

## SINCRONIZAÇÃO DE BASES DE DADOS DE DISPOSITIVOS MÓVEIS: UM CASO DE ESTUDO NO SECTOR DO CALÇADO

João Pinho  
10735@ufp.edu.pt

Nuno Magalhães Ribeiro  
Professor Associado - Faculdade de Ciência e Tecnologia  
nribeiro@ufp.edu.pt

### RESUMO

Este artigo desenvolve um caso de estudo sobre sincronização de bases de dados de duas aplicações distintas, uma aplicação *Web* para um computador *desktop* e uma aplicação móvel para Pocket PC, partindo de uma análise dos mecanismos de sincronização de bases de dados existentes para o efeito. Descreve-se o processo de análise e desenvolvimento de duas aplicações para a gestão de stocks no sector do calçado. Apresenta-se a seguir um estudo empírico que pretendeu avaliar as aplicações e a respectiva influência no desempenho dos funcionários de uma empresa do sector. Observa-se que a utilização de tais aplicações contribui para aumentar a eficiência do atendimento ao cliente, tendo-se ainda verificado uma grande aceitação por parte dos funcionários. Conclui-se que a introdução das aplicações contribui para melhorar a gestão de informação e diminuir o tempo de resposta no atendimento ao cliente.

### PALAVRAS-CHAVE

Aplicações móveis, bases de dados, sincronização de informação.

### ABSTRACT

This article develops a case study on the synchronization of databases for two applications, one *Web* based application for a desktop computer and a Pocket PC application, starting with an analysis of currently available mechanisms for database synchronization. It describes the applications developed for a case study in the context of footwear industry. It also presents an empirical study carried out in order to assess the applications' influence in the efficiency of employees in a company where they were tested. We observed a great level of acceptance and we conclude that the adoption of such applications has contributed to improve information management and decrease response times of employees when interacting with customers.

### KEYWORDS

Mobile applications, databases, information synchronization.

## 1. INTRODUÇÃO

Na pequena empresa comercial, de compra e venda de saltos no sector do calçado, que foi alvo do caso de estudo apresentado neste artigo verifica-se que não existem meios informáticos para a consulta e gestão da informação relacionada com as existências em stock. Os funcionários trabalham com saltos que se dividem em secções que, por sua vez, contêm saltos de determinadas referências e tamanhos. No entanto, apesar de necessitarem de contar centenas de saltos quando efectuam uma venda, nunca anotam o número de existências de determinado tamanho ou referência. Por isso, sempre que um cliente se desloca (ou telefona) ao ponto de venda para efectuar uma encomenda, o funcionário tem que se deslocar ao armazém, localizar a caixa contendo o salto pretendido e efectuar uma nova contagem, obrigando o cliente a suportar tempos de espera elevados, resultando num processo de atendimento extremamente ineficiente.

Neste contexto, este artigo propõe uma solução baseada na introdução de duas aplicações de *software* complementares de modo a automatizar a gestão de *stocks*, dando uma ênfase particular aos modos como se pode solucionar o problema da sincronização de bases de dados "móveis". Inicialmente, faz-se uma revisão de tecnologias para o desenvolvimento de aplicações móveis, focando-se em particular a questão da sincronização de bases de dados. Em seguida, descreve-se a concepção e a implementação de uma aplicação *Web* e de uma aplicação de computação móvel para Pocket PC, identificando-se os mecanismos de replicação e sincronização de bases de dados utilizados por cada uma das aplicações, e mencionando as soluções para os problemas que surgem durante a sincronização de bases de dados. Pretendeu-se que as aplicações implementassem um determinado conjunto de requisitos definidos pela empresa com o intuito de melhorar a eficiência das actividades desenvolvidas pelos funcionários, melhorando igualmente o tempo de resposta ao cliente. Estes objectivos foram posteriormente avaliados mediante a realização do estudo empírico aqui descrito. Este trabalho foi desenvolvido no âmbito do projecto de conclusão do curso de Licenciatura em Engenharia Informática da Universidade Fernando Pessoa.

## 2. TECNOLOGIAS PARA O DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS

Actualmente, o uso de computadores pessoais (PCs) e de dispositivos de computação móvel, tais como PDAs (*Personal Digital Assistants*), proporcionam potencialidades que, quando bem exploradas, permitem facilitar a organização e a consulta da informação. Nesta secção descreve-se o conjunto de tecnologias disponíveis para o desenvolvimento de aplicações móveis para Pocket PC que foram utilizadas no desenvolvimento da solução aqui proposta.

A **plataforma .NET** é uma tecnologia da Microsoft, orientada para os serviços *web*, que proporciona uma plataforma única para o desenvolvimento e execução de sistemas e aplicações (Microsoft, 2005). O código gerado para .NET pode ser executado em qualquer dispositivo onde a plataforma se encontre instalada, abrangendo outras tecnologias de *software*, tais como o ASP.NET - um conjunto de serviços para a criação de sítios *web* e o conjunto de classes ADO.NET para a criação de interações com bases de dados.

A **plataforma .NET Compact Framework** (NETCF) para dispositivos móveis é um subconjunto da biblioteca de classes da plataforma .NET, juntamente com classes concebidas exclusivamente

para Pocket PC, permitindo utilizar os mesmos elementos de programação presentes na plataforma .NET, tais como a *common intermediate language* (CIL) e a *common language runtime* (CLR). Com a CIL os executáveis são escritos para um conjunto de instruções que não depende da plataforma. Estas instruções são posteriormente colocadas em memória através da CLR, que faz a gestão de memória relacionada com os objectos e reclama a memória utilizada por objectos inacessíveis (*garbage collection*) (Wigley et al., 2003).

O **Microsoft ASP.NET** permite criar aplicações *web* de uma forma dinâmica através de uma linguagem de programação integrada na plataforma .NET. De acordo com Abreu (2006), o ASP.NET proporciona uma camada de abstracção que permite trabalhar valores obtidos em pedidos HTTP (*Hypertext Transfer Protocol*) no lado do servidor. A geração de "eventos servidor" derivados de "eventos cliente" realiza-se através de formulários e "controles servidor". O ASP.NET disponibiliza "controles servidor" (*Web Controls*), "controles de validação" (*Validator Controls*), e controlos *Data Source* que permitem fazer a ligação a uma fonte de dados e adquirir os dados dessa fonte para, posteriormente, serem apresentados de forma automática pelos controlos *Data Bound* (Abreu, 2006). Neste trabalho, utilizou-se ainda a tecnologia **HTML** (*HyperText Markup Language*) e o **servidor web IIS** (*Internet Information Services*) que foi utilizado para testar a aplicação *web*, denominada **SaltoFixo**, e para realizar a sincronização da base de dados do PC com a base de dados do Pocket PC. O desenvolvimento da aplicação SaltoFixo e da aplicação **SaltoMóvel**, destinada ao Pocket PC, recorreu à linguagem C# e ao ASP.NET. Para o desenvolvimento das bases de dados utilizou-se o *Microsoft SQL Server 2005* e o *SQL Server 2005 Mobile Edition*.

