

Faculdade de Engenharia da Universidade do Porto



Guia turístico em PDA com GPS

Daniel Moreira Gonçalves

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Telecomunicações

Orientador: Prof. Dr. Eurico Carrapatoso

Junho de 2008

© Daniel Gonçalves, 2008

Resumo

Com o estado actual da sociedade o tempo é um bem precioso e a autonomia um objectivo a alcançar. No turismo, as visitas guiadas por vezes limitam o indivíduo na sua disponibilidade em visitar locais de interesse, ficando condicionado a horários e ao funcionamento do grupo.

Para solucionar este problema, foi estudada uma alternativa que consiste numa aplicação multimédia, com vista a oferecer ao indivíduo opções de escolha que permitam devolver a autonomia e gestão do tempo necessário para conhecer e visitar os locais de interesse que mais lhe convém.

Foi desenvolvido um guia turístico electrónico num PDA, no qual o utilizador pode carregar roteiros pré-concebidos e visitar os locais de interesse neles presentes, com o auxílio de um receptor GPS que o orienta até ao local sem qualquer tipo de mapas instalados no dispositivo. A orientação requer acesso à internet para as imagens dos mapas serem transferidas para o dispositivo pelo serviço Google Maps.

Ao longo da sua visita, pode ser criado um álbum multimédia onde serão registadas as fotografias tiradas durante o trajecto, assim como os comentários pretendidos.

Após a chegada ao local o utilizador ir-se-á deparar com um monitor exibindo um Media Center que contém bastantes informações sobre o sítio, tais como imagens, vídeos, músicas e outro conteúdo elucidativo referente ao local. Deste modo, o visitante pode, através do PDA, aceder e navegar via Wi-Fi num centro de multimédia, adquirindo assim informação de uma maneira inovadora, divertida e de certo modo excêntrica. Pode inclusive, no local de interesse no qual se encontra, comprar souvenirs do próprio sítio através de uma *e-shop* acessível no PDA, podendo levantar o produto, previamente encomendado, à saída.

O resultado obtido foi a implementação de uma aplicação multimédia que simula um guia turístico electrónico com as funcionalidades descritas anteriormente. Para ter acesso a esta, o utilizador terá de se ligar a um servidor, autenticar-se, escolher um roteiro, navegar até ao local e controlar um Media Center. O utilizador pode assim escolher os seus itinerários preferidos, sem horários de grupo nem compromissos, tendo deste modo a capacidade de gerir o seu tempo e a opção de escolha entre viajar sozinho ou em grupo.

Abstract

With the current state of society, time is a precious asset and autonomy an objective to achieve. Tourism sometimes limits the person in his willingness to visit places of interest, being conditioned to the schedules and the most of the group.

To solve this problem, we studied an alternative that is a multimedia application, in order to offer the individual choices that allow the return of autonomy and management of the time to meet and visit the most convenient places of interest.

It was developed an electronic tour guide in a PDA, where the user can load pre-designed tours and visit places of interest in them, with the aid of a GPS receiver that guides the user to the location without any maps installed on the device. The orientation process requires Internet access to download the images of the maps to the device by the Google Maps service.

Throughout his visit, it can be created a multimedia album which will record the photos taken during the journey, as well as the optional comments.

Upon the arrival, the user will find a monitor displaying a Media Center which contains much information of the site, such as pictures, videos, music and other content concerning the place where the visitor through the PDA, can access and navigate via Wi-Fi, the Media Center and will acquire information from an innovative way, fun and somewhat eccentric.

Even in the place, the user can buy some souvenirs of the site through an e-shop accessible in the PDA, where in the exit way, he can get the product previously ordered.

The result was the implementation of a multimedia application that simulates an electronic tour guide with the features described above, where you will have to connect to a server, log in, choose a tour map, navigate to the place and control a Media Center. This allows the user to choose itineraries, without schedules or commitments, and thus has the ability to manage their time and the option to choose between travel alone or in groups.

Agradecimentos

Esta dissertação representa muito mais do que as horas de estudo, pesquisa e trabalho desenvolvido ao longo de um semestre. Representa todo um curso cheio de experiências óptimas que estão prestes a acabar. Dentro de pouco tempo, ficará consumada uma realização pessoal... Ser Engenheiro.

Desejo agradecer a todas as pessoas que, directa ou indirectamente, contribuíram para a realização desta Tese.

Agradeço aos meus pais e irmã pelo apoio, carinho e compreensão ao longo deste tempo.

Agradeço aos meus amigos de sempre que me ajudaram a ultrapassar os obstáculos.

Agradeço à Daniela, que me ajudou e contribuiu para o sucesso desta tese, ao Rui Humberto "Master" que me acompanha há alguns anos, ao Bruno Ribeiro "Tim", ao Pedro Leite, ao Luís Sousa, ao Carlos Ferreira, e a muitos mais amigos que me acompanharam ao longo do percurso universitário, um muito obrigado, cheguei onde cheguei graças a todos eles.

Agradeço por fim ao meu orientador, Professor Doutor Eurico Carrapatoso, pela sua disponibilidade ao longo desta dissertação.

Índice

Resumo	iii
Abstract	v
Agradecimentos	vii
Índice	ix
Lista de figuras	xi
Lista de tabelas	xiii
Abreviaturas	xiv
Capítulo 1	
Introdução.....	1
1.1 - Contexto e motivação.....	1
1.2 - Objectivos.....	2
1.3 - Estrutura	2
Capítulo 2	
Estado da Arte.....	3
2.1 - Multimédia	3
2.1.1 - O que é a Multimédia	3
2.1.2 - Interactividade e Usabilidade	5
2.2 - Navegação.....	6
2.2.1 - Introdução	6
2.2.2 - GPS	8
2.2.2.1 - Método de Funcionamento	8
2.2.2.2 - Descrição Técnica.....	9
2.2.3 - Protocolo NMEA e SiRF	9
2.2.3.1 - NMEA 0183	9
2.2.3.2 - NMEA 2000	12
2.2.3.3 - SiRF	12
2.3 - Dispositivos Móveis	13
2.3.1 - PDA	13
2.3.2 - Características de Hardware.....	14
2.4 - Conceitos e Programação.....	16
2.4.1 - Linguagem de Programação.....	16
2.4.2 - Human-Cumputer Interaction	16

2.4.3 - Context-Awareness	17
2.4.4 - Location Awareness	18
2.4.5 - Travel Technology	18
2.4.6 - Google Maps API	19
Capítulo 3	
Apresentação	22
3.1 - Requisitos	30
3.2 - Especificação	23
Capítulo 4	
Implementação.....	31
4.1 - Material Envolvido na Simulação do Sistema	32
4.2 - Comunicação	33
4.3 - Criação da Interface	34
4.4 - Base de Dados	35
4.5 - Criação de Utilizadores e Autenticação	36
4.6 - Criação da Itinerários	37
4.7 - Comunicação PDA com GPS	39
4.8 - Orientação ao Local e WebMap	40
4.9 - Informação Local e Interação com Media Center.....	41
Capítulo 5	
Conclusão e Desenvolvimentos Futuros.....	44
Referências	46
Anexos	48

Lista de figuras

Figura 2.1 - Latitude e Longitude	7
Figura 2.2 - Trilateração.....	8
Figura 2.3 - Ligação típica de uma rede NMEA 2000.....	12
Figura 2.4 - Funcionamento da ligação do SiRFIII com um dispositivo	13
Figura 2.5 - Modelo de contexto	17
Figura 2.6 - Mapa centrado	20
Figura 2.7 - Mapa centrado com diferentes dimensões	20
Figura 2.8 - Mapa com marcadores	21
Figura 2.9 - Mapa com linhas desenhadas interligando marcadores	21
Figura 3.1 - Sistema	23
Figura 3.2 - Ligação 3G/UMTS	24
Figura 3.3 - Registo de utilizador	25
Figura 3.4 - Autenticação de utilizador	25
Figura 3.5 - Ligação entre receptor e satélite GPS	26
Figura 3.6 - Troca de informação entre Google Maps e aplicação	27
Figura 3.7 - Ligação do PDA por WIFI ao Servidor local	27
Figura 3.8 - Ligação do PDA ao Servidor local por socket via WIFI.....	28
Figura 3.9 - Funcionalidades do sistema.....	29
Figura 4.1 - Fluxograma do sistema	32
Figura 4.2 - Comunicação Socket	33
Figura 4.3 - Divisão de funcionalidades.....	35
Figura 4.4 - Árvore de ficheiros	36

Figura 4.5 - Interface de autenticação	36
Figura 4.6 - Primeiro passo do <i>Google Static Map Wizard</i>	37
Figura 4.7 - Segundo passo do <i>Google Static Map Wizard</i>	38
Figura 4.8 - Terceiro passo do <i>Google Static Map Wizard</i>	38
Figura 4.9 - Interface da escolha de roteiros	39
Figura 4.10 - Interface de orientação	40
Figura 4.11 - Interface do WebMap.....	41
Figura 4.12 - Interface de informação	42
Figura 4.13 - Interface de controlo remoto	43
Figura 4.14 - Interface do MediaPortal	43

Lista de tabelas

Tabela 2.1 - Mensagens do Protocolos NMEA	10
Tabela 2.2 - Mensagem GPGGA.....	11
Tabela 3.1 - Base de dados do Utilizador.....	24
Tabela 3.2 - Base de dados do Roteiro.....	26
Tabela 3.3 - Base de dados das encomendas	28

Abreviaturas

Lista de abreviaturas (ordenadas por ordem alfabética)

3G	Third Generation phone technology.
API	application programming interface.
ASCII	American Standard Code for Information Interchange.
AVI	Audio Video Interleave.
BMP	BMP file format.
C#	C-sharp Program Language.
CAN	Controller Area Network.
CCD	Charge-coupled device.
CD-ROM	Compact Disc read-only memory.
CMOS	Complementary metal-oxide-semiconductor.
CSS	Cascading Style Sheets.
DOC	Document format file.
DVD	Digital Versatile Disc.
GIF	Graphics Interchange Format.
GLONASS	GLObal'naya NAVigatsionnaya Sputnikovaya Sistema.
GNSS	Global Navigation Satellite System.
GPGGA	Global positioning system fixed data.
GPS	Global Positioning System.
GUI	Graphic User Interface.
HCI	Human-Computer Interaction.
HSDPA	High-Speed Downlink Packet Access.
HTML	HyperText Markup Language.
HTTP	Hypertext Transfer Protocol.
IP	Internet Protocol.
IRNSS	Indian Regional Navigational Satellite System.
J2ME	Java Platform, Micro Edition.

JPEG	Joint Photographic Experts Group.
LCD	Liquid Crystal Display.
MP3	MPEG-1 Audio Layer 3.
MPEG	Moving Picture Experts Group.
NMEA	National Marine Electronics Association.
PDA	Assistente Digital Pessoal.
PHP	Hypertext Preprocessor.
PNG	Portable Network Graphics.
RFID	Radio-frequency identification.
RTCM	Radio Technical Commission for Maritime Services.
TCP	Transmission Control Protocol.
TXT	Text format file.
UMTS	Universal Mobile Telecommunications System.
URL	Uniform Resource Locator.
USB	Universal Serial Bus.
VB	Visual Basic.
WAVE	Waveform audio format.
WPA	Wi-Fi Protected Access.
XHTML	Extensible Hypertext Markup Language.

Capítulo 1

Introdução

Este capítulo contém a introdução à dissertação, justificando os motivos da sua escolha, inserindo o leitor no contexto em que foi realizado e, por fim, explica a estrutura do documento.

1.1 - Contexto e motivação

O trabalho insere-se na área de informação turística recorrendo a novas tecnologias, quer ao nível de hardware, quer de software. Surgiu a oportunidade de implementar esta aplicação multimédia para criar uma nova abordagem de como fazer turismo, sendo inovadora na forma como coloca o foco sob o indivíduo e na satisfação das suas necessidades individualizadas e concretas, sendo tal desiderato atingido pelo uso de novos equipamentos e tecnologias ou pela inovadora perspectiva de uso de equipamentos e sistemas já existentes, nomeadamente PDA e GPS. Com isto proporciona-se ao utilizador uma maior liberdade e autonomia para escolher roteiros de interesse particular sem estar dependente de horários ou grupos de visita, podendo deste modo obter uma maior satisfação que os restantes visitantes. Um projecto, sem dúvida interessante, que poderá alterar a forma como utilizamos e visualizamos a informação a nível turístico.

É uma grande motivação quando se pretende implementar uma aplicação para talvez no futuro ser comercializada e, se possível, com fins lucrativos, para continuar a sustentar a mesma, melhorando com equipamentos de topo e adicionando funcionalidades ainda mais inovadoras.

Este trabalho, hipoteticamente, poderá desviar-se para outras atenções não só focando o interesse turístico, como também o apoio à ciência arqueológica de modo a registar novas descobertas de locais arqueológicos e partilha dessa informação com outros através de dispositivos electrónicos preparados para o efeito.

1.2 - Objectivos

O objectivo desta dissertação é efectuar um estudo aprofundado de tecnologias actuais, criando uma aplicação que possa proporcionar ao utilizador total controlo da sua viagem com o auxílio de um guia turístico electrónico. “Escolho, visito, conheço como quero e o que quero” este é o paradigma deste projecto, dando a opção ao utilizador de seleccionar um roteiro à disposição, deslocar-se até lá e interagir com o local através do PDA, obtendo plena satisfação em termos de informação turística *on site*.

Deste modo, o utilizador poderá optar por um roteiro de interesse, tanto a nível de locais históricos como monumentos, museus, locais de culto e religiosos ou obras de foro arquitectónico, como a nível de roteiros de percursos pedestres em contacto com a natureza, visita de locais geológicos e arqueológicos, com o objectivo de enriquecer a própria cultura e até mesmo escapar à monotonia e ao fastidioso dia-a-dia.

É também essencial oferecer ao indivíduo novas formas de visitar locais desconhecidos. Isso requer a criação de funcionalidades revolucionárias de interacção com o meio ambiente, com a finalidade de suscitar e captar a atenção das pessoas e incentivá-las a utilizar as tecnologias multimédia.

1.3 - Estrutura

Esta dissertação encontra-se estruturada em cinco capítulos contendo ainda uma secção de anexos.

O primeiro capítulo corresponde à introdução, na qual estão referidos o contexto em que se insere o trabalho e a motivação que impulse o seu desenvolvimento, assim como os objectivos a alcançar.

No segundo capítulo será apresentado um levantamento do estado da arte, no que diz respeito a equipamentos de informação pessoal, sistemas de comunicações e sistemas geográficos de informação e localização, fornecendo ao utilizador conceitos de tecnologias existentes hoje em dia.

O terceiro capítulo faz a apresentação da aplicação multimédia que vai ser desenvolvida no âmbito desta dissertação.

O quarto capítulo contém a implementação propriamente dita da aplicação desenvolvida.

Por fim, no quinto capítulo, estarão deduzidas as conclusões gerais desta tese, sendo analisados os resultados principais e efectuada uma comparação crítica entre a proposta inicial e os resultados conseguidos.

Capítulo 2

Estado da Arte

Neste capítulo é apresentada uma pesquisa e um estudo do estado da arte no campo da multimédia, com vista a divulgar o que há no mercado assim como tecnologias usadas, abrangendo os dispositivos electrónicos, aplicações e os equipamentos GPS.

2.1 - Multimédia

2.1.1 - O que é a Multimédia

Multimédia [1] é uma forma de apresentar informação usando diferentes tipos de meios de informação combinados entre si. Nesta panóplia podemos incluir texto, grafismo, animação, som, ou vídeo. O *software Multimédia* organiza e junta estes elementos e fornece ao utilizador modos de interagir com este tipo de informação.

Ainda não há muito tempo atrás, multimédia exigia recursos de *hardware* muito dispendiosos. Com a contínua evolução vertiginosa nesta área tecnológica os custos foram reduzidos, de tal modo que a esmagadora maioria dos recentes sistemas dos computadores actuais (sejam eles computadores pessoais, portáteis ou PDA's) já são *multimedia-ready*, ou seja, já vêm equipados e preparados para este novo tipo de tecnologia, permitindo a sua fácil manipulação via software, e armazenamento utilizando os inúmeros suportes físicos existentes tais como, vídeo digital, CD-ROMs, e DVDs, etc.

