



**Marco Rodrigues
Miranda**

**Recomendação de pontos de interesse em tempo
real com base em contexto**

**Marco Rodrigues
Miranda**

**Recomendação de pontos de interesse em tempo
real com base em contexto**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas de Informação, realizada sob a orientação científica do Prof. Doutor José Manuel Matos Moreira e coorientação científica do Prof. Doutor Hélder Troca Zagalo, Professores auxiliares do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho à minha família e amigos que sempre me apoiaram ao longo do meu percurso académico.

o júri /the jury

presidente / president

Prof. Doutora Ana Maria Perfeito Tomé
Professora Associada da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor João Pedro Mendes Moreira
Professor Auxiliar, da Faculdade de Engenharia da Universidade do Porto
(Arguente Principal)

Prof. Doutor José Manuel Matos Moreira
Professor Auxiliar da Universidade de Aveiro (Orientador)

Agradecimentos / acknowledgements

Sirvo-me deste espaço para agradecer a todos aqueles, que ao longo do meu percurso académico, me ajudaram e motivaram para que os meus objetivos se concretizassem.

Como não poderia deixar de ser gostaria em primeiro lugar de agradecer ao meu orientador, o Prof. Doutor José Manuel Matos Moreira e ao meu coorientador Prof. Doutor Hélder Troca Zagalo por se disponibilizarem sempre com o maior empenho e dedicação na orientação do trabalho desenvolvido, bem como agradecer a enorme paciência demonstrada ao longo de todo este processo.

Agradeço também de igual modo a todos os membros da empresa Ubiwhere, em especial ao Eng. Rui Costa que me recebeu de braços abertos e possibilitou a realização deste trabalho num contexto empresarial, bem como agradecer às pessoas que trabalharam mais próximas de mim neste mesmo contexto, o Eng. Edgar Antunes e o Eng. Daniel Sousa, que me foram ajudaram e orientaram ao longo de todo o processo.

Gostaria também de agradecer à minha família pelo apoio demonstrado, não só durante o tempo de elaboração deste trabalho mas sim ao longo de todo o meu percurso académico, pois mostraram-se incansáveis a todos os níveis, nunca me faltando uma estrutura sólida para que realizasse este mesmo percurso.

Aos meus amigos, os que já trazia antes de iniciar este percurso e os que fui conhecendo neste meio, um muito obrigado por todos os bons momentos, conselhos e apoio que me foram dando sempre que assim foi necessário.

palavras-chave

Sistemas de recomendação, contexto, turismo.

resumo

A tecnologia nos dias de hoje assume um papel predominante na vida das pessoas. Um exemplo dominante disto é a *internet*, um canal de comunicação acessível ao utilizador comum que permite o acesso a uma vasta quantidade de informação. Dada a vastidão da informação acessível aos utilizadores através do uso destes canais de comunicação, torna-se urgente que esses utilizadores consigam extrair de forma rápida e personalizada a informação que pretendem. Para tal existem sistemas de recomendação que permitem combinar várias técnicas computacionais de forma a selecionar *itens* personalizados de acordo com as características de cada utilizador. Estas características dizem respeito ao perfil, padrões de comportamento, similaridade ou situação contextual dos utilizadores.

O objetivo deste trabalho passa pelo desenvolvimento de um sistema de recomendações com base em contexto inserido no domínio do turismo. Pretende-se que um utilizador possa através deste sistema obter recomendações de pontos de interesse turístico elaboradas a partir da consideração de várias condições contextuais que irão ser avaliadas neste trabalho.

Pretende-se com a implementação deste sistema realizar um estudo exploratório sobre a relevância do contexto associado às recomendações de forma a perceber como é que este poderá influenciar o resultado de uma recomendação. Irão ser analisadas diversas abordagens tradicionais utilizadas na implementação de sistemas de recomendação e como é que a componente contextual poderá ser adicionada de forma a trazer melhorias substanciais na performance deste tipo de sistema. Irão também ser realizados testes sobre o sistema desenvolvido focados na componente contextual.

Estes sistemas permitiram realizar uma análise sobre o contexto considerado a partir da qual será possível identificar que condições contextuais terão uma maior relevância na realização de recomendações, e a partir dela afinar o modelo implementado, através da atribuição de pesos às diferentes condições presentes na componente contextual.

keywords

Recommender systems, context, tourism.

Abstract

The technology nowadays assumes an important role in people's lives. A good example is the internet, a communication channel accessible to ordinary users that allows them the access to vast amounts of information. Given the huge amount of information available to the users it is important that they can extract the information they want in a fast and personalized way. For this purpose there are recommender systems that allow combining several computational techniques in order to select custom items according to characteristics based on each user. These characteristics are related to the profiles, behavior pattern, similarity or contextual situation of the users.

This work aims at the development of a contextual recommender system in touristic domain, so that a user can obtain recommendation of points of tourist interest made from the consideration of several contextual conditions that will be evaluated in this work.

It is also intended with the implementation of this system to perform an exploratory study about the relevance of the context associated to the recommendations that are made, in order to understand how context can influence the result of a recommendation. There will be an analysis of several recommender systems approaches, and how the contextual component can be added in a way that allows an improvement in the performance of this type of system. Some tests are done focusing in the contextual component.

These tests will allow identifying which contextual conditions have more relevance in the results of a recommendation, and tuning the weights assigned to the conditions considered in the system contextual component.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Glossário.....	v
Capítulo 1	1
Introdução.....	1
1.1. Motivação e Contexto	1
1.2. Objetivos	2
1.3. Estrutura do Documento.....	3
Capítulo 2	5
Sistemas de recomendação orientados ao contexto	5
2.1. Contexto e orientação ao contexto	5
2.1.1. Introdução	5
2.1.2. Definição de Contexto	5
2.1.3. Categorias associadas a Contexto	6
2.1.4. Sistemas orientados ao contexto	7
2.2. Sistemas de recomendação.....	9
2.2.1. Recomendações orientadas a contexto	9
2.2.2. Recomendações em tempo real.....	11
2.3. Projetos relacionados.....	11
2.3.1. COMPASS.....	12
2.3.2. Cyberguide.....	13
2.3.3. GUIDE.....	14
2.3.4. INTRIGUE.....	14
2.3.5. UbiquiTo.....	15
2.3.6. Gimbal SDK	16
Capítulo 3	19
Requisitos e Arquitetura do Sistema	19
3.1. Motivação.....	19
3.2. Requisitos do Sistema	20
3.2.1. Estrutura do Sistema	20
3.2.2. Módulo do servidor.....	21
3.2.2.1. Funções do servidor	21
3.2.3. Módulo do cliente	22
3.2.3.1. Funções do cliente.....	22
3.3. Características dos utilizadores	23
3.4. Recomendações	23
3.5. Atores	23
3.6. Requisitos específicos	24
3.6.1. Modelos de Casos de Utilização	24
3.6.1.1. Diagrama do módulo do servidor.....	24
3.6.1.2. Diagrama do módulo do cliente	28
3.7. Requisitos não funcionais.....	30
3.8. Arquitetura do sistema.....	30
3.8.1. Cliente web de testes.....	31
3.8.2. Módulo Recomendações Pontos de Interesse com base em contexto	32
3.8.3. Base de dados de testes	35
3.8.4. API's Externas	36
Capítulo 4	37
Implementação do Sistema.....	37
4.1. Servidor.....	37
4.1.1. Modelo de dados	37
4.1.2. Modelo Contextual.....	39
4.1.3. Obtenção de dados	41

4.1.4.	Atualização de dados	42
4.1.5.	Desenvolvimento	43
4.1.6.	Comunicação entre o servidor e o cliente	44
4.2.	Cliente	45
	Capítulo 5	49
	Discussão dos Resultados.....	49
5.1.	Metodologia	49
5.1.1.	Cenários de teste	50
5.1.2.	Resultado dos testes	52
5.2.	Resultados	53
	Capítulo 6	59
	Conclusões e trabalho futuro	59
6.1.	Conclusão.....	59
6.2.	Trabalho Futuro.....	60
	Referências	63

Lista de Figuras

Figura 2.1: <i>Screenshots</i> da aplicação COMPASS	12
Figura 2.2: Interface visual Cyberguide e unidade GPS com aplicação <i>Cyberguide</i>	13
Figura 2.3: Recomendação de atrações para desktop e para dispositivo móvel	15
Figura 2.4: UbiquiTo user interfaces: PC (esquerda) e PDA (direita).....	16
Figura 2.5: Ilustração do conceito <i>Geofence</i>	17
Figura 2.6: Ilustração do conceito <i>Interest Sensing</i>	17
Figura 2.7 - Ilustração do conceito <i>Image Recognition</i>	18
Figura 3.1: Diagrama de casos de uso do servidor	24
Figura 3.2: Diagrama de casos de uso do cliente.....	28
Figura 3.3: Arquitetura do Sistema.....	30
Figura 3.4: Exemplo prático do conceito de dominância	34
Figura 4.1: Modelo de dados	38
Figura 4.2: Valores possíveis para cada condição de contexto.....	41
Figura 4.3: Organização dos ficheiros no servidor	43
Figura 4.4: Formulário web de requisição de recomendação com base em contexto	46
Figura 4.5: Resposta a pedido de recomendação em formato JSON	47
Figura 5.1: Gráfico relativo à média de erro calculada para testes com uma condição de contexto.....	53
Figura 5.2: Gráfico relativo à média de erro calculada para testes com duas condições de contexto	54
Figura 5.3: Gráfico relativo à média de erro calculada para testes com três condições de contexto	55
Figura 5.4: Gráfico relativo à média de erro calculada para testes com todas as condições de contexto excepto uma	56
Figura 5.5: Gráfico relativo à média de erro calculada para testes com todas as condições de contexto e sem nenhuma condição de contexto	56

Glossário

GPS – O *Global Positioning System* é um sistema de navegação por satélite que permite obter informação relativamente à localização e tempo. É um sistema que é gerido pelo governo dos Estados Unidos e que é acedido livremente por qualquer pessoa que possua um recetor com GPS.

PostgreSQL – O PostgreSQL é um Sistema de Gestão de Base Dados . É um sistema *open-source*, que teve início em 1985, estando atualmente na versão 9.X. Permite diversas opções aos seus utilizadores, tais como consultas complexas, uso de chaves estrangeiras, *triggers*, integridade nas transições, suporte a ambientes concorrentes, estrutura para suportar dados georreferenciados, entre outros.

REST – *Representational State Transfer* é um estilo de arquitetura usado para a construção e desenvolvimento de *web services*. É uma palavra que surgiu no ano 2000, sendo atualmente o modelo predominante no que diz respeito ao desenho e implementação de *web services*.

SOAP – O Simple Object Access Protocol é um protocolo para a implementação de *web services*, mais propriamente para a troca de mensagens entre sistemas distribuídos, baseando-se por exemplo no protocolo Remote Procedure Call, e as suas mensagens trocadas obedecem ao formato XML.

Web Service – Método que permite a comunicação entre dispositivos eletrónicos através da *web*. Essa comunicação é realizada através de um qualquer endereço da rede com uma função bem definida, permitindo assim interoperabilidade entre sistemas distintos.

Framework – é uma abstração que une códigos comuns entre vários projetos de *software* oferecendo uma funcionalidade genérica. Um *framework* pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Ao contrário das bibliotecas, é o *framework* quem dita o fluxo de controlo da aplicação.

API – *Application Programming Interface* (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um *software* para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.

Capítulo 1

Introdução

1.1. Motivação e Contexto

Com a tecnologia a ter um lugar cada vez mais predominante no dia-a-dia das pessoas, um dos fatores que daí advém é a quantidade de informação que existe em torno das mesmas. Esta informação encontra-se disponível através de canais de comunicação, sendo o canal predominante a *internet*, que permite ao utilizador ter acesso a uma vasta quantidade de informação. No entanto, dada esta enorme quantidade de informação e o facto de esta necessitar da capacidade de análise e de decisão da pessoa que pretende ter acesso à mesma, poderá tornar a tarefa bastante difícil no que toca ao objetivo do mesmo em recolher a informação que deseja [1].

