

Universidade do Minho
Departamento de Informática

Engenharia Web
Deliverable 1:
Projeto *Web Ratio* SPWS

Leandro Costa pg41083

Paulo Jorge pg17918

14 de abril de 2020

Índice

1. Modelo IFML	3
2. <i>Domain Model</i>	4
3. Ações das operações CRUD.....	7
4. Vistas geradas pelo <i>Web Ratio</i>	8

Índice de figuras

Figura 1 – modelo IFML do projeto web ratio	3
Figura 2 - modelo de domínio	4
Figura 3 - criação do procedure	5
Figura 4 - criação do trigger	5
Figura 5 - tabela passadeira antes da inserção de carros/pedestres.....	5
Figura 6 - tabela passadeira_carro	6
Figura 7 - tabela passadeira depois da inserção de carro	6
Figura 8 - tabela passadeira_pedestre	6
Figura 9 - ação adicionar passadeira	7
Figura 10 - ação editar passadeira	7
Figura 11 - ação apagar passadeira	8
Figura 12 - passadeiras main view	8
Figura 13 - passadeira details view	8
Figura 14 - add passadeira view	9
Figura 15 - edit passadeira view.....	9

2. Domain Model

O modelo de domínio nesta fase é algo simples, pois o foco está essencialmente na gestão de passeadeiras. A relação entre as passeadeiras e pedestres/carros possui um atributo complementar que diz respeito à distância entre estes.

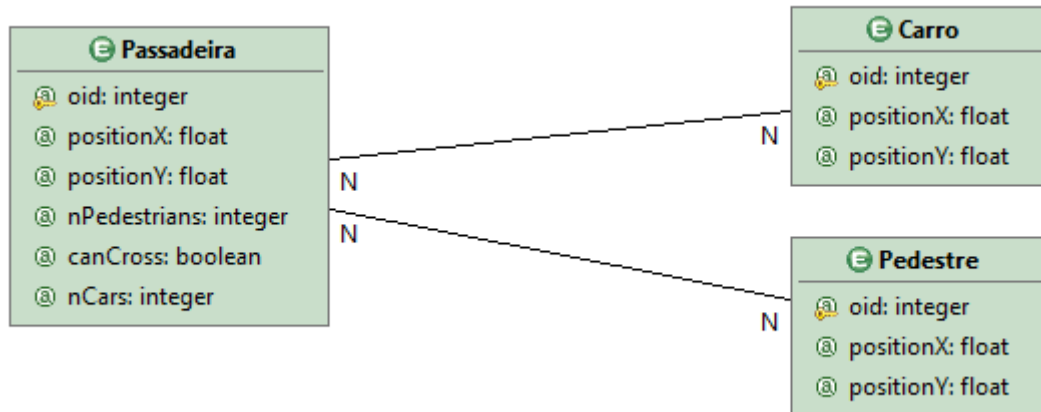


Figura 2 - modelo de domínio

Nesta fase, como ainda não existem os componentes responsáveis pela atualização progressiva das posições dos veículos e pedestres, estas foram simuladas fazendo uso de procedimentos e *triggers* na base de dados.

Quando é inserido um novo veículo ou pedestre é calculada a distância a que este novo objeto se encontra das passeadeiras existentes e esta é registada nas tabelas de relação entre as passeadeiras e carro/pedestre. Com a distância calculada é verificada se esta é inferior a 10 metros, se sim, o número de carros/pedestres (atributo da passeadeira) irá aumentar. Caso a distância seja inferior a 5 metros, o estado do atributo `canCross` passa a *false*, devido à proximidade do veículo à passeadeira.

Poderíamos permitir que estes valores de distância e número de carros/pedestres fossem estáticos e simplesmente inseridos pelo utilizador na vista do web ratio. Porém, decidimos nesta fase aproximar-nos um pouco mais do âmbito final do projeto em que estes valores não são inseridos manualmente, e sim, registados na base de dados por meio de outros componentes que não o SPWS.

