

# PIPE: Uma infra-estrutura genérica de serviços para ambientes de computação ubíqua

Bruno Félix, Nuno Preguiça

CITI / Dep. de Informática - Faculdade de Ciências e Tecnologia  
Universidade Nova de Lisboa,  
Quinta da Torre, 2829 -516 Caparica, Portugal

**Resumo** Nas últimas décadas tem-se assistido a um crescimento significativo do número e poder dos diferentes dispositivos computacionais. Este crescimento traduz-se na coexistência de duas grandes classes de equipamentos. Por um lado aqueles que permeiam de forma quase ubíqua os espaços que frequentamos e por outro lado, um grande número de equipamentos móveis que acompanham no utilizadores no seu dia-a-dia. No entanto, estas duas classes de dispositivos têm existido em mundos largamente separados. Neste artigo apresenta-se o PIPE, um sistema ubíquo para espaços públicos que permite a utilizadores equipados com dispositivos móveis usufruir dos recursos (e.g., ecrãs partilhados, capacidade computacional) fornecidos pelo sistema, podendo correr um conjunto de aplicações pré-definidas, ou as suas próprias aplicações na infra-estrutura, estendendo a capacidade dos dispositivos portáteis.

**Key words:** computação ubíqua, dispositivos móveis, ecrãs partilhados, execução remota.

**Resumo** In the last decades we have witnessed a significant growth in the number and power of computing devices. These can be divided in two broad categories. First, the devices that pervade most of the spaces we use, and second the mobile equipments carried by the users in their day-to-day life. However, up until recently both these classes of devices coexisted without much interaction. In this document, we present PIPE, an infrastructure aimed at public spaces that allows users equipped with mobile devices, to take advantage of the provided resources (e.g. shared displays, computing power), allowing them to run existing, or their own distributed applications, in machines that are part of the infrastructure.

**Key words:** pervasive computing, mobile devices, shared displays, cyber foraging.

## 1 Introdução

A crescente ubiquidade dos dispositivos computacionais é uma realidade que tem vindo a alterar profundamente o nosso modo de vida. Estes têm-se tornado

cada vez mais pequenos, potentes e baratos, sendo por isso bastante comum encontrar espaços onde coexistem dispositivos pertencentes à infra-estrutura (computadores, ecrãs, projectores, etc.) e dispositivos pessoais dos utilizadores (telemóveis, PDAs, etc) [1].

Até ao momento estas duas classes de dispositivos têm estado largamente dissociadas. No entanto, a sua combinação pode abrir oportunidades interessantes para criar novas aplicações e serviços. Estas aplicações, podem usar os dispositivos móveis como meios privilegiados para aceder aos recursos que existam num determinado espaço e para manter informação pessoal do utilizador, ao mesmo tempo que tiram partido do poder de processamento, resolução e largura de banda que os dispositivos instalados na infra-estrutura oferecem.

A união destas duas classes de dispositivos levanta um conjunto de novos desafios importantes [11,8]. Entre estes, encontra-se a necessidade de combinar um conjunto diversificado de meios computacionais, que podem estar distribuídos por diferentes dispositivos de forma a fornecer a informação desejada ao utilizador final. Este problema consiste não só na definição da arquitectura que permite compor aplicações a partir dum conjunto distribuído de recursos, mas também na forma como estes serviços podem ser instanciados e migrados de forma a executarem nos dispositivos mais apropriados. Por exemplo, uma aplicação que apresente ao utilizador informação obtida a partir de múltiplas fontes web pode ter um módulo a executar na infra-estrutura para executar este processamento, usufruindo de uma melhor conectividade e capacidade de processamento.

Neste contexto, este artigo propõe uma plataforma que permita a criação de aplicações que utilizam os recursos disponíveis na infra-estrutura para fornecer um melhor serviço aos utilizadores finais. Este sistema, designado de PIPE (Pervasive Infrastructure for Public Environments), permite que um utilizador munido apenas do seu dispositivo móvel possa tirar partido dos recursos computacionais disponibilizados pelas várias máquinas que compõem esta infra-estrutura.