No intuito de ajudar o utilizador que recorre à *internet* a extrair de forma mais rápida e personalizada a informação que pretende, existem sistemas que permitem realizar recomendações [2], com base nos seus perfis, interesses, padrão de comportamento, semelhança entre utilizadores e situação contextual, como por exemplo o local em que se encontra, a situação temporal e especificação do tipo de informação que pretende adquirir. Este tipo de recomendações permite ajudar os utilizadores no que toca à filtragem de informação com base no que pretendem dentro de um contexto definido por este, tornando o acesso à informação existente, sobretudo na *web*, mais personalizado.

Pegando nesta temática dos sistemas de recomendação, pretende-se perceber até que ponto a situação contextual dos utilizadores poderá influenciar numa recomendação requisitada por estes, se estas contribuem para a obtenção de melhores resultados por parte dos utilizadores.

De forma a poder analisar tal vertente, da qual os sistemas de recomendação poderão estar talhados, foi considerada uma temática onde este estudo se adequaria: **recomendação de pontos de interesse (turístico) em tempo real com base em contexto.**

Um dos fatores determinantes para esta opção, foi o contexto empresarial no qual o desenvolvimento deste trabalho se insere, pois ficou definido que o estudo a realizar teria como base um projeto já existente nos quadros da empresa *Ubiwhere*, o *Pervasive Tourism*, uma API ligada à área do turismo, que permite ter acesso a diversas informações como por exemplo pontos de interesse turísticos e informações relacionadas aos mesmos, e cujo seu foco é a utilização das suas funcionalidades no desenvolvimento de aplicações móveis ligadas à área do turismo, nos moldes de uma arquitetura cliente/servidor.

Este trabalho visa estudar as diversas possibilidades de dotar esse mesmo projeto da capacidade de realizar recomendações de pontos de interesse turísticos com base em

contexto, e desenvolver um cenário que permita obter essas mesmas recomendações e testar relevância das mesmas.

A ideia passa por considerar a situação contextual dos utilizadores ao longo do tempo, como por exemplo o local em que se encontra, a situação temporal, condições meteorológicas e especificação do tipo de informação que pretende adquirir, e através da convergência de todas estas informações de contexto realizar recomendações mais criteriosas e de encontro às necessidades dos utilizadores no exato momento em que estes as requisitam.

1.2. Objetivos

Pretende-se com este projeto desenvolver um módulo que permita a realização de recomendações com base em contexto, dentro da temática do turismo. Estas recomendações irão incidir na recomendação de pontos de interesse com base na localização do utilizador, os seus interesses e vários parâmetros de contexto.

Tendo como a API *Pervasive Tourism*, e dado o facto de esta ainda não se encontrar dotada de dados relativos a pontos de interesse e histórico de *check-ins*¹ de utilizadores nesses mesmos pontos, é necessário criar um conjunto de dados (reais) de acordo com as necessidades do trabalho a desenvolver. Para tal será necessário recolher dados relativos a pontos de interesse e *check-ins* de utilizadores nesses mesmos pontos a partir de fontes de dados externas, e armazená-los numa base de dados.

O contexto a introduzir nas recomendações será definido por vários parâmetros sendo que estes estarão associados ao estado do tempo, altura do dia, dia da semana, estação do ano, geolocalização, informação relativa aos pontos de interesse como, por exemplo, a categoria do mesmo. Para a obtenção de informação, para alguns dos valores relativos aos parâmetros de contexto considerados, será necessário recorrer a fontes de dados externas. Esta informação estará diretamente associada aos *check-ins* dos utilizadores nos diferentes pontos de interesse.

O módulo desenvolvido deverá estar de acordo com as tecnologias utilizadas na API *Pervasive Tourism* já desenvolvida, bem como de acordo com o modelo relacional sobre o qual esta API opera, para que este seja facilmente integrado no mesmo.

Para que se possa testar e validar as capacidades deste módulo pretende-se também desenvolver uma aplicação de testes que permita testar as capacidades de recomendação e a influência das variáveis de contexto consideradas nessas mesmas recomendações.

¹ Momento em que o utilizador ao nível aplicacional dá entrada num determinado ponto de interesse.

1.3. Estrutura do Documento

Este documento encontra-se dividido em 6 capítulos:

- Este primeiro capítulo, que faz uma introdução e contextualização do trabalho e dos objetivos que o mesmo visa atingir;
- O capítulo 2, que aborda o estado da arte tendo como temas os conceitos teóricos predominantes em torno deste trabalho como, contexto e orientação ao contexto, sistemas associados a contexto e sistemas de recomendação. Este capítulo focar-se-á também sobre uma análise sobre algumas aplicações que recorrem a estes conceitos;
- O capítulo 3, apresenta os requisitos do sistema, a arquitetura e os respetivos componentes;
- O Capítulo 4 apresenta e explica o processo de implementação do sistema, sendo abordados a implementação do servidor e a implementação da aplicação de testes que permitirá validar o modelo considerado;
- O capítulo 5, apresenta os resultados obtidos relativos aos testes realizados sobre o sistema implementado, de forma a concluir a validade do modelo implementado através da análise da relevância do contexto associado às recomendações efetuadas;
- Por fim o capítulo 6, serve como conclusão e revisão de todo o trabalho desenvolvido, e onde se traçam possíveis metas para trabalho futuro de forma a enriquecer o mesmo.

Capítulo 2

Sistemas de recomendação orientados ao contexto

2.1. Contexto e orientação ao contexto

2.1.1. Introdução

Na computação interativa tradicional, os utilizadores têm mecanismos algo inadequados para fornecer *inputs* aos computadores, logo consequentemente estes não são capazes para tirar total vantagem do contexto existente no diálogo humano-computador [1].

Através do aperfeiçoamento do acesso dos computadores a informação de contexto, o enriquecimento das comunicações nas interações humano-computador aumentaram e tornaram possível a produção de serviços computacionais mais úteis aos utilizadores [1].

A necessidade de os utilizadores exprimirem explicitamente toda a informação relevante para uma dada situação não é de todo o objetivo da computação ligada à orientação ao contexto, pois forçar os utilizadores de forma consciente a aumentar a quantidade de informação da qual tem que estar cientes, torna-se para estes mais difícil e entediante. O objetivo deverá passar por tornar a interação com os computadores mais fácil. Poderá também acontecer em algumas situações que os utilizadores não saibam qual o tipo de informação que poderá ser potencialmente relevante, e por conseguinte, qual o tipo de informação a que devem dar prioridade.

Para combater as situações onde os utilizadores não sabem qual o tipo de informação que poderá ser potencialmente relevante, a introdução de orientação ao contexto pode ajudar. Uma abordagem baseada no mesmo foca-se na recolha contextual da informação recorrendo a meios automatizados e tornando-a disponível em tempo real, permitindo assim a decisão sobre qual a informação relevante e como lidar com a mesma.

2.1.2. Definição de Contexto

A definição de contexto é complexa e pode ter várias definições. Para desenvolver uma definição específica que possa ser usada na perspetiva da área de computação ligada à sensibilidade ao contexto, é necessário fazer uma análise de como diversos investigadores definiram contexto nos seus trabalhos.

Existem definições de contexto que associam o conceito a localização, identidade de pessoas em torno do utilizador, o dia da semana, a estação do ano, temperatura, etc. [3, 4]. Outras definem contexto como a localização do utilizador, identidade e hora [5], e ainda

outras que enumeram contexto como o estado emocional do utilizador, nível de atenção, localização e orientação, data e hora, objetos e pessoas no ambiente em torno do utilizador [6].

Estes tipos de definições de contexto através de exemplo tornam-se bastante difíceis de aplicar. Quando é necessário prever se um determinado tipo de informação não listado numa definição em específico se trata de contexto ou não, não se torna claro se é possível ou não usar a definição para resolver o problema.

Outras definições simplesmente providenciam sinónimos para contexto, como por exemplo referindo-se a contexto como o ambiente ou uma situação específica associada ao utilizador [7, 8]. Algumas consideram que o contexto é o ambiente em torno do utilizador, enquanto outras consideram-no a aplicação que esse meio ambiente poderá ter, mas definições que simplesmente recorrem a sinónimos para explicar o conceito de contexto são muito difíceis de aplicar na prática.

Existem algumas definições que são mais próximas na sua essência de uma definição operacional. Estas definições defendem que os aspetos importantes associados ao contexto são: onde estamos, com quem estamos, e que recursos se encontram perto. Estes aspetos definem o contexto como algo que está constantemente a mudar a forma de interagir com o ambiente que rodeia o utilizador [6, 9].

Existem também definições que atestam que contexto está diretamente ligado ao estado físico, social, emocional ou informativo do utilizador, e que a função do contexto é ser um subconjunto de estados físicos e conceptuais que interessam a uma entidade em particular. Estas definições são muito específicas. Contexto tem tudo a ver com toda a situação relevante para uma aplicação e o seu conjunto de utilizadores. Não podem ser enumerados quais os aspetos de todas as situações que são importantes, pois estes irão mudar de situação para situação. Por esta razão estas definições poderão não ser as mais apropriadas [6].

Segundo o mesmo trabalho uma melhor definição para contexto tendo como base as definições acima apresentadas seria: contexto é informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade pode ser uma pessoa, um lugar ou um objeto considerado relevante na interação entre um utilizador e uma aplicação, incluindo o utilizador e a aplicação em si.

Definindo contexto desta forma ajudará no desenvolvimento de uma aplicação, pois existirá uma maior facilidade na enumeração do contexto para um dado cenário aplicacional. Se uma determinada informação pode ser usada para caracterizar uma situação de um participante numa interação, então essa informação será contexto.

2.1.3. Categorias associadas a Contexto

A categorização dos tipos de contexto poderá ajudar no desenvolvimento de uma aplicação, pois esta permitirá estratificar quais os diferentes elementos de contexto que serão úteis. Duas das definições anteriores relativas a contexto defendem o

desenvolvimento de tipos de contexto. Uma das definições sugere tipos de contexto como localização, meio envolvente, identidade e hora [5]. Outra vinca a importância de aspetos de contexto como a localização do utilizador, com quem este se encontra e que recursos se encontram por perto [9].

As aplicações associadas a contexto visam sempre responder às seguintes questões em relação às entidades existentes: “quem?”, “onde?”, “quando?” e “o quê?”. Com a informação proveniente das respostas a estas questões as aplicações tentam determinar o porquê de uma determinada situação ocorrer. Na realidade uma aplicação não determina o porquê de uma determinada situação ocorrer, mas sim o responsável pelo desenvolvimento dessa mesma aplicação [10].

Segundo o mesmo trabalho, existem certos tipos de contexto que são na prática considerados mais importantes do que outros, como é o caso da localização, identidade, atividade e a hora. Estes tipos podem ser considerados primários no que toca a caracterização de uma determinada situação relativa a uma entidade, pois esta para além de responder às questões “quem?”, “onde?”, “quando?” e “o quê?”, também atua como índices para outras fontes de informação contextual. Por exemplo com a localização de uma determinada entidade, é possível determinar que outros objetos ou pessoas estão perto dessa entidade e que tipo de atividade está a ocorrer perto da mesma. Logo por aqui é evidente que elementos de contexto primários para uma entidade podem ser utilizados como índices para encontrar elementos de contexto secundário (por exemplo um endereço de email) para a mesma entidade, bem como elementos de contexto primário para outras entidades relacionadas (por exemplo outras pessoas na mesma localização).

Neste tipo de categorização, existe um sistema simples de dois níveis. Os quatro elementos primários de contexto já identificados encontram-se no primeiro nível (“quem?”, “onde?”, “quando?” e “o quê?”). Todos os outros tipos de contexto encontram-se no segundo nível. Os elementos de contexto secundários partilham uma característica comum, estes podem ser indexados por elementos primários de contexto pois estes são atributos de entidades associadas a contexto primário. Em algumas situações é necessário recorrer a múltiplos elementos primários de contexto para tornar possível a indexação de uma determinada informação. Um exemplo disso poderia ser a informação meteorológica, sendo que esta é contexto, por exemplo, num cenário de um agendamento de visitas turísticas em zonas ao ar livre para os seus utilizadores. Para obter as condições meteorológicas, tanto a localização onde se pretende que a previsão seja feita como a data são necessárias. Esta caracterização ajuda na medida em que permite a seletividade do contexto a usar nas aplicações a desenvolver, a organização do mesmo, e na pesquisa de mais contexto relevante. Os considerados elementos de contexto primários indicam os tipos de informação necessária para a caracterização de uma determinada situação e o seu uso como índices permite o uso e organização do contexto.