Fotografia

A fotografia é um slide estático que é criado a partir de um dispositivo formador de imagem como, câmara fotográfica, câmara fotográfica digital, webcam, câmara de filmar

com a opção de capturar imagens, scanners, etc. A fotografia pode ser obtida por meios mecânicos e químicos ou digitais.

No caso das aplicações multimédia usa-se a fotografia digital. Na fotografia digital é utilizado um sensor CCD ou CMOS, que por sua vez converte a luz num código electrónico digital, e cria uma matriz de números digitais, denominados pixéis, que depois é armazenada num suporte de memória. Pode, posteriormente, ser manipulado por programas de edição de imagem de modo a melhorar a qualidade e até mesmo imprimir para papel. Formatos mais conhecidos: BMP, JPEG.

Texto

O texto é uma manifestação de ideias transmitidas por caracteres alfanuméricos. O texto pode ser escrito em dois modos: - modo físico, que consiste em possuir um instrumento e uma área onde se possa aplicar o texto, como por exemplo uma caneta e um papel ou um pincel e uma tela; - modo digital, que recorre a dispositivos computacionais, tais como um computador, PDA, ou qualquer aparelho que possua um processador ou editor de texto, ficando assim armazenada a informação em formato digital. Formatos mais conhecidos: TXT, DOC.

Som

O som é um fenómeno ondulatório perceptível pelos organismos vivos que possuem receptores auditivos. É resultado de um movimento vibratório da matéria transmitido através de meios materiais e elásticos. As ondas sonoras são ondas mecânicas, porque precisam de um meio material para se propagarem, longitudinais e tridimensionais. Por serem longitudinais, são ondas de pressão, e caminham no meio de propagação através de sucessivas compressões e rarefacções das partículas do meio. As ondas ao se propagarem através de um meio elástico alcançam o ouvido causando a sensação sonora. O aparelho auditivo humano é sensível a sons cujas frequências estão compreendidas na região de 20 Hz a 20 kHz, conhecida como *espectro de audiófrequências*.

Este fenómeno pode ser gravado por meios analógicos ou digitais que consistem em registar as frequências obtidas, em formato analógico ou directamente em formato digital. Formatos mais conhecidos: WAVE, MP3.

Vídeo

O vídeo é uma tecnologia de processamento de sinais electrónicos analógicos ou digitais para capturar, armazenar, transmitir ou apresentar imagens em movimento. Consiste numa sequência de fotografias consecutivas que produzem um efeito de movimento. Ao vídeo pode ser adicionado o som e o texto para se obter, por exemplo, uma sequência de imagens consecutivas com voz e legendas. O vídeo analógico obtido por câmaras de filmar mais

antigas era armazenado em cassetes de vídeo. Hoje em dia existem câmaras de filmar digitais que permitem o armazenamento em suporte digital, facilitando assim a sua manipulação por software. O vídeo digital é um dos formatos mais populares para desenvolver aplicações multimédia. Formatos mais conhecidos: AVI, MPEG.

Animação

A animação é uma subforma do vídeo, de mais curta duração, e visa criar uma repetição na sequência de imagens. São os chamados loops. Não costumam ter sons, e geralmente são ficheiros muito pequenos. Formatos mais conhecidos: GIF.

Para usarmos todos estes recursos de modo a criar uma aplicação multimédia, tanto para apresentação, como para simulação e jogos educacionais ou de lazer, precisamos de programas preparados para suportar todos esses tipos de formatos como, por exemplo PowerPoint, Macromedia Flash ou Director MX.

A esta ligação entre o utilizador e a aplicação, chamamos Interactividade.

2.1.2 - Interactividade e Usabilidade

A interactividade [2] é uma forma de comunicação estruturada e ramificada, pois consiste na apresentação sucessiva de um leque de opções de acordo com as consecutivas decisões do sujeito visando a concretização do objectivo delineado à priori. Ou seja, é estabelecida uma relação acção-opção-decisão entre o utilizador e o dispositivo. O utilizador consegue abrir e fechar janelas, chamar menus, navegar por uma sequência de imagens, sons e textos contidas na aplicação através do rato, dedo, caneta ou comando remoto. Contudo, estas ligações dinâmicas (ramificações) são finitas, o que significa que passado algum tempo todas elas são percorridas por completo, estando implícito no princípio da interactividade dado que, através de uma dada sinalização, o utilizador pode interpretar os sinais presentes na navegação, realizando um maior ou menor esforço intelectual que será decorrente da qualidade do modelo interactivo e da sua interface comunicacional.

O termo evoluiu para a área multimédia a partir da relação entre os seres humanos na vida real, copiando para a realidade virtual tudo o que podemos fazer no dia-a-dia. Em termos de curto prazo é difícil imaginar que qualquer sistema multimédia informático possa substituir de forma eficiente o processo multifacetado e riquíssimo da interacção humana que estrutura o essencial das nossas relações. Todavia, numa opinião pessoal, baseada também noutras opiniões, a longo prazo e num futuro distante, as máquinas terão um papel fulcral nas nossas vidas podendo até substituir algumas das nossas funções mais importantes.

Nicholas Negroponte, director no Media Lab do M.I.T. apresenta uma opinião bastante futurista: *«No que diz respeito à interface, o meu sonho é ver os computadores assemelharem-se aos homens. Alguns poderão criticar a ideia por excesso de*

romantismo. Eu acho que essas pessoas têm falta de ambição. (...) Aquilo que hoje se chama «interface orientada para personagens» vai transformar-se no modo dominante através do qual as pessoas dialogarão com as máquinas.

Existirão pontos precisos no tempo e no espaço onde os bits serão convertidos em átomos e vice-versa. Que isso suceda através da transmissão de cristais líquidos ou pela reverberação de um sintetizador, pouco interessa. A interface precisa de possuir uma configuração, uma forma, uma cor, um tom de voz e todo o poder de atracção sensorial.».

A usabilidade [3] é o termo usado para caracterizar a facilidade com que o utilizador consegue manusear uma certa ferramenta ou qualquer outro artefacto com um objectivo peculiar. Serve juntamente para estudar e medir a usabilidade em relação à eficiência e aspecto visual de uma aplicação em particular.

Em relação aos computadores e à interactividade humana, usabilidade refere-se à simplicidade, transparência e aspecto visual de uma aplicação multimédia ou um site de internet. Denomina-se *user-friendly* quando uma aplicação é fácil e simples de usar, isto é, todas as pessoas que experimentarem não terão grande dificuldade em navegar pela aplicação. Também tem que ser intuitiva, ou seja, mesmo não sabendo de antemão trabalhar com a aplicação, consegue com sucesso reproduzir aquilo que se imagina na aplicação em questão, isto é, utiliza-se o senso comum para tentar atingir os objectivos.

2.2 - Navegação

2.2.1 - Introdução

Navegação é a arte de planear, registar e controlar os movimentos de um barco, avião, automóvel, ou qualquer outro veículo de uma origem para um destino. Ao longo dos tempos as técnicas de navegação foram evoluindo muito graças à troca de informação entre culturas, que veio enriquecer os conhecimentos diversos sobre esta arte, desenvolvendo-a.

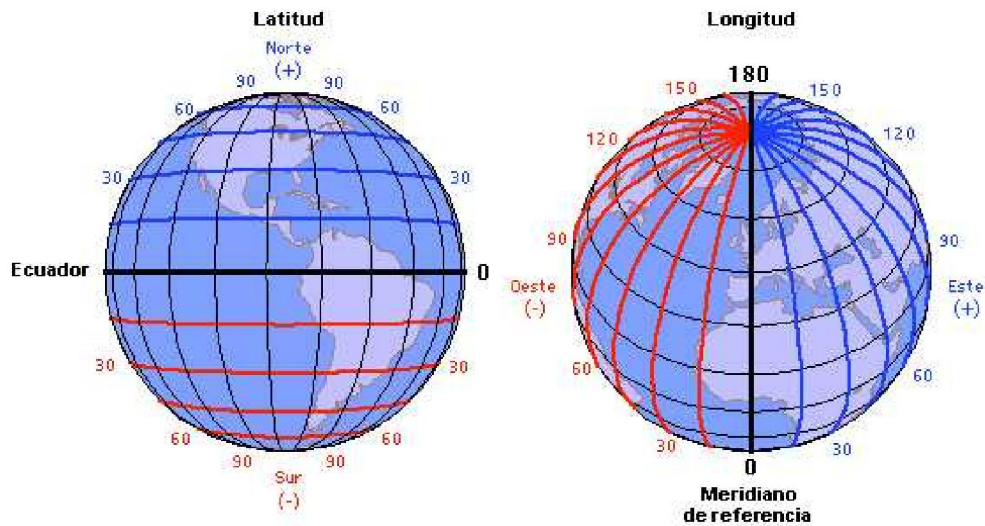


Figura 2.1 - Latitude e Longitude

Basicamente, a navegação é conseguida pelo controlo de duas variáveis espaciais, são elas a Latitude e a Longitude, como podemos observar na Figura 2.1. A Latitude refere-se à distância angular terrestre entre o Pólo Norte ou Pólo Sul e o Equador. Representa-se em ângulos desde o Equador (0°) ao Pólo Norte (90°) e ao Pólo Sul (-90°). A Longitude tem como referência o meridiano de Greenwich e os ângulos são medidos na horizontal, tomando o valor de 0° a 180° para Oeste e 0° a 180° para Este.

Nos tempos antigos a navegação sofreu uma grande impulsão aquando da navegação marítima, pois esta era bem mais difícil que a terrestre devido à ausência de pontos de referência, fazendo com que as civilizações desenvolvessem técnicas próprias para poder contornar o problema. Com esta evolução apareceram instrumentos como a bússola, e o sextante, e mais tarde o altímetro.

A bússola possui uma agulha magnética que indica o norte, devido ao campo magnético terrestre.

O sextante é um instrumento que mede a abertura angular vertical de um astro acima do horizonte.

O altímetro é um instrumento para medir a altura do ponto onde o aparelho se encontra.

Hoje em dia todos estes instrumentos referidos vieram a ser substituídos pelos satélites de navegação. *Global Navigation Satellite System* ou GNSS, é assim denominado o sistema de navegação por satélite, e permite uma cobertura geo-espacial autónoma. Este sistema permite que um pequeno receptor electrónico determine a sua posição (latitude, longitude e altura).

O GPS é um exemplo comum deste sistema de navegação.

2.2.2 - GPS

Global Positioning System [4], este é o nome do programa de posicionamento por satélite desenvolvido pelo Departamento de Defesa Norte Americano. O seu nome oficial é NAVSTAR [5] e foi inicialmente desenvolvido para fins militares, tornando-se mais tarde aberto para uso civil. Desde que foi aberto ao público como ficando totalmente operacional em 1995, revolucionou a navegação a nível mundial, tornando-se uma ferramenta imprescindível para a criação de mapas, estudos geográficos, traçado de rotas e usos científicos. Com um receptor de GPS obtemos todas as coordenadas terrestres, latitude, longitude e altitude, e ainda outras grandezas como a velocidade e intervalos de tempo. Neste momento mais de vinte e quatro satélites orbitam a terra fazendo com que nenhum receptor ligado fique sem acesso a, no mínimo, seis satélites GPS, colocando assim à disposição um excelente sistema de navegação.

Existem concorrentes ao GPS como o sistema russo GLONASS, o novo Galileu europeu, o proposto COMPASS chinês e o IRNSS indiano.

2.2.2.1 - Método de Funcionamento

O método de funcionamento é muito simples. Baseia-se num princípio matemático chamado Trilateração e consiste na determinação da distância entre um ponto, o receptor, a outros de referência, os satélites. Sabendo a distância que nos separa de três pontos podemos determinar a nossa posição relativa a esses mesmos pontos através da intersecção de três circunferências cujos raios são as distancias medidas entre o receptor e os satélites adquirindo-se assim dois pontos comuns, um espacial e outro terrestre, escolhendo aquele que está presente na superfície terrestre. Este é o ponto no qual se encontra o receptor. O quarto satélite serve para realizar a necessária sincronização entre os relógios dos satélites e o do receptor. Com estes quatro satélites é possível resolver o sistema constituído por quatro coordenadas: x , y , z e t .

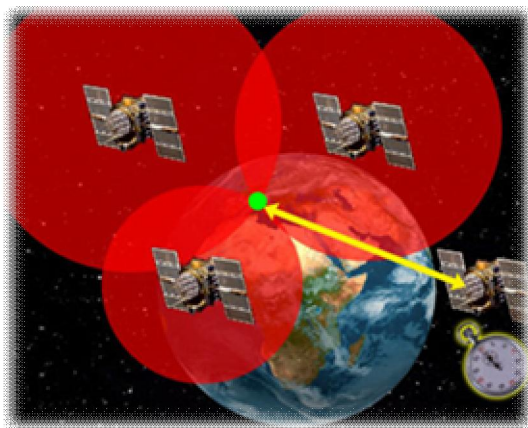


Figura 2.2 - Trilateração

2.2.2.2 - Descrição Técnica

Dentro do sistema GPS existem três segmentos importantes, o segmento espacial, o segmento de controlo e o segmento do utilizador.

O segmento espacial consiste nos satélites que orbitam em torno da Terra. Inicialmente estava previsto existirem vinte e quatro satélites, oito em cada plano orbital, mas foi alterado para quatro satélites em seis planos orbitais. Hoje em dia, são trinta e um satélites em órbita, o que significa que sete satélites estão em *standby*, prontos para entrarem em funcionamento no caso de alguns desses satélites falhar.

O segmento de controlo serve para controlar a órbita dos satélites existentes, isto é, se algum destes estiver a funcionar mal ou com órbita desviada, pode ser controlado a partir de postos de controlo terrestres. É enviada informação para estes pararem de transmitir sinal, evitando assim erros nos cálculos de quem está a utilizar um receptor que possa apanhar esses satélites.

O segmento do utilizador é constituído pelo receptor GPS de um utilizador. O receptor GPS possui uma antena que sintoniza as frequências transmitidas pelos satélites. Estes apresentam no visor as informações captadas pelo sinal transmitido. O receptor é caracterizado pelo número de satélites que pode sintonizar simultaneamente. Quanto mais satélites sintonizar ao mesmo tempo maior é a precisão da sua localização. Estes podem conter o formato RTCM SC-104 que serve para corrigir a precisão eliminando alguns erros. Muitos dos receptores de GPS usam o protocolo NMEA. O protocolo NMEA foi desenvolvido pela *National Marine Electronics Association* para ser possível comunicar com um PC, PDA ou *Smartphone* por USB, *Bluetooth* ou Porta Série, obtendo assim os dados enviados pelos emissores. Existem protocolos de *open-source* (gpsd) que se baseiam neste, podendo ser alterados sem violar os direitos de propriedade do NMEA.

2.2.3 - Protocolo NMEA e SiRF

2.2.3.1 - NMEA 0183

O protocolo NMEA 0183 [6] é uma especificação combinada de dispositivos eléctricos e de transmissão de dados, que interligam componentes electrónicos como sonares, receptores GPS, bússolas, giroscópios, sensores dos tanques de combustível, casa das máquinas, entre outros. Este protocolo foi desenvolvido e é controlado pela *National Marine Electronics Association*. É usado o código ASCII e a comunicação série para transmitir dados de um emissor para um receptor num instante. Prevê um emissor único e unidireccional que consegue transmitir apenas uma mensagem para vários receptores ao mesmo tempo.

A sua implementação varia entre ligações RS-232, RS-423, RS-422. É aconselhável testar todas as implementações possíveis para eliminar situações de incompatibilidade. No entanto, correndo a interface a uma velocidade relativamente pequena há grandes hipóteses de tudo se desenrolar sem problemas. Há que ter em atenção, especialmente quando se está a navegar, que a ligação eléctrica à terra não é muito segura, e por conseguinte deve-se proteger as portas série do PC com isoladores ópticos, que permitem a transmissão de electricidade num só sentido.

