UMA IMPLEMENTAÇÃO DA ÁLGEBRA PARA PONTOS MÓVEIS USANDO POSTGRESQL

Valtania Ferreira da SILVA (1); Fausto Véras Maranhão AYRES (2)
(1) Polícia Militar do Estado da Paraíba, e-mail: fvaltania@hotmail.com
(2) Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), e-mail: fausto.ayres@ifpb.edu.br

RESUMO

Atualmente, vem crescendo o número de aplicações que envolvem o monitoramento de pontos móveis – entidades cuja localização geográfica muda continuamente ao longo do tempo (p.ex: movimento de um carro). Em tais aplicações é necessário modelar pontos móveis e armazená-los num banco de dados. Uma alternativa para isso é a utilização da Álgebra para Pontos Móveis, criada por Ralf Guting, que modela pontos móveis através de tipos de dados espaço-temporais permitindo operações sobre pontos móveis, tais como o cálculo da trajetória de um ponto móvel e o cálculo da localização deste ponto num determinado instante. Essa álgebra foi implementada no Sistema Gerenciador de Banco de Dados (SGBD) SECONDO, permitindo a construção de bancos de dados para pontos móveis. O presente trabalho seguiu uma outra abordagem que consistiu na implementação dessa álgebra através do SGBD PostgreSQL e de sua extensão PostGIS, produzindo um conjunto de tabelas relacionadas e de funções armazenadas num banco de dados relacional e oferecendo o suporte ao desenvolvimento de aplicações voltadas para pontos móveis.

Palavras-chave: álgebra para pontos móveis, banco de dados espaço-temporais, geoprocessamento

1 INTRODUÇÃO

O advento das Geotecnologias, em particular, do Sistema Global de Posicionamento (GPS) e das redes sem fio, motivou o aparecimento de aplicações que necessitam do armazenamento e manipulação de informações relativas a pontos móveis (*moving points*) - entidades que se movem continuamente ao longo do tempo. Como exemplos de tais aplicações têm-se os sistemas baseados em localização (PELEKIS, 2006) que necessitam lidar com as posições geográficas ocupadas pelos pontos móveis, em tempo real ou não, e com o histórico dos seus movimentos. Para tanto, é necessário utilizar um banco de dados que suporte o armazenamento e a manipulação de pontos móveis.

Neste contexto, uma questão importante é como modelar e armazenar pontos móveis. Algumas soluções para modelar pontos móveis são encontradas em (SISTLA et al, 1997, YI et al, 2002). Uma outra solução é a Álgebra para Pontos Móveis – um subconjunto da Álgebra para Objetos Móveis criada por Guting et al (2000), a qual permite modelar a abstração de pontos móveis através de tipos de dados espaciais, temporais e espaço-temporais, para se obter informações sobre seus movimentos, tais como: a trajetória do movimento do ponto móvel; a duração do movimento do ponto móvel; a distância entre dois pontos móveis; a localização do ponto móvel num determinado instante e o instante que o ponto móvel atingiu uma determinada localização. Esta álgebra foi implementada no ambiente SECONDO (GUTING et al, 2005) – um Sistema Gerenciador de Banco de Dados (SGBD) extensível e modular que permite a implementação de diferentes álgebras.

O presente trabalho apresenta uma implementação da Álgebra para Pontos Móveis através do SGBD PostgreSQL (2010) e de sua extensão espacial PostGIS (2010). A abordagem utilizada consiste na implementação dos principais tipos de dados espaço-temporais da álgebra num banco de dados relacional, produzindo um conjunto de tabelas e funções armazenadas. A escolha deste SGBD deve-se ao fato do mesmo oferecer as operações espaciais e temporais requeridas na implementação das operações espaço-temporais, além de ser um software livre. Como resultado, obtém-se um banco de dados para armazenar pontos móveis, oferecendo um suporte para o desenvolvimento de aplicações que os utilizam. Até o presente momento não encontramos na literatura nenhuma implementação dessa álgebra, utilizando tal abordagem.