## 3. REPLICAÇÃO E SINCRONIZAÇÃO DE BASES DE DADOS

O processo de sincronização de um PDA permite trocar dados entre um Pocket PC e um PC *desktop* para actualização ou realização de cópias de segurança. Existem várias ferramentas que permitem a sincronização de dados com base em protocolos normalizados, definidos pela OMA (*Open Mobile Alliance*), que permitem assegurar que os dados existentes em ambos os lados são idênticos. Exemplos de tais ferramentas incluem o *Hotsync* para dispositivos Palm OS e o *ActiveSync* da Microsoft para dispositivos Windows Mobile, que sincronizam dados de agenda, contactos, notas textuais, mensagens de e-mail, ficheiros genéricos e documentos pertencentes a outras aplicações da Microsoft tais como as aplicações do Office.

No entanto, existem cenários de utilização em que se pretende outro tipo de sincronização dos dados, tal como no caso em que se utilizam bases de dados. Na realidade, com este tipo de aplicações é necessário um processo de sincronização mais sofisticado que permita assegurar a validade dos dados em ambos os lados da sincronização. Para resolver este problema o *SQL Server Mobile* inclui dois mecanismos para a ligação e troca de dados entre sistemas: o RDA (*Remote Data Access*) e o MR (*Merge Replication*) que tiram partido do uso de classes disponíveis na plataforma .NET.

O **RDA** proporciona um mecanismo de recepção e envio de dados entre as bases de dados *SQL Server* e *SQL Server Mobile*. A arquitectura do RDA (Figura 1) envolve um "agente cliente" e um "agente servidor" que são disponibilizados pelo *SQL Server Mobile*. O primeiro encontra-se no dispositivo móvel entre a aplicação .NET e a base de dados *SQL Server Mobile*, e efectua pedidos HTTP para o "agente servidor" que se encontra no servidor IIS. Em seguida, o "agente servidor" faz a ligação ao servidor SQL para replicar e sincronizar dados ou executar

instruções. O “agente servidor” é, na realidade, um ficheiro DLL (*Dynamic Link Library*) que se instala no directório virtual que é criado para a sincronização no servidor IIS, e funciona como uma extensão de funcionalidades para o IIS (Yao e Durant, 2004).

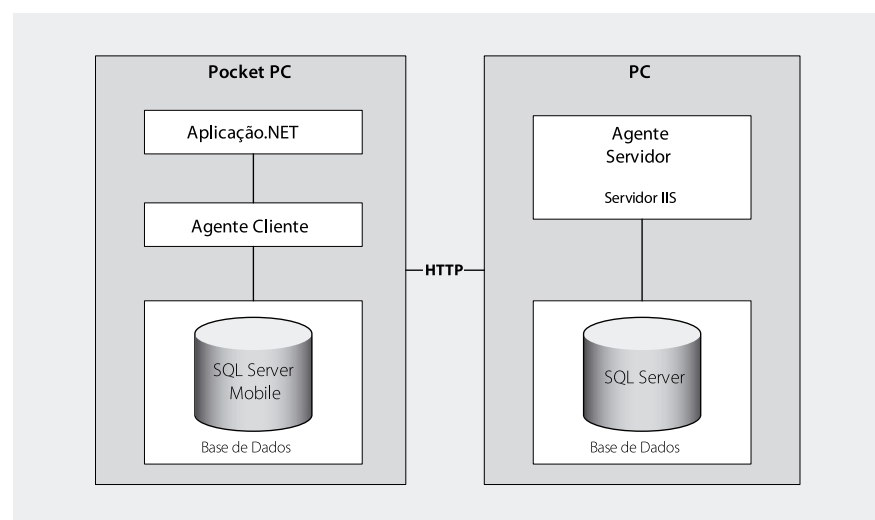


Figura 1 – Arquitectura do RDA (adaptado de Yao e Durant, 2004)

Apesar do RDA ser simples de utilizar e implicar requisitos de gestão reduzidos (uma vez que a ligação à base de dados é gerida pelo “agente servidor” e, em seguida, o “agente cliente” cria automaticamente as tabelas necessárias sem a intervenção directa do programador), apresenta limitações importantes: não permite a sincronização simultânea de várias tabelas, transferindo individualmente os dados para cada tabela no dispositivo móvel e obrigando o programador a escrever código para fazer vários pedidos diferentes para cada tabela, e limita muito a resolução de conflitos: caso um cliente mude uma linha de uma tabela que já foi modificada pelo servidor, ao fazer-se a sincronização, esta nova modificação substitui a anterior porque o RDA modifica os dados mesmo que eles tenham mudado após a replicação.

Por outro lado, o mecanismo **MR** permite que um servidor *SQL Server (Publisher)* disponibilize as suas bases de dados e tabelas a outras bases de dados *SQL Server (Subscribers)*, quer existam no PC quer em dispositivos móveis. Isto consegue-se através da publicação desses objectos e dos dados que contêm, sendo posteriormente acedidos por outros servidores *SQL Server* mediante a criação de réplicas sobre as quais se trabalham os dados e, quando necessário, se sincronizam os dados. De uma forma geral, neste processo estão envolvidas três bases de dados: *Publisher*, *Distributor* e o *Subscriber*. O *Publisher* torna os dados disponíveis para replicação através da criação de publicações, o *Distributor* contém os dados necessários para a gestão da replicação e o *Subscriber*, subscreve uma publicação e recebe os dados replicados (Wigley et al., 2003). A publicação consiste num conjunto de artigos (tabelas, *Stored Procedures* ou funções) que são colocadas para publicação no *SQL Server*. Quando um *Subscriber* cria uma subscrição a partir de uma publicação, um “depósito de dados” com tabelas é transferido para o dispositivo a partir do *Distributor*, através do “agente servidor” e, em seguida, estas tabelas são criadas no dispositivo através do “agente cliente”. Numa

sincronização de dados, o “agente cliente” extrai as modificações efectuadas nos dados das tabelas da subscrição e envia-as para o “agente servidor” que cria um ficheiro com as inserções, alterações e eliminações efectuadas no dispositivo móvel, tal como se ilustra na Figura 2. Ainda no servidor IIS, um processo designado por *merge agent* carrega um *provider* que faz a leitura do ficheiro com as instruções e informa o *merge agent* das alterações a efectuar na publicação. Finalmente, o *merge agent* comunica ao *provider* as alterações que devem ser feitas na subscrição e este cria um ficheiro de saída com essas alterações que serão enviadas pelo “agente servidor” para o “agente cliente”, que é responsável por efectuar as respectivas modificações na subscrição. O *merge agent* é igualmente responsável por detectar e resolver conflitos que ocorram durante a sincronização dos dados.