No sistema PIPE executa um conjunto de serviços e aplicações. Um serviço disponibiliza uma funcionalidade específica que pode ser utilizada apenas por outros serviços ou aplicações. As aplicações fornecem um serviço ao utilizador final. Enquanto os serviços executam apenas nos nós da infra-estrutura, as aplicações podem também executar nos dispositivos dos utilizadores. Quer as aplicações quer os serviços podem ser instaladas dinamicamente nos dispositivos apropriados.

A nossa visão é que este sistema possa executar num espaço público (e.g. um centro comercial), onde existem à partida uma grande quantidade de dispositivos computacionais espalhados pela infra-estrutura e um número substancial de utilizadores. O sistema incluirá um conjunto pré-definido de aplicações e serviços disponibilizados pelo espaço público em que executam. Os utilizadores podem aceder a estes serviços, fazendo o carregamento dinâmico das aplicações nos seus dispositivos se necessário.

Ao contrário de outras soluções para este tipo de ambiente [7,6,10], a nossa solução permite ainda carregar dinamicamente aplicações do utilizador na infra-

estrutura. Esta funcionalidade possibilita uma maior flexibilidade, não ficando o utilizador limitado a usar as aplicações disponibilizadas pelo espaço público, mas podendo usar as suas próprias aplicações.

O resto deste documento está estruturado da seguinte forma. A secção 2 descreve o trabalho relacionado. A secção 3 discute brevemente os requisitos relevantes para um sistema desta natureza. A secção 4 apresenta o desenho do sistema PIPE. Na secção 5 descrevem-se duas aplicações que usam o sistema PIPE, seguindo-se a avaliação do sistema e algumas conclusões finais nas secções 6 e 7 respectivamente.

## 2 Trabalho relacionado

Os avanços tecnológicos permitiram a criação de sistema de computação ubíqua [14,11]. Estes sistemas exibem uma grande variabilidade, sendo possível identificar duas categorias especialmente relevantes para o tipo de plataforma apresentada neste artigo. Por um lado, aplicações específicas que tiram partido de computadores presentes no espaço em conjunto com os dispositivos móveis disponíveis. Por outro lado, plataformas ubíquas para o fornecimento de serviços, que permitem aos utilizadores aceder a um conjunto mais vasto de recursos e aplicações.

Na primeira categoria podemos englobar diferentes trabalhos - e.g. [13,5]. Um exemplo representativo é o sistema Bluscreen [5], que permite mostrar publicidade, ou outros conteúdos informativos em ecrãs públicos. Estes conteúdos são escolhidos tendo em conta os dispositivos bluetooth detectados nas proximidades. Este sistema não requer software específico por parte dos utilizadores, tendo em consideração nos conteúdos mostrados o facto dos utilizador já os terem observado anteriormente ou não. Assim, este sistema funciona em situações em que à partida não se conhecem os interesses e tipologia dos utilizadores. O sistema PIPE permite criar uma solução semelhante.

A segunda categoria engloba sistema ubíquos mais complexos e com diferentes objectivos. O Celadon [7] é um exemplo deste tipo de plataformas, que visa criar uma infra-estrutura ubíqua que permita aos dispositivos móveis presentes num dado espaço tirar partido dos recursos e serviços que a infra-estrutura fornece. Os serviços disponibilizados encontram-se agregados em zonas, que correspondem a espaços concretos (e.g. estações de metro, centros comerciais, etc).

O AlfredO[10] apresenta uma arquitectura orientada aos serviços, para suportar o controlo e execução de serviços fornecidos pela infra-estrutura a partir do dispositivo pessoal do utilizador. Os serviços são decompostos em três camadas (dados, lógica e visualização), podendo uma ou várias destas camadas ser transferidas e executadas no cliente móvel, oferecendo um bom grau de flexibilidade.

O PIPE foi influenciado pela abordagem seguida por estes dois sistemas, na medida em que oferece uma solução baseada em aplicações disponibilizadas pelo sistema, que podem ser consumidas pelos utilizadores. No entanto, permite também que estes usem a infra-estrutura para instalar e correr as suas próprias aplicações.

Recentemente, vários trabalhos exploraram a possibilidade dos utilizadores de computadores portáteis usarem os recursos disponíveis na infra-estrutura para executarem computações arbitrárias - e.g., [3,12].