Os *procedure* e *trigger* implementados para teste foram os seguintes (para a passeadeira, para o caso dos pedestres é bastante semelhante, excetuando a verificação que altera a coluna `canCross`):

```

DELIMITER ;;
• CREATE PROCEDURE ADD_PASSADEIRA_CARRO(IN newCarID INT)
BEGIN
  DECLARE n INT DEFAULT 0;
  DECLARE i INT DEFAULT 0;
  DECLARE px,py,cx,cy,dist FLOAT DEFAULT 0.0;
  SELECT COUNT(*) FROM passadeira INTO n;
  SET i=0;
  WHILE i<n DO

    SELECT positionX, positionY FROM passadeira WHERE oid=i+1 INTO px, py;
    SELECT positionX, positionY FROM carro WHERE oid=i+1 INTO cx, cy;

    SELECT ROUND(sqrt(power(px-cx,2) + power(py-cy,2))) INTO dist;

    INSERT INTO passadeira_carro(passadeiraID, carroID, distancia) VALUES(i+1, newCarID, dist);

    UPDATE passadeira SET nCars = nCars + 1 WHERE oid = i+1 and dist<10;
    UPDATE passadeira SET canCross = false WHERE oid = i+1 and dist<3;

    SET i = i + 1;

  END WHILE;
End;
;;
DELIMITER ;

```

Figura 3 - criação do procedure

```

DELIMITER #
• CREATE TRIGGER new_car
AFTER INSERT ON carro
FOR EACH ROW
BEGIN
  CALL ADD_PASSADEIRA_CARRO(NEW.oid);
END#
DELIMITER ;

```

Figura 4 - criação do trigger

De modo a testar, foram criadas 3 passadeiras:

	oid	positionx	npedestrians	canCross	ncars	positiony
▶	1	5	0	1	0	6
	2	10	0	1	0	5.6
	3	15	0	1	0	2
*	NULL	NULL	NULL	NULL	NULL	NULL

Figura 5 - tabela passadeira antes da inserção de carros/pedestres

Pode-se observar as suas posições (x e y) e que nesta fase não existem veículos nem pedestres por perto, portanto os seus valores são 0 e o canCross encontra-se a *true*.

Agora adicionando 2 veículos é calculada automaticamente a distância e esta é inserida na tabela `passadeira_carro`, que foi automaticamente preenchida com todas as combinações entre passadeiras e carros:

	passadeiraID	carroID	distancia
▶	1	2	8
	1	3	8
	2	2	3
	2	3	3
	3	2	4
	3	3	18
*	NULL	NULL	NULL

Figura 6 - tabela `passadeira_carro`

Desta feita, foi “disparado” o *trigger* e irão ser feitas verificações aos valores de distância. Como foi mencionado acima, se este for inferior a 10, será contabilizado na coluna `nCarros` da tabela `passadeira`. Adicionalmente, caso seja inferior a 3, o estado da variável `canCross` irá passar para *false*:

	oid	positionx	npedestrians	canCross	ncars	positiony
▶	1	5	0	1	2	6
	2	10	0	1	2	5,6
	3	15	0	1	1	2
*	NULL	NULL	NULL	NULL	NULL	NULL

Figura 7 - tabela `passadeira` depois da inserção de carro

Da mesma forma, se for adicionado um pedestre, é calculada automaticamente a distância e que este se encontra de todas as passadeiras presentes no sistema. Se os pedestres adicionados estiverem a uma distância inferior a 10, a coluna `nPedestrians` irá incrementar.

	passadeiraID	pedestreID	distancia
▶	1	1	12
	1	2	12
	2	1	9
	2	2	6
	3	1	12
	3	2	13
*	NULL	NULL	NULL

Figura 8 - tabela `passadeira_pedestre`

3. Ações das operações CRUD

Relativamente às operações CRUD, estas foram descritas no modelo e foram implementadas da seguinte forma:

