



Departamento de Engenharia Informática

Interface móvel de acesso ao sistema DomoBus

Ambientes Inteligentes

2009/2010

David Possidónio nº57361

DomoMobile v1.0

Índice

Sumário	3
A Solução	3
A Architectura	7
A Implementação	8
Avaliação da Implementação	10
Referências	11

Sumário

No âmbito da cadeira de Ambientes Inteligentes, foi proposto o desenvolvimento de uma aplicação para um dispositivo móvel, que permita monitorizar e comandar um sistema DomoBus. Para a demonstração do funcionamento desta aplicação foi também pedida uma interface simples para PC. A aplicação realizada é suportada por dispositivos móveis (PDA) com Windows Mobile 6, que comunica com a interface do sistema Domobus através da rede Wi-Fi. Através da aplicação é possível navegar pela estrutura física da habitação, aceder a dispositivos e às suas propriedades para conhecer o seu valor ou para o alterar. Outra funcionalidade requerida é incluir sistema de favoritos em que o utilizador pode explicitar dispositivos que usa com mais frequência, podendo acedê-los de um modo mais rápido.

A solução

A solução implementada contempla os conceitos e ideias fundamentadas nos Documentos de Especificação do Sistema DomoBus*. Assim sendo, alguns conceitos que irão ser mencionados pressupõe-se que já são conhecidos.

Para a construção de uma solução eficaz foi proposto a divisão do sistema **DomoMobile** em quatro partes: a aplicação cliente para PDA¹, a aplicação cliente para Desktop², o Servidor³ e o Modelo base.

1. Os Clientes PDA e Desktop

São duas aplicações independentes que tem a mesma função, apenas operam em ambientes diferentes, como o nome indica uma é para dispositivos móveis e a outra para computadores usuais. Apesar da diversidade na implementação, devido às diferentes tecnologias usadas, ambos tem o mesmo comportamento. Comunicam com o servidor para fazer pedidos e não guardam estado do sistema. Cada vez que o utilizador pretende saber o estado de um propriedade ou alterá-la, terá que fazer um pedido ao Servidor para saber qual o seu estado e só após fazer o pedido de alteração.

A aplicação Cliente após efectuar uma ligação ao servidor e se autenticar, recebe uma lista de Casas ao qual tem acesso. Após o utilizador escolher a casa pretendida, o Servidor envia um ficheiro XML com a especificação da casa. O Cliente faz dinamicamente o carregamento da Casa em memória e esta pronto e interagir com ela. Se o utilizador pretender controlar uma casa diferente o processo repete-se. Foi feita a abordagem do carregamento individual de cada casa para diminuir o processamento do Cliente PDA.

*A Lista completa dos documentos usados estão presentes na Bibliografia.

1 – Designação utilizada ao longo do documento que representa a aplicação cliente para o dispositivo móvel.

2 – Designação utilizada ao longo do documento que representa a aplicação cliente para o computador pessoal.

3 – Designação utilizada ao longo do documento que representa a aplicação Servidor, que comunica com os clientes.

2. O Servidor

A aplicação Servidor recebe os pedidos provenientes dos vários Clientes e responde aos pedidos de cada um. Ele comunica com a rede **Domobus** e guarda estado sobre os dispositivos de uma ou várias casas (Poderá ser útil separar uma casa física em várias casas no sistema).

As operações suportadas pelo Servidor são:

- Echo – Mensagem de resposta a um pedido de ligação.
- Login – Autêntica os utilizadores perante o sistema.
- GetHouses – Envia uma lista de casas aos clientes. A lista só contém as casas para o qual o cliente está registado.
- GetHouseDescription – Envia a especificação em XML da casa escolhida pelo cliente.
- Set – Operação que altera o valor de uma propriedade de um dispositivo. Verifica se o utilizador tem permissões para aceder ao dispositivo ou se este se encontra bloqueado para escrita por parte de outro utilizador com nível de acesso superior. Envia para o Cliente o resultado da operação, o resultado também pode ser um erro de comunicação com o dispositivo.
- Get – Verifica também as permissões do utilizador e envia para o Cliente o valor correspondente ao estado do dispositivo.
- SetFavorite – Envia para o servidor a informação referente a um dispositivo ou divisão da casa que o utilizador pretenda adicionar à sua lista de favoritos (Como a aplicação Cliente não guarda estado sobre quaisquer operações depois de desconectado do servidor, esta operação é útil para que os favoritos sejam guardados no servidor e quando o utilizador se voltar a ligar ao servidor (independentemente da aplicação Cliente utilizado) carregar a especificação da casa que traz a informação dos dispositivos favoritos.