2.1.4. Sistemas orientados ao contexto

Os sistemas orientados ao contexto foram primeiramente pensados para ser *software* que se adapta de acordo com a sua localização de utilização, e tendo em consideração o conjunto de pessoas e objetos próximos, bem como as alterações registadas nesses objetos

ao longo do tempo [9]. Desde essa altura que têm surgido diversas tentativas no intuito de definir o que é a computação orientada ao contexto.

A primeira definição dada às aplicações orientadas ao contexto restringia a definição, pois existem aplicações que simplesmente informavam sobre contexto e aplicações que se adaptavam ao contexto. De certo modo poderá dizer-se que a orientação ao contexto cresceu a par com outros termos tais como, adaptativo, reativo, responsivo, situado, sensível a contexto e direcionado para o meio envolvente. As definições anteriores de computação orientada ao contexto acabam por convergir apenas em duas categorias: o uso de contexto e a adaptação a esse mesmo contexto.

Um caso mais geral do uso de contexto em sistemas orientados a contexto é a capacidade que certos aparelhos computacionais possuem para detetar, interpretar e responder a aspetos relacionados com o meio envolvente do utilizador e os próprios aparelhos em si. Outra definição que define orientação a contexto é a assunção de que esta possui a capacidade de providenciar a maior flexibilidade a um serviço computacional baseado na deteção em tempo real de contexto [9].

Muitos investigadores definem as aplicações orientadas a contexto como aplicações que de forma dinâmica se modificam ou adaptam o seu comportamento baseado no contexto da aplicação e do utilizador, mais especificamente o comportamento que permite que estas aplicações monitorizem *inputs* a partir de sensores no meio envolvente, e permitir aos utilizadores escolher o alcance de contexto lógico e físico, de acordo com os seus interesses ou atividades. Esta definição é um pouco mais restritiva, pois identifica o método através do qual a aplicação atua após ser considerado o contexto [11].

As definições que se enquadram mais na categoria de “adaptação a contexto” requerem que o comportamento de uma aplicação tenha de ser modificado para que esta possa ser considerada orientada a contexto. A aplicação destas definições para estabelecer uma aplicação orientada a contexto gera então a conclusão de que não se adequam. Algumas definições mais gerais que mais se adequariam seriam: Um sistema é orientado a contexto se este usar contexto para providenciar informação relevante ou serviços para o utilizador, onde a relevância depende das tarefas do utilizador.

Uma aplicação que simplesmente mostra o contexto do meio envolvente do utilizador ao mesmo, não modifica o seu comportamento, mas é orientada a contexto. Se as definições menos gerais forem usadas, uma aplicação não seria classificada de orientada a contexto. O único requisito é a resposta ao contexto, permitindo a deteção e interpretação a efetuar por outras entidades computacionais. Este requisito difere de outras definições gerais acima mencionadas, através do foco no utilizador, não limitando a orientação ao contexto apenas à interface aplicacional, não sendo necessário às aplicações realizarem serviços de forma automática, e não requerendo a aquisição de contexto em tempo real.

2.2. Sistemas de recomendação

Recomendações não são mais do que sugestões personalizadas (*itens*), gerados por um sistema de recomendação que tem como base informação relacionada com as preferências dos utilizadores associados a esse sistema (comunidade) [12]. A tarefa principal de um sistema de recomendação é tentar prever a avaliação subjetiva que um utilizador dará a um determinado *item* [13]. Os sistemas de recomendação podem dividir-se nas seguintes categorias [14]:

- **Filtragem colaborativa** – Os utilizadores podem dar feedback sobre a sua opinião em relação a um *item* recomendado. Este tipo de sistemas utiliza a relação utilizador-*item*-rating para que se torne possível prever que tipo de recomendações realizar a utilizadores que sejam similares em termos de preferências e comportamento [15].

- **Baseados em conteúdo (do inglês: *content-based*)** – Esta categoria também se baseia no *feedback* dos utilizadores, no entanto quando um utilizador solicita uma recomendação este tipo de sistemas apenas se foca nas preferências e comportamentos do utilizador que a requisita e não na dos outros utilizadores pertencentes a essa comunidade na qual o sistema se encontra inserido. Este tipo de sistemas considera apenas os *itens* que este utilizador gostou no passado e compara a sua descrição com *itens* que se possam adequar a uma possível recomendação para esse utilizador. Se for encontrada similaridade entre os *itens* avaliados positivamente no passado pelo utilizador e os que poderão ser recomendados, estes poderão surgir na recomendação [16].

- **Baseados em conhecimento (do inglês: *Knowledge-based*)** – Um problema que surge com os sistemas que se inserem nas categorias de filtragem colaborativa e baseados em conteúdo trata-se do facto de estes não conseguirem realizar boas recomendações enquanto não existir um número grande de utilizadores na comunidade sobre a qual o sistema atua, o chamado “*cold start*” [17] ou “*ramp-up*” [18]. Os sistemas de recomendação baseados em conhecimento evitam este problema, pois não dependem na avaliação dos utilizadores, mas sim numa estrutura de conhecimento que permite inferir sobre as necessidades dos utilizadores e as suas preferências. Uma técnica importante relacionada com esta categoria de sistemas de recomendação designa-se por *case-based reasoning* [19], onde todas as sessões de recomendações geradas são armazenadas dando origem a um conjunto de informação base. Novas recomendações são geradas, a partir de sessões de recomendações similares já existentes nesse conjunto base [13].

- **Híbridos** – Sistemas que combinam duas ou mais das categorias acima mencionadas de forma a obter uma melhor performance, pois conseguem combater as limitações das outras abordagens quando utilizadas de forma única [13, 14].

2.2.1. Recomendações orientadas a contexto

Uma recomendação orientada ao contexto terá como base uma ou mais categoria(s) das mencionadas no ponto anterior, podendo dizer-se assim que este tipo de sistema de recomendação será uma extensão de uma ou mais dessas categorias. Por exemplo, para um

sistema de recomendação baseado numa abordagem de filtragem colaborativa temos a função F que estima uma avaliação para os pares de valores (*Utilizador*, *Item*) que não foram considerados anteriormente [20].

$$F: Utilizador \times Item \rightarrow Avaliação$$

Recorrendo a esta função é possível devolver os n *itens* com maior avaliação para um determinado utilizador. Pode-se assim afirmar que estes n *itens* são aqueles que o sistema de recomendação em causa considera serem os de maior valor para esse utilizador. Estes tipos de sistemas de recomendação são os designados por tradicionais ou a duas dimensões [20].

Por vezes esta abordagem mais tradicional sobre os sistemas de recomendação não é suficiente ou apropriada para a realização de recomendações. Os interesses dos utilizadores mostram ser um elemento estável quando utilizado na realização de recomendações inserido no domínio do turismo, no entanto a decisão por parte dos utilizadores na escolha de um determinado ponto de interesse para visitar, poderá variar tendo em conta múltiplos fatores (variáveis), ou seja contexto [17, 21].

Vamos considerar por exemplo a recomendação de cafés em Portugal. Um café perto ou na praia em pleno Inverno tratar-se-ia, por norma, de uma recomendação fraca, pois dadas as condições meteorológicas registadas habitualmente nesta estação (chuva, frio e vento) as pessoas tentam evitar este tipo de destinos. Em contraste com um sistema de recomendação baseado numa abordagem tradicional de filtragem colaborativa, um sistema de recomendações orientado a contexto é dotado da capacidade de detetar uma situação idêntica à descrita e não lhe atribuir muita ou nenhuma relevância na realização de uma dada recomendação. Isto deve-se ao facto de um sistema de recomendação assente numa abordagem tradicional de filtragem colaborativa não considera o contexto no seu algoritmo.

Um sistema de recomendação orientado a contexto tendo como base a filtragem colaborativa na realização de recomendações, pretende conjugar essa mesma abordagem com a introdução de contexto, tentando perceber a influência em termos analíticos do contexto na realização de recomendações. Assim sendo a função F descrita anteriormente terá de ser estendida através da inclusão de contexto:

$$F: Utilizador \times Item \times Contexto \rightarrow Avaliação$$

Utilizador e *Item* são os domínios dos utilizadores e dos *itens* respetivamente. A avaliação é domínio das avaliações, e o contexto especifica a informação contextual associada à aplicação [20].

Pela função F acima descrita é possível constatar que o contexto tem influência na avaliação de um determinado *item*, podendo assim ser dividido em vários componentes. Separando a função F podemos considerar que uma subavaliação diz respeito a uma média relativa à avaliação de um determinado *item* i , enquanto as restantes subavaliações permitem extrair a informação relativa à relevância contextual de cada uma das condições

de contexto consideradas na avaliação de um determinado *item* numa dada recomendação. Em termos teóricos teremos algo como:

$$Rating = \bar{i} + \sum_{x \in C} contexto_x + \sum_{x \in C} contexto_{xi}$$

O \bar{i} consiste na predição da avaliação e é obtido computacionalmente como a avaliação média para o *item* i na base de dados. O $contexto_x$ consiste na influência global que uma determinada condição de contexto com índice x tem, sendo que o peso desta condição é igual independentemente do *item* em causa. Quanto ao $contexto_{xi}$ não é mais do que a influência de uma condição contextual x mas tendo em consideração a avaliação feita sobre um determinado *item* i . É preciso ter também em conta que fazem parte do somatório apenas as condições contextuais consideradas ativas por parte do utilizador na aplicação aquando a requisição da recomendação pretendida.

De forma a ensaiar este modelo de avaliação contextual, é necessário possuir dados reais que permitam a partir dos mesmos extrair a informação contextual necessária [22, 23]. Assim obtém-se um conjunto de dados iniciais que assenta no modelo descrito (*item*, contexto).

2.2.2. Recomendações em tempo real

Este tipo de recomendações encontra-se associado a características do contexto e a preferências do utilizador, pois este tipo de recomendações é afetado pela variável tempo. Um sistema que permite este tipo de recomendações consegue reagir e adaptar-se consoante as variações desta variável, permitindo que uma recomendação facultada por este tipo de sistema esteja atualizada. Estas atualizações permitem que numa dada recomendação possam ser adicionados, removidos ou substituídos *itens*, ou até que a sua relevância se modifique. No entanto este tipo de capacidade pode ter os seus aspetos negativos, pois se estas ocorrem de forma automática, o utilizador poderá não se aperceber das mesmas, e isso pode ser algo indesejado por parte do mesmo. Por este motivo este tipo de sistema deve, aquando estas alterações, possuir mecanismos que permitam ao utilizador identificar que existe a possibilidade de atualização de informação e que a decisão desta mesma alteração seja tomada pelo mesmo [20].

2.3. Projetos relacionados

Neste ponto irão ser abordados alguns projetos, inseridos no domínio do turismo, já existentes que de alguma forma se servem de alguns conceitos acima mencionados para desempenhar as suas funções. Irá ser feita uma breve descrição de algumas aplicações que se encontram diretamente ligadas à temática das recomendações com base em contexto. Irá ser também abordada de forma descritiva as potencialidades de uma plataforma de desenvolvimento que permite dotar aplicações móveis que a utilizem da capacidade de aliar o contexto nas suas funcionalidades.

2.3.1. COMPASS

A aplicação COMPASS tem esta designação pois trata-se de um acrónimo para Context-aware Mobile Personal ASSistant [24]. Trata-se de uma aplicação desenvolvida em 2004 cujas funcionalidades se encontram inseridas no domínio do turismo, e permitem que turistas tenham acesso a informação e serviços (desde pontos de interesse a amigos) num determinado contexto do interesse desses mesmos turistas, tendo em conta os seus objetivos num dado momento.

Por exemplo um turista que tenha interesse em história e arquitetura, e que utilize esta aplicação irá ter acesso a informação personalizada consoante os seus interesses, como monumentos. Outro exemplo seria um turista desejar encontrar um sítio para passar a noite. Neste caso a aplicação iria facultar uma lista de informações relativa a hotéis e outro tipo de locais onde a hospedagem seria possível, cujas características vão de encontro às preferências do turista.