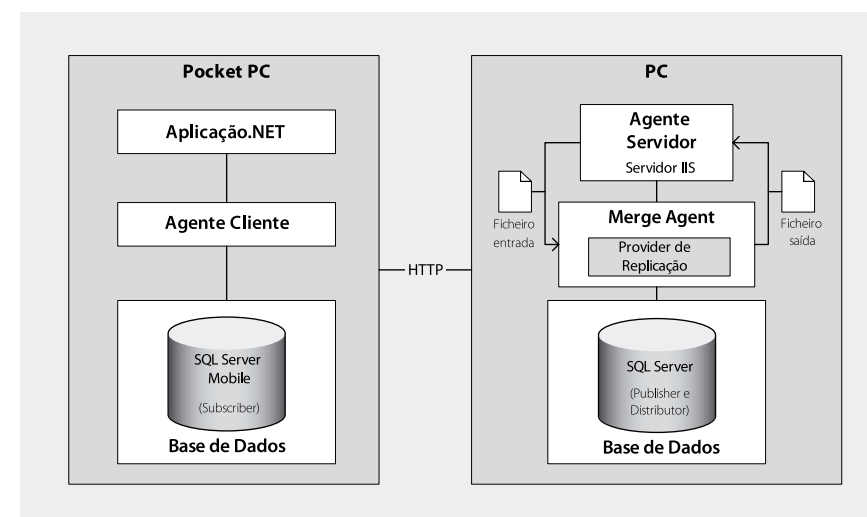


Figura 2 – Arquitectura do Merge Replication (adaptado de Fox e Box, 2003)

Pelas suas características, o MR foi o mecanismo escolhido para realizar a replicação e sincronização de dados das aplicações desenvolvidas. De facto, com o MR a gestão dos artigos é realizada no servidor, que executa igualmente a gestão automática de conflitos sem a intervenção do programador, exceptuando os casos em que o programador tem que construir o seu próprio *conflict resolver*. No âmbito do trabalho desenvolvido, foi necessário criar um *conflict resolver* para resolver um problema de sincronização, uma vez que nenhum dos “*resolvers*” disponíveis por omissão permitia resolver o problema que se descreve adiante. No entanto, de acordo com Fox e Box (2003), a característica mais importante do MR é o suporte que proporciona para a criação de várias tabelas (durante a replicação da base de dados) e para a respectiva sincronização (durante a sincronização da base de dados). Finalmente, o MR permite efectuar a gestão dos ID’s das chaves primárias que têm incrementação automática. No MR criam-se blocos de números de determinada dimensão no *Publisher* e nos *Subscribers*, evitando deste modo o surgimento de colunas com ID’s repetidos e que causariam a ocorrência de erros (Tiffany, 2003). Isto significa que o *Publisher* e os *Subscribers* podem incrementar as respectivas colunas sem existir o perigo de criar um registo (ou uma linha) com o mesmo ID.

Por omissão, o MR recorre a uma lógica baseada em prioridades para **resolver os conflitos**: se não se escolher um *resolver*, o *Publisher* vai “vencer” sempre que existir um conflito, perdendo-se assim as alterações que foram efectuadas nos clientes. Se forem efectuadas alterações no *Publisher* e não no *Subscriber* (ou vice-versa), não são detectados conflitos, e as bases de dados são actualizadas correctamente. Mas se existirem alterações em ambas as bases de dados, poderá ocorrer um conflito de actualização (*update*) quando for efectuada a sincronização. Depois de detectado um conflito, o *merge agent* executa o respectivo *conflict resolver* e determina o vencedor do conflito. Em seguida, a linha “vencedora” é aplicada tanto no *Publisher* como no *Subscriber*. Como se afirmou, para além do uso dos *conflict resolvers*, o programador pode igualmente criar novos “*resolvers*” através do uso de *Stored Procedures* ou de *Business Logic Handlers*. De facto, pode-se criar um “*resolver*” recorrendo à linguagem SQL, sob a forma de uma *Stored Procedure*, registá-lo no servidor e associá-lo a um determinado artigo. Durante a sincronização, esta *Stored Procedure* é executada sempre que exista um conflito no artigo a que foi associada. Por outro lado, a utilização do *Business Logic Handler* permite escrever uma *assembly* que permite fazer chamadas a métodos durante a sincronização, para cada linha que tenha sido alterada, independentemente da existência de um conflito (Microsoft, 2007).

#### 4. ANÁLISE E CONCEPÇÃO DE APLICAÇÕES PARA GESTÃO DE STOCKS DE SALTOS

Neste caso de estudo, um salto é identificado através de várias características, como se pode verificar na Figura 3, contendo uma Referência, um Tamanho e uma Cor. Os primeiros dois números da referência indicam a respectiva secção e, em simultâneo, a altura do salto. Um salto de determinada altura está sempre na mesma secção, independentemente do seu tamanho ou cor. Os saltos existem em determinadas quantidades de pares e possuem um preço por par. O preço é atribuído através da referência e cor, não interessando o tamanho. Uma amostra de um salto pode encontrar-se num mostruário para facilitar a escolha dos clientes.

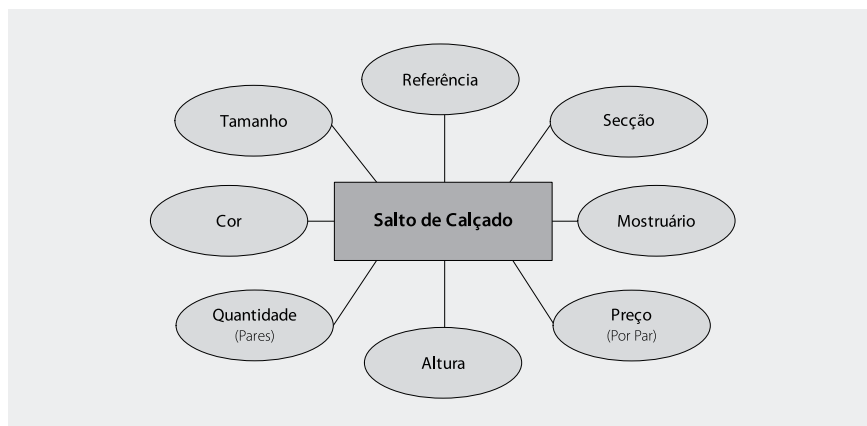


Figura 3 – Características de um Salto de Calçado

Em termos dos **requisitos funcionais**, a aplicação do PC (SaltoFixo) deve permitir operações de inserção, consulta, edição e eliminação das várias características referidas acima. A aplicação móvel para o Pocket PC (SaltoMóvel) deve proporcionar funcionalidades de consulta e edição dos dados que existem na aplicação SaltoFixo. Uma vez que os tamanhos e as secções são na maioria fixos e são replicados da base de dados principal logo na primeira vez que se corre a aplicação, não é necessário disponibilizar uma funcionalidade de inserção e eliminação de tamanhos e secções no Pocket PC. Finalmente, deve ser possível replicar e sincronizar as bases de dados de ambas as aplicações, de modo que as aplicações trabalhem sobre bases de dados idênticas. Para além disso, após a realização de alterações em alguma das bases de dados deve ser possível sincronizar a informação de forma a reflectir as alterações em ambas as bases de dados e, por consequência, em ambas as aplicações.