2 ÁLGEBRA PARA PONTOS MÓVEIS

A Álgebra para Pontos Móveis está centrada no conceito de trajetória, obtida a partir do movimento contínuo, no tempo e no espaço, de um ponto móvel. Essa álgebra é parte integrante da Álgebra para Objetos Móveis (GUTING et al, 2000), a qual é utilizada para modelar tanto pontos móveis (*moving points*) quanto regiões móveis (*moving regions*). O uso de regiões móveis está fora do escopo deste trabalho.

A álgebra representa o movimento de um ponto móvel de forma discreta, utilizando instantes de tempos associados às unidades de movimento. A Figura 1 ilustra a representação discreta do movimento de um ponto móvel no plano *xy* em função do tempo. Para se calcular a trajetória de um ponto móvel basta se projetar o seu movimento no plano espacial, resultando numa linha.

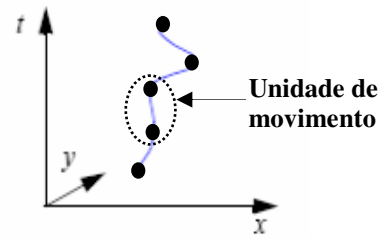

Figura 1 - Representação discreta do movimento do ponto móvel

A álgebra modela os pontos móveis através de tipos de dados espaço-temporais, os quais se baseiam em tipos espaciais e temporais (LEMA et al, 2003). Os tipos de dados espaciais são *Point* e *Line*, onde *Point* é definido por uma coordenada no plano espacial (*xy* ou *xyz*) e *Line* é definido como um conjunto de elementos do tipo *Point*. Exemplos de operações espaciais da álgebra são: *length(Line), distance(Line,Line)* e *intercection(Line, Line).*

Os tipos de dados temporais são *Instant* e *Interval*, compreendendo, respectivamente um instante e um intervalo de tempo, onde *Interval* é composto por dois *Instant* (instante inicial e instante final). Exemplos de operações espaciais da álgebra são: *before(Interval, Interval)* e *contains(Interval*, *Interval).*

Os tipos de dados espaço-temporais são *UPoint* e *MPoint*. O tipo *UPoint* define uma unidade de movimento de um ponto móvel num intervalo de tempo, a qual é composta por um ponto inicial, um ponto final e um intervalo de tempo associado ao deslocamento entre esses dois pontos. O tipo *MPoint* define um conjunto de ocorrências do tipo *UPoint*. Logo, o tipo *MPoint* representa todo o movimento de um ponto móvel. A Tabela 1 lista as principais operações envolvendo os tipos espaço-temporais:

Tabela 1 – Principais operações espaço-temporais da álgebra

Operação	Resultado	Descrição
trajectory(MPoint)	Line	obtém a trajetória do ponto móvel
locations(MPoint)	Points	obtém todas as posições do ponto móvel
mdistance(MPoint,MPoint)	MReal[1]	obtém as distâncias entre dois pontos móveis
mduration(MPoint)	Duration	obtém a duração do movimento de um ponto móvel
atposition(MPoint, Instant)	Instant	obtém a posição do ponto móvel num dado instante
atinstant(MPoint, Point)	Point	obtém o instante em que ponto móvel passou pela posição

3 MATERIAIS E MÉTODOS

A metodologia adotada no desenvolvimento deste trabalho seguiu os seguintes passos:

1. Elaboração do modelo conceitual do banco de dados, para modelar os tipos de dados espaço-temporais da álgebra;

[1] MReal (*moving real*) é um conjunto de números reais

2. Realização do mapeamento do modelo conceitual para o modelo lógico, resultando na criação das tabelas do banco de dados e da funções armazenadas no banco de dados;
3. Teste do banco de dados, consistindo da inserção de dados de pontos móveis e de consultas utilizando as funções armazenadas. Será utilizado um cenário para teste com três pontos móveis, cujos movimentos possuem trajetórias diferentes, onde a trajetória do primeiro ponto móvel é perpendicular às trajetórias dos demais pontos, que, por sua vez, são paralelas entre si.