Configuração da ligação série:

- *Baud Rate: 4800 bit/seg*
- *Data Bits: 8*
- *Parity: N/A*
- *Stop Bits: 1 ou mais*
- *Handshake: N/A*

Regras de aplicação do protocolo:

- Cada mensagem começa pelo símbolo \$
- Os cinco caracteres seguintes indicam o tipo de mensagem do Emissor,
- Todos os campos são diferenciados por vírgulas,
- O primeiro caracter depois da mensagem do Emissor é um asterisco,
- A seguir ao asterisco são apresentados dois números que representam o controlo de erros, designado por *checksum*,
- No final de cada mensagem NMEA são acrescentados os seguintes caracteres de terminação: <CR><LF>.

Exemplos do protocolo:

Sentence	Description
\$GPGGA	Global positioning system fixed data
\$GPGLL	Geographic position - latitude / longitude
\$GPGSA	GNSS DOP and active satellites
\$GPGSV	GNSS satellites in view
\$GPRMC	Recommended minimum specific GNSS data
\$GPVTG	Course over ground and ground speed

Tabela 2.1 - Mensagens do Protocolos NMEA

\$GPGGA Sentence (Fix data)

Exemplo de sinal não recebido:

\$GPGGA,235947.000,0000.0000,N,00000.0000,E,0,00,0.0,0.0,M,,,,0000*00

Exemplo de sinal recebido:

\$GPGGA,092204.999,4250.5589,S,14718.5084,E,1,04,24.4,19.7,M,,,,0000*1F

Field	Example	Comments
Sentence ID	\$GPGGA	
UTC Time	092204.999	hhmmss.sss
Latitude	4250.5589	ddmm.mmmm
N/S Indicator	S	N = North, S = South
Longitude	14718.5084	dddmm.mmmm
E/W Indicator	E	E = East, W = West
Position Fix	1	0 = Invalid, 1 = Valid SPS, 2 = Valid DGPS, 3 = Valid PPS
Satellites Used	04	Satellites being used (0-12)
HDOP	24.4	Horizontal dilution of precision
Altitude	19.7	Altitude in meters according to WGS-84 ellipsoid
Altitude Units	M	M = Meters
Geoid Separation		Geoid separation in meters according to WGS-84 ellipsoid
Separation Units		M = Meters
DGPS Age		Age of DGPS data in seconds
DGPS Station ID	0000	
Checksum	*1F	
Terminator	CR/LF	

Tabela 2.2 - Mensagem GPGGA

2.2.3.2 - NMEA 2000

Podemos dizer que o protocolo NMEA 2000 [7] é uma evolução do NMEA 0183. Esta evolução consistiu no aumento do débito para 250000 bit/seg e tem também a capacidade de transmitir numa configuração de multi-emissor para multi-receptor, visto que o anterior previa um débito de 4800 bit/seg e tinha apenas um emissor. Este protocolo consegue comunicar com dispositivos usando a tecnologia CAN [8] e é baseado no protocolo SAE J1939; contudo cria as suas próprias mensagens de código.

Na figura seguinte está representada uma instalação típica de uma rede NMEA 2000, onde todos os cabos e as ligações têm de ter certas especificações para constituir sem problemas a rede NMEA 2000.

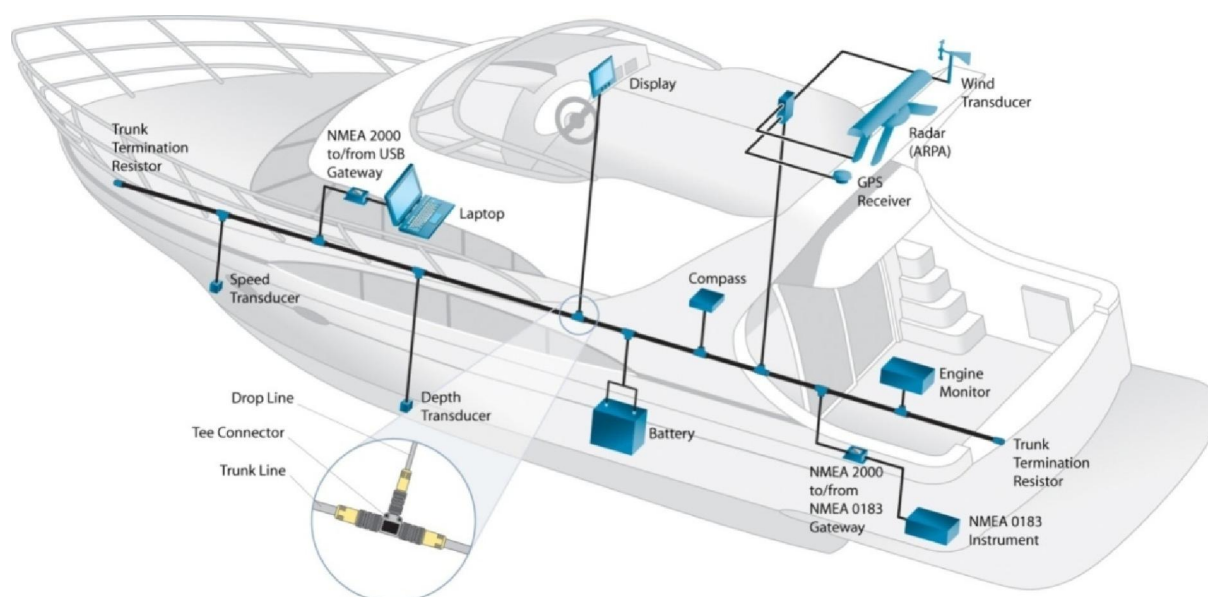


Figura 2.3 - Ligação típica de uma rede NMEA 2000

2.2.3.3 - SiRF

A SiRF [9] é uma companhia que fabrica microcontroladores e software para receptores de GPS.

Neste momento o chip mais utilizado é o SiRFstarIII, que é de alta sensibilidade e recebe sinais directamente dos satélites GPS, determina assim a posição do local onde se encontra o receptor.

Estes componentes podem já vir instalados em alguns aparelhos, ou podem ser adicionados por meio de cabos ou cartões *CompactFlash* para, por exemplo, serem introduzidos nos PDAs com ranhura para esse tipo de cartão.

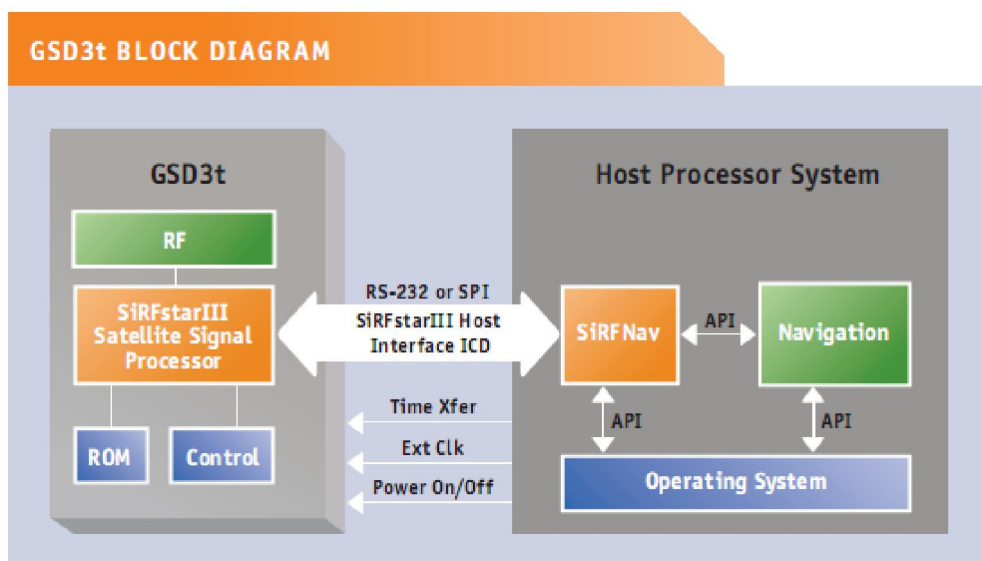


Figura 2.4 - Funcionamento da ligação do SiRFIII com um dispositivo

2.3 - Dispositivos Móveis

2.3.1 - PDA

PDA ou Assistente Digital Pessoal [10], é um computador pessoal de dimensões reduzidas, pensado para ser utilizado constantemente como agenda pessoal, relógio, calendário, incluindo várias aplicações, como no caso dos PDAs com Windows Mobile, que possuem Word, Excel, PowerPoint entre outras. Hoje em dia todos permitem a ligação à internet por Wireless [11] possibilitando o envio e recepção de e-mails e têm um conjunto de características que permitem realizar as mesmas tarefas comparativamente a um computador pessoal. Está equipado com colunas de som e pode ter rádio e um receptor GPS incluído. Basicamente é um dispositivo multimédia muito bem equipado e muito versátil, com grandes potencialidades para quem o usa diariamente, tornando-se um instrumento imprescindível para alguns profissionais.

As características técnicas variam muito de equipamento para equipamento, mas possuem em comum um ecrã táctil que permite a interacção entre utilizador e máquina de forma intuitiva, rápida e bastante divertida. Usualmente é utilizada uma "stylus pen" para a navegação no sistema operativo, e mais recentemente com a evolução da sensibilidade e da própria tecnologia, é possível utilizar o dedo para obter os mesmos fins sem haver erros de

interpretação. Este sistema também é capaz de reconhecer a escrita do utilizador que utiliza a "stylus pen". Vêm equipados também com uma interface de leitores de cartões de memória flash.

Uma grande vantagem é possuírem um módulo receptor GPS, normalmente o chip SiRFstarIII referido anteriormente. Nesta tese esta tecnologia é um requisito, pois torna-se assim possível determinar a localização do PDA, e fazer com que este reaja com o meio envolvente a partir do acesso à internet.

Também será de referir que para ter acesso na maior parte do território é necessário que o PDA esteja equipado com a tecnologia UMTS para permitir a ligação à internet onde quer que o utilizador se encontre, dentro ou fora de Zonas Wi-Fi [12].

Sistemas operativos mais conhecidos compatíveis com estes dispositivos são o Palm OS, Windows Mobile, Blackberry OS, Symbian OS.

Alguns exemplos de software de navegação por GPS que podem ser utilizados em diversas plataformas e sistemas operativos, incluem tracky, oziexplorer, tomtom, ndrive, garmin, navicore, entre outros [13].

2.3.2 - Características de Hardware

Como já foi referido anteriormente, existem certas especificações que um PDA deve satisfazer. Estas incluem: - módulo de GPS; - ligação sem fios à internet, neste caso à internet móvel UMTS/HSDPA [14]; - um ecrã de dimensões generosas, de modo a favorecer a observação de conteúdos; - um bom processador que permita um rápido processamento das aplicações; - protocolos de comunicação como *Bluetooth* e Wi-fi; - boa capacidade de memória ROM; - e, ranhuras de expansão de memória.

Depois de uma pesquisa efectuada sobre PDAs no mercado, serão aqui apresentados alguns equipamentos que possuem essas características.



HTC P6500

Operative System: Windows Mobile 6
 CPU: 32bit Qualcomm MSM7200
 CPU clock: 400 MHz
 ROM capacity: 1024 MB
 RAM capacity: 256 MB
 Display: TFT 65k colour, 3.5", 240x320
 Network Data Link: GPRS, EDGE, UMTS, HSDPA
 Expansion Slots: MMC, SD
 USB: USB 1.1 mini
 Bluetooth: Bluetooth 2.0
 Wireless LAN: 802.11b, 802.11g
 Built-in GPS: NMEA 0183
 Digital Camera: 3.1 MP, LED flash
 Battery: Lithium-ion, 1600 mAh



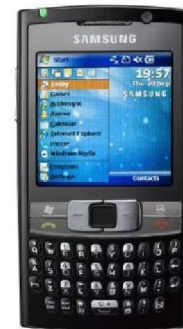
ASUS P750

Operative System: Windows Mobile 6
 CPU: 32bit Intel Xscale PXA270
 CPU clock: 520 MHz
 ROM capacity: 256 MB
 RAM capacity: 64 MB
 Display: TFT 65k colour, 2.6", 240x320
 Network Data Link: GPRS, EDGE, UMTS, HSDPA
 Expansion Slots: microSD
 USB: USB 1.1 mini
 Bluetooth: Bluetooth 2.0
 Wireless LAN: 802.11b, 802.11g
 Built-in GPS: SiRF Star III
 Digital Camera: 3.1 MP
 Battery: Lithium-ion, 1300 mAh



HTC P3650

Operative System: Windows Mobile 6
 CPU: 32bit Qualcomm MSM7200
 CPU clock: 400 MHz
 ROM capacity: 256 MB
 RAM capacity: 128 MB
 Display: TFT 65k colour, 2.8", 240x320
 Network Data Link: GPRS, EDGE, UMTS, HSDPA
 Expansion Slots: microSD
 USB: USB 2.0 mini
 Bluetooth: Bluetooth 2.0
 Wireless LAN: 802.11b, 802.11g
 Built-in GPS: NMEA 0183
 Digital Camera: 3.1 MP
 Battery: Lithium-ion, 1350 mAh



SAMSUNG SGH-I780

Operative System: Windows Mobile 6
 CPU: 32bit Marvell PXA310
 CPU clock: 624 MHz
 ROM capacity: 256 MB
 RAM capacity: 128 MB
 Display: TFT 262k colour, 2.5", 320x320
 Network Data Link: GPRS, EDGE, UMTS, HSDPA
 Expansion Slots: microSD
 USB: USB 2.0 mini
 Bluetooth: Bluetooth 2.0
 Wireless LAN: 802.11b, 802.11g
 Built-in GPS: NMEA 0183
 Digital Camera: 1.9 MP
 Battery: Lithium-ion, 1000 mAh

2.4 - Conceitos e Programação

2.4.1 - Linguagem de Programação

Existem várias linguagens de programação para ambientes mobile. Entre elas as mais conhecidas são o J2ME (Java) [15] que pode ser usado para vários sistemas operativos, pois corre sobre uma plataforma que pode ser instalada em todos os dispositivos, existe também o C# e VB .NET que correm na plataforma .NET Compact Framework [16] e ainda o código nativo C++. Todas estas linguagens são orientadas a objectos. O C# e VB .NET só correm em ambiente Windows pois neste momento a plataforma .NET só pode ser instalada neste sistema operativo, mas espera-se que futuramente também seja possível instalar .NET em PalmOS.

2.4.2 - Human-Cumputer Interaction

O foco do conceito de *Human-Computer Interaction* [17] é o estudo da capacidade de interacção do utilizador com o computador. Esta é estabelecida a partir da Interface, que é composta por software e hardware.

O desenvolvimento deste conceito visa melhorar a interacção entre o utilizador e a máquina, tornando assim o computador mais fácil de usar e intuitivo. As características de maior relevância são:

- I Processos de estruturação de interfaces e metodologias
- I Métodos de implementação de interfaces
- I Técnicas de avaliação e comparação com outras interfaces
- I Desenvolvimento de novas interfaces e novas técnicas

Um objectivo a longo prazo do HCI é desenvolver um sistema que permita reduzir a barreira entre a tarefa que o utilizador pretende realizar e o que o computador é capaz de interpretar.

No campo da investigação do HCI o interesse é desenvolver novas metodologias de arquitectura, experimentando novos hardwares e softwares para estudar o paradigma da interactividade e desenvolver modelos e teorias de interacção.

2.4.3 - Context-Awareness

Context-Awareness [18] é um termo usado para definir a interacção do meio ambiente com dispositivos electrónicos móveis ou estáticos. Tais dispositivos podem tirar proveito desta ciência usando aplicações Context-aware para ter noção do ambiente onde se encontram e com isto responderem de modo automático a situações específicas previamente programados.

Para estruturar um modelo de Context-Aware tem-se em conta as seguintes características:

- O Contexto descreve a situação e o ambiente onde o dispositivo ou o utilizador estão;
- O Contexto é identificado apenas por um nome;
- Um conjunto de funcionalidades define cada contexto;
- Para cada funcionalidade é determinado pelo contexto (explicitamente ou implicitamente) um conjunto de valores.