Em termos da **especificação do modelo da base de dados**, e de acordo com as relações identificadas, foram criadas quatro tabelas como mostra na Figura 4. As tabelas Secção e Tamanho que apenas contêm o nome da secção e o tamanho do salto, a tabela Referência que contém o preço e a chave estrangeira de Secção, e a tabela Salto que contém as restantes características de um salto, bem como as chaves estrangeiras das tabelas Referência e Tamanho, de forma a evitar a repetição e os relacionamentos de muitos para muitos. Esta escolha da criação da tabela Salto tem origem na necessidade de eliminar os relacionamentos *N-N* que se verificavam entre as tabelas Referência e Tamanho, mantendo, deste modo, as tabelas normalizadas. Todas as tabelas contêm identificadores com incrementação automática que constituem as respectivas chaves primárias.

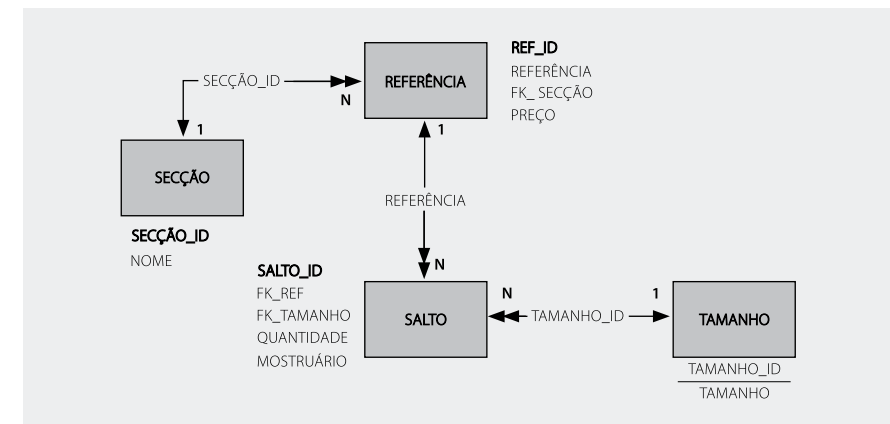


Figura 4 – Modelo entidade-associação da base de dados

No que diz respeito à **escolha do mecanismo de sincronização de dados**, apesar da configuração do RDA ser um pouco mais simples, o MR proporciona características de controlo automático da incrementação das chaves primárias e resolução de conflitos, tal como se explicou na secção 3. Ora, o modelo da base de dados especificado acima recorre à incrementação das chaves, e algumas tabelas na base de dados contêm quantidades que podem ser editadas. Assim, estes dois requisitos justificam a escolha do mecanismo MR para a sincronização de dados, uma vez que ambas as bases de dados causam a ocorrência de situações de conflito quando se pretende realizar a replicação e a sincronização de dados.

Para além disso, a utilização do MR permite ainda definir artigos através da definição de publicações e sincronizar vários artigos em simultâneo, o que, recorrendo ao RDA, teria que ser feito programaticamente como se explicou anteriormente.

### 5. IMPLEMENTAÇÃO DAS APLICAÇÕES E BASE DE DADOS

A aplicação *web SaltoFixo* foi implementada recorrendo à ferramenta de desenvolvimento *Visual Studio 2005*, utilizando ASP.NET e a linguagem de programação C# para a escrita do código. As Figuras 5 e 6 ilustram os ecrãs mais importantes da aplicação SaltoFixo, enquanto que a Figura 7 apresenta o mapa de navegação que determina a estrutura de ecrãs para a interacção com o utilizador, seguindo a notação definida por Ribeiro (2004).

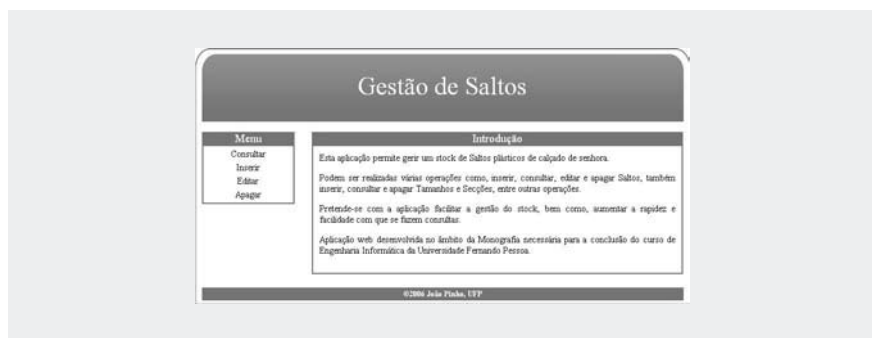


Figura 5 – Página principal da aplicação SaltoFixo

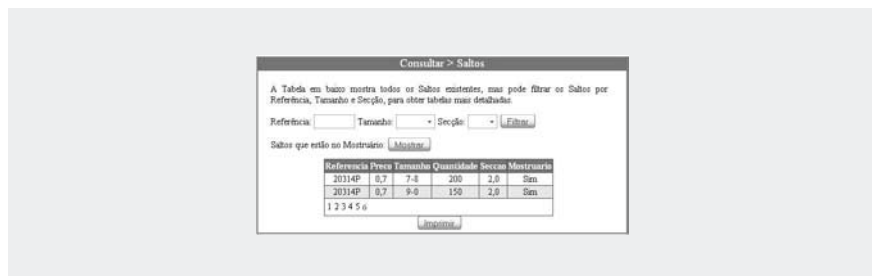


Figura 6 – Página de consulta de Saltos da aplicação SaltoFixo

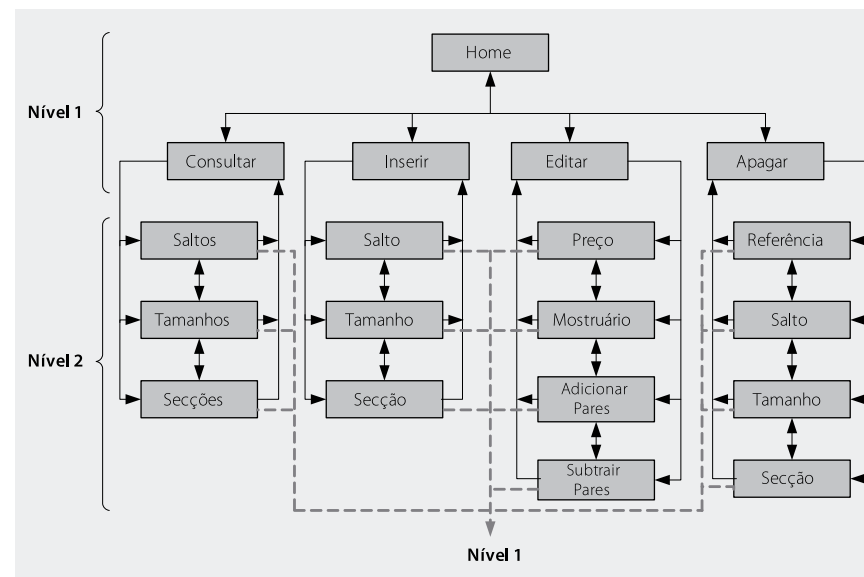


Figura 7 – Mapa de navegação da aplicação SaltoFixo

A aplicação *SaltoMóvel* foi implementada recorrendo à ferramenta de desenvolvimento *Visual Studio 2005*, juntamente com *Windows Mobile 5 Pocket PC SDK (Software Development Kit)* que permite a criação de aplicações para dispositivos móveis e utilizando a linguagem de programação C#. As Figuras 8 e 9 mostram os ecrãs mais importantes da aplicação SaltoMóvel e o respectivo mapa de navegação.