O Slingshot [12] permite a execução de código arbitrário em máquinas de um *hot-spot*, através da replicação de máquinas virtuais em que correm as aplicações dos utilizadores. Essas réplicas são obtidas a partir de um servidor, bem conhecido pelo utilizador, que actua como réplica primária e é usado como repositório “confiável” do estado da aplicação. A solução proposta neste artigo é mais leve, e baseada na execução de programas Java. Adicionalmente, ao contrário do Slingshot, as novas aplicações ficam integradas no ambiente de computação ubíqua, podendo aceder aos serviços disponibilizados no sistema.

### 3 Requisitos funcionais

Nesta secção iremos abordar brevemente alguns dos requisitos funcionais que identificámos e que foram determinantes em algumas escolhas que fizemos aquando do processo de desenho da nossa solução.

**Custo:** A implementação de uma arquitectura para o fornecimento de serviços ubíquos não deve comportar um custo muito elevado. De preferência deve poder tirar partido de muito do hardware que já existe espalhado pelos diferentes locais, e também não deve exigir equipamento específico da parte dos clientes. Mesmo nestas condições, uma infra-estrutura deste tipo terá alguns custos. O estudo dos modelos de negócio associados à disponibilização deste tipo de infra-estrutura está fora do âmbito deste trabalho, mas poderia passar pelo suporte pelo espaço público como forma de atracção dos cliente, usando receitas de publicidade apresentada na infra-estrutura ou usando um modelo de partilha de custos com os utilizadores.

**Flexibilidade:** Uma plataforma desta natureza deve não só contemplar os mecanismos para fazer uma gestão fácil dos serviços que a infra-estrutura oferece, mas também permitir ao utilizador executar as suas próprias aplicações ou serviços.

**Tempo de resposta:** Tendo em conta que esta plataforma tem um modelo que é mais facilmente aplicável a espaços públicos, como por exemplo centros comerciais, a questão do tempo de resposta é bastante importante. Neste tipo de ambiente, não é expectável que o utilizador tenha disposição para esperar bastante tempo para interagir com a infra-estrutura, pelo que esse facto deverá ser considerado.

**Segurança e auditabilidade:** Tendo em conta que há a possibilidade dos utilizadores correrem as suas próprias aplicações a questão da segurança torna-se particularmente premente. Neste contexto existem vários problemas distintos, dos quais se destacam os seguintes.

Primeiro, a protecção da infra-estrutura face à execução de software arbitrário. Este foi o único problema tratado de forma completa no nosso sistema, usando um modelo de *sandbox*, em que as aplicações executam de forma isolada recorrendo aos mecanismos de segurança do sistema Java.

Segundo, o controlo de acessos de uma aplicação relativamente aos serviços que pode utilizar. Para lidar com este problema seria possível combinar a utilização do sistema *Kerberos* com um sistema de *capacidades*, em que um cliente apresentasse aos vários serviços um *ticket* com as suas permissões.

Finalmente, é interessante providenciar mecanismos que permitam certificar ou auditar os resultados das computações efectuadas na infra-estrutura, de forma a evitar ou identificar utilizações ilícitas dos recursos disponibilizados (e.g. download de conteúdos ilegais), eventualmente procedendo à responsabilização dos utilizadores envolvidos.

## 4 Desenho

A nossa solução visa implementar uma infra-estrutura ubíqua, que faça uso de equipamento comum (e barato) que satisfaça os requisitos apresentados anteriormente. O objectivo consiste em criar uma arquitectura que dê suporte a um cenário em que os utilizadores interagem com a infra-estrutura presente através dos seus dispositivos móveis, tirando partido das aplicações e recursos existentes, e escondendo algumas das limitações dos seus equipamentos pessoais. Esta, permite não só a existência de um conjunto bastante vasto de serviços pré-definidos, mas também dá a possibilidade ao utilizador para executar as suas próprias aplicações tirando partido dos recursos disponíveis.

A arquitectura do PIPE pode ser decomposta em duas grandes componentes. Os nós de computação propriamente ditos (*infrastructure nodes*) e os dispositivos móveis dos utilizadores, que através de uma aplicação específica podem ter acesso a todos os recursos e funcionalidades fornecidos pela infra-estrutura. Esta última componente não será obrigatória já que poderão existir aplicações que não requeiram uma interacção explícita com o utilizador (ou o seu dispositivo) ou que inclusivamente usem outras técnicas (e.g. interacção gestual) para o fazer.