Figura 2.1: Screenshots da aplicação COMPASS [24]

Esta aplicação permite que um utilizador possa visualizar num mapa a sua localização atual recorrendo a recetores GPS. Em função do perfil do utilizador, do seu objetivo e proximidade, é possível visualizar no mapa ou na forma de lista objetos relevantes para o utilizador, como por exemplo edifícios e pessoas relacionadas com o utilizador. Estes objetos e o próprio mapa apresentado ao utilizador estão em constante mudança à medida que o perfil do utilizador, posição e objetivos variam. A figura 2.1 exemplifica a interface da aplicação.

Estas funcionalidades proporcionadas pela aplicação são possíveis através da pesquisa de serviços que se encontram ligados à entrega de objetos relacionados com o contexto associado ao utilizador. Em termos gerais a plataforma que opera sob a aplicação serve-se de serviços que vão de encontro aos critérios estipulados pelo contexto associado ao utilizador e o seu objetivo.

2.3.2. Cyberguide

O projeto Cyberguide [25] foi desenvolvido em 1996 no *Graphics, Visualization and Usability Center Laboratory, Georgia Institute of Technology*, nos Estados Unidos da América, e era composto por uma série de protótipos de guias turísticos acessíveis a partir de dispositivos móveis. Inicialmente o desenvolvimento deste projeto focou-se mais numa parte muito específica do contexto que pode estar associado a um utilizador dos protótipos desenvolvidos, a localização e orientação do mesmo. Os protótipos desenvolvidos foram pensados para terem o conhecimento da localização dos utilizadores bem como registar um histórico dessas mesmas localizações. Estes protótipos estão também preparados para serem multiplataforma em diversos dispositivos móveis bem como para funcionarem tanto num registo *indoor* como *outdoor*.

Sendo o domínio deste projeto o turismo, os primeiros protótipos desenvolvidos visavam servir como assistente para um tipo de turista muito específico – utilizadores do protótipo que pretendem realizar uma visita no “GVU Center Lab”², nos dias abertos ao público. Por norma este tipo de visitante tem acesso a um mapa (em papel) dos vários laboratórios existentes no centro de investigação e a um conjunto de informações (também em papel) que descrevem os projetos que estão a ser demonstrados em vários locais. Com a introdução deste protótipo deu-se a substituição de todos estes *itens* em papel por os mesmos *itens* mas acessíveis a partir de uma aplicação instalada num dispositivo móvel.

O objetivo a longo prazo deste projeto é tornar estes protótipos numa aplicação final, que permita a um turista perceber onde se encontra, para o que está a olhar e possa também prever as respostas às questões que o que o rodeia lhe possa suscitar. Pretende-se também que permita a interação entre utilizadores da mesma aplicação. A curto prazo o objetivo foi desenvolver versões dos protótipos Cyberguide e colocá-los em funcionamento em PDA's (*personal digital assistants*) e PC's (*personal computers*), onde a orientação ao contexto apenas é introduzida através da consideração da localização física e orientação da unidade onde o protótipo se encontra a funcionar, como mostra a figura 2.2. Através do acesso a localização dos utilizadores (no caso dos dispositivos móveis) e do registo de histórico destas localizações é possível enriquecer a forma como os utilizadores interagem e comunicam com o sítio onde estão e as pessoas que visitam esse mesmo sítio.

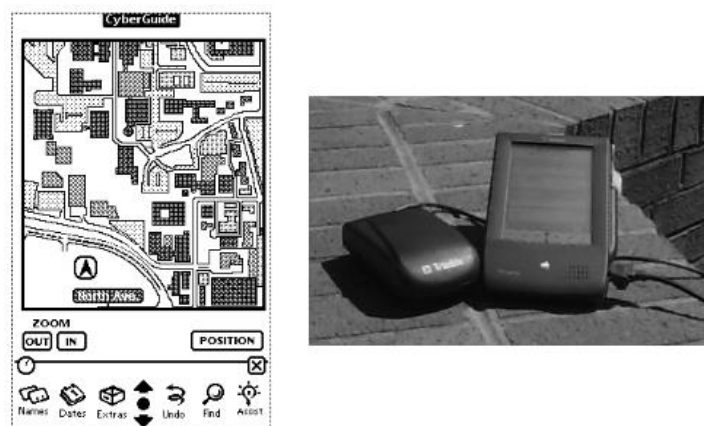


Figura 2.2: Interface visual Cyberguide e unidade GPS com aplicação *Cyberguide* [25]

² Graphics, Visualization and Usability Center Laboratory, Georgia Institute of Technology, Atlanta, EUA

2.3.3. GUIDE

O projeto GUIDE [26] desenvolvido em 2000 e consiste num guia turístico *context-aware* vocacionado para turistas de uma determinada cidade. Os visitantes de uma cidade podem aceder através de uma aplicação instalada num dispositivo móvel a informação *context-aware* relativa a essa cidade. Esta informação que é apresentada aos utilizadores da aplicação, é trabalhada com base nos perfis destes mesmos utilizadores e na informação contextual como por exemplo a localização física dos mesmos.

A abordagem para este projeto surge com base em outro projeto já mencionado o Cyberguide, sendo que outros aspetos e melhoramentos foram focados na implementação do mesmo.

A aplicação GUIDE permite que um utilizador tenha acesso, a partir do dispositivo móvel, a conteúdos *web* baseados no contexto atual do utilizador, sendo que também permite que um utilizador tenha acesso a conteúdos *web* sem considerar a componente contextual.

A aplicação permite também que um utilizador possa planear uma visita com base num determinado número de pontos de interesse, sendo que estes pontos são determinados pelo mesmo com base nas suas preferências. Para tal a aplicação realiza um questionário ao utilizador onde aborda questões como por exemplo as categorias relativas a pontos de interesse que mais aprecia e quais das atrações populares da cidade teria interesse em visitar. Com base nesta informação a aplicação consegue gerar um roteiro turístico relevante da cidade para aquele utilizador em específico, sendo que o utilizador tem sempre a possibilidade de rejeitar ou aceitar as propostas sugeridas no roteiro, ou até mesmo adicionar destinos extra.

Através desta aplicação é também possível aceder a serviços interativos como por exemplo realizar a reserva num hotel, ou até mesmo verificar quais os filmes que estão em exibição e os horários das sessões num cinema perto de si, e caso lhe interesse, reservar bilhetes. É dotada também da capacidade de comunicação através de mensagens entre utilizadores permitindo que utilizadores que estejam a realizar a visita em conjunto possam se distanciar sem nunca perderem o contacto entre eles.

2.3.4. INTRIGUE

INTRIGUE [27] (INteractive TouRist GUIdE) trata-se de um projeto desenvolvido em 2003 cujo objetivo passava por desenvolver um protótipo de um servidor relativo a informação turística que possibilitasse disponibilizar informação personalizada sobre variadas atrações turísticas num determinada área geográfica bem como gerar itinerários que conjugam essas mesmas atrações turísticas, usando recomendações requisitadas através de uma interface gráfica (figura 2.3). Este protótipo para a realização destas recomendações tem em consideração alguns fatores de contexto como por exemplo o tipo de turistas que pretendem visitar as atrações turísticas (por exemplo famílias com crianças

ou pessoas idosas), os horários de funcionamento das atrações turísticas e categorias preferidas dos utilizadores.

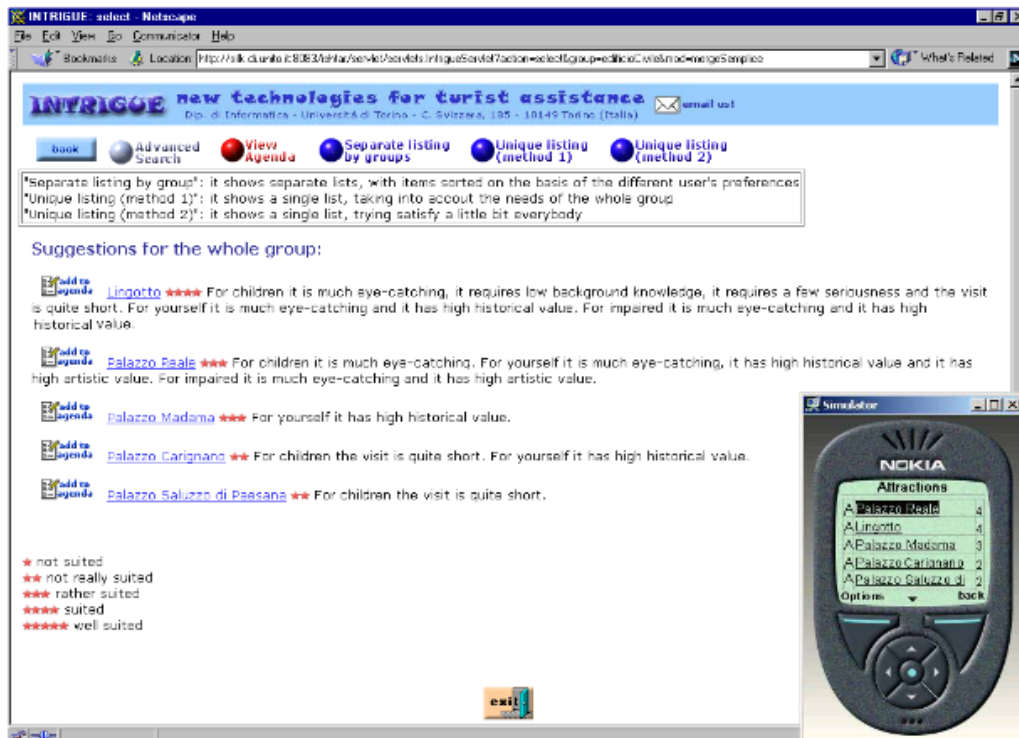


Figura 2.3: Recomendação de atrações para desktop e para dispositivo móvel [27]

2.3.5. UbiquiTo

UbiquiTo [28] um guia turístico para dispositivos móveis à semelhança de alguns já descritos. O UbiquiTo desenvolvido em 2004 trata-se de um protótipo que permite filtrar a informação e torná-la acessível ao utilizador de uma forma mais apropriada, tendo em conta diversos fatores como serviços direcionados para turistas tendo em consideração a localização destes, exploração do perfil dos utilizadores em aspetos como os seus interesses, preferências e visitas prévias em locais onde o sistema atua (neste caso a cidade de Turim em Itália), permitindo assim o acesso a sugestões personalizadas por parte do UbiquiTo.

As interações dos utilizadores com o UbiquiTo oferecem um certo grau de adaptabilidade, pois o UbiquiTo toma em consideração um conjunto de condições de contexto como a data e hora e a localização do utilizador, permitindo assim que este realize recomendações relevantes aos utilizadores. Foram projetadas versões do UbiquiTo para funcionamento em PDA's (*personal digital assistants*) e PC's (*personal computers*), mostrando a figura 2.4 um exemplo dessas mesmas *interfaces*.



Figura 2.4: UbiquiTo user interfaces: PC (esquerda) e PDA (direita) [28]

2.3.6. Gimbal SDK

O Gimbal [29] trata-se de uma plataforma *context awareness* desenvolvida para Android e iOS pela Qualcomm Labs. Esta plataforma permite que desenvolvedores possam criar aplicações mais inteligentes, tendo como base fornecedores de conteúdos, e que desta forma estas possam oferecer conteúdos relevantes, de forma personalizada e oportuna, a audiências móveis onde e quando estas mais precisam.

Esta plataforma atua em tempo real, tendo em conta parâmetros como a localização, interesses e situação contextual dos utilizadores destinados a esta plataforma.

Como principais características o Gimbal SDK foca-se nos seguintes aspetos:

- *Geofence*;
- *Interest Sensing*;
- *Image Recognition*;
- *Communicate*;
- *Manager*.

Geofence

Através da delimitação digital de fronteiras em torno de espaços físicos, é possível ao *Gimbal Geofence* criar ligações com os aparelhos móveis dos diferentes utilizadores de forma a criar um sistema orientado ao contexto geográfico simplificado e otimizado em termos de consumo de bateria, permitindo assim que as mensagens dos diferentes fornecedores de conteúdos sejam entregues aos diferentes utilizadores.