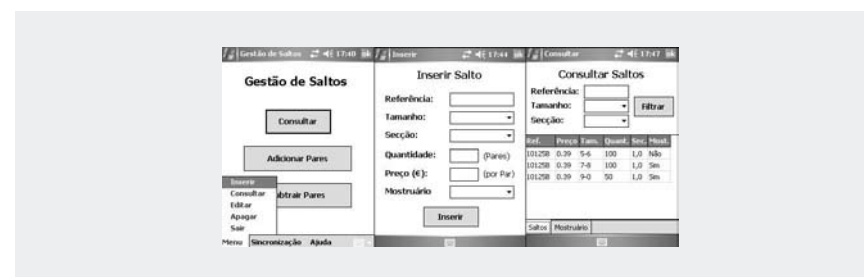


Figura 8 – Ecrãs da aplicação SaltoMóvel

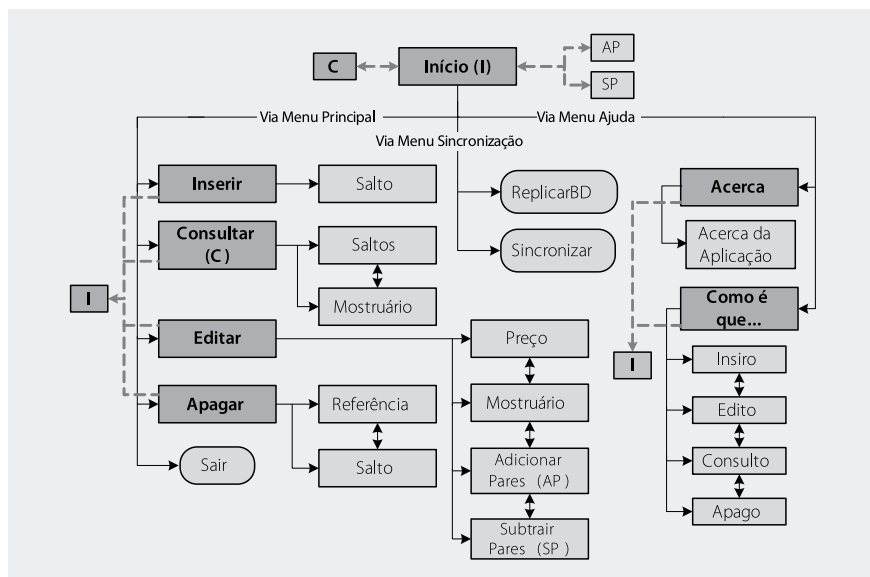


Figura 9 – Mapa de navegação da aplicação SaltoMóvel.

A **base de dados** foi implementada de acordo com o modelo introduzido na Secção 4. No entanto, na tabela Salto foram criadas novas colunas, as colunas **AadicionarAPub** e **AsubtrairAPub** para efectuar cálculos de quantidades sempre que ocorrer um conflito na sincronização. As novas colunas apenas podem ser alteradas pela aplicação SaltoMóvel. Caso se trate de uma adição de quantidades, actualiza-se a coluna de quantidade e coloca-se o valor adicionado na coluna **AadicionarAPub**. Caso se trate de uma subtracção de quantidades, actualiza-se a coluna de quantidade e coloca-se o valor subtraído na coluna **AsubtrairAPub**. Desta forma, se tanto o *Publisher* como o *Subscriber* alterarem quantidades de saltos antes de ser efectuada uma sincronização (causando um conflito), não ocorre a substituição duma coluna de quantidade por outra, mas sim a soma da coluna quantidade da Publicação com a subtracção das colunas adicionais, isto é efectuando-se o seguinte cálculo:

Quantidade final = Quantidade da Publicação + (**AadicionarAPub** - **AsubtrairAPub**).

Finalmente, no que diz respeito à **resolução de conflitos**, e uma vez que não existe nenhum *Conflict Resolver* no *SQL Server* que seja capaz de resolver o problema das quantidades descrito anteriormente, procedeu-se à implementação de um *resolver* do tipo *Business Logic Handler*, que foi posteriormente registado no *SQL Server*. Assim, para além de resolver um conflito de actualização quando este ocorre, é igualmente possível “limpar” as colunas auxiliares quer exista um conflito, quer não, de forma a evitar quaisquer outros tipos de conflitos que poderiam surgir. A figura 10 apresenta um exemplo de um método que efectua os cálculos dos valores a colocar nas colunas das quantidades depois de uma sincronização com conflito em que tanto o *Publisher* como o *Subscriber* fizeram alterações na mesma coluna.

```
public override ActionOnUpdateConflict UpdateConflictsHandler(...)
{
    customDataSet = publisherDataSet.Copy();

    int qPub = Int32.Parse(publisherDataSet.Tables[0].Rows[0]["Quantidade"]);
    int qSubA=Int32.Parse(subscriberDataSet.Tables[0].Rows[0]["AadicionarAPub"]);
    int qSubS=Int32.Parse(subscriberDataSet.Tables[0].Rows[0]["AsubtrairAPub"]);

    int qFinal = qPub + (qSubA - qSubS);
    customDataSet.Tables[0].Rows[0]["Quantidade"] = qFinal;
    customDataSet.Tables[0].Rows[0]["AadicionarAPub"] = 0;
    customDataSet.Tables[0].Rows[0]["AsubtrairAPub"] = 0;

    return ActionOnUpdateConflict.AcceptCustomConflictData;
}
```

Figura 10 – Método UpdateConflictsHandler.

## 6. AVALIAÇÃO DAS APLICAÇÕES

Para a avaliação das aplicações foi efectuado um estudo empírico que consistiu da utilização das aplicações pelos funcionários da empresa para a realização das respectivas tarefas. Pretendeu-se obter conclusões sobre a utilidade das aplicações desenvolvidas, averiguando igualmente sobre a receptividade a novas tecnologias por parte destes profissionais.