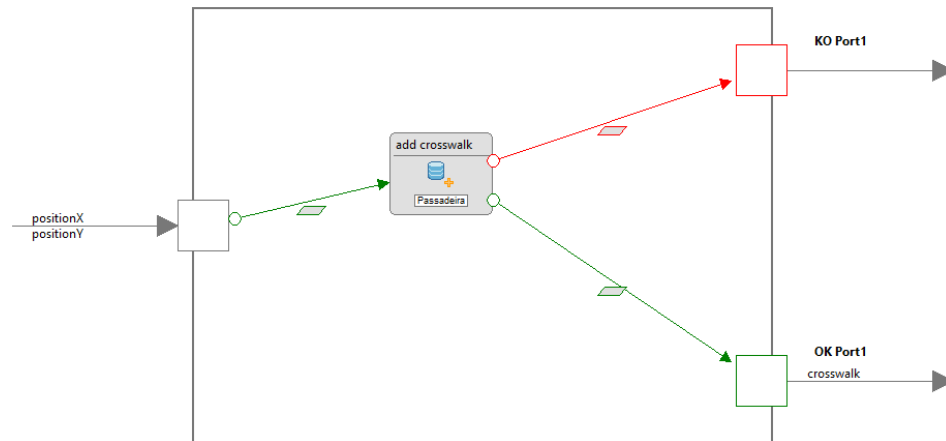


Figura 9 - ação adicionar passadeira

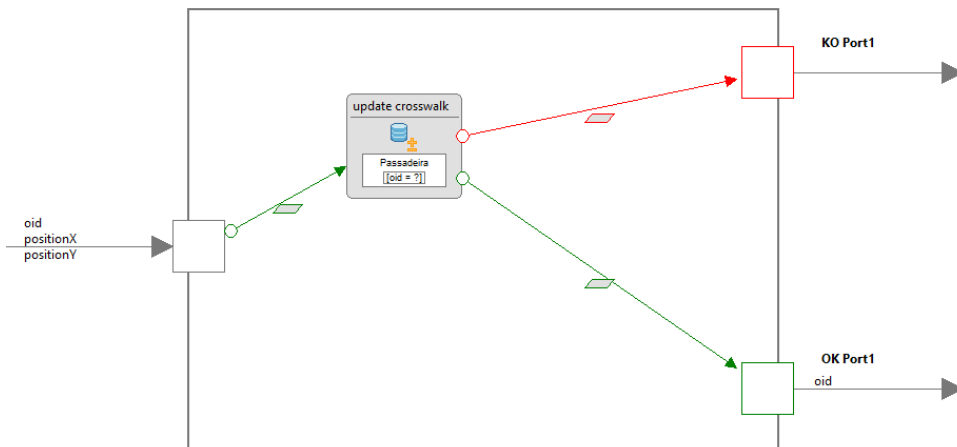


Figura 10 - ação editar passadeira

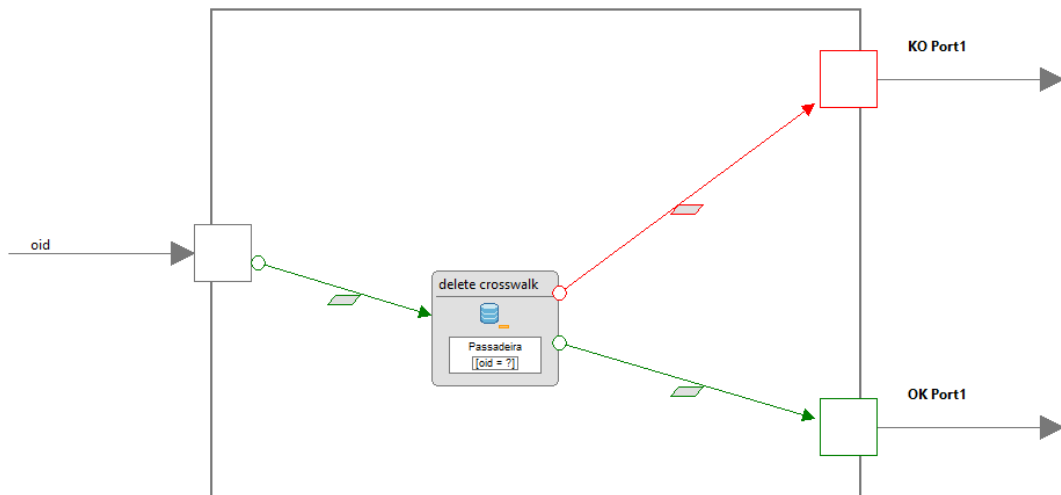


Figura 11 - ação apagar passadeira

4. Vistas geradas pelo *Web Ratio*

No final, é possível a gestão de passadeiras utilizando as vistas fornecidas pelo web ratio, mediante o modelo desenvolvido.