É possível encontrarem-se vários Clientes ligados ao mesmo tempo ao Servidor e a alterar o estado das mesmas propriedades, uma vez que os pedidos são efectuados à medida que chegam a última alteração é que será observada no dispositivo.

O estado do sistema é simulado através de uma entidade denominada de XMLStateParse que carrega as especificações das casas, em que é dado um valor a cada propriedade dos dispositivos.

3. O Modelo

O modelo usado na especificação do sistema é mesmo o dado pelo *Modelo de Especificação e Programação de um Sistema Domótico [Figura1]*

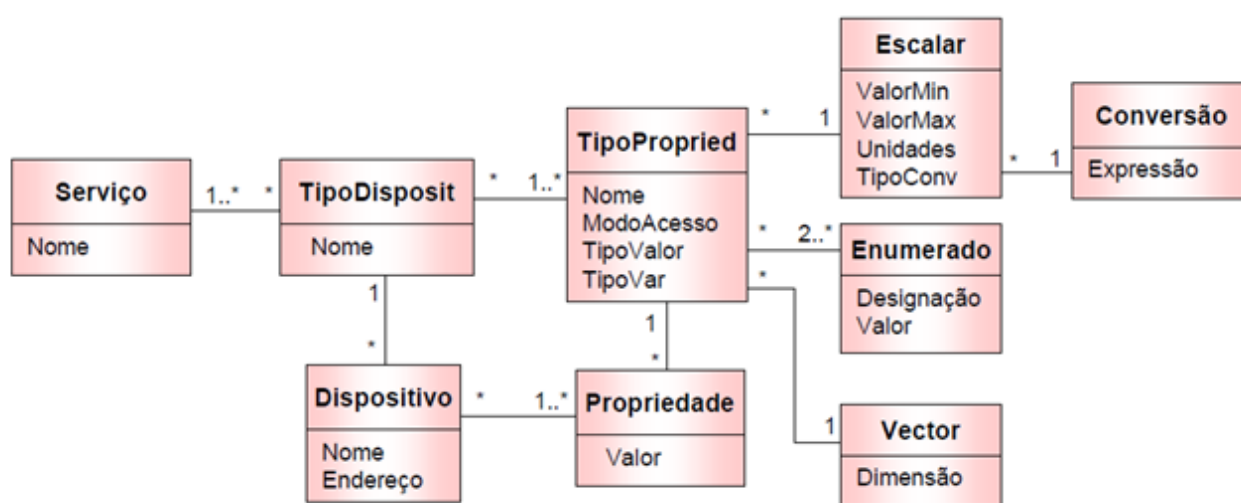


Figura 1. Modelo de um dispositivo domótico [1]

Para fazer uma aplicação que funcione com a mesma lógica que o Sistema Base DomoBus, foi preferido implementar uma solução que não fosse muito diferente deste. Este facto contribui também para facilitar o controlo sobre cada componente e principalmente criar uma solução genérica e o mais reutilizável possível.

Para representar o Utilizador foi adicionado ao modelo a Entidade “User”. Esta é constituída pelo Nome(*Name*), Password e Nível de Acesso(*AccessLevel*) com pode ser visto na Figura 2.

O Modelo Base da especificação do Sistema de uma habitação que também é discriminado na documentação do DomoBus esta presente na Figura 2, mas com algumas sugestões de alteração para inquirir a funcionalidade relativamente aos Favoritos.

