

Licenciatura em Engenharia Informática e de Computadores

Programação na Internet, INVERNO de 2014/15, 10 De Fevereiro 2015

Trabalho Final

Grupo 43

31181 – Rui Dinis

28499 – Victor Morais

33357– João Caselli

Lisboa, 07 de Dezembro de 2014

ÍnDice

[Modelo EA 4](#_Toc405834578)

[ALÍNEA 1 5](#_Toc405834579)

[ALÍNEA 2 6](#_Toc405834580)

[ALÍNEA 3 7](#_Toc405834581)

[ALÍNEA 4 8](#_Toc405834582)

[ALÍNEA 5 9](#_Toc405834583)

[ALÍNEA 6 10](#_Toc405834584)

[ALÍNEA 7 11](#_Toc405834585)

[ALÍNEA 8 12](#_Toc405834586)

[ALÍNEA 9 14](#_Toc405834587)

[ALÍNEA 10 15](#_Toc405834588)

[ALÍNEA 11 17](#_Toc405834589)

[ALÍNEA 12 18](#_Toc405834590)

# Modelo EA

Na modelação do problema proposto chegou-se á seguinte solução com o diagrama entidade-associação anexo.

Assim podemos observar no mesmo que optamos por criar duas entidades independentes para modelar clientes e funcionários, sendo que a primeira tem duas especializações para clientes individuais e empresariais. Poderíamos ter optado por criar uma entidade que modelasse todas as entidades existentes no sistema e os seus atributos comuns tais como nome, NIF, morada, contacto, não obstante a opção foi esta devido ao facto de neste problema específico dada a pouca interação entre clientes e funcionários foi vantajoso ter as duas entidades separadas.

Modelamos os contactos e as moradas em duas entidades para o efeito, e existem entidades-associações para os contactos dos funcionários e para os contactos dos clientes, possibilitando assim que cada uma destas entidades tenha mais do que um contacto.

A entidade Funcionário Função Loja foi criada para associar um funcionário a uma função desempenhada numa determinada loja. Existe uma entidade “Funções Associadas” que não é mais do que metadata que define as funções acumuláveis entre si. Deste modo torna-se fácil alterar as regras de negócio sem ser necessário alterar código. A mesma estratégia foi seguida para os tipos de serviços disponíveis, os estados possíveis para um serviço assim como para os possíveis resultados das entregas. Esta metadata está modelada nas entidades Tipo de serviço, Estado serviço e situação de entrega respetivamente.

A entidade serviço tira partido desta metadata tendo toda a informação necessária a um serviço que tenha sido criado. Aos serviços correspondem tentativas de entrega que são modeladas na entidade Entrega. Podemos dizer que a entidade Entrega é uma entidade fraca de Serviço, uma vez que sem um serviço não existem entregas. De referir ainda que a entidade Serviço tem uma propriedade preço, que não sendo uma chave estrangeira para o valor do tipo de serviço efetuado, permite atualizar a metadata dos tipos de serviços, sem que os serviços anteriores sejam alterados. Foi também criado um campo “Data Expectável”, que através de uma função insere nesta entidade Serviço um “timestamp” do limite do prazo expectável para a execução do serviço. Por último foi criada a entidade Controlo que guarda as “reservas” feitas no sistema para um determinado cliente.

# ALÍNEA 1

As funcionalidades pedidas nesta alínea são obtidas através dos procedimentos :

Insere Cliente Individual

Insere Cliente Empresarial

Remove Cliente

Atualiza Dados Cliente

As primeiras duas fazem a inserção de um cliente de um determinado tipo. Conforme já mencionado na descrição do EA, e uma vez que não foi nossa opção modelar o problema com uma entidade única para todas as “pessoas” existentes no sistema, foi necessário criar estes dois procedimentos para cada um dos tipos de cliente. Estes dois procedimentos tiram partido de um procedimento “Insere Morada” genérico que criamos para tratar de todas as inserções de moradas no sistema.

A remoção de um cliente do sistema é efetuada através do procedimento “Remove Cliente”. A única situação que merece ser referida nesta situação prende-se com o facto de que a remoção de um cliente da base de dados faz com que todos os serviços registados na mesma pelo cliente removido, passem a constar como serviços do cliente genérico. Esta mesma alteração é efetuada neste procedimento. O procedimento Atualiza Dados Cliente é o procedimento genérico para através de um id de cliente passado por parâmetro ao procedimento atualizar a informação desse mesmo cliente.

# ALÍNEA 2

Procedimentos:

Insere Funcionário

Atualiza Funcionário

Remove Funcionário

Á semelhança da inserção e atualização de clientes no sistema, os funcionários são inseridos e os seus dados atualizados de uma forma muito semelhante tirando partido dos procedimentos genéricos Insere Morada e Insere Contacto. De referir aqui que a remoção de um funcionário do sistema não remove os seus dados do sistema, mas apenas atualiza o seu estado para inativo. Esse estado é obtido através de um campo da tabela Funcionário de nome “Ativo”.

# ALÍNEA 3

Esta funcionalidade é executada no procedimento Atualiza Função Funcionário.