A figura 2.5 apresenta um modelo em que existe uma hierarquia organizada para desenvolver o contexto.

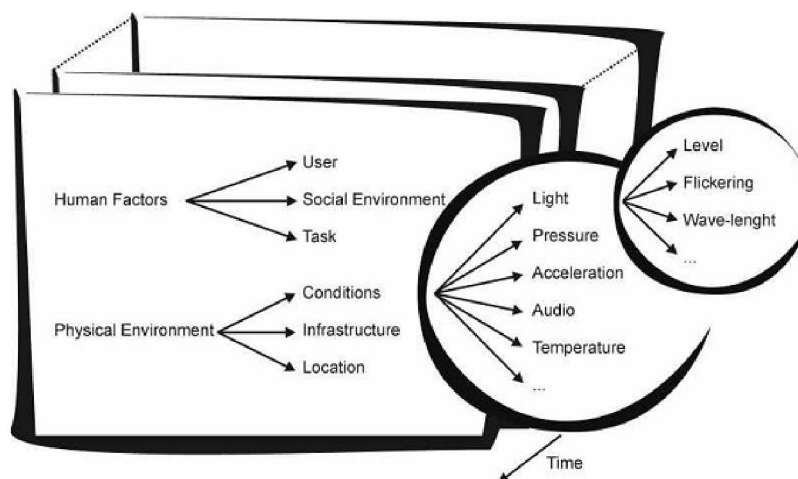


Figura 2.5 - Modelo de contexto

Os Factores Humanos estão divididos por três categorias: a informação do utilizador (habitat, estado emocional, biorritmo, condições fisionómicas), o ambiente onde o utilizador se encontra e as tarefas que este pretende executar (objectivo, actividades espontâneas, tarefas organizadas). Também o contexto relativo ao Ambiente Físico é dividido em três categorias: local (posição relativa e absoluta), infra-estrutura circundante (dispositivos computacionais em redor, tipos de comunicações) e condições físicas do ambiente (ruído, claridade, pressão).

Este modelo define a estrutura para uma obtenção do Contexto.

Para obter uma aproximação do ambiente recorre-se a sensores (*phidgets*) que possam transmitir informação para o dispositivo, por exemplo para um PDA. O GPS é um sistema que constitui um ambiente inteligente pois estamos constantemente a receber informação da nossa posição em tempo real.

2.4.4 - Location Awareness

Location Awareness [19] é o termo usado basicamente para determinar a localização de algo ou alguém.

Esta localização só pode ser feita a partir de dispositivos electrónicos, como o radar, smartphones, GPS, RFID e outros aparelhos com a mesma função.

Um exemplo deste conceito em prática é um conjunto de utilizadores com equipamentos suportando esta “tecnologia”, estarem inscritos numa base de dados onde é possível ver as distâncias entre eles e ver todas essas posições num mapa virtual, podendo ainda cada utilizador individualmente receber informação de outros dentro da sua rede. Pode-se ter uma noção real deste método conhecendo o sistema Dodgeball [20]. Este consiste em localizar amigos e amigos de amigos, enviando mensagens através de telemóvel para toda a comunidade situada relativamente perto da localização do emissor.

Estes dois conceitos fazem parte do Ubiquitous Computing [21], conseguindo obter um projecto de Ambientes de Sistemas Inteligentes.

2.4.5 - Travel Technology

Travel technology [22] é uma expressão usada para descrever aplicações de tecnologias da informação em viagens, turismo e indústrias hoteleiras. É uma referência à tecnologia do turismo pois é uma grande ajuda nesta área, visto estar em evolução contínua. Com esta tecnologia foi criada uma forma de turismo electrónico, o chamado e-Tourism. Isto permite visitas virtuais pela internet onde se pode simular a visita local a partir de um computador ou de um equipamento electrónico e ainda pode ser usado como um auxílio à viagem de um explorador.

Alguns exemplos do uso desta tecnologia em vários formatos são por exemplo o GPS que auxilia o utilizador a se deslocar fazendo de guia turístico com o software apropriado e um dispositivo electrónico como um PDA. Com o mesmo dispositivo também se pode armazenar informação para ser usada como livro turístico, e até mesmo como guia sonoro que dá a informação sobre o local, muito utilizado em autocarros turísticos.

2.4.6 - Google Maps API

O Google Static Maps API serve para introduzir imagens de mapas em páginas Web ou desenvolver aplicações que usem esta API. Para se poder usar é preciso obter uma chave para ser introduzida no URL para que o mapa seja visualizado correctamente. Por dia só é possível visualizar 1000 imagens de mapas diferentes. Esta restrição visa impedir abusos de utilização e assim conseguir um bom desempenho do servidor no atendimento a todos os pedidos feitos pelos utilizadores. Se este limite for superado o serviço pode ficar sem funcionar temporariamente. O Google Maps API retorna vários tipos de imagens GIF, PNG e JPEG como resposta a um HTTP request via URL. Neste URL é possível introduzir marcadores, trajectos, tipo de imagem, zoom, tamanho de imagem, tipo de mapa e a chave de acesso ao serviço.

O URL assume o seguinte formato: "http://maps.google.com/staticmap?parameters". Alguns dos parâmetros são opcionais e são todos separados por este carácter "&". Os parâmetros que são identificados pelo serviço são:

- **center** - permite centrar o mapa nas coordenadas, latitude e longitude, a especificar.
- **zoom** - permite requisitar o mapa com o aumento pretendido.
- **size** - define o tamanho do mapa a disponibilizar.
- **format** - é opcional e determina o formato da imagem do tipo PNG, GIF ou JPEG.
- **maptype** - é opcional e define o tipo de mapa. Especifica se vai ser indicado para tipo *roadmap* (predefinido) ou do tipo *mobile* para ser visualizado em ecrãs de dispositivos móveis.
- **markers** - é opcional e serve para marcar qualquer tipo de local. É definido por latitude longitude e os marcadores são separados pelo carácter "|".
- **path** - é opcional e serve para desenhar linhas rectas que podem ser ligadas por pontos comuns com outras rectas desenhadas. É útil para por exemplo desenhar trajectos de orientação. São também separados pelo carácter "|".
- **key** - serve para autenticar o uso do serviço Google Static Maps API.

Como exemplo, serão apresentadas nas figuras 2.6, 2.7, 2.8 e 2.9 imagens elucidativas em relação a um mapa centrado num determinado ponto, um tamanho diferente do mapa, um mapa com marcadores, e um mapa com rectas desenhadas interligando marcadores.



http://maps.google.com/staticmap?center=41.148414,-8.610020&zoom=14&size=500x300&key=YOUR_KEY_HERE

Figura 2.6 - Mapa centrado



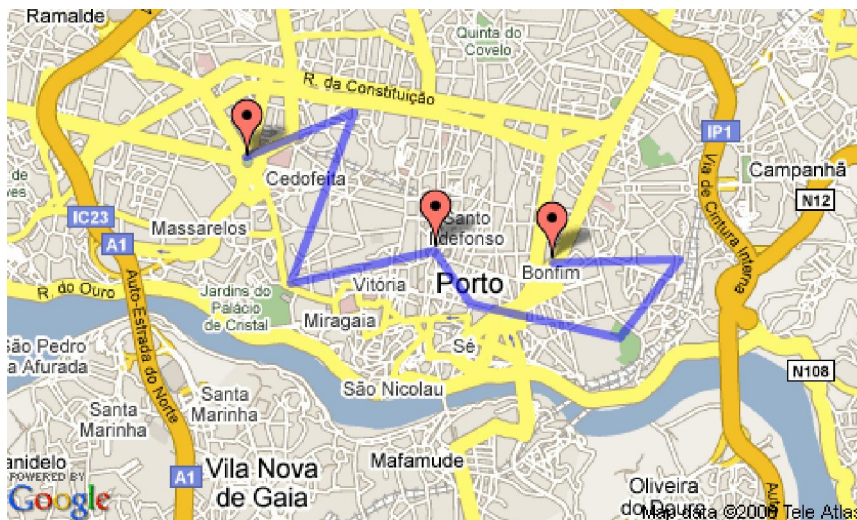
http://maps.google.com/staticmap?center=41.148414,-8.610020&zoom=14&size=200x200&key=YOUR_KEY_HERE

Figura 2.7 - Mapa centrado com diferentes dimensões



http://maps.google.com/staticmap?center=41.149968,-8.610243&markers=41.158237,-8.629074,red|41.150482,-8.598518,red|41.151387,-8.610191,red&zoom=14&size=500x300&key=YOUR_KEY_HERE

Figura 2.8 - Mapa com marcadores



http://maps.google.com/staticmap?center=41.149968,-8.610243&markers=41.158237,-8.629074,red|41.150482,-8.598518,red|41.151387,-8.610191,red&path=rgba:0x0000FF80,weight:5|41.15824,-8.62925|41.16160,-8.61843|41.14867,-8.62478|41.15113,-8.61036|41.14712,-8.60676|41.14454,-8.59182|41.15048,-8.58582|41.15022,-8.59852&zoom=13&size=500x300&key=YOUR_KEY_HERE

Figura 2.9 - Mapa com linhas desenhadas interligando marcadores

Capítulo 3

Apresentação

A aplicação apresentada no Capítulo 4 pretende ser a resposta a uma necessidade na sociedade actual que é a gestão do tempo, e neste caso concreto a gestão do tempo em actividades lúdicas de turismo. Tal como na vida quotidiana e na vida profissional, o factor tempo em turismo torna-se cada vez mais um factor chave. Um recurso de inegável valor que importa saber gerir correctamente tendo em vista uma satisfação plena do cidadão enquanto turista. Pode assim planear as suas rotas e passeios turísticos, organizando as suas visitas de acordo com os seus interesses individuais.

Irá ser implementado um sistema que permite a um utilizador registar-se para poder utilizar a aplicação, através de um browser e uma base de dados. Quando o utilizador estiver registado e tentar fazer o login na aplicação a correr no PDA, o cliente ir-se-á ligar a um servidor através de uma ligação 3G/UMTS e recorrendo à base de dados irá testar se o utilizador está autorizado a utilizar a aplicação.

Quando o login for bem sucedido o utilizador terá à sua escolha um leque variado de roteiros possíveis a visitar, e escolherá o que lhe for mais interessante. A aplicação encarrega-se de indicar o caminho até ao destino escolhido.

Ao chegar ao destino o utilizador é informado da sua chegada ao local por uma mensagem no visor do PDA. Será agora possível consultar a informação disponibilizada no local através de um Webserver, via Wi-Fi no PDA, dando a oportunidade ao utilizador de navegar num browser incluído na aplicação com acesso a toda a descrição do local, a sua história e se pretender comprar souvenirs, recorrer a um carrinho de compras online para os poder levantar à saída. Para isso terá de possuir um crédito positivo na sua conta de utilizador. Também poderá aceder e navegar num Media Center a partir do PDA via Wi-Fi caso o local onde se encontre possua um LCD ligado a um servidor que corre o Media Center. Nesse caso, poderá consultar vídeos, músicas, fotos e qualquer outro tipo de formato multimédia presente. Depois de acabada a visita do local onde se encontra poderá passar para o próximo local do roteiro.

Ao longo de todo o trajeto o utilizador pode com esta aplicação captar imagens do percurso com a câmara fotográfica incorporada no PDA, e automaticamente é gerado um álbum fotográfico de viagem, onde o utilizador pode criar os comentários que desejar.

3.1 - Especificação

3.2.1 - Constituição do Sistema

O sistema representado na Figura 3.1, é constituído por vários módulos que se irão interligar de modo a obter um resultado final de acordo com o pretendido.

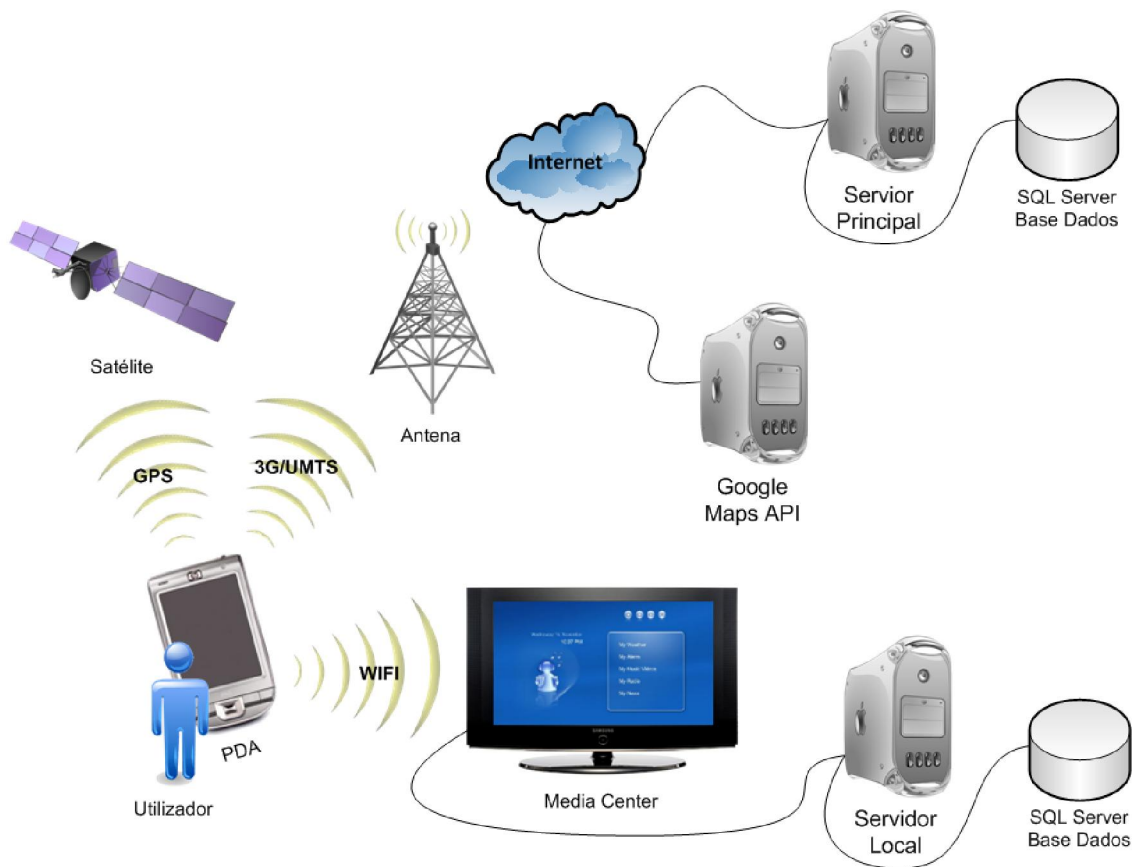


Figura 3.1 - Sistema

3.2.2 - Ligação do PDA ao Servidor Principal por 3G/UMTS

A ligação do PDA ao servidor Principal terá de ser feita por 3G/UMTS. Este protocolo serve para fazer ligações de alta velocidade permitindo assim ao utilizador ter acesso à internet móvel de banda larga. Esta ligação, representada na Figura 3.2, servirá para ligar a aplicação ao servidor e trocar informação com este, como por exemplo, para autenticar o utilizador e aceder à base de dados para escolher o roteiro a percorrer.

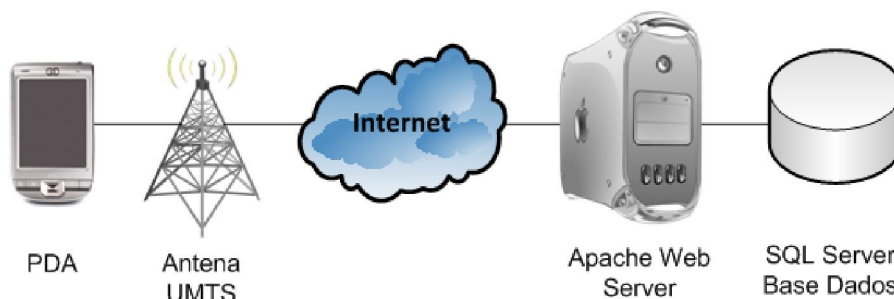


Figura 3.2 - Ligação 3G/UMTS

3.2.3 - Criação de Perfil e Autenticação de Utilizadores

O utilizador pode-se registar através de um Web browser inserindo os seus dados pessoais. Os dados pessoais introduzidos irão ser enviados para uma base de dados no Servidor Principal com o perfil previamente criado.