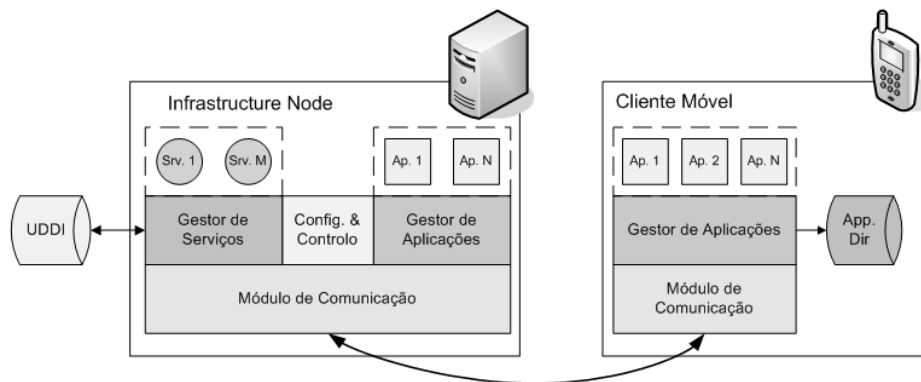
Tipicamente um utilizador que disponha apenas do seu dispositivo móvel com a aplicação cliente instalada acede ao sistema através da ligação a um *infrastructure node*. Após estar ligado ao sistema, o cliente passa a poder utilizar as funcionalidade disponibilizadas (não só os serviços existentes em qualquer nó da infra-estrutura, mas também a possibilidade de instalar novas aplicações).

De seguida, detalham-se as componentes do sistema.

### 4.1 Infrastructure Node

O *infrastructure node* corresponde a uma máquina presente na infra-estrutura e na qual podem executar serviços e aplicações. Esta componente foi implementada usando a linguagem Java, e pode dividir-se em três módulos (fig. 1): módulo de comunicação, gestor de aplicações e gestor de serviços.

O **módulo de comunicação** é responsável por gerir as comunicações com os clientes móveis, sendo actualmente suportadas comunicações via bluetooth ou TCP/IP (sobre Wi-Fi, GPRS, etc.)



**Figura 1.** Arquitectura interna do infrastructure node e do cliente móvel

O **gestor de aplicações** é responsável por fazer toda a gestão das aplicações residentes na infra-estrutura e das aplicações transferidas pelos utilizadores. Estas aplicações podem fazer uso dos diversos serviços que são fornecidos pela plataforma (e.g. serviço de câmaras, serviço de reconhecimento de faces), implementando o sistema uma arquitectura orientada aos serviços.

Para permitir o controlo por parte do dispositivo móvel do utilizador, as aplicações podem definir uma interface Web que fica disponível quando a aplicação é iniciada, ou podem ter associada uma aplicação móvel (Java 2 ME) que é instalada automaticamente caso seja necessário.

Todas as aplicações executam num contexto protegido, tendo associado um conjunto de permissões diferentes, consoante serem aplicações da infra-estrutura ou instaladas dinamicamente. Para tal, usam-se os mecanismo de segurança do Java.

A instalação dinâmica de uma aplicação ocorre por pedido dum cliente móvel. No pedido, especifica-se um conjunto de fontes alternativas do código a instalar - tipicamente, o URL dum servidor e o próprio cliente móvel.

Finalmente, o **gestor de serviços** gere os serviços disponíveis às aplicações, sendo responsável pelo controlo dos serviços que correm localmente no nó, bem como pelo seu registo no directório global de serviços. Na nossa implementação, este directório é um servidor de registo UDDI [2], global à infra-estrutura em que o sistema executa.

Quando uma aplicação necessita dum serviço, esta envia um pedido ao gestor de serviços local ao *infrastructure node* onde se encontra a correr, que por seu turno verifica se tal serviço consta no directório. Se tal acontecer, então é feita a ligação entre o código cliente do serviço que faz parte da aplicação e o serviço propriamente dito. Na operação de pedido de serviço, a aplicação poderá especificar algumas preferências, em particular se o serviço requerido deverá ser local ao nó onde esta corre ou não. É ainda suportada a composição de serviços, pelo

que os serviços podem usar outros serviços de forma a fornecer às aplicações operações de mais alto nível.