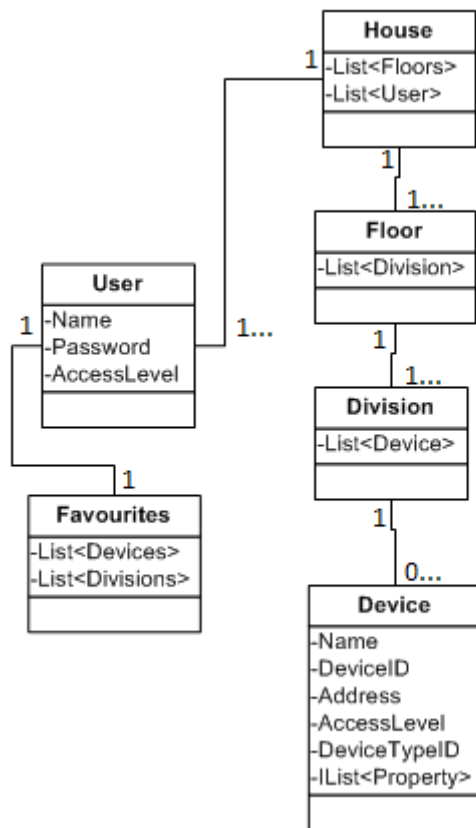


Figura 2. Modelo Base de uma habitação e respectivos dispositivos

Para simplificar a implementação é considerado que cada casa tem utilizadores registados, assim se um utilizador estiver registado em duas casas diferentes, pode ter IDs diferentes. Para guardar e aceder aos favoritos foi adicionado a Entidade Favoritos (*Favorites*). Esta é constituída pelos conjuntos de referências para Dispositivos(*Devices*) e Divisões(*Divisions*).

Para especificação XML do Sistema DomoBus é apresentada a seguinte alteração:

```

<FavoriteList>
  <Favorite FavoriteType="DEVICE" RefUser="1" FavoriteTypeID="1"/>
  <Favorite FavoriteType="DIVISION" RefUser="1" FavoriteTypeID="2"/>
</FavoriteList>

```

Nas quais o FavoriteType designa o tipo do favorito, o RefUser é o identificador do utilizador e o FavoriteTypeID é o identificador no contexto do favorito (no caso de ser um dispositivo ou uma divisão).

A Arquitectura

Cada módulo principal está dividido em componentes de forma a tornar a implementação modular e para facilitar no desenvolvimento e testes.