Neste procedimento são efetuadas uma série de verificações antes da atribuição de uma nova função a um funcionário, através da inserção de uma nova entrada na tabela Funcionário Função Loja. Em primeiro é verificado através de consulta nesta mesma tabela se o funcionário já tem duas funções atribuídas, sendo este o máximo de funções que um funcionário pode ter. Se o funcionário em questão não tiver qualquer função atribuída é de imediato atribuída a função pretendida, uma vez que essa atribuição não irá violar qualquer regra de negócio estabelecida. Em segundo verifica-se se a inserção pretendida não viola a regra de um funcionário não poder trabalhar em duas lojas diferentes. Por último verifica-se se a atribuição pretendida não viola as funções que podem ser acumuladas. Para isso usamos a tabela Funções Associadas, que contém a metadata destas regras de negócio. Qualquer violação de uma destas verificações faz com que o procedimento aborte e a atribuição não tenha lugar.

# ALÍNEA 4

Executado na função tabelar “Get Address”.

Esta função retorna uma tabela com toda a informação de um determinado código postal de 7 digitos, encontrada na base de dados dos CTT. Será uma tabela com uma única entrada, mas ainda assim achamos mais conveniente modelar esta função com o retorno em forma de tabela.

# ALÍNEA 5

Com o procedimento Processar Serviços, atribui-se a um estafeta aos serviços que estejam no estado “Analise”, passando esses mesmos serviços para o estado “Progresso”.

Este procedimento tira partido da vista “Numero de Serviços Por estafeta” criada no sistema, que produz o número de serviços atualmente afetados a cada um dos estafetas da BIT. Assim para cada serviço á espera de estafeta, procuramos na vista referida, estafetas disponíveis para aceitar mais encomendas, desde que eles estejam afetos á loja a quem pertence o serviço a ser processado, e que não estejam esses estafetas com serviços reservados por clientes. Estas reservas que são possíveis efetuar no sistema constam da tabela Controlo, e este mecanismo é abordado mais á frente na alínea 7. Caso seja encontrado um estafeta que cumpra os requisitos, é-lhe atribuído o serviço e o mesmo serviço é passado ao estado de “Progresso”.

# ALÍNEA 6

No procedimento Simular Entrega, criamos um número aleatório de 1 a 100 e usamos esse número para simular a probabilidade de uma entrega não ser bem-sucedida. Se o número for superior a 95 a entrega vai ter o estado de “INSUCESSO”, caso contrário o estado de “SUCESSO”. Neste procedimento apenas controlamos este número e inserimos na tabela Entrega com o código de estado da entrega correspondente. De referir que a tabela entrega tem associado um gatilho AFTER para inserções, cuja função é depois de cada inserção avaliar se a referida inserção foi a terceira inserção sem sucesso e atualizar a tabela Serviço passando o serviço a que a entrega se destinava para o estado cancelado. Assim como atualizar os serviços cujas entregas tenham tido sucesso para o estado concluído.

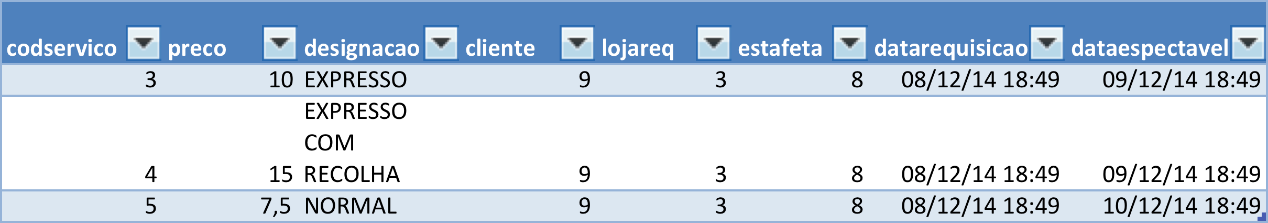
# ALÍNEA 7

Para fazer a simulação dos serviços disponíveis numa loja foram criados os procedimentos armazenados *spEstimaServicos* e *spReservaEstafeta*.

O procedimento *spReservaEstafeta* seleciona um estafeta que esteja disponível para receber um serviço e de seguida reserva-o usando para tal a tabela Controlo guardando um token que é válido por 1 minuto e está associado a esse estafeta, á loja da requisição e ao próprio cliente.

Os serviços urgentes só são mostrados caso o prazo limite de contratualizaçãoo seja superior á data da simulação do serviço.

Exemplo executado depois das 18h:



> EXEC dbo.spEstimaServicos 9,3

@cliente=9, @loja=3

# ALÍNEA 8

Para fazer a inserção de serviços foi criado o procedimento *spInsereServico* e as funções auxiliares *GetDeadLineDate* *GetStoreInSameCounty* e *IsAddressFromSameCounty*.

Este procedimento verifica se o tipo de serviço a criar está disponível para ser efetuado num conselho diferente do conselho da requisição e caso esteja verifica se existe uma loja no conselho de entrega, caso o serviço não esteja disponível em conselhos diferentes ou não existe uma loja que possa assegurar a entrega no conselho pretendido a inserção é abortada.