## 4.2 Cliente Móvel

A aplicação do sistema que executa no cliente móvel gere as aplicações dos utilizadores e as suas comunicações com a infra-estrutura. No nosso protótipo esta foi implementada em Java 2 ME. Internamente é composta por dois módulos (fig. 1): um módulo de comunicações e um módulo de gestão de aplicações.

O módulo de comunicações é responsável por gerir as ligações do dispositivo móvel com a infra-estrutura e todas as subsequentes comunicações, usando os meios de comunicação disponíveis para o efeito.

O gestor de aplicações tem duas funções principais. Primeiro, é responsável por manter as aplicações dos utilizadores. Quando iniciadas pelo utilizador, estas aplicações podem executar localmente ou na infra-estrutura. Caso seja necessário executar as aplicações (ou parte delas) na infra-estrutura, este módulo é responsável por interagir com a infra-estrutura, de forma a iniciar a aplicação num *infrastructure node* apropriado.

Segundo, é responsável por contactar a infra-estrutura e apresentar ao utilizador a lista de aplicações que estão disponíveis. Como se explicou anteriormente, o acesso a uma destas aplicações pode-se efectuar usando uma interface Web ou através da execução duma aplicação móvel instalada automaticamente.

## 5 Aplicações de teste

Nesta secção apresentam-se duas aplicações demonstrativas da utilização do sistema desenvolvido.

### 5.1 Aplicação de publicidade

A primeira aplicação tem como objectivo apresentar publicidade numa zona comercial (e.g. centro comercial). Para tal, assume-se que existe um conjunto de *infrastructure nodes* distribuídos pelo espaço. Adicionalmente, existem alguns ecrãs que podem ser usados para apresentar informação e que são controlados pelo sistema. Finalmente, existem câmaras de vigilância que cobrem a zona comercial.

Para que um sistema deste tipo seja o mais eficaz possível, a aplicação de visualização de publicidade deve usar a informação sobre os utilizadores detectados nas proximidades dos diferentes nós da infra-estrutura para adequar os conteúdos mostrados ao público presente [9]. Neste caso, a escolha dos anúncios tem como base as características do local onde o nó se encontra e o histórico da deslocação dos utilizadores no espaço.

A publicidade pode ser mostrada de duas formas distintas, num ecrã público associado ao *infrastructure node*, ou directamente no telemóvel do utilizador. Pensamos que estas duas técnicas se podem complementar na medida em que

numa situação em que são detectados poucos utilizadores (e.g 2 ou 3), pode não ser desejável mostrar a publicidade no ecrã público, já que é fácil perceber a quem é que esse anúncio se dirige. Nesta situação, consideramos que é desejável que os conteúdos sejam enviados directamente para o dispositivo móvel do utilizador. Actualmente os conteúdos mostrados assumem a forma de imagens estáticas, mas a aplicação está preparada para utilizar outros formatos multimédia.

A aplicação executa na infra-estrutura e faz uso de dois serviços: um serviço de publicidade e um serviço de adaptação de conteúdos.

O primeiro, combina um serviço de câmaras, que recolhe imagens das proximidades do nó, com um serviço de detecção de faces baseado na solução desenvolvida por Grangeiro et. al. [4], a executar num nó da infra-estrutura, de forma a contabilizar o número de pessoas num dado local. Adicionalmente é usado um serviço de detecção bluetooth que permite descobrir os dispositivos próximos do *infrastructure node*. O serviço de publicidade oferece à aplicação uma listagem dos utilizadores detectados (identificados por endereço bluetooth) e respectivas preferências com base no histórico das suas deslocações.

Esta informação é usada no contexto da aplicação para efectuar a escolha do anúncio a mostrar no ecrã público, ou caso sejam detectados poucos utilizadores, para escolher para cada utilizador o anúncio a enviar. Nesta última situação é utilizado o serviço de adaptação de conteúdos, que permite adaptar os anúncios a enviar antes de iniciar a transferência para um dispositivo móvel. No protótipo actual, uma vez que se tratam de imagens, essa adaptação consiste na redução do seu tamanho original.