Os *softwares* utilizados no desenvolvimento deste trabalho foram:

- O Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL (2010), na versão 8.3, e de sua extensão espacial PostGIS (2010), para criação das tabelas e das funções armazenadas do banco de dados proposto. Para a criação das funções utilizou-se do SGBD: a linguagem procedural *plpgsql,* os operadores temporais e as funções espaciais do PostGIS (p.ex: *length2d, makeline*, *addpoint,, x, y* e *astext*). Para postar os comandos SQL, utilizou-se o utilitário PgAdminIII.
- O Sistema de Informação Geográfica (SIG) Jump (2010), na versão 1.1, para visualizar os resultados das consultas espaciais sobre o banco de dados criado.

4 RESULTADOS E DISCUSSÕES

Nesta seção, são apresentados os resultados produzidos em cada etapa do desenvolvimento deste trabalho, bem como os respectivos comentários.

4.1 Modelo Conceitual do Banco de Dados

A modelagem conceitual do banco de dados foi feita a partir dos principais tipos de dados espaço-temporais *MPoint*, *UPoint*, *Point*, *Interval* e *Instant* (descritos na Seção 2), resultando no Diagrama de Entidade-Relacionamento apresentado na Figura X, o qual é composto por entidades homônimas aos tipos algébricos. Sendo assim, a entidade MPOINT representa a abstração de ponto móvel, a entidade UPOINT representa uma unidade de deslocamento de um ponto móvel num intervalo de tempo definido pela entidade INTERVAL, a qual se relaciona com duas ocorrências da entidade INSTANT, compreendendo os instantes inicial e final da unidade. Além disso, a entidade UPOINT está relacionada com duas ocorrências da entidade POINT, compreendendo as localizações inicial e final da unidade. .

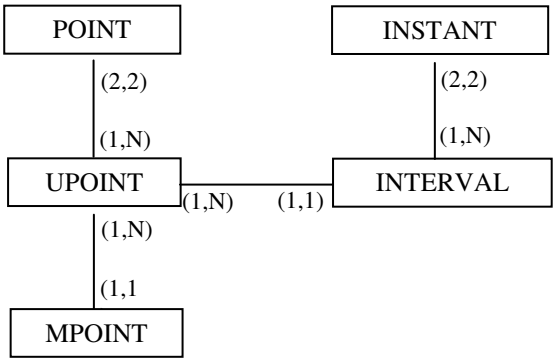

Figura 3 – Modelo conceitual do banco de dados

4.2 Modelo Lógico do Banco de Dados

Para cada entidade do modelo conceitual apresentado na seção anterior, foi criada uma tabela no banco de dados utilizando os seguintes comandos escritos na linguagem SQL:

```
CREATE TABLE TB_POINT  (
    nu_point  int primary key);
SELECT AddGeometryColumn ('','TB_POINT','point_geom',-1,'POINT',2);
CREATE TABLE TB_INSTANT (
    nu_instant int not null primary key,
    ts_instant timestamp without time zone  not null);
```

```
CREATE TABLE TB_INTERVAL (
    nu_interval int not null primary key,
    nu_instant_1 int,
    nu_instant_2 int,
    constraint tb_interval_inst1_fk foreign key (nu_instant_1) references tb_instant (nu_instant),
    constraint tb_interval_inst2_fk foreign key (nu_instant_2) references tb_instant (nu_instant));
CREATE TABLE TB_UPOINT (
    nu_upoint    int not null primary key ,
    nu_point_1    int,
    nu_point_2    int,
    nu_interval   int,
    constraint tb_upoint_point1_fk  foreign key (nu_point_1)  references tb_point (nu_point),
    constraint tb_upoint_point2_fk  foreign key (nu_point_2)  references tb_point (nu_point),
    constraint tb_upoint_interval_fk foreign key (nu_interval) references tb_interval (nu_interval));
CREATE TABLE TB_MPOINT (
    nu_mpoint  int,
    nu_upoint   int,
    constraint tb_mpoint_nu_mpoint_pk primary key (nu_mpoint, nu_upoint),
    constraint tb_mpoint_upoint_fk foreign key (nu_upoint) references tb_upoint (nu_upoint));
```