Figura 12 - passadeiras main view

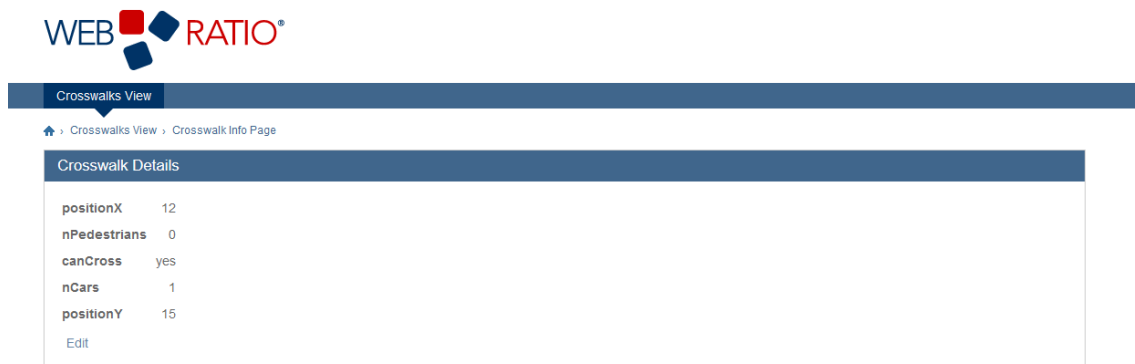


Figura 13 - passadeira details view

The screenshot shows the 'WEB RATIO' logo at the top left. Below it is a dark blue header bar with the text 'Crosswalks View'. Underneath the header is a breadcrumb trail: 'Crosswalks View > Add Crosswalk'. The main content area has a title bar 'New Crosswalk'. Below this, there are two input fields: 'positionX' and 'positionY'. At the bottom left of the form is an 'Add' button.

Figura 14 - add passadeira view

The screenshot shows the 'WEB RATIO' logo at the top left. Below it is a dark blue header bar with the text 'Crosswalks View'. Underneath the header is a breadcrumb trail: 'Crosswalks View > Edit Crosswalk'. The main content area has a title bar 'Edit Crosswalk'. Below this, there are two input fields: 'positionX' with the value '2' and 'positionY' with the value '3'. At the bottom left of the form is an 'Edit' button.

Figura 15 - edit passadeira view

Todas as operações CRUD às passadeiras foram testadas e comprovamos a sua funcionalidade, que estava de acordo com o esperado. Se for inserido um pedestre ou veículo, as informações das passadeiras são atualizadas e é possível visualizá-las através das vistas do web ratio.

Consideramos que não faria sentido permitir a inserção de campos como nPedestrians ou canCross aquando a criação de uma passadeira, uma vez que estes não são valores que devem ser inseridos manualmente. A simulação utilizando o *trigger* na base de dados simula uma aproximação da aplicação final, em que os valores de número de pedestres e veículos próximos serão dinamicamente alterados mediante a localização destes. Uma vez que nesta fase apenas estamos a focar-nos no SPWS, tínhamos a opção de permitir a sua edição manual, mas acabamos por utilizar o *trigger* e automatizar o processo (mesmo que este não venha a ser o que vai acontecer no projeto final).

Sendo assim, ao criar uma passadeira os únicos dados que podem ser inseridos (e posteriormente editáveis pelo utilizador) são apenas as coordenadas da passadeira, pois são os únicos dados da passadeira que não dependem de outros objetos.

Nesta fase, ainda não possuímos os componentes responsáveis por simular informação de pedestres e veículos que calculem as distâncias, etc, de forma dinâmica. No projeto final, com todos os componentes funcionais, a atualização de dados, comunicação da posição quando entre num raio definido, e respetivas constantes verificações serão da responsabilidade *backend*, cuja arquitetura ainda não está completamente definida.