Esta aplicação é exemplificativa duma aplicação que corre na infra-estrutura, sem necessidade duma interacção explícita por parte do utilizador. Adicionalmente, demonstra a criação duma aplicação que utiliza um conjunto variado de serviços, alguns dos quais executam necessariamente numa localização (e.g. detecção de bluetooth, câmara), enquanto outros podem executar em qualquer localização do sistema (e.g. serviço de adaptação).

## 5.2 Aplicação de informação estendida sobre produtos

Esta aplicação visa demonstrar a viabilidade de aplicações que corram as componentes computacionalmente mais exigentes na infra-estrutura. Neste caso concreto, a aplicação móvel é capaz de recolher, através da câmara do dispositivo, a informação do código de barras de um produto e enviar esses dados para a componente que reside na infra-estrutura. Esta contacta diversas fontes de dados presentes na Web e oferece um mashup dessa informação ao cliente. Após estes dados serem recebidos na aplicação móvel, o utilizador pode ver informação adicional sobre o produto em causa, bem como produtos relacionados ou vídeos que sejam relevantes.

A aplicação móvel foi construída em Java 2 ME, e pode subdividir-se em três grandes módulos: captura, pesquisa e mashup.



O módulo de captura, faz uso da biblioteca ZXing<sup>1</sup> para efectuar o reconhecimento dos códigos de barras. O módulo de pesquisa, recebe a informação do módulo anterior e com base nas preferências do utilizador contacta o módulo de mashup que pode correr localmente ou na infra-estrutura e é responsável pela obtenção dos dados relativos ao produto e vídeos relacionados.

No módulo de mashup, quando é recebido um pedido relativo a um produto, este contacta os serviços da Amazon<sup>2</sup> para obter informação relativa ao produto em causa, incluindo os produtos relacionados, e os serviços do YouTube<sup>3</sup> para obter vídeos relacionados.

Adicionalmente, é possível ao utilizador, efectuar o download dos diferentes vídeos fornecidos. Caso escolha fazer o download do vídeo quando o módulo de mashup corre na infra-estrutura, este faz a adaptação dos conteúdos para um formato adequado ao dispositivo móvel em causa, usando um serviço de adaptação. Mais concretamente, como se tratam de vídeos o serviço de adaptação usa a biblioteca ffmpeg<sup>4</sup> para o efeito.

## 6 Avaliação

Nesta secção avalia-se o benefício de executar parte duma aplicação na infra-estrutura, usando como teste a aplicação de informação estendida sobre produtos. Os resultados apresentados comparam o acesso à mesma informação no dispositivo móvel, acedendo directamente às fontes de dados ou transferindo parte da aplicação para a infra-estrutura.

Foram realizados dois testes distintos que mediram o tempo até à apresentação da informação e a quantidade de dados transferidos no processo. No primeiro teste, a informação acedida era relativa a um produto. No segundo teste, a informação acedida consistia num vídeo.

### 6.1 Contexto da experiência:

Nos testes foram utilizados os seguintes equipamentos. Como cliente móvel usou-se um telemóvel Nokia N82, com o sistema operativo Symbian OS 9.2. Neste caso, usou-se apenas um *infrastructure node*, a executar num PC equipado com um processador AMD Turion 64x2, 2GB de RAM e Windows Vista. O computador estava equipado com um adaptador bluetooth (versão 2.0 EDR), um adaptador wi-fi (802.11b/g) e com uma ligação à Internet de 20Mbps, através de wi-fi.

### 6.2 Testes

**Teste 1:** O objectivo do primeiro teste foi avaliar o tempo que demora a mostrar a informação relativa a um produto, quantificando os dados que foram transferidos para o dispositivo móvel e para a infra-estrutura (caso se aplique). Esta

<sup>1</sup> <http://code.google.com/p/zxing/>

<sup>2</sup> <http://docs.amazonwebservices.com/AWSEcommerceService/2005-03-23/>

<sup>3</sup> <http://code.google.com/intl/pt-PT/apis/youtube/overview.html>

<sup>4</sup> <http://www.ffmpeg.org>

	Tempo (ms)	Transferências (KB)	
		Infra-estrutura	Móvel
Stand-alone	28441	0	147546
Bluetooth	2442	147546	13256
Wi-fi	4671	147546	13256