Neste contexto, é importante observar que a tabela TB_Upoint referencia as tabelas TB_Interval e TB_Point, e a tabela TB_MPoint referencia a tabela TB_Upoint, conforme especificado no modelo conceitual do banco de dados. Utiliza-se a função nativa a*ddgeometrycolumn* para incluir, na tabela TB_Point, a coluna espacial point_geom, para armazenar pontos geográficos no espaço 2D, usando um sistema de coordenadas local (-1). A tabela TB_Mpoint possui uma chave primária concatenada, não permitindo a inclusão duplicada de uma mesma unidade de movimento (UPoint) para um mesmo ponto móvel (MPoint).

As operações algébricas descritas na Seção 2 foram implementadas como funções armazenadas no banco de dados, identificadas, respectivamente, por: F_MTrajectory, F_MPoint, F_MDistance, F_MDuration, F_MPosition e F_MInstant. A implementação da função F_MTrajectory, na linguagem plpg/sql, é apresentada a seguir.

```
CREATE FUNCTION F_MTrajectory(integer) returns geometry as $$
declare  registro  record;  linha geometry;  cont integer := 0;
begin
  select m.nu_mpoint, p1.point_geom as ponto1_geom, p2.point_geom  as  ponto2_geom
  from tb_mpoint m, tb_upoint u, tb_point p1, tb_point p2
  where m.nu_mpoint= $1and m.nu_upoint=u.nu_upoint and u.nu_point_1=p1.nu_point and
        u.nu_point_2=p2.nu_point
  order by m.nu_upoint
  loop
    cont = cont + 1;
    if cont = 1
        then  linha := (makeline(registro.ponto1_geom, registro.ponto2_geom));
        else   linha := (addpoint(linha, registro.ponto2_geom));
    end if;
  end loop;
  return  linha;
end;
$$ language plpgsql;
```

A Função F_MTrajectory recebe, como parâmetro, o identificador do ponto móvel (representado por $1) e retorna uma geometria do tipo *linestring* (representado pela variável linha), formada por todos os pontos geográficos do respectivo ponto móvel, obtidos através da junção das tabelas TB_Point, TB_UPoint e TB_MPoint. Para construir essa geometria, são utilizadas duas funções do PostGIS*: makeline e addpoint.*. A função *makeline* é usada para criar o *linestring* com os dois pontos primeiros pontos do ponto móvel e a função *addpoint* é usada para adicionar ao *linestring* os demais pontos.

4.3 Teste do Banco de Dados

A Figura 4 ilustra o cenário de teste, contendo as trajetórias e os sentidos dos movimentos dos três pontos móveis, onde: (i) a seqüência de movimentos de cada ponto móvel é identificada por números (1, 2, etc.), para uma melhor visualização, e inicia-se no instante 10:02:00; (ii) o intervalo entre cada movimento é a mesma e é igual 10 minutos e (iii) a distância entre cada movimento é constante, porém ela varia entre os pontos móveis (o primeiro usa distância=3, o segundo usa distância=2 e o terceiro usa distancia=4). Como exemplo disso, o ponto móvel 1 termina a sua seqüência de 4 movimentos no instante 10:42:00, contendo 5 pontos geográficos, totalizando uma distância de 12.