O *Gimbal Geofence* define os locais que mais importam numa determinada aplicação e interage com os clientes nessas localizações. Um utilizador ao entrar numa *Geofence* definida, torna possível escolher entre enviar conteúdo para o seu dispositivo móvel ou simplesmente usar essa informação para refinar as ofertas com base nas preferências pessoais desse utilizador. Sempre que o *Geofence* estiver ativado pelo utilizador este

funcionará em *background* mesmo quando a aplicação estiver fechada. A figura 2.5 ilustra o conceito de *Geofence*.

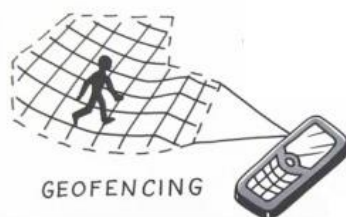


Figura 2.5: Ilustração do conceito *Geofence* [29]

Interest Sensing

Através do recurso a heurísticas bem otimizadas, uma aplicação dotada das capacidades disponibilizadas pela *Gimbal* será capaz de criar um perfil de utilizador bastante intuitivo. O uso do *Interest Sensing* permite que uma aplicação possa realizar um *matching* mais preciso das preferências específicas de um cliente (utilizador), tendo em conta o tempo, a sua localização e o interesse que este possa ter naquele momento. O *Gimbal* permitirá que esta capacidade seja de livre acesso para a aplicação permitindo assim que esta possa gerar as suas inferências com base no mesmo conjunto de dados que o *Gimbal* usa, para a aplicação realizar recomendações para os utilizadores. A figura 2.6 ilustra o conceito de *Interest Sensing*.



Figura 2.6: Ilustração do conceito *Interest Sensing* [29]

Image Recognition

Permite que um cliente possa realizar scans a imagens com o seu dispositivo móvel de forma a obter conteúdos digitais relevantes, como por exemplo ver vídeos ou até mesmo realizar uma compra no momento.

Esta característica apresenta um grau de precisão elevado permitindo o reconhecimento de imagens programadas quase de forma instantânea. Uma vez realizado o *upload* de uma

imagem para o *Gimbal Manager*, a tecnologia realiza um *scan* a todas as características e pixéis da imagem de forma a garantir um grau de precisão elevado no seu reconhecimento e avaliação. Assim, um cliente terá em seu poder um conjunto de opções aprimoradas para escolher. Desta forma será possível oferecer conteúdos adicionais relativos à imagem sobre a qual se realiza o reconhecimento, desde a compra de produtos, visualização de vídeos e partilhas nas redes sociais. A figura 2.7 ilustra o conceito de *Image Recognition*.

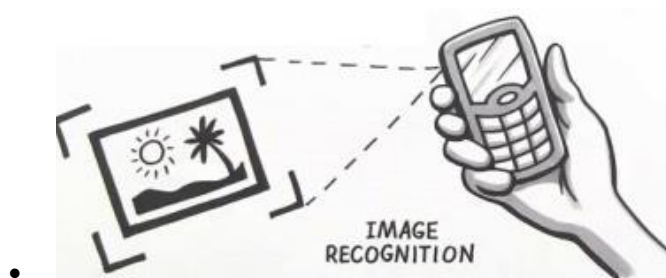


Figura 2.7 - Ilustração do conceito *Image Recognition* [29]

Communicate

Trata-se de uma solução que permite alcançar de forma personalizada os conteúdos de uma aplicação e permite um equilíbrio nas comunicações *push*. Ao fundir a localização, tempo e preferências relevantes de clientes (comunicações *push* associadas a contexto), os clientes passam de destinatários passivos a participantes ativos.

Comunicações contextuais funcionam através da fusão das funcionalidades de contexto do *Gimbal* para que seja possível entregar conteúdos para os clientes que se preocupam no preciso momento em que estes conteúdos lhes cheguem. O *Gimbal Communicate* permite a filtragem da audiência através de segmentação e interesses, permitindo que uma determinada mensagem seja entregue à audiência certa, no local certo e à hora certa.

Manager

O *Gimbal* é dotado de um *web-based manager* que permite ter, na vida real, uma visão do comportamento dos clientes. São considerados parâmetros como localização, altura do dia, histórico *web*, aplicações e o seu uso, de forma a inferir comportamentos e hábitos naturais dos clientes. Através destes parâmetros é possível perceber quais os locais visitados, a duração de uma visita a um local, o número de visitas, interações com comunicações contextuais e outros indicadores demográficos.

Dotado de uma interface intuitiva e *user-friendly*, o *Gimbal Manager* permite ir a fundo na análise de qualquer evento reportado, esteja este associado a *geofencing*, notificações contextuais ou *image recognition*. As capacidades analíticas permitem que um utilizador possa exportar todos os dados, para uma análise posterior.

Capítulo 3

Requisitos e Arquitetura do Sistema

Este capítulo tem como objetivo especificar os requisitos, que são descritos sobre a forma de casos de uso, para o desenvolvimento de um módulo que permita realizar recomendações de pontos de interesse com base em contexto, no domínio do turismo, e que este esteja pensado para uma possível integração numa API já existente designada por ‘*Pervasive Tourism*’, desenvolvida pela empresa Ubiwhere. Este módulo visa possibilitar a recomendação de pontos de interesse com base nas preferências de pesquisa dos utilizadores, como o caso do raio em torno dos mesmos, a categoria dos pontos de interesse e condições de contexto e sua relevância. De forma a avaliar o contexto associado aos pontos de interesse as recomendações irão ter em consideração o padrão de comportamento contextual dos utilizadores nas visitas aos diferentes pontos de interesse, bem como a localização geográfica dos utilizadores que requisitam recomendações. Para além do módulo mencionado deve também ser desenvolvido um (cliente) de forma a possibilitar a introdução/configuração dos parâmetros necessários para a realização de recomendações.

Para além da definição dos requisitos do sistema, este capítulo fornece também as informações necessárias (descrição do sistema, restrições, utilizadores) para o desenvolvimento do sistema e para a sua implementação.

Este capítulo está essencialmente dedicado à contextualização e apresentação das funcionalidades do módulo e cliente a desenvolver. Numa primeira fase, será apresentada uma visão Macro das funções do módulo e cliente a desenvolver. Numa segunda fase, mais detalhada, serão apresentados os requisitos do sistema assim como diagramas de casos de uso que servem de base às funcionalidades a implementar.

3.1.Motivação

Para o desenvolvimento deste módulo e tendo em consideração as várias propostas de implementação de sistemas de recomendação com base em contexto já apresentadas no capítulo 2, há que focar um aspeto essencial, o tipo e quantidade de dados que estarão disponíveis para ensaiar o modelo a desenvolver, pois só assim este poderá ser validado. De ressaltar que a natureza destes dados deverá ser real. Dado que a API ‘*Pervasive Tourism*’ não contém dados relativos a *check-ins* de utilizadores em diversos pontos de interesse, é necessário angariar um conjunto de dados que cumpram os requisitos necessários para a implementação do módulo de recomendações. Será a partir do histórico de *check-ins* dos utilizadores nos diferentes pontos de interesse que a informação de contexto será extraída para a realização de recomendações.

A angariação destes mesmos dados trata-se de um processo moroso, pois é necessário uma grande quantidade de informação de *check-ins* de forma a obter um conjunto de dados

para validação do modelo a implementar. O importante será não só obter quantidade mas também variedade no que toca a pontos de interesse e valores para as condições de contexto a considerar. A solução passa então pela utilização de um conjunto de dados já existente que possibilite trabalhar os requisitos necessários à validação do modelo proposto.

Ao recorrer a uma solução deste tipo surgem algumas limitações ao nível do modelo de recomendações a implementar, sendo que a mais evidente é o facto de o modelo ficar limitado ao tipo de informação que é possível extrair dos dados disponíveis no conjunto de dados utilizado. Um sistema de recomendação associado a contexto assente numa abordagem colaborativa tem como elemento predominante a consideração de aspetos de perfil de utilizador, e o tipo de dados associados ao perfil de utilizador que por norma são de carácter privado, ou seja, não se encontram disponíveis. Esta limitação na obtenção de dados dificulta a implementação de um modelo do tipo (utilizador, *item*, contexto) onde o elemento utilizador não terá o peso desejado no modelo por falta de dados, sendo que existirá sempre algo associado a este elemento. Assim sendo o foco do modelo terá que inevitavelmente incidir sobre os restantes elementos: o *item* (pontos de interesse) e o contexto (todas as condições de contexto que são possíveis considerar tendo em conta os dados disponíveis).

Utilizando dados como ponto de partida para a validação do modelo e consequente realização de recomendações, será válido também afirmar que a abordagem a seguir assenta também num cenário baseado em conhecimento (secção 2.2), por seguinte estamos na presença de um modelo híbrido.

Em conclusão, terá que se assumir que o modelo a implementar, contextualizando na temática, terá de incidir em aspetos como os pontos de interesse turísticos e dados relativos aos mesmos, e condições de contexto, que são possíveis de apurar, associadas aos *check-ins* dos utilizadores nesses mesmos pontos, tendo sempre em consideração que a diversidade de dados relativa aos utilizadores que realizam os *check-ins* nos pontos de interesse turísticos será mínima dada a natureza privada dos dados de utilizador.

3.2. Requisitos do Sistema

3.2.1. Estrutura do Sistema

No âmbito deste trabalho o produto divide-se em duas partes essenciais. O módulo associado às recomendações com base em contexto, a considerar numa possível integração na API '*Pervasive Tourism*' (lado servidor) e o cliente que permitirá testar as funcionalidades do módulo (lado cliente). O sistema baseia-se portanto numa arquitetura cliente-servidor.

3.2.2. Módulo do servidor

A pretensão com o desenvolvimento deste módulo passa por enriquecer a API já existente, dotando-a da capacidade de poder realizar recomendações a utilizadores de pontos de interesse com base na sua geolocalização, características desses mesmos pontos, históricos de visitas dos utilizadores e todo o contexto registado na altura dessas visitas. Para tal, o módulo desenvolvido deverá permitir o armazenamento de toda a informação necessária para a realização destas recomendações, passando assim pelo registo dos *check-ins* nos diferentes pontos de interesse e pelo registo de todo o contexto considerado na altura desses mesmo *check-ins*.

Deverá também permitir a gestão e armazenamento de informação relativa a pontos de interesse, bem como possíveis alterações no futuro. Deverá também possuir a capacidade para devolver uma lista de pontos de interesse ordenada com base num *score* calculado a partir das informações relativas aos pontos de interesse e com base na experiência por parte dos utilizadores que já os frequentaram.

Toda a informação necessária inerente aos pontos de interesse, ao contexto, registo de histórico de *check-ins*, será obtida com recurso a API's já existentes que disponibilizam serviços para o efeito, como é o caso das API's do *Foursquare* [30] e o *Open Weather Map* [31]. Quanto ao armazenamento de dados, este é feito recorrendo a um modelo de dados de testes e que tem como base o modelo já existente associado à API '*Pervasive Tourism*', desenvolvido em PostgreSQL [32].

Em termos de comunicação com o módulo desenvolvido, esta será feita recorrendo ao uso de *web services*. Desta forma o cliente a desenvolver poderá comunicar com o servidor. A comunicação poderá ser feita recorrendo a uma API Rest ou SOAP.

3.2.2.1. Funções do servidor

- Leitura de informação a partir de fontes externas ao projeto;
- Processamento e filtragem da informação obtida externamente;
- Registo de *check-ins* de utilizadores relativamente a pontos de interesse tendo em conta o contexto considerado, num repositório de dados;
- Cálculo de um *score* para um determinado conjunto de pontos de interesse, com base em informação relativa a contexto associado a pontos de interesse e a visitas de utilizadores a esses mesmos pontos, possibilitando recomendações;
- Envio de informação gerada/armazenada para um cliente;
- Comunicação de informação proveniente de um cliente com servidor.

