database EVENTOS;
```

use EVENTOS;

```
/*=====*/
/* Table: COMENTARIOS */
/*=====*/
create table COMENTARIOS
(
    IDCOMENTARIO int not null,
    IDUTILIZADOR int not null,
    IDEVENTO int not null,
    TITULO text not null,
    TEXT text not null,
    DATEHORA datetime not null,
    primary key (IDCOMENTARIO)
);
```

```
/*=====*/
/* Table: EVENTOS */
/*=====*/
create table EVENTOS
(
    IDEVENTO int not null,
    IDTIPO int not null,
    IDUTILIZADOR int not null,
    TITULO text not null,
    DESCRICAO text not null,
    DATAHORAFIM datetime not null,
    DATAHORAINICIO datetime not null,
    GPSLAT float not null,
    GPSLON float not null,
    CONFIRMARDO bool not null,
    STRCONF char(50) not null,
```

```

        primary key (IDEVENTO)
    );

/*=====*/
/* Table: EVENTOSNEWSLETTER */
/*=====*/
create table EVENTOSNEWSLETTER
(
    IDPROXNEWSLWTTER int not null,
    IDEVENTO          int not null,
    IDNEWSLETTER      int not null,
    primary key (IDPROXNEWSLWTTER)
);

/*=====*/
/* Table: FAVORITOS */
/*=====*/
create table FAVORITOS
(
    IDFAVORITO      int not null,
    IDUTILIZADOR    int not null,
    IDEVENTO        int not null,
    DATAHORA       datetime not null,
    primary key (IDFAVORITO)
);

/*=====*/
/* Table: LOGINS */
/*=====*/
create table LOGINS
(
    IDLOGIN         int not null,
    IDUTILIZADOR    int not null,
    STRSESSAO       char(50) not null,
    EXPIRE          datetime not null,
    COOKIE          bool not null,
    primary key (IDLOGIN)
);

/*=====*/
/* Table: NEWSLETTER */
/*=====*/
create table NEWSLETTER
(
    IDNEWSLETTER    int not null,
    TITULO          text not null,
    DATAHORA       datetime not null,
    primary key (IDNEWSLETTER)
);

/*=====*/
/* Table: TIPOS */
/*=====*/

```

```

/*=====*/
create table TIPOS
(
    IDTIPO          int not null,
    NOME            char(60),
    primary key (IDTIPO)
);

/*=====*/
/* Table: UTILIZADORES */
/*=====*/
create table UTILIZADORES
(
    IDUTILIZADOR    int not null,
    NOME            char(60),
    EMAIL          char(256) not null,
    FOTO           text,
    GPSLAT         float not null,
    GPSLON         float not null,
    HPASSWORD      char(25) not null,
    CRIADO         datetime not null,
    primary key (IDUTILIZADOR)
);

alter table COMENTARIOS add constraint FK_EVETOSSAOCOMENTADOS foreign key
(IDEVENTO)
references EVENTOS (IDEVENTO) on delete restrict on update restrict;

alter table COMENTARIOS add constraint FK_UTILIZADORESCREVECOMENTARIO foreign key
(IDUTILIZADOR)
references UTILIZADORES (IDUTILIZADOR) on delete restrict on update restrict;

alter table EVENTOS add constraint FK_EVENTOTEMTIPO foreign key (IDTIPO)
references TIPOS (IDTIPO) on delete restrict on update restrict;

alter table EVENTOS add constraint FK_UTILIZADORCRIAEVENTOS foreign key (IDUTILIZADOR)
references UTILIZADORES (IDUTILIZADOR) on delete restrict on update restrict;

alter table EVENTOSNEWSLETTER add constraint FK_EVENTOSPUBLICITADOSNEWSLETTER
foreign key (IDEVENTO)
references EVENTOS (IDEVENTO) on delete restrict on update restrict;

alter table EVENTOSNEWSLETTER add constraint FK_NEWSLETTERTEMEVENTOS foreign key
(IDNEWSLETTER)
references NEWSLETTER (IDNEWSLETTER) on delete restrict on update restrict;

alter table FAVORITOS add constraint FK_EVENTOSSAOMARCADOS foreign key (IDEVENTO)
references EVENTOS (IDEVENTO) on delete restrict on update restrict;

alter table FAVORITOS add constraint FK_UTILIZADORTEMFAVORITOS foreign key
(IDUTILIZADOR)
references UTILIZADORES (IDUTILIZADOR) on delete restrict on update restrict;

```

```
alter table LOGINS add constraint FK_UTILIZADORESLOGINS foreign key (IDUTILIZADOR)  
references UTILIZADORES (IDUTILIZADOR) on delete restrict on update restrict;
```

Texto escrito conforme o Acordo Ortográfico - convertido pelo Lince.