Como **instrumentos para a recolha de dados** foi medido o desempenho dos funcionários através do tempo dispendido na consulta do stock de Saltos e do número de deslocações às secções de armazenamento para efectuar a consulta. Foi também desenvolvido e aplicado um questionário, de forma a avaliar as opiniões dos participantes. As Tabelas 1, 2 e 3 apresentam o questionário aplicado, estando divididas em três grupos: o primeiro grupo corresponde ao perfil sócio-demográfico do utilizador, o segundo à facilidade de uso e funcionalidades e o terceiro à utilidade das aplicações. A Tabela 4 mostra as escalas utilizadas.

Característica	Questões
Perfil do utilizador	1) Indique, por favor, a sua idade.
	2) Indique, por favor, a frequência com que utiliza: a) Um computador; b) Um Pocket PC.
	3) Indique ainda o local em que normalmente utiliza o: a) Computador; b) Pocket PC.

Tabela 1 – Primeiro grupo de questões.

Característica	Questões
Adequação do vocabulário	1) O vocabulário utilizado em ambas as aplicações é adequado ao serviço.
Facilidade de Uso	2) A localização dos botões e do texto que se apresenta no ecrã facilita a manipulação da informação: a) Na aplicação do computador; b) Na aplicação do Pocket PC.
	3) Qual foi o grau de dificuldade que sentiu na introdução da informação: a) No Computador; b) No Pocket PC.
	5) A função dos botões das aplicações é fácil de identificar.
	6) A localização da informação nos ecrãs da aplicação é fácil de recordar: a) Na aplicação do computador; b) No Pocket PC.
	7) Qual foi o grau de dificuldade que sentiu na utilização da aplicação: a) Do computador; b) Do Pocket PC.
Utilidade das funcionalidades proporcionadas	4) As aplicações fornecem a informação necessária para executar as tarefas.
	8) Qual é a sua opinião em relação às tarefas que realizou recorrendo às aplicações? a) Acho que a aplicação do computador facilitou o meu trabalho. b) Acho que a aplicação do Pocket PC facilitou o meu trabalho.

Tabela 2 – Segundo grupo de questões.

Característica	Questões
Utilização das aplicações	1) Indique qual das aplicações é que mais utilizou e diga quais os motivos.
	2) O que acha das aplicações em termos de utilidade?
	3) Qual é a sua opinião em relação à futura implementação e utilização das aplicações no serviço?

Tabela 3 – Terceiro grupo de questões.

Grupo	Questões	Escala			
1	2a e 2b	1	2	3	4
		Nunca utilizei	1 vez por semana	2 a 3 vezes por semana	Diariamente
2	1	<div style="border: 1px solid black; padding: 5px; text-align: center;">           Para a recolha de opiniões, nas questões do grupo 2 utilizou-se uma escala do tipo Likert com os cinco pontos seguintes:         </div>			
	2a e 2b				
	4				
	5				
	6a e 6b				
	8a e 8b				
	3a e 3b				
7a e 7b	Muito Difícil	Difícil	Mediano	Fácil	Muito Fácil

Tabela 4 – Escalas utilizadas nas questões.

A experiência durou duas semanas e foi realizada pelas seis pessoas que trabalham na empresa, tendo consistido da utilização de ambas as aplicações para a realização das tarefas decorrentes da actividade normal de atendimento aos clientes.

De forma a avaliar o **desempenho**, para trinta consultas efectuadas, registou-se o tempo de consulta de Saltos. O número de deslocações efectuadas é igual ao número de consultas: trinta deslocações. A Tabela 5 mostra o intervalo de Saltos consultados e o respectivo tempo de contagem antes da introdução das aplicações. O tempo de resposta do funcionário foi determinado como sendo o tempo que demora a deslocar-se a uma secção, efectuar a contagem dos Saltos e regressar para responder ao pedido do cliente. Com a utilização das

aplicações, o tempo de resposta foi calculado tendo em conta o tempo que o utilizador demorou a introduzir a informação e a obter a resposta por parte da aplicação. Na Tabela 6 podem observar-se os resultados recolhidos mediante a utilização das aplicações.

Número de Pares	Consultas / Deslocações	Tempo de Contagem (minutos)
[10-50[	8	± 17s a ± 1m20s
[50-100[	11	± 1m20s a ± 2m43s
[100-150[	6	± 2m43s a ± 4m
[150-200[	1	± 4m a ± 5m30s
[200-250[	1	± 5m30s a ± 7m
[250-350[	2	± 7m a ± 9m15s
[350-500]	1	± 9m15s a ± 14m

Tabela 5 – Resultados antes da introdução das aplicações.

Número de Pares	Deslocações	Tempo de resposta do utilizador com aplicação web	Tempo de resposta do utilizador com aplicação móvel
N pares	0	Entre 15 a 20 segundos	Entre 25 a 30 segundos

Tabela 6 – Resultados depois da introdução das aplicações.

Após a análise das Tabelas 5 e 6 pode concluir-se que a introdução das aplicações na empresa permite diminuir substancialmente o tempo associado à resposta do funcionário aos pedidos do cliente, podendo igualmente permitir que se elimine por completo as deslocações às secções do armazém onde se encontra o Salto a consultar.

Para a característica **“adequação do vocabulário”**, e como se observa na Figura 11, pode considerar-se que os participantes ficaram bastante satisfeitos com o vocabulário que se utiliza nas aplicações e consideraram-no adequado ao tipo de trabalho que desenvolvem. No que diz respeito à **facilidade de uso**, os resultados obtidos no questionário foram agrupados por aplicação e são apresentados nas Figuras 12, 13, 14 e 15.

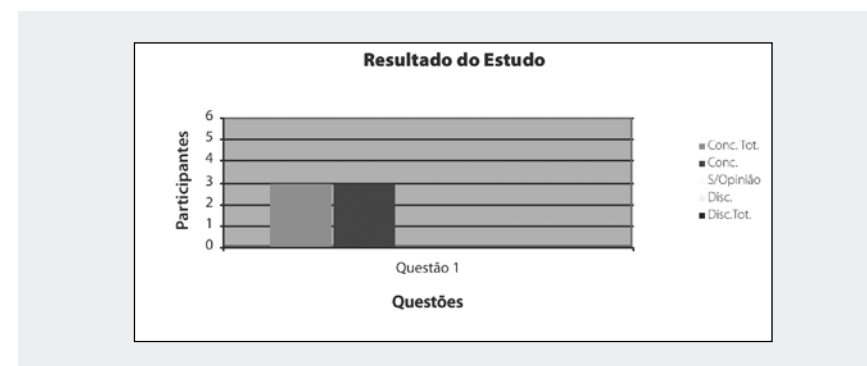


Figura 11 – Gráfico da característica “Adequação do vocabulário”.