Poderão ser introduzidos dados pessoais assim como características de personalidade que ajudarão a identificar e a traçar linhas de interesse que permitam apresentar ao utilizador notícias e novidades relevantes nas áreas em comum.

Na tabela 3.1 podem ser observados exemplos de campos do utilizador a serem criados na base de dados.

Atributo	Tipo	Finalidade
<i>Nome</i>	Caracteres	Dados pessoais
<i>Morada</i>	Caracteres	Dados pessoais
<i>Data de nascimento</i>	Data	Dados pessoais
<i>e-mail</i>	Caracteres	Dados pessoais
<i>Nome de utilizador</i>	Caracteres	Autenticação
<i>Password</i>	Caracteres	Autenticação
<i>Tipos de interesses</i>	Caracteres	Cruzamento de dados
<i>Saldo disponível</i>	Numérico	Compra de souvenirs

Tabela 3.1 - Base de dados do Utilizador

O nome de utilizador e a palavra passe irão servir para o utilizador se autenticar no sistema podendo assim usar a aplicação legalmente.

Para a criação do perfil do utilizador é usado um digrama de seqüências na figura seguinte.

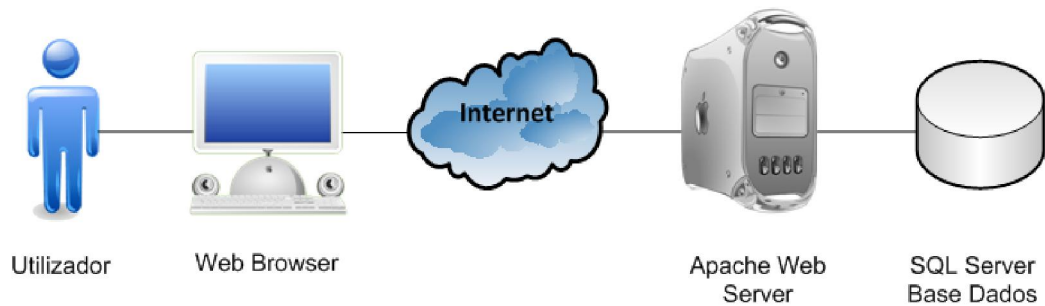


Figura 3.3 - Registo de utilizador

Para a autenticação do utilizador, o diagrama é semelhante, sendo agora efectuada a partir do PDA correndo a aplicação.

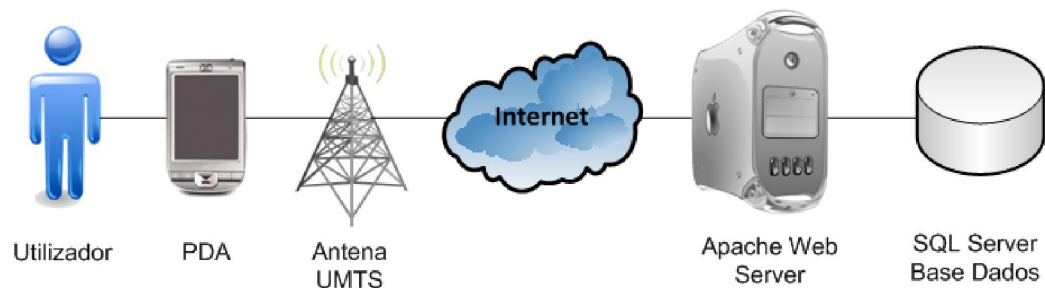


Figura 3.4 - Autenticação de utilizador

3.2.3 - Criação de Itinerários dos Roteiros

Para a criação de itinerários ir-se-á utilizar a API do Google Static Maps. Baseado no Google Maps para dispositivos móveis, esta API interpreta uma linha URL com determinada informação e retorna para um Web browser a imagem do local de acordo com as coordenadas contidas no referido URL.

Nesta API, como se observa na secção do Google Static Maps do Capítulo 2, podem-se introduzir marcadores nos locais de interesse e depois desenhar rectas de modo a descrever um percurso interligando os marcadores.

O URL obtido na configuração do itinerário irá ser introduzido na base de dados do Servidor Principal.

Na tabela seguinte podem ser vistos exemplos de campos do roteiro a serem criados na base de dados.

Atributo	Tipo	Finalidade
<i>Nome</i>	Caracteres	Identificação do roteiro
<i>URL do itinerário</i>	Caracteres	Visualização do mapa
<i>Data de criação</i>	Data	Dados do roteiro
<i>Autor</i>	Caracteres	Dados do roteiro

Tabela 3.2 - Base de dados do Roteiro

3.2.5 - Criação do Álbum Fotográfico

A aplicação irá permitir que o utilizador tenha a liberdade de fotografar enquanto visita. À medida que o utilizador tira fotografias elas vão sendo enviadas, juntamente com comentários que este queira acrescentar, para a base de dados do Servidor Principal e são gravadas no álbum fotográfico da sua própria conta. No fim da sua viagem poderá aceder a esse álbum multimédia a partir de um Web browser. Autentica-se e visualiza tudo o que gravou durante todo o tempo do trajecto.

3.2.6 - Ligação entre PDA e receptor GPS

A ligação entre o PDA e o receptor GPS é feita através do protocolo NMEA. O receptor precisa de estar em contacto com, no mínimo três a quatro satélites, para que possa ser calculada a sua posição correctamente. O PDA activa o receptor e simula uma porta COM, por onde vai receber toda a informação disponibilizada pelo receptor, que por sua vez recebe informação dos satélites.

A aplicação que está a correr no PDA actualiza a posição constantemente para fornecer as coordenadas ao utilizador com o objectivo de este visualizar no mapa a sua posição e assim manter-se nas linhas do itinerário.



Figura 3.5 - Ligação entre receptor e satélite GPS

3.2.7 - Orientação do Utilizador ao Local

A orientação recorrerá ao carregamento do itinerário gravado na base de dados do roteiro. O URL será carregado pelo PDA e enviado pela internet para servidor do Google, que interpretará o que foi enviado, devolvendo o mapa com todas as características anteriormente gravadas, como marcadores e linhas desenhadas. Depois de o mapa ter sido carregado com sucesso, o utilizador com o auxílio do GPS, só terá de percorrer as linhas desenhadas com o objectivo de o levar ao local.

A ligação do PDA ao Servidor do Google é pode ser observada no esquema seguinte.

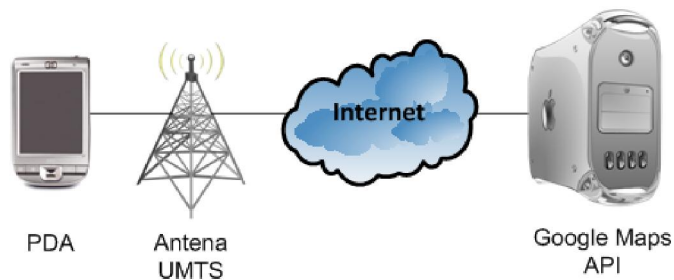


Figura 3.6 - Troca de informação entre Google Maps e aplicação

3.2.8 - Ligação do PDA ao Servidor Local

A ligação do PDA ao servidor local será feita pelo protocolo comum WIFI. Quando o PDA estiver no local, detectará via placa wireless uma rede sem fios. Assim poderá entrar na rede privada através de um controlo de segurança específico que a rede tem previamente configurado, como por exemplo o protocolo de segurança WPA, que obriga o utilizador a introduzir um nome de utilizador e uma palavra passe para ser autenticado correctamente. No servidor local, que possui um servidor apache, irá estar a correr o programa WAMP5 que se vai encarregar de disponibilizar a página Web da aplicação. Assim que o utilizador tiver acesso, vai poder ver e navegar pela informação disponibilizada. A ligação será baseada no esquema seguinte abaixo representado.



Figura 3.7 - Ligação do PDA por WIFI ao Servidor local

3.2.9 - Interação do Utilizador com o Media Center

Depois de o utilizador conseguir a ligação à rede wireless, na aplicação aparecerá um botão que permitirá controlar o Media Center presente no local. O protocolo de ligação será por socket TCP/IP. A aplicação irá estabelecer contacto com o servidor local através de um IP e uma porta. No programa MediaPortal será adicionado um plug-in que irá detectar e receber todas as ligações por socket via WIFI, permitindo assim ao utilizador entrar em contacto com este programa e interagir com o seu menu, podendo visualizar filmes, fotos, e músicas. O esquema seguinte mostra como é feita a ligação socket via WIFI.



Figura 3.8 - Ligação do PDA ao Servidor local por socket via WIFI

3.2.10 - Compra de Souvenirs *on-line*

Quando o utilizador se liga à rede wireless local irá ter acesso à página Web que possui um *link* que abrirá uma nova página Web. Quando esta for carregada, será apresentado um carrinho de compras on-line onde está apresentado um leque variado de souvenirs que pode encomendar, desde que a sua conta tenha um saldo positivo. Essas encomendas serão gravadas numa base de dados local referente à conta do utilizador, podendo ser obtidas à saída do local, onde um assistente com acesso às encomendas na base de dados tratará de entregar o produto. Exemplo de campos das encomendas a serem criadas na base de dados.

Atributo	Tipo	Finalidade
<i>Número de encomenda</i>	Numérico	Dados da encomenda
<i>Nome do utilizador</i>	Caracteres	Dados do utilizador
<i>Data de encomenda</i>	Data	Dados da encomenda
<i>Hora da encomenda</i>	Hora	Dados da encomenda
<i>Nome do produto</i>	Caracteres	Dados da encomenda
<i>Quantidade</i>	Numérico	Dados da encomenda
<i>Preço</i>	Numérico	Dados da encomenda

Tabela 3.3 - Base de dados das encomendas

3.2.11 - Interface

Todo este sistema será utilizado através de uma interface criada propositadamente para este fim. Está dividida por abas em que cada uma corresponde às partes que constituem este sistema. São elas:

- Autenticação
- Carregamento dos roteiros
- Orientação do utilizador para os locais
- Criação do álbum multimédia
- Visualização da presente localização no mapa e interacção
- Visualização da informação do local e compra on-line
- Controlo e interacção com Media Center



Figura 3.9 - Funcionalidades do sistema

3.2 - Requisitos

Os requisitos para este sistema correr, estão identificados a baixo:

Requisitos recomendados do PDA:

- Sistema operativo Windows Mobile 2003 ou superior
- Velocidade do processador 400 MHz ou superior
- Espaço livre em memória RAM de 10MB ou mais
- Ecrã de 240x320 ou 320x320
- Ligação UMTS, HSDPA
- Placa Wireless LAN 802.11 b/g
- GPS incorporado com protocolo NMEA 0183
- Microsoft .NET Compact Framework 3.5
- Câmara fotográfica digital de 3.1 MP

Requisitos recomendados do Servidor principal:

- Processador Intel Xeon 3065 dual-core 2,33Ghz
- Memória RAM 2GB
- Disco rígido SATA de 250Gb 3,5" a 7200 rpm
- Placa Ethernet 10/100Mbps
- Sistema operativo Windows Server 2008
- Programa PostgreSQL 8.3.3 ou MySQL 5.0
- Programa Wamp5

Requisitos recomendados dos Servidores locais:

- Processador Intel Xeon 3065 dual-core 2,33Ghz
- Memória RAM 2GB
- Disco rígido SATA de 250GB 3,5" a 7200 rpm
- Placa Ethernet 10/100Mbps
- Sistema operativo Windows Server 2008
- Programa Wamp5
- Programa MediaPortal
- Placa Wireless LAN 802.11 b/g

Capítulo 4

Implementação

Neste capítulo está apresentada a implementação de uma aplicação demonstrativa de como o projecto pode funcionar. Serão apresentadas todas as justificações para as opções tomadas. Ficou decidido que nesta aplicação se iria utilizar a linguagem de programação C#.NET. Esta decisão foi tomada porque o sistema operativo da Microsoft no PDA evita que haja conflitos de compatibilidade de software e hardware. O facto de haver vários programadores que partilham e disponibilizam a informação na internet também facilita a aprendizagem da própria linguagem por ser de alto nível.

Como o material de trabalho e o tempo eram escassos para fazer uma aplicação que simulasse a proposta previamente descrita, algumas características do sistema sofreram umas ligeiras alterações de modo a ser possível a utilização do material disponível, e adequar o tempo disponível ao tempo necessário de implementação. Contudo, apesar das alterações as funcionalidades mantiveram-se. É preciso sublinhar que o trabalho implementado é apenas uma simulação da aplicação proposta, demonstrando as suas reais potencialidades.

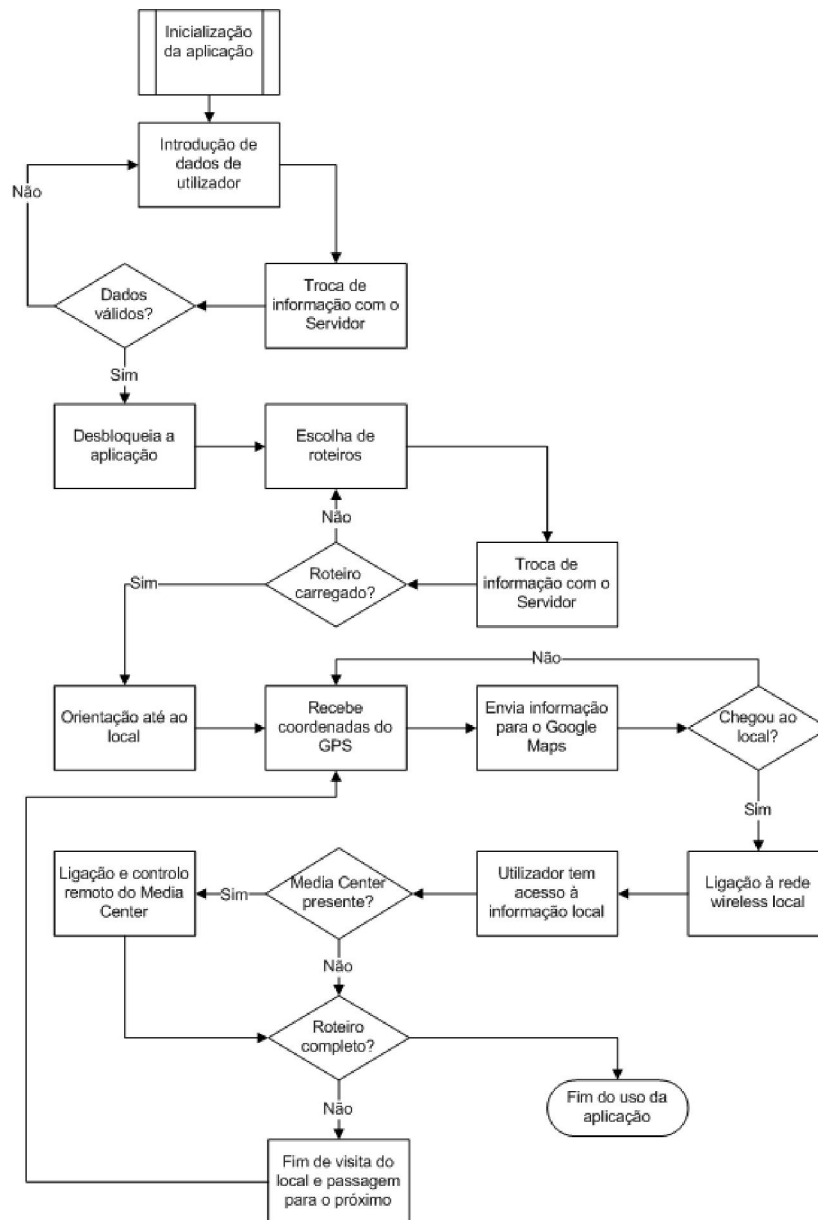


Figura 4.1 - Fluxograma do sistema

4.1 - Material Envolvido na Simulação do Sistema

Neste projecto foram usados dispositivos electrónicos e várias aplicações informáticas. Abaixo estão apresentados todos os equipamentos e programas que foram utilizados para a implementação do projecto:

Hardware:

- Um computador portátil HP Pavilion dv4000
- Um PDA HP iPAQ h5550
- Um receptor GPS Fortuna Clip-On bluetooth
- Ligação do computador portátil à internet (banda larga)

Software:

- Windows Vista Business Edition (HP Pavilion)
- Windows Pocket PC 2003 (HP iPAQ)
- .NET Compact Framework
- Microsoft Visual Studio 2008
- WMAP5
- MediaPortal

4.2 - Comunicação

A primeira parte do trabalho consistiu em desenvolver código para que um cliente e um servidor pudessem comunicar entre si. Para isso foi utilizada uma comunicação socket TCP/IP. Esta ligação permite que duas máquinas possam trocar informação entre si, enviando e recebendo bytes de uma forma rápida e síncrona. Consiste em ter um servidor a correr um código que abrirá uma porta e estará a detectar ligações no seu IP. Deste modo o cliente quando quiser aceder ao servidor terá de iniciar uma ligação TCP/IP, referindo o IP do servidor ao qual se quer ligar e à respectiva porta onde este está à escuta, abrindo assim um canal de comunicação que irá permitir a troca de informação.