3.2.3. Módulo do cliente

A função por parte do cliente deve permitir testar e afinar as capacidades de recomendação do módulo a implementar. Trata-se de uma aplicação *web* que irá interagir com o módulo de recomendação desenvolvido, recorrendo a *web services*. Esta aplicação permitirá através de uma interface de utilizador minimalista (dado o seu carácter de testes) receber recomendações de pontos de interesse com base nos filtros de pesquisa considerados pelo utilizador na altura do pedido da recomendação, bem como com base na sua geolocalização e condições de contexto ideais dos pontos de interesse a considerar. Estas recomendações serão feitas através da devolução e visualização de uma lista de pontos de interesse ordenada por relevância, sendo que a aplicação permitirá também a visualização da informação associada a cada um dos pontos de interesse devolvidos numa determinada recomendação.

Estando o modelo de recomendação proposto a considerar um parâmetro como a geolocalização do utilizador que requisita a recomendação, será natural levantar a questão relativamente à natureza do cliente ser a de uma aplicação *web* e não *mobile* visto que os dispositivos móveis estão dotados da capacidade de utilização de GPS, o que poderia contribuir com a disponibilização das coordenadas geográficas do utilizador na posse do dispositivo móvel que realizaria as recomendações. A justificação para tal deve-se à natureza dos dados utilizados na validação do modelo. Estes não se encontram em concordância com a condição geográfica da localização onde este modelo foi validado, aspeto que torna a obtenção automática de coordenadas geográficas através de GPS irrelevantes nos testes de validação do modelo. Por conseguinte os parâmetros relativos à geolocalização do utilizador que requisita a recomendação terão de ser introduzidos de forma manual para que estes estejam de acordo com a zona geográfica abrangida pelos pontos de interesse considerados. Além disso, para avaliar a solução é necessário simular um grande número de recomendações, e para tal é necessário um conjunto de dados bastante abrangentes, o que num cenário de utilização real não seria viável, pois seria necessário recolher os dados necessários ao longo do tempo de desenvolvimento do trabalho, e estes nunca seriam suficientemente abrangentes para avaliar o modelo implementado, visto que um sistema de recomendação depende sempre de uma grande quantidade de dados de forma a realizar boas recomendações.

3.2.3.1. Funções do cliente

A aplicação *web* para efeitos de teste deverá ser capaz de responder às seguintes necessidades:

- Visualização de parâmetros a considerar na realização de uma recomendação;
- Edição de parâmetros a considerar na realização de uma recomendação;
- Visualização de resultados de uma recomendação efetuada;

- Visualização da informação individual relativa a um *item* dos resultados de uma recomendação, neste caso pontos de interesse.

3.3. Características dos utilizadores

Um produto que utilize as capacidades do módulo de recomendação a desenvolver destina-se ao utilizador-tipo associado sobretudo à área turística, e que na posse de um dispositivo móvel (*smartphone*) com ligação à internet e de uma aplicação inserida no domínio *web* que implementa esse mesmo módulo, pretenda obter informação referente a possíveis pontos de interesse para a zona onde se encontra tendo em conta as suas preferências, contexto associado aos pontos de interesse, contexto este que se encontrar associado ao comportamento geral dos restantes utilizadores em relação aos mesmos pontos de interesse.

3.4. Recomendações

- Dado o foco numa abordagem baseada em conhecimento (secção 2.2) para a implementação do módulo de recomendações, estas serão altamente dependentes, em termos de eficácia, da existência de um grande volume de informação relativa aos *check-ins* de utilizadores em diversos pontos de interesse;
- O sucesso de um sistema de recomendações dependerá sempre do seu número de utilizadores, do número de *itens* e do número de interações dos utilizadores com os *itens*.

3.5. Atores

Este *item* descreve todos os atores envolvidos no sistema.

- **Utilizador:** Tem como principal responsabilidade a utilização da aplicação de testes, que permite testar e analisar as capacidades que o módulo de recomendações permite explorar.
- **Servidor:** Tem como responsabilidade a gestão dos conteúdos presentes na aplicação. Este recorre a fontes externas de informação de forma a recolher toda a informação necessária para o bom funcionamento do módulo. É responsável também pelo armazenamento de todos os conteúdos necessários à realização de recomendações. Permite também executar todo o processo de avaliação dos pontos

de interesse, o que permite devolver informação ao cliente para que esta possa ser organizada de forma relevante para o utilizador tendo em conta as suas escolhas e outro tipo de contexto associado.

- **Fontes Externas:** Tem como principal responsabilidade interagir com o servidor. Essa interação, que é iniciada pelo servidor, visa a obtenção de informação relevante proveniente das fontes externas.

3.6. Requisitos específicos

3.6.1. Modelos de Casos de Utilização

De seguida são apresentados os requisitos funcionais sob a forma de casos de usos, referentes às Figuras 3.1 e 3.2. Todos terão uma prioridade atribuída, que indicará o grau de importância com que os casos de usos devem ser implementados, sendo essa prioridade um valor da seguinte lista: Alta, Média, Baixa. Os requisitos funcionais que apresentam uma prioridade Alta deverão ser de implementação obrigatória.

3.6.1.1. Diagrama do módulo do servidor

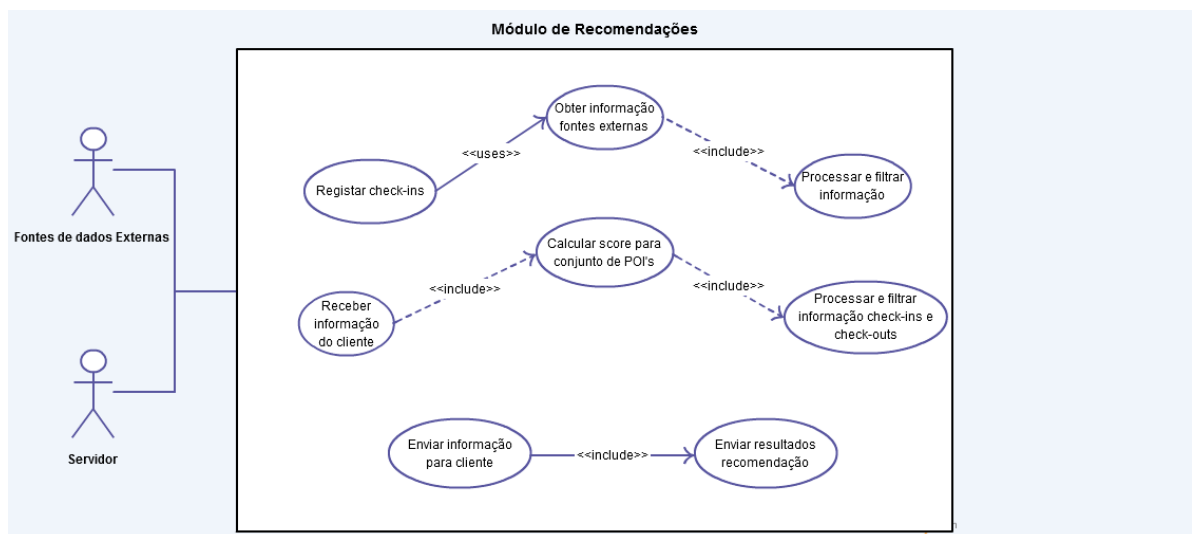


Figura 3.1: Diagrama de casos de uso do servidor

A Figura 3.1 demonstra quais os casos de uso do servidor. Este irá comunicar com o ator “Fonte de dados Externas”, de forma a obter dados relevantes no âmbito deste trabalho, neste caso dados relativos a *check-ins* de utilizadores em diversos pontos de interesse, bem como os valores das condições de contexto registados no momento desses *check-ins*. Esses dados, após terem sido obtidos, irão ser processados pelo servidor, que irá por exemplo ler informação da base de dados de forma a visualizar se os dados obtidos já existem na mesma. Só após este processamento é que os dados obtidos são guardados na base dados.

O servidor estará também dotado da capacidade de realizar recomendações de pontos de interesse com base no comportamento dos utilizadores e de várias condições contextuais consideradas, através de um algoritmo desenvolvido que permite devolver um conjunto de resultados, referente a pontos de interesse, devidamente ordenado consoante o contexto a considerar e as preferências do utilizador que requisitou uma determinada recomendação. Para a realização das recomendações o utilizador recorrerá a um cliente de testes, que permitirá a interação com o servidor, enviando informação pedida por este, tendo que obter os dados guardados na base de dados, aplicar o algoritmo desenvolvido e devolver os resultados da recomendação, satisfazendo o pedido efetuado pelo cliente *web*.

Recolha de informação relativa a pontos de interesse recorrendo a *check-ins* nesses mesmos pontos

Tipo: *Script*

Prioridade: *Alta*

Descrição: O servidor deverá ser capaz de recolher informação relativa a diversos pontos de interesse, através de dados de *check-ins* de utilizadores nesses mesmos pontos. Através de dados como as coordenadas geográficas onde o *check-in* foi efetuado (latitude e longitude), bem como a data e hora do mesmo, e com o auxílio da API externa *Foursquare* [30], obter os dados relativos aos pontos de interesse onde esses *check-ins* foram efetuados, permitindo assim criar um repositório de dados referente a pontos de interesse e informação relacionada aos mesmos.

Atores: servidor

Pré-condições: Servidor com conexão à *internet*

Fluxo de trabalho:

1. O servidor analisa informação relativa ao *check-in* de um utilizador;
2. O servidor com base na latitude e longitude dos *check-ins* recolhe toda a informação necessária relativa ao ponto de interesse onde o *check-in* foi realizado, a partir da fonte de dados externa *Foursquare* [30];
3. O utilizador verifica se a informação para aquele ponto de interesse já existe.

Resultados esperados:

1. O servidor armazena a informação adquirida sobre o ponto de interesse caso esta ainda não exista.

Recolha de informação de contexto

Tipo: *Funcionalidade*

Prioridade: *Alta*

Descrição: O servidor deverá ser capaz de recolher informação relativa a valores de condições de contexto a utilizar para a realização de recomendações, recorrendo à comunicação com fontes externas de informação (como é o caso da API *Open Weather Map* [31]). Recorrendo aos serviços disponibilizados por estas API's é possível então realizar a recolha dessa informação.

Atores: servidor

Pré-condições: Servidor com conexão à *internet*

Fluxo de trabalho:

1. O servidor comunica com uma fonte externa ao projeto;
2. O servidor requisita determinada informação fornecida pela fonte externa com a qual iniciou a comunicação;
3. O servidor aguarda a obtenção de resposta.

Resultados esperados:

1. O servidor vai receber a informação disponibilizada pelo serviço externo;
2. Caso a fonte externa não forneça a informação pedida pelo servidor, este aguarda um determinado tempo, tentando mais tarde voltar a pedir a mesma informação.

Processamento e filtragem de informação de contexto

Tipo: Funcionalidade

Prioridade: *Alta*

Descrição: O servidor deverá ser capaz de processar e filtrar a informação requisitada com recurso a fontes externas. Esta informação deverá estar de acordo com uma nomenclatura definida para o módulo de recomendação sendo necessário realizar a transição da informação recolhida para um formato que o módulo possa interpretar e assim executar as suas funcionalidades. Um exemplo disso poderá ser a temperatura, onde o valor devolvido pela fonte de informação externa será algo como um valor numérico referente a uma temperatura em graus *Celcius* ou *Fahrenheit*. Tratando-se a temperatura de uma variável de contexto considerada nas recomendações onde os seus valores definidos são finitos e nominais (*hot, warm, cold*); as temperaturas terão de ser convertidas para estes valores definidos na nomenclatura aplicada.

Atores: servidor

Pré-condições: O servidor obteve informação de fontes externas ao projeto;

Fluxo de trabalho:

1. O servidor recebe informação de fontes externas ao projeto;
2. O servidor analisa, processa e filtra a informação recebida, adaptando-a à nomenclatura definida.

Resultados esperados:

1. O servidor apenas armazena a informação útil e que irá ser utilizada no futuro.

Processamento e filtragem da informação relativa a *check-ins*

Tipo: Funcionalidade

Prioridade: *Alta*

Descrição: O servidor deverá ser capaz de através de uma análise periódica ao histórico de *check-ins*, verificar para cada ponto de interesse quais os valores predominantes para cada um dos atributos de contexto associados aos pontos de interesse, permitindo assim saber quais os valores mais frequentemente registados para cada um desses atributos em relação a cada um dos pontos de interesse.