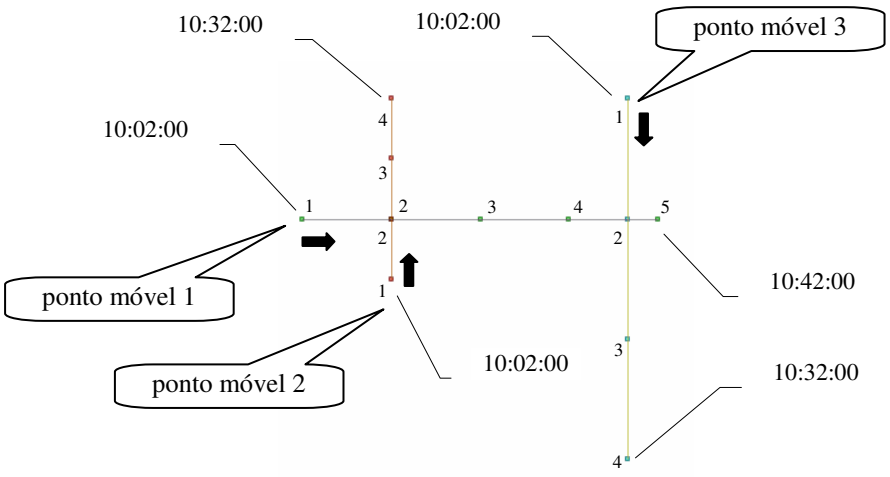

Figura 4 – Cenário de teste com três pontos móveis

A carga do banco de dados de teste foi feita através de comandos SQL, exemplificados por*:*

```
INSERT into tb_point (nu_point, point_geom) values (1, geometryfromtext('point(0 8)',-1));
INSERT into tb_point (nu_point, point_geom) values (2, geometryfromtext('point(3 8)',-1));
INSERT into tb_instant (nu_instant, ts_instant) values (1, '01-01-2010 10:02:00');
INSERT into tb_instant (nu_instant, ts_instant ) values (2,'01-01-2010 10:12:00');
INSERT into tb_interval(nu_interval, nu_instant_1, nu_instant_2) values (1, 1, 2);
INSERT into tb_upoint (nu_upoint,nu_point_1, nu_point_2, nu_interval) values (1, 1, 2,1);
INSERT into tb_mpoint (nu_mpoint, nu_upoint) values (1, 1);
```

Esses comandos armazenam no banco de dados o primeiro ponto móvel (nu_mpoint = 1) e sua primeira unidade de movimento (nu_upoint = 1) que está associada aos pontos geográficos (0,8) e (3,8), os quais estão associados, respectivamente, ao instante inicial “01-01-2010 10:02:00” e final “01-01-2010 10:12:00”.

A Figura 5 mostra os dez movimentos do cenário de teste com seus respectivos pontos geográficos, como resultado de uma consulta ao banco de dados (através da ferramenta *PgAdminIII)*, envolvendo a junção de todas as tabelas do banco, onde as colunas nu_mpoint e nu_upoint identificam os pontos móveis e suas unidades de movimento e as colunas p1, p2, instant_1 e instant_2, identificam, em cada unidade, os pontos geográficos e seus respectivos instantes de movimento.

```
SELECT m.nu_mpoint ,m.nu_Upoint,(x(p1.point_geom),y(p1.point_geom)) AS P1 ,
       (x(p2.point_geom),y(p2.point_geom)) AS P2,
       i1.ts_instant as instant_1,i2.ts_instant as instant_2
from tb_mpoint m, tb_upoint u, tb_point p1, tb_point p2,  tb_interval t, tb_instant i1,
     tb_instant i2
where  m.nu_upoint = u.nu_upoint
      and   u.nu_point_1 = p1.nu_point
      and   u.nu_point_2  = p2.nu_point
      and   u.nu_interval = t.nu_interval
      and  t.nu_instant_1 = i1.nu_instant
      and   t.nu_instant_2 = i2.nu_instant
order by m.nu_mpoint ,m.nu_Upoint,t.nu_instant_1
```