Todo este processo de comunicação estará a correr em background sem a visualização do utilizador.

Podemos ver no esquema seguinte como a ligação é feita e quando a informação é trocada.

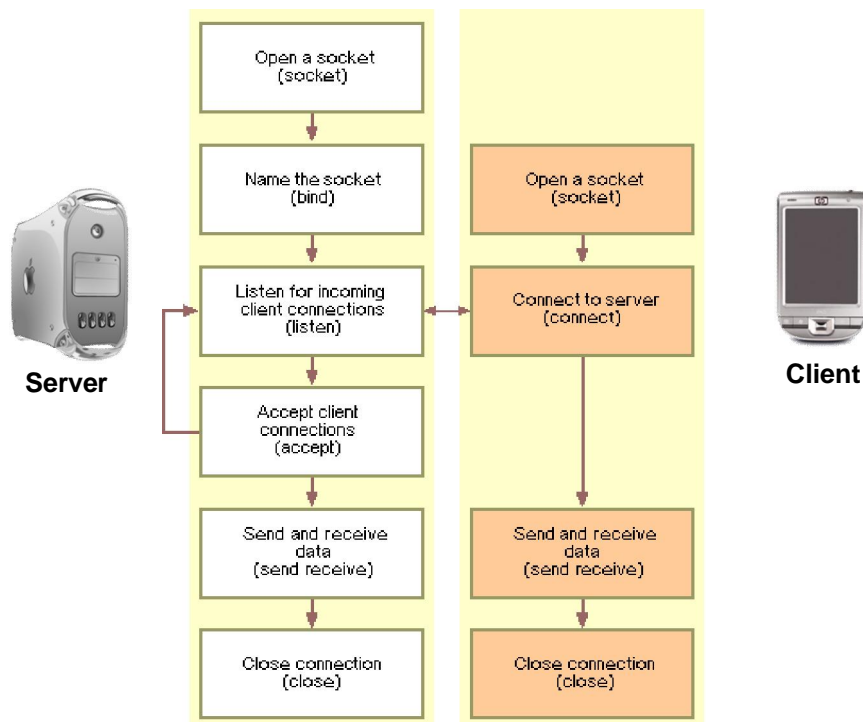


Figura 4.2 - Comunicação Socket

No Servidor e Cliente são utilizados Namespaces, classes e métodos para se implementar o código responsável por detectar e criar ligações nas Sockets e a consequente transferência de dados:

System.Net.Sockets - é um namespace que funciona como uma library, que permite manipular a interface das sockets do Windows (Winsock) para ter controlo sobre o acesso à rede.

TcpListener - É uma classe presente no Servidor e fornece métodos simples que escutam e aceitam pedidos de ligação feitas pelo cliente, de modo síncrono.

Start e *Stop* - São métodos incluídos na classe *TcpListener*. Estes irão começar e parar a escuta no IP e porta introduzida.

TcpClient - É uma classe presente no Cliente e fornece métodos simples para se ligar, enviar e receber fluxo de dados numa ligação de modo síncrono.

Socket - É uma classe que fornece um conjunto de métodos e propriedades para a comunicação na rede.

AcceptSocket e *Close* - São métodos da classe *Socket* que servem para aceitar e fechar as ligações anteriormente criadas.

NetworkStream - A classe fornece métodos para enviar e receber dados sobre ligação por sockets.

StreamWriter e *StreamReader* - São classes em que a sua função é ler e escrever caracteres de uma determinada codificação.

Flush - É um método que serve para descarregar o conteúdo do buffer, que anteriormente foi "enchido" com caracteres, para serem enviados para a socket.

Nos anexos encontra-se o código desenvolvido, assim como o uso destas classes e métodos.

4.3 - Criação da Interface

Depois de implementado o código para ligar o Cliente ao Servidor, é agora necessário um interface que irá permitir ao utilizador interagir com o programa. Toda a interface foi desenhada no Visual Studio 2008 através de um construtor GUI próprio, preparado para o formato das janelas do Windows Pocket PC.

Pensou-se em construir uma interface simples e atraente de modo a proporcionar ao utilizador uma maneira fácil e intuitiva de navegar pela aplicação.

Na interface, as funcionalidades foram divididas por separadores, e à medida que se avança no roteiro, são disponibilizadas novas abas para o utilizador explorar.

A aplicação está assim dividida da seguinte forma:

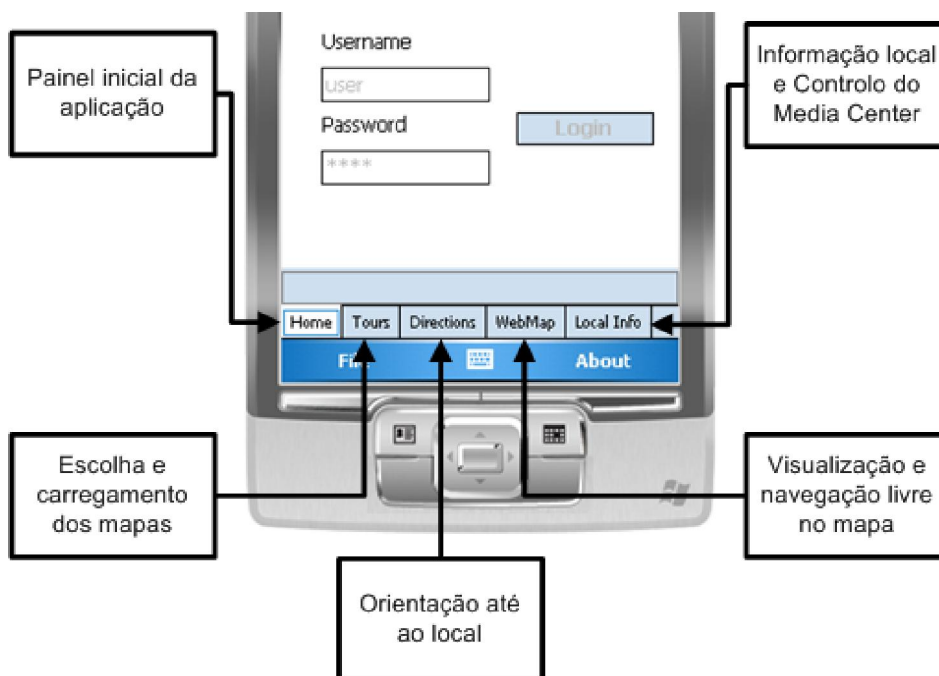


Figura 4.3 - Divisão de funcionalidades

À medida que se avançar nas funcionalidades da aplicação ir-se-á explicar todos os campos, botões e imagens presentes nesta.

4.4 - Sistema de dados em Ficheiros

O sistema de dados em ficheiros foi implementado de uma maneira simples. Como é uma aplicação para demonstrar que é possível o funcionamento do sistema optou-se por introduzir informação em ficheiros presentes no disco do Servidor. Os ficheiros estão divididos por pastas contendo a informação de utilizadores, coordenadas de mapas e imagens de mapas. A explicação do conteúdo dos ficheiros estará presente nas funcionalidades seguintes.

Um esquema da árvore de ficheiros no disco rígido está ilustrado na Figura 4.4.

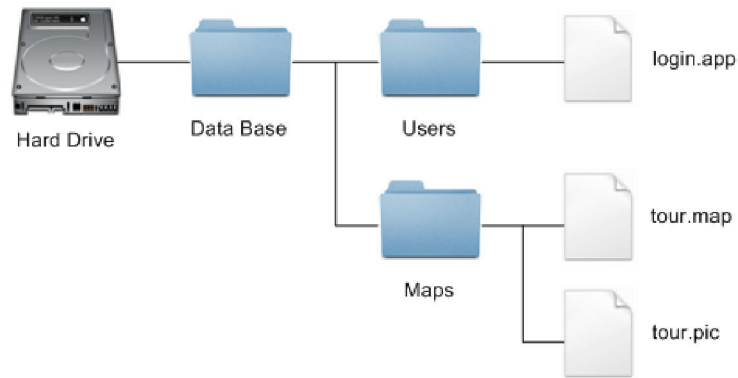


Figura 4.4 - Árvore de ficheiros

4.5 - Criação de Utilizadores e Autenticação

Para criar um utilizador basta abrir o ficheiro *login.app* na pasta *Users* e introduzir o nome de utilizador e a respectiva palavra passe, separados por ";". Ficando, por exemplo, o seguinte texto no ficheiro:

```
>user;pass
>user1;pass1
```

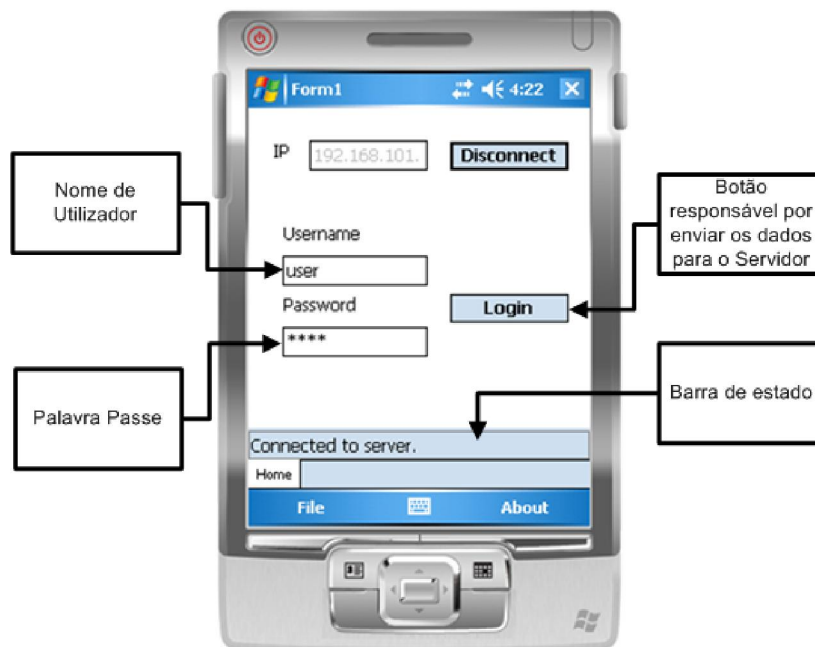


Figura 4.5 - Interface de autenticação

Os dados a introduzir na aplicação, como está representado na Figura 4.5, para autenticar o utilizador, irão ser inseridos através da interface no Cliente. Este depois de estabelecer a ligação por sockets irá enviar o nome de utilizador e a palavra passe para o

Servidor. Finda a transmissão do Cliente para o Servidor, este abre o ficheiro *login.app* e determina se os dados recebidos correspondem aos dados presentes na base de dados.

Em caso afirmativo, envia uma mensagem de boas vindas para o Cliente a confirmar a autenticação, dando autorização ao utilizador para explorar a aplicação. Em caso negativo, envia uma mensagem com a informação de que os dados estão incorrectos, e a aplicação aconselha o utilizador a introduzir novamente os dados.

4.6 - Criação de Itinerários

A criação de itinerários foi desenvolvida com a ajuda do *Google Static Map Wizard* presente na página do *Google Static Maps API*. Este *wizard* ou assistente tem como objectivo, através de uma interface mais *user-friendly*, construir o URL com todas as características e parâmetros da API. Pode-se criar marcadores, desenhar caminhos, determinar o tamanho do mapa, e ainda alterar as características dos marcadores e dos caminhos. Contudo este assistente apenas ajuda no essencial, ficando ao critério do programador a introdução de alterações ou até mesmo a inclusão de características e parâmetros complementares. Para isto terá de alterar o URL automaticamente criado.

Este assistente é constituído por três partes. Na primeira, escolhe-se o local, são criados os marcadores e também é desenhado o caminho. Na segunda, personalizam-se os marcadores e os caminhos, visualizando o aspecto final do mapa. E finalmente na terceira, tem-se acesso ao URL automaticamente criado momentos antes na primeira e segunda parte.

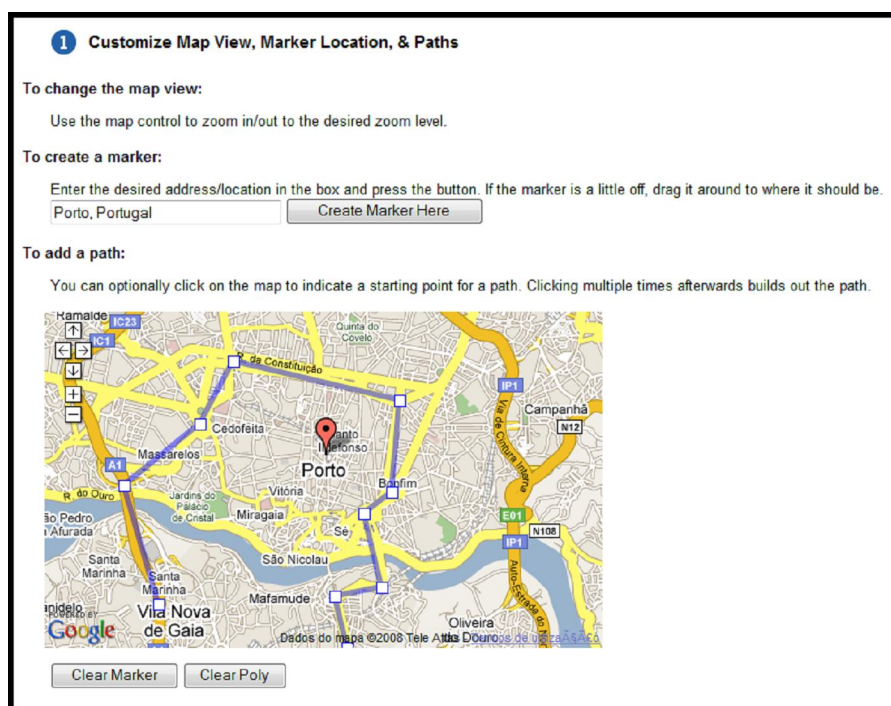


Figura 4.6 - Primeiro passo do *Google Static Map Wizard*

Relativamente à escolha dos roteiros, a sua implementação consiste em primeiro lugar, facultar ao utilizador, por parte da aplicação, os roteiros disponíveis na base de dados. Depois da escolha efectuada, o Cliente pede os dados contidos no ficheiro do roteiro escolhido. O Servidor irá ler os dados do roteiro escolhido e envia-os para o Cliente juntamente com a imagem que corresponde ao ficheiro requerido. A imagem será visualizada pelo utilizador, enquanto os dados do ficheiro "Porto.map" por exemplo, vão ser utilizados em background pela aplicação.