Atores: servidor

Pré-condições: Deverá existir informação relativa a histórico de *check-ins* de utilizadores nos pontos de interesse para que esta possa ser processada e filtrada.

Fluxo de trabalho:

1. O servidor analisa o histórico de registo de *check-in* dos utilizadores, sendo que para cada ponto de interesse são analisados os valores associados a cada atributo de contexto, tendo como base todos os registos para cada ponto de interesse no histórico;
2. Para cada um dos atributos de contexto associado a um ponto de interesse é verificado qual o valor predominante;
3. Os valores predominantes para cada atributo de contexto referente a cada ponto de interesse são registados.

Resultados esperados:

1. Armazenamento no repositório de dados, de toda a informação relativa aos valores predominantes inerentes aos atributos de contexto de cada um dos pontos de interesse.

Sistema de avaliação relativo a um conjunto de pontos de interesse

Tipo: Funcionalidade

Prioridade: *Alta*

Descrição: O servidor deverá ser capaz de para um conjunto de pontos de interesse associados a uma recomendação a devolver a um utilizador, calcular um *score* com base no contexto ideal associado a cada um dos pontos de interesse, e escolhas do utilizador que solicita a recomendação, bem como a geolocalização do utilizador no momento da solicitação da recomendação, permitindo assim que os resultados associados a uma determinada recomendação sejam ordenados consoante o que mais se adequa ao momento e ao utilizador que a solicita.

Atores: servidor

Pré-condições: O servidor deverá ter uma conexão à *internet* que permita comunicar com os serviços externos. Deverá também existir acesso a informação relativa ao contexto ideal associado a cada ponto de interesse, sendo que a mesma já deve estar filtrada e processada.

Fluxo de trabalho:

1. O servidor analisa informação relativa ao contexto associado a cada ponto de interesse associado a uma recomendação;
2. O servidor compara a informação de contexto ideal à informação de contexto registada no momento da solicitação da recomendação;
3. Com base na existência de *matching* ou não, relativo à comparação anterior é atribuído um peso a cada uma das condições de contexto;
4. Os pesos atribuídos ao contexto associado a cada ponto de interesse darão origem a um *score* relativo a cada um desses pontos.

Resultados esperados:

1. Permitirá organizar os pontos de interesse inerentes a uma recomendação consoante o que mais interessa e mais se adequa no momento da solicitação da recomendação, com base num sistema de avaliação.

3.6.1.2. Diagrama do módulo do cliente

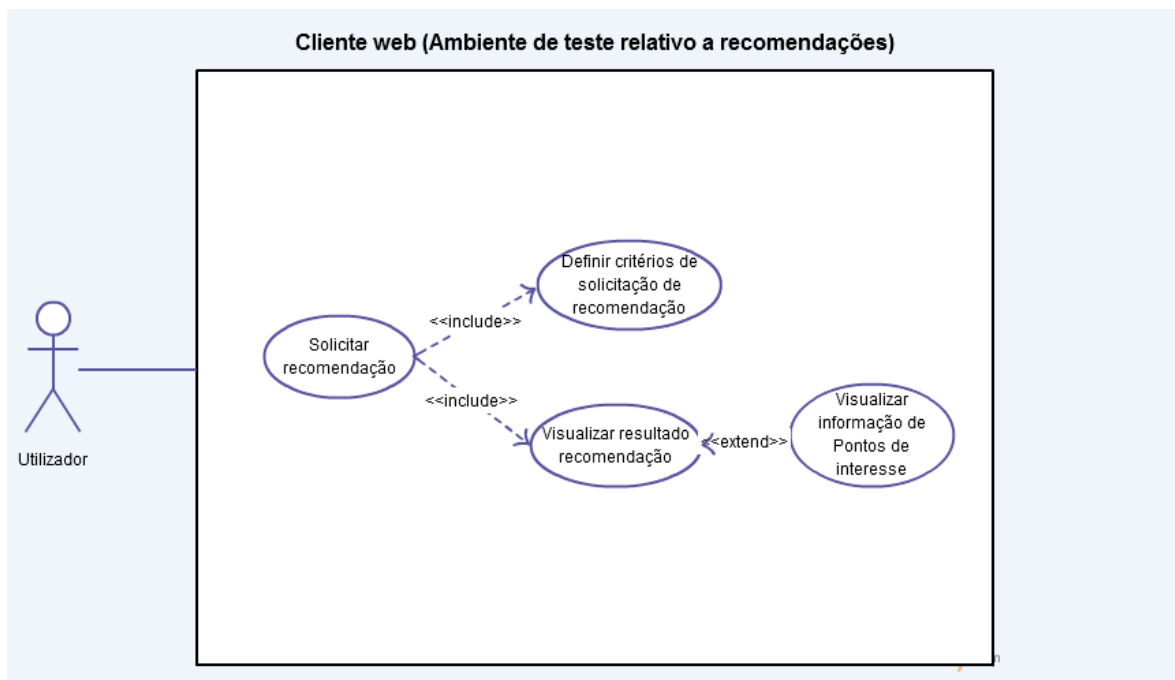


Figura 3.2: Diagrama de casos de uso do cliente

Acesso à localização atual

Tipo: Funcionalidade

Prioridade: *Alta*

Descrição: A aplicação de testes deverá permitir ao utilizador introduzir a sua localização geográfica, visto que se trata de uma aplicação *desktop* não tendo assim GPS. Esta opção prende-se ao facto de permitir também simular em número muito elevado situações de posicionamento geográfico do utilizador, para efeitos de testes.

Atores: utilizador

Pré-condições: Introdução de coordenadas geográficas válidas

Fluxo de trabalho:

1. A aplicação permitirá aceder às coordenadas geográficas simuladas do utilizador que se encontra a solicitar uma recomendação através do cliente de testes.

Resultados esperados:

1. Localizações aleatórias cobrindo toda a área geográfica em análise.

Visualização de critérios de solicitação de recomendação

Tipo: Funcionalidade

Prioridade: *Média*

Descrição: A aplicação deverá permitir que o utilizador visualize as diferentes opções que este tem quando pretende solicitar uma recomendação. Este poderá decidir quais os atributos de contexto que querará considerar na solicitação de uma recomendação e no caso de para alguns deles ser necessário a introdução de *input*, permiti-lo.

Atores: utilizador

Pré-condições: Localização geográfica válida.

Fluxo de trabalho:

1. O utilizador visualiza os atributos de contexto na forma de uma lista acompanhado de um botão de ativação/desativação para cada um deles;
2. O utilizador define que atributos quer considerar na sua solicitação de recomendação;
3. O utilizador para alguns dos atributos que considerou poderá ter de introduzir algum *input* de forma a definir o valor para o atributo.

Resultados esperados:

1. O utilizador fica apto a solicitar uma recomendação com base na sua configuração.

Visualização do resultado de uma recomendação solicitada

Tipo: Funcionalidade

Prioridade: *Muito Alta*

Descrição: A aplicação deverá permitir a visualização de uma lista de pontos de interesse como resultado de uma solicitação de uma recomendação.

Atores: utilizador

Pré-condições: Requisitar recomendação.

Fluxo de trabalho:

1. O utilizador inicia uma recomendação.
2. O utilizador define quais os atributos de contexto que pretende considerar para a recomendação.
3. O utilizador define valores para os atributos que assim o exigirem.

Resultados esperados:

1. O utilizador visualiza uma lista das recomendações sugeridas pelo sistema de recomendação ordenada com base nas suas preferências e no contexto considerado.

Visualização de perfil de um ponto de interesse

Tipo: Funcionalidade

Prioridade: Baixa

Descrição: A aplicação deverá permitir a visualização de informação relativa a um ponto de interesse a partir da sua seleção na lista de pontos de interesse devolvidos pelas recomendações.

Atores: utilizador

Pré-condições: O utilizador terá que ter solicitado uma recomendação.

Fluxo de trabalho:

1. O utilizador solicita uma recomendação;
2. O utilizador visualiza os resultados devolvidos pela recomendação solicitada, na forma de uma lista;
3. O utilizador seleciona um ponto de interesse dessa lista;

4. O utilizador visualiza o perfil do ponto de interesse selecionado contendo toda a sua informação;

Resultados esperados:

1. O utilizador visualiza a informação detalhada associada a pontos de interesse após sua seleção a partir de uma lista de pontos de interesse relativa a resultados de uma recomendação.

3.7. Requisitos não funcionais

Compatibilidade

Tipo: Suporte

Prioridade: *Alta*

Descrição: A aplicação deverá ser compatível e acessível através de um *browser* sendo que o sistema operativo não tem relevância. O *browser* testado foi o Mozilla Firefox versão 24.0.

Comunicação com servidor

Tipo: Interoperabilidade

Prioridade: *Muito Alta*

Descrição: A aplicação deverá ser capaz de comunicar com o servidor, de forma a requisitar serviços de obtenção e envio de dados relevantes no contexto das recomendações de pontos de interesse com base em contexto.

3.8. Arquitetura do sistema

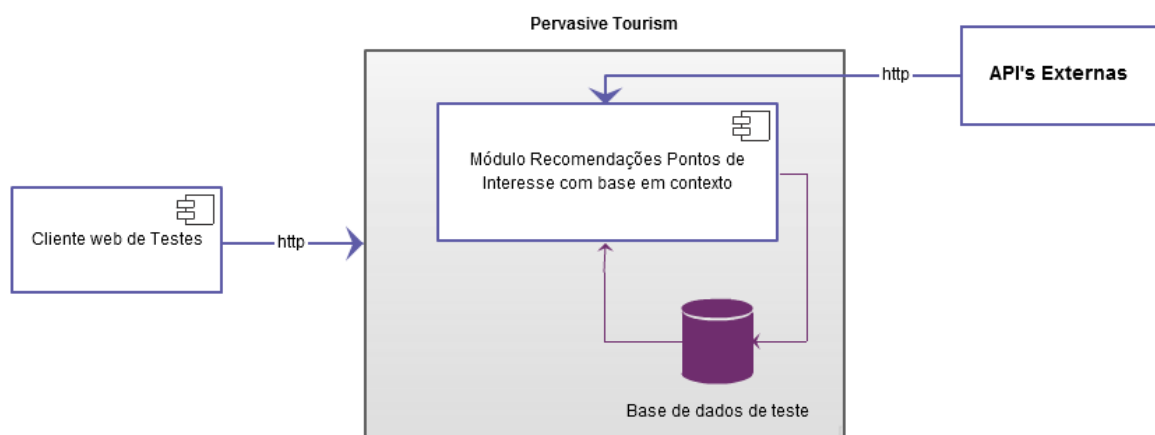


Figura 3.3: Arquitetura do Sistema

Através da análise do diagrama com a arquitetura do sistema (Figura 3.3), é possível verificar que este é constituído por quatro componentes claramente distintos e que se baseia numa arquitetura cliente-servidor.

O componente designado de “Cliente *web* de testes” é a aplicação nativa a ser desenvolvida na linguagem python recorrendo a uma *web framework* designada Django [33] e trata-se do único componente a ser usado diretamente por parte dos utilizadores na realização de testes ao módulo.

O componente “Módulo Recomendações Pontos de Interesse com base em contexto” é o servidor responsável por recolher e receber toda a informação relevante ao cenário de base deste trabalho e disponibilizar serviços ao “Cliente *web* de testes”, nomeadamente a execução e consequentes resultados de recomendações com base em contexto.

O componente “Base de dados de teste” é um repositório de dados no qual são guardados os dados processados pelo servidor. O componente “APIs Externas” consiste no conjunto de todos os serviços disponíveis na rede e que é utilizado pelo componente “Módulo Recomendações Pontos de Interesse com base em contexto” na obtenção da informação relevante.

3.8.1. Cliente *web* de testes

Este componente irá possibilitar a utilização do sistema de recomendação com base em contexto implementado, recorrendo a uma interface visual, permitindo a um utilizador introduzir os *inputs* necessários à requisição de uma recomendação e consequente visualização dos resultados.

Este componente estará em interligação com o sistema, sendo que ao nível da perceção conceptual do sistema um utilizador só terá uma noção do sistema como o cliente apenas, sendo que todas as restantes componentes terão um carácter abstrato.