ainel de saída

Saída de Dados | Explain | Mensagens | Histórico

	nu_mpoint integer	nu_upoint integer	p1 record	p2 record	instant_1 timestamp without time zone	instant_2 timestamp without time zone
1	1	1	(0,8)	(3,8)	2008-01-28 10:02:00	2008-01-28 10:12:00
2	1	2	(3,8)	(6,8)	2008-01-28 10:12:00	2008-01-28 10:22:00
3	1	3	(6,8)	(9,8)	2008-01-28 10:22:00	2008-01-28 10:32:00
4	1	4	(9,8)	(12,8)	2008-01-28 10:32:00	2008-01-28 10:42:00
5	2	5	(3,6)	(3,8)	2008-01-28 10:02:00	2008-01-28 10:12:00
6	2	6	(3,8)	(3,10)	2008-01-28 10:12:00	2008-01-28 10:22:00
7	2	7	(3,10)	(3,12)	2008-01-28 10:22:00	2008-01-28 10:32:00
8	3	8	(11,12)	(11,8)	2008-01-28 10:02:00	2008-01-28 10:12:00
9	3	9	(11,8)	(11,4)	2008-01-28 10:12:00	2008-01-28 10:22:00
10	3	10	(11,4)	(11,0)	2008-01-28 10:22:00	2008-01-28 10:32:00

Figura 5 – Consulta aos movimentos dos pontos móveis 1, 2 e 3

A Figura 6 exibe os resultados de três consultas, usando a função F_MTrajectory, para se obter, respectivamente, as geometrias das trajetórias dos pontos móveis 1, 2 e 3, onde a função *asText(geometry)* converte a geometria para o formato textual. A Figura 7 exibe os resultados de uma consulta, usando a função F_MPoint, para se obter os pontos geográficos do ponto móvel 1. As funções *X(geometry)* e *Y(geometry)* são usadas para apresentar as coordenadas no formato textual.

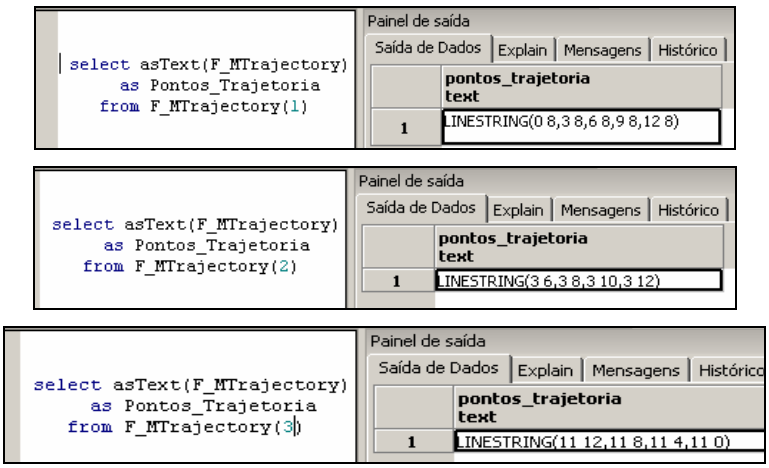

Figura 6 – Consulta usando a função F_MTrajectory sobre os pontos móveis 1, 2 e 3

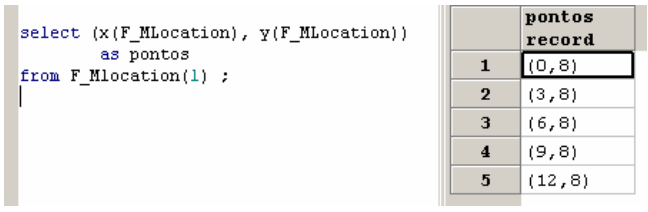

Figura 7 – Consulta usando a função F_MLocation sobre o ponto móvel 1

A partir dos pontos geográficos e das trajetórias obtidas na aplicação dessas funções sobre os três pontos móveis, foi possível visualizá-los em conjunto através do SIG Jump (ver Figura 8).