Figura 4.9 - Interface da escolha de roteiros

4.7 - Comunicação PDA com GPS

A ligação entre o GPS e o PDA foi realizada através de porta COM. Através do protocolo de comunicação *bluetooth*, o PDA e o receptor GPS são emparelhados. Simultaneamente, é criada uma porta COM virtual que vai ser responsável pela troca de informação entre estes dois dispositivos. O receptor GPS possui um protocolo NMEA 0183 e serão utilizados alguns dos parâmetros existentes no grupo GPGGA, tais como a latitude e a longitude. Para implementar o código, basta abrir a porta COM responsável pela ligação do GPS, captar a informação do grupo GPGGA que é disponibilizado pelo receptor e depois manipulá-la de maneira a obter as coordenadas de posição: latitude e longitude. O código responsável por esta ligação está na secção de anexos.

4.8 - Orientação ao Local e WebMap

A orientação até ao local é feita através das coordenadas recebidas pelo GPS e do serviço *Google Static Maps*. Depois de obtidas as coordenadas, a aplicação constrói um URL para enviar ao serviço do Google Maps via internet. Este processa a informação enviada e retorna para a aplicação uma imagem do mapa anteriormente requerido para que o utilizador possa visualizar a sua posição.

À medida que as coordenadas vão sendo alteradas devido à deslocação do utilizador, a aplicação vai actualizando, através de um *timer* responsável pelas interrupções, o URL, enviando-o para o serviço do Google. Consequentemente são facultadas ao utilizador actualizações da sua posição permitindo uma orientação em tempo real. Com esta funcionalidade o utilizador é conduzido ao local através das linhas desenhadas no mapa aquando da sua concepção. Por fim, para o utilizador chegar ao destino só terá de percorrer as referidas linhas que o orientam ao local.



Figura 4.10 - Interface de orientação

Para o utilizador ter uma visualização geral de onde se encontra, poderá consultar o WebMap, apresentado na figura 4.11, que permite visualizar o mapa da sua posição. Assim pode observar os locais em seu redor, ver se os pontos de visita estão longe ou perto da sua posição e inclusivamente consultar outros locais exteriores à visita. Para isso, possui um controlador de Zoom e um curso que permite arrastar o mapa para Norte, Sul, Este e Oeste.



Figura 4.11 - Interface do WebMap

4.9 - Informação Local e Interação com Media Center

Quando o utilizador chega ao local a aplicação detecta uma rede wireless e vai tentar entrar em ligação com a mesma. Neste momento, o utilizador insere as suas credenciais para proceder à autenticação na rede privada protegida por um protocolo de segurança específico. Quando a ligação entre o PDA e a rede wireless é efectuada, o utilizador pode carregar no botão "Information" presente na aplicação, para deste modo abrir uma página Web alojada no Servidor local.

A página Web irá ficar *on-line* com a ajuda do programa WAMP5 que corre no Servidor Local. Este programa possui um servidor HTTP chamado Apache onde pode ser alojada qualquer tipo de página Web, sendo publicada quando é requisitada. Esta página pode conter variada informação sobre o local, sendo o seu código concretizado em várias linguagens de programação, como por exemplo, HTML, PHP, CSS ou XHTML.

Caso o local possua um Media Center, o utilizador pode aceder a ele e controlá-lo remotamente via Wi-Fi. A aplicação abrirá uma ligação por socket entre o PDA e o Servidor local estabelecendo um canal de comunicação entre a aplicação e o Media Center. O programa de Media Center utilizado no Servidor local é o MediaPortal, um software *open-source* onde se podem instalar vários *plug-ins*, o qual irá servir para criar um controlo remoto Wi-Fi. O botão responsável por esse controlo é o “LCD Control”.

Podemos observar na figura seguinte a interface de informação.



Figura 4.12 - Interface de informação

Quando a ligação entre a aplicação e o Media Center é estabelecida, aparece uma outra interface gráfica, como podemos constatar na figura seguinte. O código desta interface é uma reutilização de um projecto *open-source*, o qual foi alterado de modo a ser possível incluir nesta aplicação.

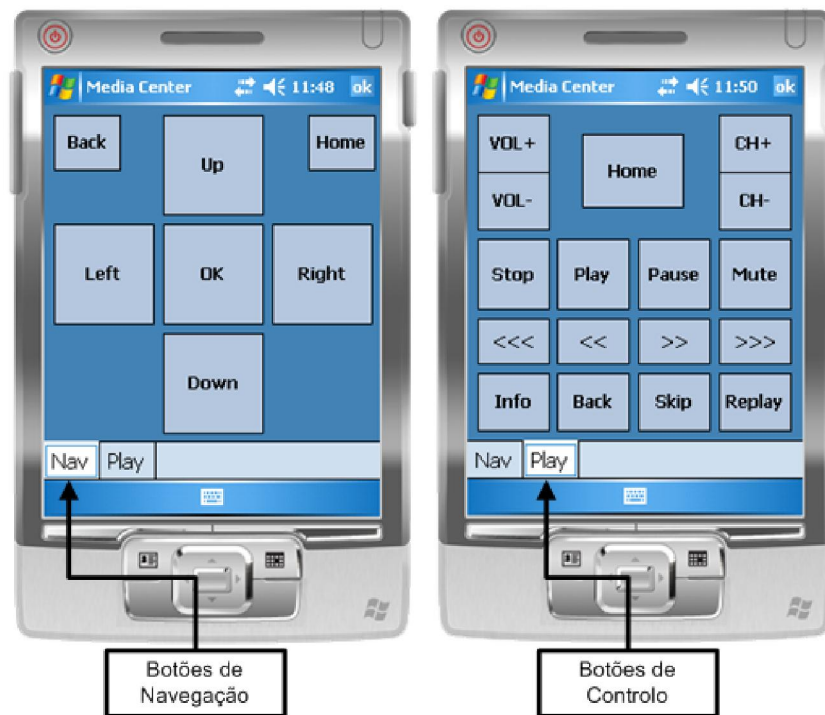


Figura 4.13 - Interface de controlo remoto

O MediaPortal corre no Servidor local onde estará ligado um LCD que apresentará ao público a sua interface. Podemos observar na figura seguinte o aspecto que terá o LCD exposto aos visitantes.



Figura 4.14 - Interface do MediaPortal

Capítulo 5

Conclusão e Desenvolvimentos Futuros

Todos os objectivos inicialmente propostos foram concretizados com êxito e o desenvolvimento desta aplicação multimédia foi um sucesso.

Apesar das dificuldades ao longo de todo o trabalho, ficou aqui patenteado que é uma aplicação válida e inovadora que pode alterar o modo como cada indivíduo vive as suas experiências turísticas e de lazer. Cada vez mais os dispositivos electrónicos fazem parte do nosso dia-a-dia, tornando-se mesmo alguns deles imprescindíveis. Foram criados com o intuito de nos fornecer ferramentas e recursos outrora inexistentes, de modo a simplificar a mais banal ou complexa tarefa do nosso quotidiano. Tempo e autonomia são duas palavras que invocam e evidenciam as potencialidades desta aplicação.

A proposta desta tese foi explicada teoricamente, apresentando ideias excelentes e viáveis para tornar esta aplicação numa das melhores no mercado do turismo electrónico. Foi implementado um sistema baseado nesta proposta, com algumas alterações devido à falta de equipamento, com o intuito de demonstrar que é possível construir aplicações divertidas e intuitivas para viajar, podendo esta oferecer interacção do utilizador com o meio envolvente e não se reduzir apenas a uma aplicação limitada ao tempo e ao espaço.

Este software tem ainda uma intrínseca margem de evolução considerável, visto que o tempo de implementação e os recursos disponíveis não foram os ideais. No entanto, tomando como ponto de partida o que já foi criado e adicionando-lhe mais recursos, quer em termos de hardware quer em termos de software, pode-se obter um produto de alto nível e de elevada procura no mercado.

Em relação a desenvolvimentos futuros, haverá algumas melhorias a fazer, como a ligação entre o Cliente e o Servidor, que deverá ter como base de comunicação um Servidor HTTP, com o objectivo de comunicar com segurança, pela porta 80. Terão também de ser implementadas as funcionalidades não desenvolvidas, como o Álbum Fotográfico e a loja de souvenirs on-line no local. Deverá ser possível ao utilizador, manipular o Google Maps API para deste modo desenhar e planear os seus próprios roteiros. Por fim, depois de todas estas funcionalidades estarem cem por cento implementadas, a construção de um dispositivo de Hardware, englobando todos os aparelhos necessários, deverá ser concretizada para uso exclusivo da aplicação com vista à sua comercialização.

Todas a funcionalidades apresentadas na implementação foram testadas e os seus resultados reais. Pode-se ver excertos do código na secção de anexos no fim desta dissertação.

Referências

- [1] Curso Multimédia. Disponível em http://www.citi.pt/Multi_sem_frames.html. Acesso em Dezembro 2007.
- [2] Curso Multimédia. Disponível em <http://www.citi.pt/multimedia/interac.html>. Acesso em Dezembro 2007.
- [3] Usabilidade e Comunicação na Internet. Disponível em <http://www.seara.com/fotos/editor2/usabilidade.pdf>. Acesso em Dezembro 2007.
- [4] How GPS Works. Disponível em <http://videos.howstuffworks.com/howstuffworks/38-how-gps-works-video.htm>. Acesso em Dezembro 2007.
- [5] NAVSTAR GPS. Disponível em http://www.spaceandtech.com/spacedata/constellations/navstar-gps_consum.shtml. Acesso em Dezembro 2007.
- [6] NMEA 0183. Disponível em <http://www.nmea.org/pub/0183/>. Acesso em Dezembro 2007.
- [7] NMEA 2000. Disponível em <http://www.nmea.org/pub/2000/>. Acesso em Dezembro 2007.
- [8] "Introduction to the Controller Area Network", Steve Corrigan. Disponível em <http://focus.ti.com/lit/an/sloa101/sloa101.pdf>. Acesso em Dezembro 2007.
- [9] SiRF. Disponível em <http://www.sirf.com/>. Acesso em Dezembro 2007.
- [10] "The Evolution of PDA", Evan Koblentz. Disponível em <http://www.snarc.net/pda/pda-treatise.htm>. Acesso em Dezembro 2007.
- [11] Wi-Fi Wireless Networks and Technology. Disponível em http://compnetworking.about.com/od/wireless/WiFi_Wireless_Networks_and_Technology.htm. Acesso em Maio 2008.

- [12] Wi-Fi. Disponível em <http://www.wi-fi.org/>. Acesso em Dezembro 2007.
- [13] PocketPC GPS and Navigation Software. Disponível em <http://www.gpsinformation.org/dale/PocketPC/wince.htm>. Acesso em Dezembro 2007.
- [14] HSDPA in W-CDMA. Disponível em <http://www.umtsworld.com/technology/hsdpa.htm>. Acesso em Dezembro 2007.
- [15] Mobile Java ou Java ME. Disponível em http://www.java.com/pt_BR/download/faq/whatis_j2me.xml. Acesso em Dezembro 2007.
- [16] What is .NET. Disponível em <http://www.on-dotnet.com/pub/a/dotnet/2005/09/06/what-is-dotnet.html>. Acesso em Dezembro 2007.
- [17] "Curricula for Human-Computer Interaction", Association for Computing Machinery, Inc. Disponível em <http://sigchi.org/cdg/>. Acesso em Dezembro 2007.
- [18] "There is more to Context than Location", Albrecht Schmidt, Michael Beigl, and Hans-W. Gellersen. Disponível em http://www.comp.lancs.ac.uk/~albrecht/pubs/pdf/schmidt_cug_elsevier_12-1999-context-is-more-than-location.pdf. Acesso em Dezembro 2007.
- [19] Location Awareness in Hardware, Brady Forrest. Disponível em <http://radar.oreilly.com/archives/2006/10/location-awareness-in-hardware.html>. Acesso em Dezembro 2007.
- [20] Dodgeball. Disponível em <http://www.dodgeball.com/>. Acesso em Dezembro 2007.
- [21] "Ubiquitous Computing: An Interesting New Paradigm", Marcia Riley. Disponível em http://www.cc.gatech.edu/classes/cs6751_97_fall/projects/say-cheese/marcia/mfinal.html. Acesso em Dezembro 2007.
- [22] Journal of Information Technology & Tourism, Cognizant.

Anexos

Excerto de código do Servidor (server.cs)

```
while (true)
{
    Int32 port = xxxxx;
    IPAddress localaddr = IPAddress.Parse("xxx.xxx.xxx.xxx");
    TcpListener tcpListener = new TcpListener(localaddr, port);
    tcpListener.Start();
    Socket socketForClient = tcpListener.AcceptSocket();
    try
    {
        if (socketForClient.Connected)
        {
            while (true)
            {
                Console.WriteLine("Client connected");
                NetworkStream networkStream = new NetworkStream(socketForClient);
                StreamWriter streamWriter = new StreamWriter(networkStream);
                StreamReader streamReader = new StreamReader(networkStream);
                string line = streamReader.ReadLine();
                Console.WriteLine("Read:" + line);
                if (line.Equals("exit"))
                {
                    break;
                }
                if (line.Equals("Are you listenig?"))
                {
                    streamWriter.WriteLine("yes");
                    streamWriter.Flush();
                }
                if (line.Equals("Map"))
                {
```

```

DirectoryInfo map = new DirectoryInfo("C:/teste/maps/");
FileInfo[] maps = map.GetFiles("*.map");
streamWriter.WriteLine(maps.Length);
streamWriter.Flush();
for (int i = 0; i <= (maps.Length)-1; i++)
{
    streamWriter.WriteLine(maps[i]);
    streamWriter.Flush();
}
}
if (line.Equals("openMap"))
{
    string map = streamReader.ReadLine();
    StreamReader file_log = File.OpenText("c:/teste/maps/" + map);
    string read = file_log.ReadToEnd();
    read = read.Replace("\r\n", ";");
    Console.WriteLine(read);
    streamWriter.WriteLine(read);
    streamWriter.Flush();
    string pic = streamReader.ReadLine();
    string[] tempPic = pic.Split('.');
    FileStream picture = File.OpenRead("c:/teste/maps/" + tempPic[0] + ".jpg");
    byte[] mapaPic = new byte[picture.Length];
    picture.Read(mapaPic, 0, mapaPic.Length);
    streamWriter.WriteLine(picture.Length);
    streamWriter.Flush();
    for (int i = 0; i <= (picture.Length) - 1; i++)
    {
        streamWriter.WriteLine(mapaPic[i]);
        streamWriter.Flush();
    }
}
}
if (line.Equals("login"))
{
    string login = null;
    string read = null;
    string s2 = "null";
    streamWriter.WriteLine("ok");
    streamWriter.Flush();
    login = streamReader.ReadLine();
    login = login.ToLower();
    StreamReader file_log = File.OpenText("c:/teste/users/login.txt");
    while ((read = file_log.ReadLine()) != null)
    {
        if (read.Equals(login))
        {

```

```

        s2 = "valid";
        break;
    }
}
if (s2.Equals("valid"))
{
    streamWriter.WriteLine("valid_login");
    streamWriter.Flush();
}
else if (s2.Equals("null"))
{
    streamWriter.WriteLine("invalid_login");
    streamWriter.Flush();
}
file_log.Close();
}
}
socketForClient.Close();
tcpListener.Stop();
Console.WriteLine("Exiting Client...");
}
catch
{
    socketForClient.Close();
    tcpListener.Stop();
    Console.WriteLine("Exiting Client...");
}
}
}