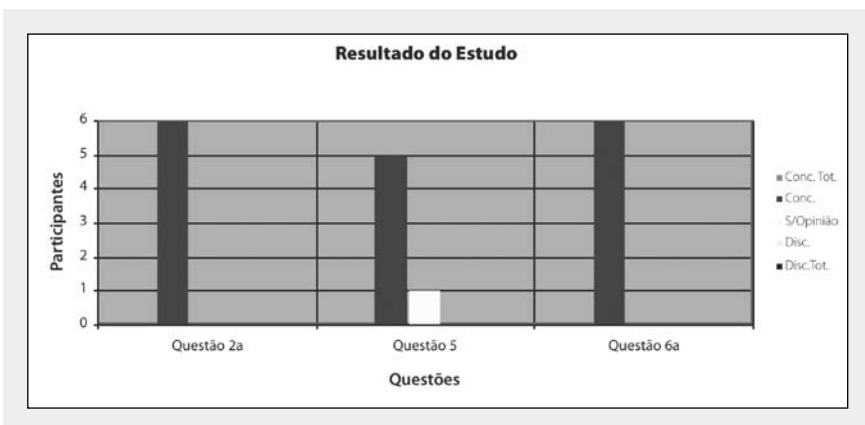


Figura 12 – “Facilidade de uso” para as questões 2, 5 e 6 para a aplicação SaltoFixo.

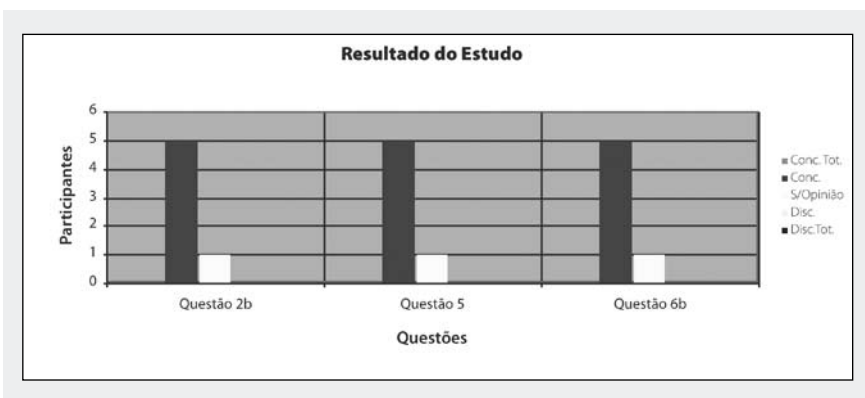


Figura 13 – “Facilidade de uso” para as questões 2, 5 e 6 para a aplicação SaltoMóvel.

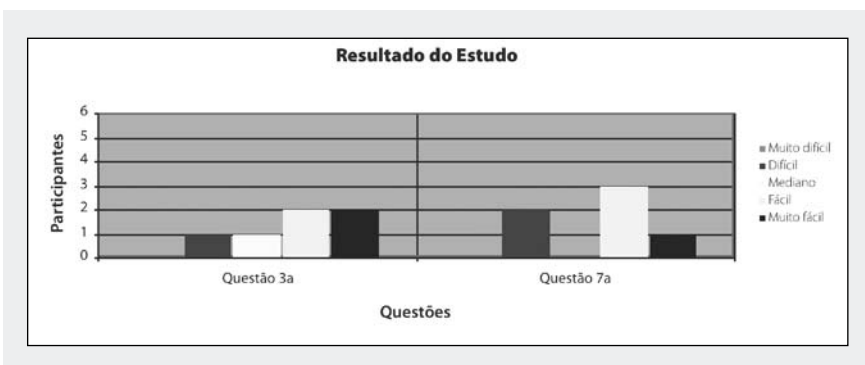


Figura 14 – “Facilidade de uso” para as questões 3 e 7 para a aplicação SaltoFixo.

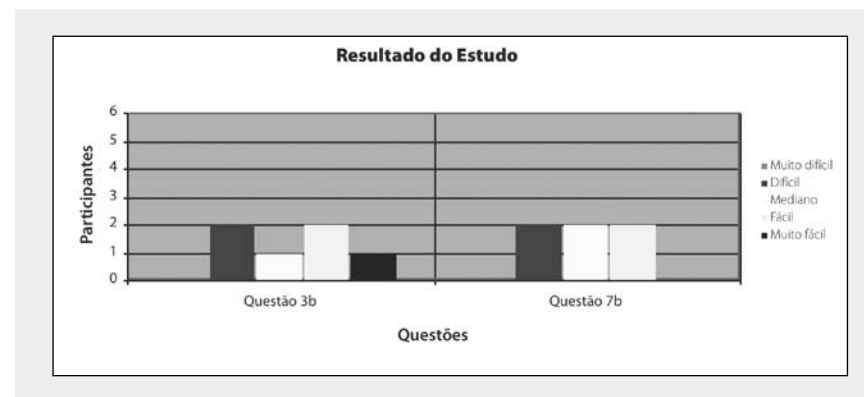


Figura 15 – “Facilidade de uso” para as questões 3 e 7 para a aplicação SaltoMóvel.

O cálculo do *alpha* de Cronbach, que apresenta um valor superior a 0,7 para os resultados das questões de ambas as aplicações, indica que é possível fazer uma agregação consistente de todas as questões associadas à característica “Facilidade de uso”, o que permite determinar uma média global e utilizar esse valor como indicador da facilidade de uso das aplicações. Assim, globalmente, os participantes concordaram que as aplicações são fáceis de usar, visto que a média global das questões 2a, 3a, 5, 6a e 7a e 2b, 3b, 5, 6b e 7b é de 3,83 e de 3,56 respectivamente, numa escala de 1 a 5. No que diz respeito à **utilidade das funcionalidades proporcionadas**, e como se verifica nas figuras 16 e 17 os participantes apresentam uma opinião bastante positiva. Assim, é possível concluir que, para estes funcionários, as aplicações foram encaradas como uma ajuda importante no desempenho das suas tarefas.

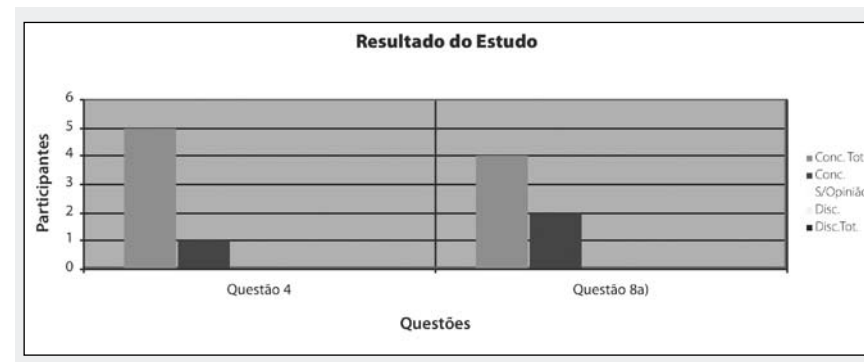


Figura 16 – “Utilidade das funcionalidades proporcionadas”: aplicação SaltoFixo.