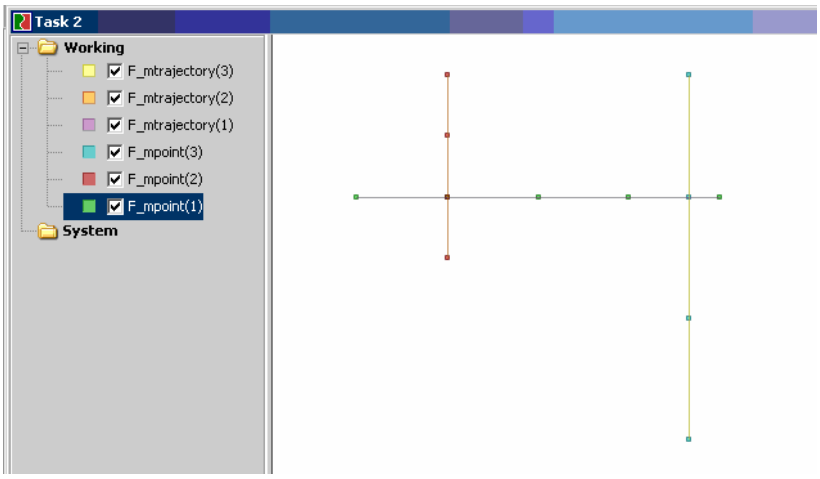

Figura 8 – Visualização dos pontos móveis 1, 2 e 3 no SIG Jump

A Figura 9 exibe os resultados de uma consulta, usando a função F_MDistance, para se obter a distância móvel entre os pontos móveis 1 e 2, pois calcula as distâncias registradas, durante todo o percurso, entre esses dois pontos móveis, em cada instante comum dos seus movimentos. Observa-se, na segunda tupla resultante, que a distância entre os pontos móveis é zero, indicando o cruzamento das suas trajetórias.

```
select f_mdistance
   as distancia_movel
from F_MDistance(1,2);
```

	distancia_ numeric
1	3.6055512
2	0
3	3.6055512

Figura 9 – Consulta usando a função F_MDistance sobre os pontos móveis 1 e 2

A Figura 10 exibe os resultados de uma consulta, usando a função F_MDuration, para se obter a o tempo de duração do movimento do ponto móvel 1, que é de 40 minutos, correspondendo a quatro unidades de movimento com 10 minutos cada uma. .

```
select f_mduration
  as duracao
from F_MDuration(1);
```

	duracao time witho
1	00:40:00

Figura 10 – Consulta usando a função F_MDuration sobre o ponto móvel 1

As Figuras 11 e 12 exibem os resultados de consultas, usando, respectivamente, a função F_MPosition, para se obter as posições dos pontos móveis 1 e 2 no instante "01-01-2010 10:12:00", e a função F_MInstant, para se obter os instantes em que esses pontos móveis estavam na posição (3,8). Tais resultados comprovam que esses dois pontos móveis se encontravam na mesma posição e no mesmo instante, caracterizando uma interseção espaço-temporal. Num cenário real, tal interseção implicaria numa colisão entre esses pontos móveis e causaria a interrrupção de seus respectivos movimentos. No entanto, neste cenário de teste, tais movimentos não foram interrrompidos.

```
select (x(f_mposition), y(f_mposition))
        as ponto
from F_MPosition(1,'01-01-2010 10:12:00')
```

	ponto record
1	(3,8)

```
select (x(f_mposition), y(f_mposition))
        as ponto
from F_MPosition(2,'01-01-2010 10:12:00')
```

	ponto record
1	(3,8)