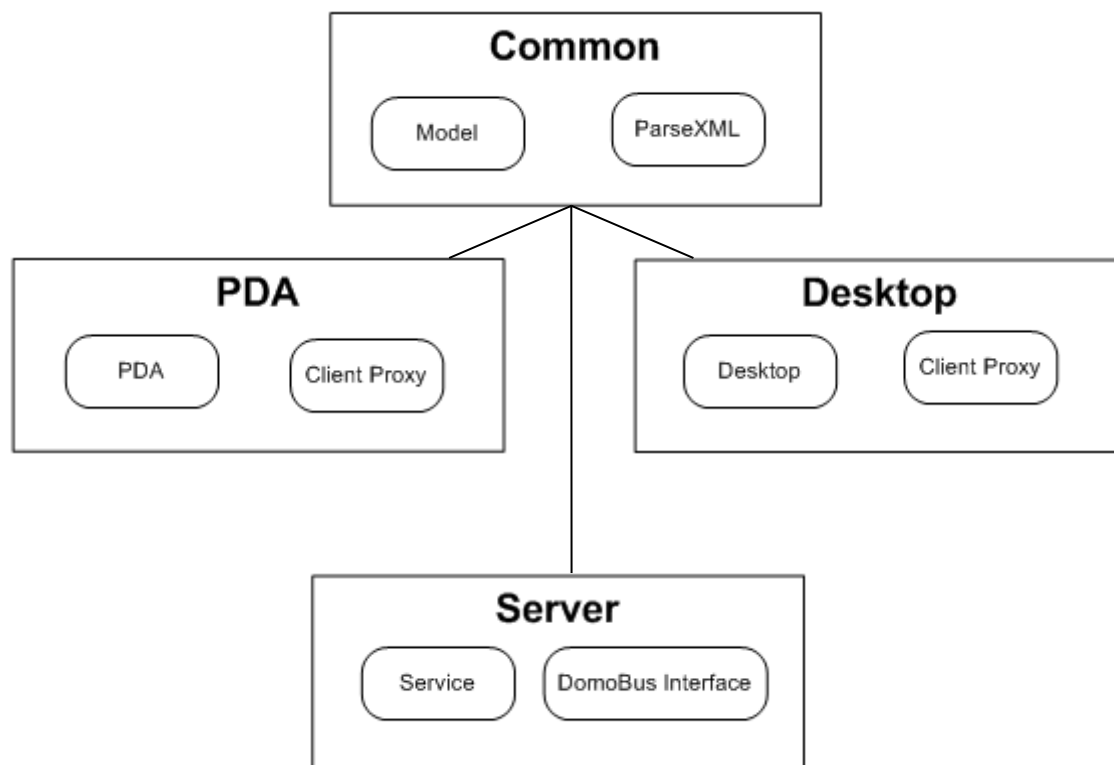


Figura 3. Arquitectura do Sistema DomoMobile

No topo temos o módulo **Common**, que contém o Modelo de Domínio, que vai ser usada por todas as aplicações, juntamente com a componente responsável pelo Parse* dos ficheiros XML que abrangem especificações das casas.

No módulo **PDA** encontra-se a lógica associada às funcionalidades e à interface da aplicação. Outra componente é o Client Proxy, aqui encontram-se os mecanismos de configuração da comunicação com o Servidor.

No módulo **Desktop**, tal como acontece com o PDA, tem também divididas as componentes de configuração da comunicação e interface.

Na componente Service do módulo **Server**, reside a definição do Serviço, a lógica de tratamento de pedidos, verificação de permissões, etc. E na componente DomoBus Interface encontra-se a lógica

*Parse – procedimento que consiste em validar e fazer a conversão de um ficheiro xml em classes.

associada à conversão de dados e comunicação com a Rede DomoBus. Neste módulo podem também ser empregues extensões, para disponibilizar o acesso às funcionalidades que o Serviço oferece, das mais variadas formas (ex. Interface IIS para aceder ao mesmo serviço usando a Internet).

A Implementação

Para a implementação das aplicações foi usada a linguagem de programação C# juntamente com a plataforma .NET e .NET Compact Framework 3.5 para dispositivos com Windows Mobile.

A interface do Cliente PDA é baseada em Windows Forms e a interface do Cliente Desktop é feita com base na tecnologia WPF(*Windows Presentation Foundation*). São notórias as diferenças na implementação da lógica nas duas aplicações, no caso do Cliente PDA foi feita uma abordagem um pouco menos complexa devido às limitações da plataforma. De forma a minimizar o uso de memória na UI para WM6 foi utilizado apenas um Form, no qual os itens que o constituem são reciclados ao longo de todos os ecrãs apresentados ao utilizador. Esta abordagem faz com que a aplicação fique mais leve e que o programa seja apenas mais lento no arranque, pois todos os objectos são carregados para memória.



Figura 4. Interface DomoMobile

A *figura 4* mostra a primeira interação com o utilizador, em que é necessário especificar o endereço IP onde o Serviço se encontra disponível. Depois a aplicação tenta estabelecer uma ligação ao Servidor e se este falhar informa o utilizador, se for bem sucedido é pedido ao utilizador que introduza as suas credenciais e só depois este tem possibilidade de efectuar pedidos ao Servidor para alterar o estado de uma casa.

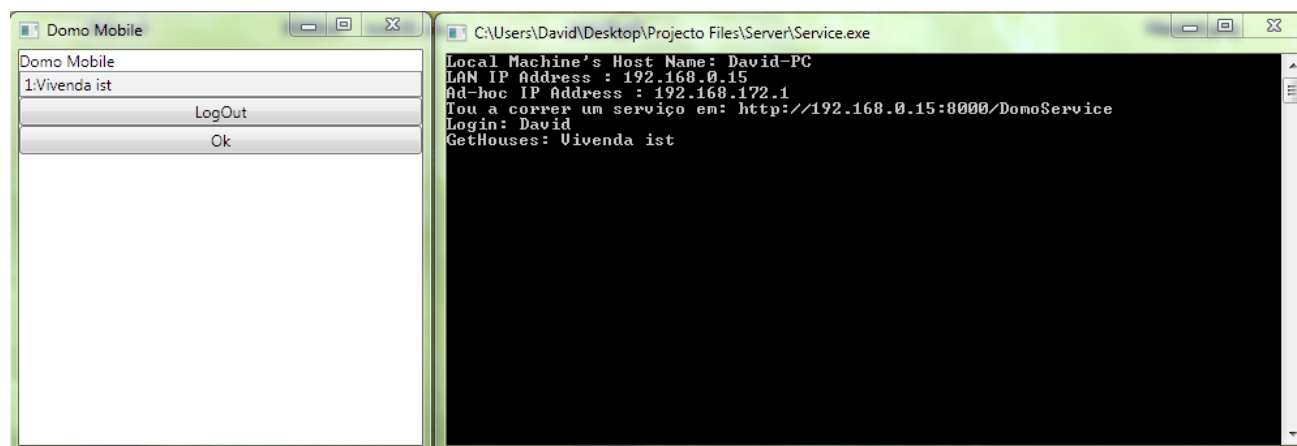


Figura 5. Interface Desktop DomoMobile(esquerda) e Servidor (direita).