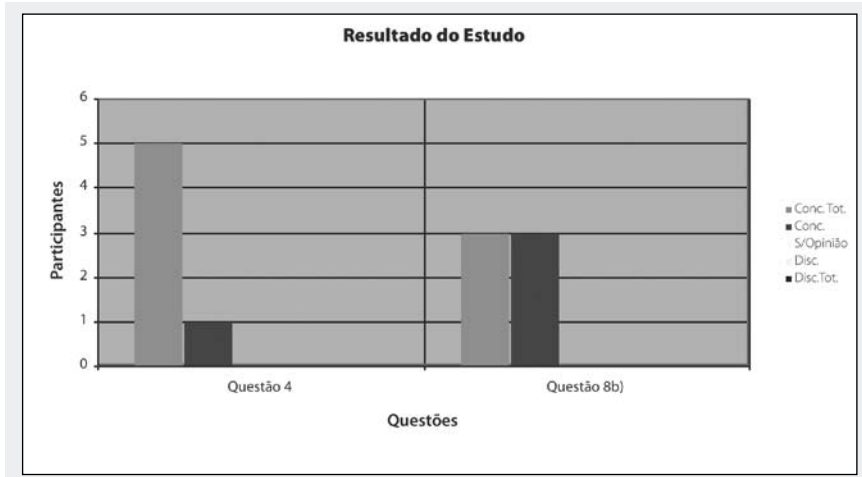


Figura 17 – “Utilidade das funcionalidades proporcionadas”: aplicação SaltoMóvel.

Em relação às questões abertas, referentes à **utilização das aplicações**, os resultados obtidos nos questionários apresentam-se nas Tabelas 7, 8 e 9.

Questão	1) Indique qual das aplicações é que utilizou mais e diga quais os motivos				
	Respostas				
Participantes	PC	Pocket PC	Prática de utilização	Devido ao local onde se encontra a trabalhar	Mobilidade
1	x		x		
2	x		x		
3	x			x	
4		x			x
5		x			x
6		x			x

Tabela 7 – Resultados para a primeira questão aberta.

Questão	2) O que acha das aplicações em termos de utilidade?			
	Respostas			
Participantes	Bom para gestão de stocks	Consulta rápida de existências	Redução do tempo de atendimento ao cliente	Facilita o meu trabalho
1	x	x	x	
2			x	x
3				x
4	x	x	x	
5	x			
6	x		x	

Tabela 8 – Resultados para a segunda questão aberta.

Questão	3) Qual a sua opinião em relação à futura implementação e utilização das aplicações no serviço?			
	Respostas			
Participantes	A sua implementação traria benefícios	Vão permitir a prestação de um serviço melhor	Vão permitir oferecer respostas imediatas às solicitações	Vão fornecer vantagens internas na gestão de stocks
1	x	x	x	x
2	x		x	
3	x		x	x
4	x		x	x
5	x		x	
6	x		x	

Tabela 9 – Resultados para a terceira questão aberta.

A análise da Tabela 7, permite verificar que a utilização das aplicações foi dividida em igual proporção. A maior parte dos participantes que utilizaram sobretudo a aplicação SaltoFixo justificam esta utilização por questões práticas. Os participantes que utilizaram mais a aplicação SaltoMóvel justificam a utilização devido à mobilidade que proporciona. Já no que diz respeito à segunda questão aberta, como se verifica na Tabela 8, os participantes apresentam opiniões mais diversificadas. Na realidade, ao analisar as respostas verifica-se que os participantes responderam de acordo com alguns dos objectivos traçados inicialmente para o desenvolvimento deste trabalho: melhorar a eficiência das actividades desenvolvidas pelos funcionários, melhorando também o tempo de resposta nas interações funcionário – cliente. Finalmente, analisando a Tabela 9, verifica-se que os participantes concordam com a implementação e utilização futura das aplicações na empresa. Para concluir, note-se que, nesta análise dos resultados, não são efectuados nenhuns cruzamentos de dados tendo em conta o perfil dos utilizadores, uma vez que a amostra é muito limitada e não dispõe de participantes suficientes para permitir a obtenção de conclusões estatisticamente relevantes. Para a consulta de dados mais detalhados sobre este estudo experimental aconselha-se o leitor interessado a consultar a monografia que deu origem a este trabalho na biblioteca da UFP.

## 7. CONCLUSÃO

O objectivo de tentar criar aplicações para a gestão da informação da pequena empresa de compra e venda de Saltos para calçado de senhora foi atingido. A recolha e análise dos resultados dos testes efectuados proporcionou informação importante, que contribui para esclarecer os tipos de necessidades dos funcionários e confirma o alcance dos objectivos propostos inicialmente. Verificou-se que a introdução das aplicações veio contribuir para uma melhor gestão de informação e para um melhoramento do tempo de resposta nas interações funcionário – cliente. Apesar de algumas pessoas sentirem algumas dificuldades na utilização das aplicações devido às poucas competências informáticas que possuem, apresentaram respostas muito positivas em relação à sua utilidade e utilização futura, manifestando a vontade de, pelo menos, tentar acompanhar a evolução tecnológica. Conclui-se assim que a utilização das aplicações no dia-a-dia destes profissionais pode constituir uma mais valia. Os dados recolhidos neste estudo permitem igualmente concluir que a mobilidade que a aplicação móvel proporciona pode constituir um factor de desequilíbrio entre uma possível escolha entre as duas aplicações.

Finalmente, este trabalho contribuiu igualmente para aprofundar o estudo dos tipos de mecanismos de sincronização de bases de dados que existem actualmente, quais as suas características e qual o que melhor se enquadra no âmbito das aplicações desenvolvidas.

## BIBLIOGRAFIA

- Abreu, L.** (2006). *ASP.NET 2.0 Curso Completo*. Lisboa, FCA – Editora de Informática.
- Fox, D. e Box, J.** (2003). *Building Solutions with the Microsoft .NET Compact Framework: Architecture and Best Practices for Mobile Development*. USA, Addison Wesley.
- Microsoft** (2005). *Basics of .NET*. [Em linha]. Disponível em <<http://www.microsoft.com/net/basics.mspx>>. [Consultado em 05/12/2006].
- Microsoft** (2007). *How to: Implement a Business Logic Handler for a Merge Article (Replication Programming)*. [Em Linha]. Disponível em <<http://msdn2.microsoft.com/en-us/library/ms147911.aspx>>. [Consultado em 09/01/2007].
- Ribeiro, N.** (2004). *Multimédia e Tecnologias Interactivas*. Lisboa, FCA
- Tiffany, R.** (2003). *SQL Server CE Database Development with the .NET Compact Framework*. Estados Unidos da América, APress.
- Wigley, A., Wheelwright, S., Burbidge, R., MacLeod, R. e Sutton, M.** (2003). *Microsoft .NET Compact Framework (Core Reference)*. Estados Unidos da América, Microsoft Press.
- Yao, P e Durant, D.** (2004). *.NET Compact Framework Programming with C#*. Estados Unidos da América, Prentice Hall PTR.