**Tabela 1.** Tempos médios de execução e quantidade de dados transferida no decorrer do primeiro teste.

informação consiste numa imagem do produto, o seu nome e respectiva classificação, uma listagem de outros produtos relacionados, bem como informação detalhada sobre o vídeo sugerido (título, duração e imagem) para além de uma lista de outros vídeos relacionados. Este teste foi efectuado em três modalidades distintas, primeiro com todo o processamento efectuado no dispositivo móvel. Seguidamente usando a infra-estrutura para contactar os diversos serviços e transferindo os resultados através de bluetooth e por último, usando também a infra-estrutura mas transferindo os dados via wi-fi. Os resultados obtidos apresentam-se na tabela 1.

Estes resultados mostram uma redução considerável do tempo necessário a obter a informação pretendida quando se usa a infra-estrutura. Esta redução no tempo de resposta fica a dever-se a dois factores: por um lado a transferência de uma menor quantidade de dados para o dispositivo móvel, reduzindo por isso o peso das comunicações. Este facto deve-se a que apenas uma parte da informação obtida nos serviços web acedidos é relevante para a aplicação. Por outro lado, o processamento desses dados, que são XML, requer algum poder computacional e por isso também é mais moroso no cliente móvel, sendo bastante mais rápido quando efectuado na infra-estrutura.

**Teste 2:** O segundo teste avalia os potenciais ganhos de se usar a infra-estrutura para efectuar a adaptação de conteúdos com alta qualidade, antes destes serem transferidos para o dispositivo móvel. Este teste obrigou a uma ligeira alteração da aplicação, na medida em que considerámos mais interessante efectuar as medições com base no download de um vídeo em alta definição relativamente a vídeos do YouTube. À semelhança do teste anterior efectuámos esta experiência em três modalidades distintas. Na primeira descarregámos directamente o vídeo para o dispositivo móvel, ao passo que nas seguintes, o dispositivo móvel contactou a infra-estrutura que por seu turno efectuou o download e adaptou o vídeo ao dispositivo móvel. Na segunda experiência a transferência dos conteúdos adaptados foi feita usando bluetooth, ao passo que na terceira usámos Wi-Fi. Na tabela 2 encontra-se o resumo dos dados obtidos.

Mais uma vez, estes resultados mostram uma redução considerável do tempo necessário a obter a informação pretendida quando se usa a infra-estrutura. Os resultados obtidos confirmam que as comunicações para o telemóvel incorrem num custo considerável, mesmo quando se usa Wi-Fi. Como se pode verificar, para o download do mesmo ficheiro de cerca de 108MB, o tempo de transferência

	Tempo (s)				Transferências (MB)		
	Total	Download	Transcoding	Transf.p/móvel	Total	Infra-estrutura	Móvel
<b>Stand-alone</b>	<b>351,81</b>	351,81	0,00	0,00	<b>107,91</b>	0,00	107,91
<b>Bluetooth</b>	<b>222,56</b>	54,88	98,21	69,47	<b>112,88</b>	107,91	4,97
<b>Wi-fi</b>	<b>178,58</b>	54,88	98,21	25,49	<b>112,88</b>	107,91	4,97

**Tabela 2.** Tempos médios e quantidade de dados transferida para as várias entidades no segundo teste.

na infra-estrutura é cerca de um quinto. Este teste evidencia também o potencial da utilização de um serviço de adaptação de conteúdos, já que neste caso concreto, foi possível transformar um vídeo com alta qualidade e com cerca de 108MB, num vídeo em formato MPEG-4 com cerca de 5MB, sendo obviamente a sua transferência para o dispositivo móvel muito mais rápida, mesmo quando se considera o tempo necessário à adaptação dos dados, a qual na nossa solução apenas pode ser iniciado após concluir o download completo do ficheiro para a infra-estrutura.

Em suma, os resultados obtidos confirmam que a utilização da infra-estrutura para executar as componentes da aplicação mais exigentes em termos computacionais, ou de comunicação traz efectivamente vantagens. Essas vantagens traduzem-se numa redução bastante grande do tempo de resposta das aplicações e a redução das comunicações efectuadas pelos dispositivos móveis, o que se traduz igualmente numa potencial redução da energia consumida por estes dispositivos.