Para o cliente Desktop foi tomada uma abordagem um pouco mais complexa que requer mais processamento, os objectos são criados dinamicamente e posteriormente destruídos quando já não são necessários. Mais especificamente foi utilizado o padrão **Model-View-ViewModel** que é um padrão específico para a tecnologia **WPF**. Neste padrão a interface gráfica é suportada por uma classe ViewModel que tem toda a lógica de negócio. A classe ViewModel usa as classes do modelo de dados para representar a informação. Assim os ecrãs encontram-se definidos em **xml** e cada um deles tem “por trás” uma classe com a sua lógica. Este conceito também se aplica a elementos dentro dos ecrãs, no caso dos itens que aparecem na listagem estes têm o seu visual definido em **xml** e uma classe que contém a sua lógica.

Na comunicação usado é **WCF** (Windows Communication Foundation) que é uma nova tecnologia que permite a comunicação entre ambientes .NET.

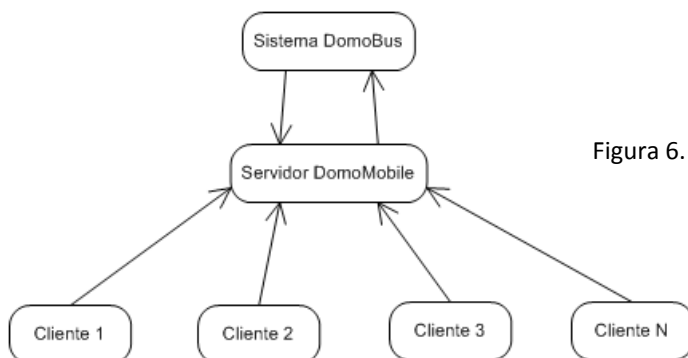


Figura 6. Comunicações entre componentes

Na *figura 6* encontra-se representadas as comunicações entre as várias componentes do sistema, notar que as setas indicam a direcção na qual a comunicação é feita. Os Clientes podem ser do tipo PDA ou Desktop.

Para criar os módulos de proxy dos clientes foram utilizadas as ferramentas: **SvcUtil.exe** (*ServiceModel Metadata Utility Tool*) para o cliente Desktop e **NetcfSvcUtil.exe** para o cliente PDA. As ferramentas reconhecem o serviço criado e geram automaticamente as configurações necessárias para estabelecer a ligação ao Servidor.

Avaliação da Implementação

O sistema base encontra-se a funcionar mas poderiam ser acrescentadas um maior número de funcionalidades num curto espaço de tempo. O factor que condicionou o avanço da implementação da solução foi a aprendizagem das tecnologias.

Algumas condicionantes do projecto são:

- O servidor poderia guardar o estado dos dispositivos numa base de dados, e assim, em caso de falha poderia recuperar o estado sem ter que aceder a todos os dispositivos.
- A comunicação é feita apenas num sentido e um pouco restritiva.
- A implementação dos favoritos não esta completa ao nível da interface.
- A apresentação na interface dos dispositivos pode ser melhorada, podendo ser possível mostrar em qualquer altura a lista de todos os dispositivos presentes na casa ou piso.

Referências

[1] Modelo de especificação e Programação de um Sistema Domótico

<http://domobus.net/papers/>

[2] Especificação XML de um Sistema DomoBus

<http://domobus.net/papers/>

[3] Paul Yao, David Durant (2009) *Programming .NET Compact Framework 3.5, Second Edition*
Addison-Wesley

[4] An Introduction to WCF for Device Developers

<http://community.opennetcf.com/articles/cf/archive/2007/11/29/an-introduction-to-wcf-for-device-developers.aspx>

[5] Chris Sells, Ian Griffiths (2007) *Programming WPF, Second Edition*
O'Reilly Media