Figura 11 – Consulta usando a função F_MPosition sobre os pontos móveis 1 e 2

```
select f_minstant as instante
from F_MInstant(1, 3, 8)
```

	instante timestamp without time zo
1	2010-01-01 10:12:00

```
select f_minstant as instante
from F_MInstant(2, 3, 8)
```

	instante timestamp without time zo
1	2010-01-01 10:12:00

Figura 12 – Consulta usando a função F_MInstant sobre os pontos móveis 1 e 2

Outras consultas foram formuladas, combinado-se as funções implementadas neste trabalho com as funções nativas do PostGIS ou do PostgreSQL. Por exemplo, as consultas "select from length2d(f_mtrajectory(1))" e "select min(f_mdistance) from f_mdistance(1,2)" obtém, respectivamente, o comprimento (12) da trajetória do ponto móvel 1 e a menor distância (0) entre os pontos móveis 1 e 2.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma implementação da álgebra para pontos móveis, utilizando o SGBD PostgreSQL e sua extensão espacial PostGIS. Os tipos de dados espaço-temporais algébricos foram implementados através de tabelas e funções armazenadas num banco de dados relacional. Para testar essa implementação, foram armazenados num banco de dados três pontos móveis, projetados a partir de um cenário com dez movimentos em sentidos diferentes, usando um sistema local de coordenadas *2D*. Além disso, foram realizadas consultas ao banco de dados, usando as referidas funções, obtendo-se informações importantes sobre os pontos móveis, tais como: a trajetória do movimento; a duração do movimento; a distância entre os pontos; a localização num determinado instante e o instante para uma determinada localização. Adicionalmente, os resultados das consultas possibilitaram a determinação de uma colisão entre dois pontos móveis, caracterizada pela ocorrência de interseção espaço-temporal.

A principal contribuição deste trabalho consiste no desenvolvimento de um modelo conceitual e lógico de banco de dados relacional e sua implementação através do SGBD PostgreSQL/PostGIS, permitindo armazenar pontos móveis e realizar consultas sobre o histórico de seus movimentos e oferecendo um suporte mínimo necessário para o desenvolvimento de aplicações que utilizem pontos móveis.

REFERÊNCIAS

GÜTING, R.H., BÖHLEN, M.H., ERWIG, M., JENSEN, C.S., LORENTZOS, N.A., SCHNEIDER, M. AND VAZIRGIANNIS, M. **A Foundation for Representing and Querying Moving Objects in Databases**. ACM Transactions on Database Systems 25, 2000, 1-42.

R.H. GÜTING, V.T. DE ALMEIDA, D. ANSORGE, T. BEHR, Z. DING, F. HOFFMANN, M. SPIEKERMANN, AND U. TELLE. **SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching**. In Proc. 21st Intl. Conf. on Data Engineering (ICDE), 2005, 1115-1116.

JUMP. **The Unified Mapping Platform**. <http://www.vividsolutions.com/JUMP> Acesso em 1/07/2010.

LEMA, J. A. C., FORLIZZI, L., GÜTING, R. H., NARDELLI, E. AND SCHNEIDER, M. **Algorithms for Moving Objects Databases**. The Computer Journal 46(6): 680-712, 2003.

PELEKIS, N., THEODORIDIS, Y. **Boosting Location-Based Services with a Moving Object Database Engine**. Proceedings of MobiDE, 2006.

POSTGIS. **Manual do PostGIS**. <http://www.webgis.com.br/postgis> Acesso em 1/07/2010.

POSTGRESQL. **Site do postgresql**. <http://www.postgresql.org.br> Acesso em 1/07/2010.

SISTLA, A.P., WOLFSON, O., CHAMBERLAIN, S. AND DAO, S. **Modeling and querying moving objects.** In Proc. IEEE Intl. Conf. On Data Engeneering, pages 422–432, Birminghan, UK, 1997.

YI, B. AND MEDEIROS, C.BAUZER. **Um modelo de Dados para Objetos Móveis**. In IV Simpósio Brasileiro de GeoInformática, 33-40, 2002.