## 7 Conclusões

Neste documento descreveu-se o PIPE, uma infra-estrutura genérica de serviços para ambientes de computação ubíqua. O PIPE permite que utilizadores equipados apenas com dispositivos móveis acedam aos recursos e aplicações disponibilizadas por uma infra-estrutura de computação ubíqua disponibilizada num espaço público. Ao contrário da aproximação normal neste tipo de infra-estrutura, o PIPE permite ainda que um utilizador execute as suas próprias aplicações, ou parte delas, na infra-estrutura.

As aplicações de teste apresentadas mostram que o PIPE poderia ser usado num contexto de espaço público (e.g. centro comercial). Os resultados quantitativos obtidos mostram o benefício de executar parte das aplicações dos utilizadores na infra-estrutura, permitindo beneficiar da vantagem de uma maior capacidade de computação sem restringir o utilizador ao conjunto de serviços que a infra-estrutura disponibiliza.

Dado que o sistema PIPE se encontra ainda em desenvolvimento, como é natural, existem ainda algumas áreas a contemplar de forma a tornar esta solução mais completa. Em particular, a questão da segurança e autenticação dos utilizadores foi abordada apenas de forma preliminar. Outros aspectos interessantes

a abordar prendem-se com a utilização de diferentes técnicas para efectuar o emparelhamento do cliente móvel com o *infrastructure node* (e.g. utilizando a câmara do dispositivo móvel), ou com a partilha de ecrãs públicos entre vários utilizadores.

## Referências

1. International telecommunications union - statistics, 12 Jun. 2009. <http://www.itu.int/ITU-D/ict/statistics/>.
2. OASIS - Committees - OASIS UDDI Specifications TC, 9 Jun. 2009. <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>.
3. Sachin Goyal and John Carter. A lightweight secure cyber foraging infrastructure for resource-constrained devices. In *WMCSA '04: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 186–195, Washington, DC, USA, 2004. IEEE Computer Society.
4. Filipe Grangeiro, Rui Jesus, and N. Correia. Face recognition and gender classification in personal memories. In *ICASSP 2009 - 34th IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 04 2009.
5. M. Karam, T. Payne, and E. David. Evaluating bluscreen: Usability for intelligent pervasive displays. *Pervasive Computing and Applications, 2007. ICPCA 2007. 2nd International Conference on*, pages 18–23, July 2007.
6. Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Debbie Caswell, Philippe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris, John Schettino, Bill Serra, and Mirjana Spasojevic. People, places, things: web presence for the real world. *Mob. Netw. Appl.*, 7(5):365–376, 2002.
7. MC Lee, HK Jang, YS Paik, SE Jin, and S Lee. Ubiquitous device collaboration infrastructure: Celadon. In *SEUS-WCCIA '06: Proceedings of the The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, pages 141–146, Washington, DC, USA, 2006. IEEE Computer Society.
8. C. Narayanaswami, D. Coffman, M. C. Lee, Y. S. Moon, J. H. Han, H. K. Jang, S. McFaddin, Y. S. Paik, J. H. Kim, J. K Lee, J. W. Park, and D. Soroker. Pervasive symbiotic advertising. In *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 80–85. ACM, 2008.
9. Terry Payne, Ester David, Nicholas R. Jennings, and Matthew Sharifi. Auction mechanisms for efficient advertisement selection on public displays. In *Proceedings of European Conference on Artificial Intelligence*, pages 285–289, 2006.
10. Jan S. Rellermeyer, Oriana Riva, and Gustavo Alonso. Alfredo: an architecture for flexible interaction with electronic devices. In *Middleware '08: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, pages 22–41, New York, NY, USA, 2008. Springer-Verlag New York, Inc.
11. M. Satyanarayanan. Pervasive computing: vision and challenges. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 8(4):10–17, 2001.
12. Ya-Yunn Su and Jason Flinn. Slingshot: deploying stateful services in wireless hotspots. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 79–92, New York, NY, USA, 2005. ACM.
13. Ville Tuulos, Jürgen Scheible, and Heli Nyholm. Combining web, mobile phones and public displays in large-scale: Manhattan story mashup. *Pervasive Computing*, pages 37–54, 2007.
14. Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, 1999.