Caso contrário o serviço é inserido no estado ‘Análise’ com o preço (somado de eventuais suplementos) associado ao tipo de serviço e com uma data de entrega prevista.

Exemplo de serviços criados:



> BEGIN TRAN

SET XACT\_ABORT ON;

DECLARE @CodServico int

EXEC [dbo].[spInsereServico] 1,'Notificação Urgente no mesmo Concelho' ,8,1,31,null,@CodServico OUTPUT

EXEC [dbo].[spInsereServico] 2,'Notificação Urgente no mesmo Concelho com recolha' ,9,3,11,12,@CodServico OUTPUT

EXEC [dbo].[spInsereServico] 3,'Notificação Expresso no mesmo Concelho' ,10,4,17,17,@CodServico OUTPUT

EXEC [dbo].[spInsereServico] 3,'Notificação Expresso num Concelho diferente' ,13,4,20,null,@CodServico OUTPUT

EXEC [dbo].[spInsereServico] 5,'Notificação Normal' ,14,1,23,null,@CodServico OUTPUT

SELECT \* FROM Servico;

ROLLBACK

# ALÍNEA 9

Para cancelar um serviço foi criado o procedimento *spCancelarServico* que muda o estado do serviço indicado para o estado cancelado exceto se o serviço já estiver concluído, lançando uma exceção nesse caso.

> BEGIN TRAN

SET XACT\_ABORT ON;

DECLARE @CodServico int

EXEC [dbo].[spInsereServico] 1,'Notificação Urgente no mesmo Concelho' ,8,1,31,null,@CodServico OUTPUT

SELECT \* FROM Servico;

EXEC [dbo].[spCancelarServico] @CodServico;

SELECT \* FROM Servico;

ROLLBACK

# ALÍNEA 10

Para listar os estafetas disponíveis para aceitar entregas foi criada a vista *EstafetasDisponiveis* que fazendo uso da vista *NumeroServicosPorEstafeta* mostra os funcionários que têm menos de 10 serviços (limite máximo de serviços que um estafeta pode fazer num dado dia).

A vista *NumeroServicosPorEstafeta* verifica todos os serviços que estão no estado ‘Progresso’ devolvendo o número de serviços por cada estafeta ativo.

Exemplo do resultado da vista:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **﻿ID** | **Loja** | **Nome** | **Apelido** | **Email** |
| 6 | 1 | PEDRO RICARDO DA SILVA | PEREIRA | NULL |
| 7 | 2 | DARIO FILIPE DIAS | VARANDA | NULL |
| 8 | 3 | TIAGO | RODRIGUES | NULL |
| 9 | 4 | RICARDO JORGE FERNANDES P | CRUZ | NULL |

# ALÍNEA 11

Para listar os serviços que foram entregues fora do prazo espectável foi criada a vista *ServicosForaPrazo*, esta vista mostra todos os serviços no estado ‘Concluido’ cuja data de entrega seja superior á data prevista de entrega (calculada aquando da criação do serviço através da função *GetDeadlineDate*.

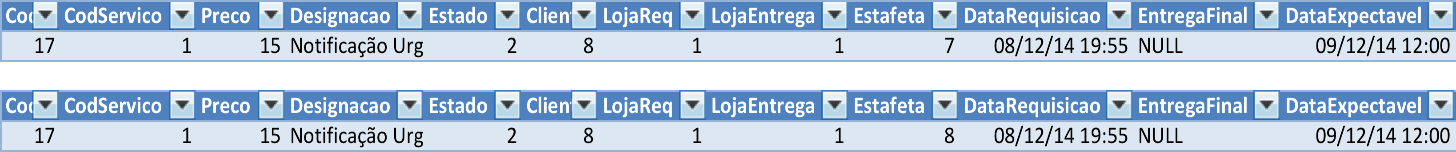
# ALÍNEA 12

Para alterar os serviços alocados a um dado estafeta garantindo a qualidade de serviço e equidade da distribuição entre os estafetas disponíveis foi criado o procedimento *spAfectarServicos.*

Este procedimento coloca o estado do funcionário indicado como inativo e de seguida remove-o de todos os serviços para os quais tenha sido alocado e coloca todos os serviços em causa no estado ‘Análise’.

Por fim é chamado o procedimento *spProcessaServicos* que garante a qualidade dos serviços e sua distribuição por entre os estafetas disponíveis.

Exemplo antes e depois da execução do procedimento *spAfectarServicos*:



> BEGIN TRAN

SET XACT\_ABORT ON;

DECLARE @CodServico int

EXEC [dbo].[spInsereServico] 1,'Notificação Urgente no mesmo Concelho' ,8,1,31,null,@CodServico OUTPUT

EXEC [dbo].[spProcessarServicos]

SELECT \* FROM Servico;

EXEC spAfectarServicos 7

SELECT \* FROM Servico;

SELECT \* FROM NumeroServicosPorEstafeta

ROLLBACK