```

Excerto de código do Cliente (app.cs)

```
TcpClient myclient;
private NetworkStream networkStream;
private StreamReader streamReader;
private StreamWriter streamWriter;
public string Latitude, Longitude, webMap, webMarkers, webPath, latDest, longDest, mediaIp;
public int Zoom = 16, mediaPort = 8100;
public double N, S, W, E, V, H;
public string[] local, coord, nameIP;

public static void Main()
{
    Application.Run(new App());
}
public App()
{
    InitializeComponent();
    Unauthorized();
}
public void socketConnect()
{
    statusBar1.Text = "Connecting...";
    try
    {
        myclient = new TcpClient(textBox3.Text, 666);
        networkStream = myclient.GetStream();
        streamReader = new StreamReader(networkStream);
        streamWriter = new StreamWriter(networkStream);
    }
    catch
    {
        Application.Exit();
    }
    try
    {
        streamWriter.WriteLine("Are you listenig?");
        streamWriter.Flush();
        string connect;
        connect = streamReader.ReadLine();
        if (connect.Equals("yes"))
        {
            statusBar1.Text = "Connected to server.";
            button2.Text = "Disconnect";
            textBox1.Enabled = true;
            textBox2.Enabled = true;
        }
    }
}
```

```

        buttonLogIn.Enabled = true;
        textBox3.Enabled = false;
    }
}
catch
{
    statusBar1.Text = "Error connecting, please try again.";
    streamReader.Close();
    streamWriter.Close();
    networkStream.Close();
}
}
private string TurnFirstToUpper(string input)
{
    string first = input.Substring(0, 1);
    string rest = input.Remove(0, 1);
    return input = first.ToUpper() + rest;
}
private void Authorized()
{
    try
    {
        tabControl1.TabPages.Add(tabPage2);
        tabControl1.TabPages.Add(tabPage3);
        tabControl1.TabPages.Add(tabPage4);
        tabControl1.TabPages.Add(tabPage5);
    }
    catch { }
}
private void Unauthorized()
{
    try
    {
        tabControl1.TabPages.RemoveAt(1);
        tabControl1.TabPages.RemoveAt(1);
        tabControl1.TabPages.RemoveAt(1);
        tabControl1.TabPages.RemoveAt(1);
    }
    catch { }
}
private void button1_Click(object sender, EventArgs e)
{
    string user = textBox1.Text;
    string pass = textBox2.Text;
    if (buttonLogIn.Text.Equals("Login"))
    {

```

```

string s = null;
statusBar1.Text = "Logging in...";
streamWriter.WriteLine("login");
streamWriter.Flush();
s = streamReader.ReadLine();
streamWriter.WriteLine(user + ";" + pass);
streamWriter.Flush();
s = streamReader.ReadLine();
if (s.Equals("valid_login"))
{
    user = TurnFirstToUpper(user);
    statusBar1.Text = "Welcome " + user + "...";
    buttonLogIn.Text = "Logout";
    textBox1.Enabled = false;
    textBox2.Enabled = false;
    Authorized();
    streamWriter.WriteLine("Map");
    streamWriter.Flush();
    s = streamReader.ReadLine();
    int i;
    string[] maps = new string[Convert.ToInt32(s)];
    for (i = 0; i < Convert.ToInt32(s);i++)
    {
        string temp = streamReader.ReadLine();
        maps[i] = temp;
        comboBox1.Items.Add(maps[i]);
    }
}
else
{
    statusBar1.Text = "Login failed, please try again.";
}
}
else if (buttonLogIn.Text.Equals("Logout"))
{
    buttonLogIn.Text = "Login";
    button7_Click(sender, e);
    comboBox1.Items.Clear();
    textBox1.Enabled = true;
    textBox2.Enabled = true;
    buttonLogIn.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    user = TurnFirstToUpper(user);
    statusBar1.Text = "Good bye " + user + "...";
    Unauthorized();
}

```

```

        //serialPort1.Close();
        timer1.Enabled = false;
    }
}
private void button2_Click(object sender, EventArgs e)
{
    if (button2.Text.Equals("Connect"))
    {
        socketConnect();
    }
    else
    {
        button2.Text = "Disconnect";
        statusBar1.Text = "Disconnected from server.";
        streamWriter.WriteLine("exit");
        streamWriter.Flush();
        button7_Click(sender, e);
        streamReader.Close();
        streamWriter.Close();
        networkStream.Close();
        button2.Text = "Connect";
        buttonLogIn.Text = "Login";
        textBox3.Enabled = true;
        textBox1.Enabled = false;
        textBox1.Text = "";
        textBox2.Enabled = false;
        textBox2.Text = "";
        buttonLogIn.Enabled = false;
        Unauthorized();
        timer1.Enabled = false;
        comboBox1.Items.Clear();
    }
}
private void button5_Click(object sender, EventArgs e)
{
    if (button5.Text.Equals("Stop"))
    {
        timer1.Enabled = false;
        button5.Text = "Get GPS";
    }
    else
    {
        try
        {
            if (serialPort1.IsOpen)
            {

```

```

        timer1.Enabled = true;
        button5.Text = "Stop";
    }
    else
    {
        serialPort1.PortName = "COM8";
        serialPort1.Open();
        timer1.Enabled = true;
        button5.Text = "Stop";
    }
}
catch { }
}
}
private void timer1_Tick(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        string data = serialPort1.ReadExisting();
        string[] strArr = data.Split('$');
        for (int i = 0; i < strArr.Length; i++)
        {
            string strTemp = strArr[i];
            string[] lineArr = strTemp.Split(';');
            if (lineArr[0] == "GPGGA")
            {
                try
                {
                    //Latitude
                    double dLat = Convert.ToDouble(lineArr[2]);
                    dLat = dLat / 100;
                    string[] lat = Convert.ToString(dLat).Split('.');
                    Latitude = lineArr[3] + lat[0] + "." + Convert.ToInt32(Convert.ToInt32(lat[1]) / .06);
                    Latitude = Latitude.Remove(0, 1);
                    //Longitude
                    double dLon = Convert.ToDouble(lineArr[4]);
                    dLon = dLon / 100;
                    string[] lon = Convert.ToString(dLon).Split('.');
                    Longitude = lineArr[5] + lon[0] + "." + Convert.ToInt32(Convert.ToInt32(lon[1]) / .06);
                    Longitude = "-" + Longitude.Remove(0, 1);
                    //Display
                    textBox4.Text = Latitude;
                    textBox5.Text = Longitude;
                    //Near the local
                    for(int j = 1; j < coord.Length; j++)
                    {

```

```

        string[] latTemp = Latitude.Split('.');
        string latitudeTemp = latTemp[0]+"."+latTemp[1].Substring(0, 3);
        string[] longTemp = Longitude.Split('.');
        string longitudeTemp = longTemp[0]+"."+longTemp[1].Substring(0, 3);
        if ((latitudeTemp + "|" + longitudeTemp).Equals(coord[j]))
        {
            timer1.Enabled = false;
            button5.Text = "Get GPS";
            medialp = nameIP[j];
            button9_Click(sender, e);
            MessageBox.Show("Voce chegou ao local: " + nameIP[j]);
            break;
        }
    }
}
catch {}
break;
}
}
}
}
private void button3_Click(object sender, EventArgs e)
{
    buttonOpenMap.Enabled = false;
    button6.Enabled = true;
    buttonCloseMap.Enabled = true;
    streamWriter.WriteLine("openMap");
    streamWriter.Flush();
    string s = comboBox1.Text;
    streamWriter.WriteLine(s);
    streamWriter.Flush();
    string temp1 = streamReader.ReadLine();
    local = temp1.Split(';');
    webMarkers = local[0];
    webPath = local[1];
    coord = new string[15];
    //Coordenadas para o inicio do percurso
    latDest = local[4];
    longDest = local[5];
    for (int i = 4, j = 1; i < local.Length; i = i + 6, j++)
    {
        //listBox1.Items.Add(local[i]+"."+local[i+1]);
        coord[j] = local[i]+"|" +local[i+1];
    }
    nameIP = new string[15];
    for (int i = 3, j = 1; i < local.Length; i = i + 6, j++)

```

```

    {
        nameP[j] = local[i];
    }
    streamWriter.WriteLine(s);
    streamWriter.Flush();
    string temp2 = streamReader.ReadLine();
    byte[] mapaPic = new byte[Convert.ToInt32(temp2)];
    for (int i = 0; i < Convert.ToInt32(temp2); i++)
    {
        string temp = streamReader.ReadLine();
        mapaPic[i] = Convert.ToByte(temp);
    }
    MemoryStream ms = new MemoryStream(mapaPic);
    Bitmap bmp = new Bitmap(ms);
    pictureBox1.Image = bmp;
}
private void button7_Click(object sender, EventArgs e)
{
    buttonOpenMap.Enabled = true;
    button6.Enabled = false;
    buttonCloseMap.Enabled = false;
    pictureBox1.Image = null;
    coord = null;
}
private void button9_Click(object sender, EventArgs e)
{
    string[] latTemp = Latitude.Split('.');
    string latitudeTemp = latTemp[0] + "." + latTemp[1].Substring(0, 3);
    string[] longTemp = Longitude.Split('.');
    string longitudeTemp = longTemp[0] + "." + longTemp[1].Substring(0, 3);
    string[] ltDestTemp = latDest.Split('.');
    string latDestTemp = ltDestTemp[0] + "." + ltDestTemp[1].Substring(0, 3);
    string[] lgDestTemp = longDest.Split('.');
    string longDestTemp = lgDestTemp[0] + "." + lgDestTemp[1].Substring(0, 3);
    if (latitudeTemp.Equals(latDestTemp) && longitudeTemp.Equals(longDestTemp))
    {
        string target = "http://maps.google.com/staticmap?center=" + Latitude + "," + Longitude + "&markers=" +
Latitude + "," + Longitude + ",midred|41.219227,-8.551548,bluec|41.217016,-8.553672,bluee|41.215579,-
8.553178,bluef&path=rgba:0x0000FF80,weight:5|41.21926,-8.55159|41.21866,-8.55103|41.21811,-
8.55230|41.21718,-8.55193|41.21703,-8.55365|41.21647,-8.55352|41.21560,-8.55318|&zoom=" + Zoom +
"&size=240x222&key=ABQIAAAA0YgPFQlikEmnRQ48fiqZKRO5gPuthJSplTW50EQVijBR6zKL3RTw5OYK2EEIo4nn_Sta
XzupVfWKag";
        webBrowser2.Navigate(new Uri(target));
    }
    else
    {

```

```

        string webTemp = "http://maps.google.com/staticmap?center=" + Latitude + "," + Longitude +
"&markers=" + Latitude + "," + Longitude + ",midred|" + latDest + "," + longDest +
",bluec&path=rgba:0xff000080,weight:5|" + Latitude + "," + Longitude + "|" + latDest + "," + longDest + "|" &zoom="
+ Zoom +
"&size=240x222&key=ABQIAAAA0YgPFQliKEmnRQ48fiqZKRO5gPuthJSpLTw50EQVijBR6zKL3RTw5OYK2EEIo4nn_Sta
XzupVfWKag";
        webBrowser2.Navigate(new Uri(webTemp));
    }
}
private void button10_Click(object sender, EventArgs e)
{
    V = Convert.ToDouble(Latitude);
    H = Convert.ToDouble(Longitude);
    Zoom = 16;
    webMap = "http://maps.google.com/staticmap?center=" + V + "," + H + "&markers=" + Latitude + "," +
Longitude + ",midred|" + webMarkers + "&path=rgba:0x0000FF80,weight:5|" + webPath +
"&zoom=" + Zoom + "&size=240x246&key=ABQIAAAA0YgPFQliKEmnRQ48fiqZKRTNNCNNEvQP7NISal3Vj_WI-
jROIRO7VSbX3Y_wEqwtcm045eV-QnNXmA";
    webBrowser1.Navigate(new Uri(webMap));
}
private void button15_Click(object sender, EventArgs e)
{
    Zoom = Zoom+1;
    webMap = "http://maps.google.com/staticmap?center=" + V + "," + H + "&markers=" + Latitude + "," +
Longitude + ",midred|" + webMarkers + "&path=rgba:0x0000FF80,weight:5|" + webPath + "&zoom=" + Zoom +
"&size=240x246&key=ABQIAAAA0YgPFQliKEmnRQ48fiqZKRTNNCNNEvQP7NISal3Vj_WI-
jROIRO7VSbX3Y_wEqwtcm045eV-QnNXmA";
    webBrowser1.Navigate(new Uri(webMap));
}
private void button16_Click(object sender, EventArgs e)
{
    Zoom = Zoom-1;
    webMap = "http://maps.google.com/staticmap?center=" + V + "," + H + "&markers=" + Latitude + "," +
Longitude + ",midred|" + webMarkers + "&path=rgba:0x0000FF80,weight:5|" + webPath + "&zoom=" + Zoom +
"&size=240x246&key=ABQIAAAA0YgPFQliKEmnRQ48fiqZKRTNNCNNEvQP7NISal3Vj_WI-
jROIRO7VSbX3Y_wEqwtcm045eV-QnNXmA";
    webBrowser1.Navigate(new Uri(webMap));
}
private void button11_Click(object sender, EventArgs e)
{
    if (Zoom > 13)
    {
        H = H - 0.0015;
    }
    if (Zoom < 14 && Zoom > 10)
    {

```

```

        H = H - 0.01;
    }
    if(Zoom < 11)
    {
        H = H - 0.2;
    }
    webMap = "http://maps.google.com/staticmap?center=" + V + "," + H + "&markers=" + Latitude + "," +
Longitude + ",midred|" + webMarkers + "&path=rgba:0x0000FF80,weight:5|" + webPath + "&zoom=" + Zoom +
"&size=240x246&key=ABQIAAAA0YgPFQlikEmnRQ48fiqZKRTNNCNNEvQP7NIsal3Vj_WI-
jROIRO7VSbX3Y_wEqwtcm045eV-QnNXmA";
    webBrowser1.Navigate(new Uri(webMap));
}
private void button13_Click(object sender, EventArgs e)
{
    if (Zoom > 13)
    {
        H = H + 0.0015;
    }
    if (Zoom < 14 && Zoom > 10)
    {
        H = H + 0.01;
    }
    if (Zoom < 11)
    {
        H = H + 0.2;
    }
    webMap = "http://maps.google.com/staticmap?center=" + V + "," + H + "&markers=" + Latitude + "," +
Longitude + ",midred|" + webMarkers + "&path=rgba:0x0000FF80,weight:5|" + webPath + "&zoom=" + Zoom +
"&size=240x246&key=ABQIAAAA0YgPFQlikEmnRQ48fiqZKRTNNCNNEvQP7NIsal3Vj_WI-
jROIRO7VSbX3Y_wEqwtcm045eV-QnNXmA";
    webBrowser1.Navigate(new Uri(webMap));
}
private void button14_Click(object sender, EventArgs e)
{
    if (Zoom > 13)
    {
        V = V + 0.0015;
    }
    if (Zoom < 14 && Zoom > 10)
    {
        V = V + 0.01;
    }
    if (Zoom < 11)
    {
        V = V + 0.2;
    }
}

```

```

        webMap = "http://maps.google.com/staticmap?center=" + V + "," + H + "&markers=" + Latitude + "," +
Longitude + ",midred|" + webMarkers + "&path=rgba:0x0000FF80,weight:5|" + webPath + "&zoom=" + Zoom +
"&size=240x246&key=ABQIAAAA0YgPFQliKEmnrQ48fiqZKRTNNCNNEvQP7NISal3Vj_WI-
jROIRO7VVSbX3Y_wEqwtcm045eV-QnNXmA";
        webBrowser1.Navigate(new Uri(webMap));
    }
    private void button12_Click_1(object sender, EventArgs e)
    {
        if (Zoom > 13)
        {
            V = V - 0.0015;
        }
        if (Zoom < 14 && Zoom > 10)
        {
            V = V - 0.01;
        }
        if (Zoom < 11)
        {
            V = V - 0.2;
        }
        webMap = "http://maps.google.com/staticmap?center=" + V + "," + H + "&markers=" + Latitude + "," +
Longitude + ",midred|" + webMarkers + "&path=rgba:0x0000FF80,weight:5|" + webPath + "&zoom=" + Zoom +
"&size=240x246&key=ABQIAAAA0YgPFQliKEmnrQ48fiqZKRTNNCNNEvQP7NISal3Vj_WI-
jROIRO7VVSbX3Y_wEqwtcm045eV-QnNXmA";
        webBrowser1.Navigate(new Uri(webMap));
    }
    private void button8_Click(object sender, EventArgs e)
    {
        FormMain f = new FormMain(mediaIp, mediaPort);
        f.Show();
    }
    private void button4_Click(object sender, EventArgs e)
    {
        string serverMachine = mediaIp;
        string info = "http://" + serverMachine + "/info/";
        webBrowser3.Navigate(new Uri(info));
    }
}

```