O cliente que define este componente será um cliente *web* desenvolvido na linguagem Python recorrendo a um *web framework* designado Django [33], e estará acessível a partir de qualquer *web browser*.

O objetivo principal deste componente é permitir que se possam testar as funcionalidades do módulo de recomendações com base em contexto desenvolvido, e a partir dos resultados dos testes realizar uma análise de modo a que se possa concluir quais as vantagens de aliar o contexto a recomendações, nomeadamente no domínio deste trabalho, o turismo.

Ao nível do funcionamento este componente comunica apenas com o componente “Módulo Recomendações Pontos de Interesse com base em contexto”. Esta comunicação com este componente permite ao cliente obter a informação necessária à realização de recomendações e visualização de resultados (lista de pontos de interesse recomendados ordenados mediante a preferência do utilizador e contexto considerado). Esta informação encontra-se armazenada no componente “Base de dados de teste” e será sempre transmitida através da componente “Módulo Recomendações Pontos de Interesse com base em contexto”, para o componente “cliente *web* de testes”. A forma como esses dados são obtidos ou escritos na componente “Base de dados de teste” é totalmente irrelevante para o componente “Cliente *web* de testes”.

A comunicação entre o componente “Cliente *web* de testes” e o componente “Módulo Recomendações Pontos de Interesse com base em contexto” é realizada através da *internet*.

3.8.2. Módulo Recomendações Pontos de Interesse com base em contexto

Este componente é o intermediário entre a aplicação de testes que apresenta uma interface gráfica, e os dados que esta disponibiliza. Este componente é o mais importante no trabalho, pois é responsável por todo o processo de interpretação dos *inputs* fornecidos pelo cliente *web* de testes e consequente obtenção e envio de recomendações com base em contexto para o cliente. De forma a compreender melhor estas tarefas segue abaixo uma breve descrição de todo o processo:

1. Este módulo apresenta-se como um *web service* e como tal tem de ser consumido de forma a desempenhar a sua tarefa. O cliente *web* de testes é responsável pelo consumo deste *web service*. Como requisitos para o consumo deste *web service* é necessário introduzir os *inputs* necessários para que o módulo de recomendações realize recomendações. Os *inputs* necessários são a geolocalização do utilizador (latitude e longitude), o raio, em metros, com centro na sua geolocalização que pretende considerar para pontos de interesses, a categoria relativa aos pontos de interesse, e as condições contextuais que pretende considerar bem como o peso que pretende atribuir a cada uma dessas condições na obtenção de recomendações.
2. Com os *inputs* necessários o módulo começa por filtrar todos os pontos de interesse armazenados na base de dados através de uma seleção dos pontos que se encontram no raio de ação centrado na geolocalização do utilizador considerado por este. Para exercer esta tarefa foi desenvolvida uma função ao nível da base de dados, que permitisse a interação com dados espaciais (latitudes e longitudes). Isto é possível graças ao sistema de gestão base de dados escolhido que possui uma extensão que permite trabalhar com dados espaciais, o PostGIS [39]. Para além desta função que lida com dados georreferenciados, também a forma como estes dados foram armazenados salvaguardava a performance do módulo, pois este filtro que se encontra a ser descrito tem de filtrar todos os pontos de interesse armazenados para saber se estes se encontram dentro do raio desejado, o que para um número muito grande de pontos de interesse pode tornar o processo muito lento. Para contornar este problema foi também utilizado um índice GIST [40] sobre a entidade responsável pelo armazenamento dos dados relativos a pontos de interesse. Este índice não é mais que uma estrutura de dados que permite construir uma árvore de pesquisa cuja sua função se destina à organização de dados numa hierarquia, sendo que esta estrutura pode ser utilizada para quase todo o tipo de dados. Neste caso em específico foi utilizado para dados georreferenciados. Neste caso a aplicação deste índice permite garantir que as pesquisas na entidade em questão apresentam uma performance consideravelmente superior.

3. Após a aplicação do filtro mencionado no ponto 2 é necessário aplicar um novo filtro que irá permitir selecionar apenas os pontos de interesse que remetem à categoria para a qual o utilizador pretende obter recomendações. Este filtro será aplicado sobre os resultados obtidos no ponto 2.
4. Em seguida vem a consideração do contexto e da sua associação aos resultados obtidos no ponto 3. Para cada um dos pontos de interesse obtidos no ponto 3 é agora necessário gerar um *ranking* com base nas condições de contexto consideradas e os respetivos pesos atribuídos a cada uma delas, aquando o pedido de recomendação de forma a organizar os resultados em termos de relevância contextual. Para que tal seja possível foi introduzido o conceito de dominância contextual associada aos pontos de interesse. Este conceito surge como uma solução para o facto de existir o *handicap* relativo aos dados dos utilizadores e esse fator impossibilitar o seguimento de uma abordagem de filtragem colaborativa na realização de recomendações. Esta situação deve-se em parte ao problema do *cold start* já explicado na secção 2.2. Algumas abordagens estudadas assentam nas recomendações baseadas em conhecimento (ver secção 2.2), e será nesta abordagem que este trabalho estará assente. Tendo como base alguns dos projetos relacionados estudados, deve existir uma fase onde é coletado o *feedback* de pessoas no intuito de se poder avaliar os *itens* a recomendar, neste caso pontos de interesse no domínio do turismo. Este processo pode ser feito através de inquéritos a um conjunto de pessoas [23]. O facto deste processo logisticamente poder ser um pouco moroso e obrigar à envolvimento de um grupo consideravelmente grande de pessoas dificulta a sua implementação. Assim sendo optou-se por uma solução estatística que possibilita estabelecer padrões em relação às condições contextuais consideradas nas visitas dos utilizadores aos diferentes pontos de interesse armazenados na base de dados. O que se sabe é que os dados existentes remetem essencialmente a pontos de interesse e *check-ins* de utilizadores nesses mesmos pontos, e sabe-se à partida que a natureza do utilizador que realizou o *check-in* é desconhecida. A solução passa então por calcular uma moda para cada condição de contexto considerada nas visitas desses mesmos utilizadores aos pontos de interesse. Considerando o universo de *check-ins* existentes na base de dados, é necessário para cada ponto de interesse distinto perceber quantos *check-ins* realizados por utilizadores existem e calcular o valor dominante (moda) para cada uma das condições de contexto considerada, permitindo assim que cada ponto de interesse tenha as suas condições de contexto ideais para visita registadas de forma persistente na base de dados. Em suma existe um valor dominante para cada condição de contexto em função de um determinado ponto de interesse. Com esta informação é possível saber, dentro do universo considerado de *check-ins* de utilizadores, qual o padrão dominante relativo às condições contextuais nas visitas aos diferentes pontos de interesse, sendo que este padrão será relevante no momento em que uma recomendação seja solicitada. Esta dominância contextual (moda) será o contributo por parte dos utilizadores no que toca ao modelo de recomendações a implementar, baseado no apresentado na secção 2.2.1. A figura 3.4 ilustra um exemplo para o cálculo da dominância, onde é possível visualizar informação relativa a *check-ins* de utilizadores em dois pontos de interesse (A e B). A estes *check-ins* está associada informação de

contexto sobre a qual irá ser aplicado o conceito de dominância. Para todos os *check-ins* relativos a cada um dos pontos de interesse (A e B) é obtido o valor que apresenta mais frequência para cada uma das condições de contexto considerada. Por exemplo no caso do “ponto A” para a “condição de contexto 1”, o valor que ocorre mais vezes é “Evening” enquanto para o “ponto B” é “Afternoon”.

Check-ins		Condição de contexto 1	Condição de contexto 2	Condição de contexto 3
		Altura do dia	Estado do Tempo	Temperatura
check in 1	ponto de interesse A	Morning	Clear	hot
check in 2	ponto de interesse A	Evening	Rainy	warm
check in 3	ponto de interesse B	Morning	Mist	cold
check in 4	ponto de interesse B	Afternoon	Rainy	cold
check in 5	ponto de interesse A	Evening	Clear	hot
check in 6	ponto de interesse B	Afternoon	Rainy	warm

Dominância dos valores associados às condições de contexto	Condição de contexto 1	Condição de contexto 2	Condição de contexto 3
	Altura do dia	Estado do Tempo	Temperatura
Ponto de interesse A	Evening	Clear	hot
Ponto de interesse B	Afternoon	Rainy	cold

Figura 3.4: Exemplo prático do conceito de dominância

Com o conceito de dominância explicado torna-se mais fácil descrever como é que o *ranking* que permite organizar os resultados obtidos no ponto 3 em termos de relevância contextual é calculado. Recorrendo aos dados obtidos a partir da aplicação do conceito de dominância é possível identificar quais os valores contextuais ideais (estatisticamente) para visitar um determinado local. Através da comparação entre os valores das condições contextuais consideradas nos *inputs* e os valores “dominantes” de contexto associados aos pontos de interesse do conjunto de resultados obtido no ponto 3 é possível atribuir um *score* a cada um dos pontos de interesse do conjunto de resultado que se traduz na avaliação contextual desses pontos para aquela recomendação. O cálculo destes *scores* traduz-se no sistema de *ranking* contextual, que permitirá atribuir uma relevância contextual aos resultados obtidos. O conjunto de pontos de interesse obtidos no ponto 3 pode ser agora ordenado pelo seu *score*, sendo que quanto maior for o seu *score* maior será a sua relevância contextual na recomendação e por conseguinte será um melhor *item* na recomendação em detrimento de outros *items* que apresentem um *score* inferior. Em suma será o contexto a ditar a relevância dos resultados recomendados, tendo como base para a avaliação contextual o conceito de dominância explicado.

5. Após a consideração e associação de contexto ao conjunto de resultados e respetiva ordenação dos mesmos, estes encontram-se preparados para serem apresentados ao utilizador que pretende obter a recomendação tendo em conta os *inputs* que introduziu. O módulo de recomendações envia então os resultados para o cliente *web* de testes para que o utilizador possa visualizar o resultado da recomendação que pretende, sabendo que os resultados se encontram organizados em termos de relevância contextual.

3.8.3. Base de dados de testes

Este componente consiste numa base de dados que permite armazenar os dados relativos aos pontos de interesse turísticos. os *check-ins* de utilizadores nesses mesmos pontos, bem como todos os dados contextuais associados a esses mesmos pontos de interesse e *check-ins* nesses mesmos pontos. Estes dados são obtidos a partir de fontes de dados externas, bem como de um conjunto de dados de *check-ins* do Foursquare [30] que foi utilizado para popular a base de dados.

Este conjunto de dados, já utilizado em outro estudo [34], contém informação relativa a *check-ins* de utilizadores em diversos pontos de interesse nas cidades de Nova Iorque e San Francisco nos Estados Unidos da América. Este conjunto de dados contém informação extraída de uma rede social baseada no conceito de localização de utilizadores bastante conhecida, o Foursquare [30]. A informação contida neste conjunto de dados remete a dados como latitude e longitude onde o *check-in* foi efetuado, bem como a respetiva data e hora em que esse evento ocorreu.

A API pública que o Foursquare [30] fornece não permite, de forma direta, aos seus utilizadores a recolha deste tipo de dados pessoais (política de privacidade). No entanto todos os utilizadores do aplicativo Foursquare [30] que partilham a informação relativa aos *check-ins* via Twitter tornam este tipo de informação pública, pois esta informação pode ser recolhida recorrendo à API pública do Twitter. Todos os *check ins* que este conjunto de dados contém estão compreendidos entre Agosto de 2010 e Novembro de 2011.

Pelo que foi descrito, pode-se constatar que a informação que é possível recolher do mesmo não é suficiente para o trabalho que se pretende desenvolver, pois este não contém informação específica relativa aos pontos de interesse, que de forma muito básica neste conjunto de dados é representada pelas coordenadas geográficas, e não contém a informação de contexto que se pretende considerar. Logo por aqui é necessário enriquecer este conjunto de dados com dados que permitam extrair mais informação para os *check-ins* que este contém. Para tal recorreu-se de novo à API pública de Foursquare [30], de forma a recolher dados relativos aos pontos de interesse identificados no conjunto de dados pelas suas coordenadas geográficas. De forma a introduzir a informação relativa às condições de contexto desejadas, associadas aos *check-ins* do conjunto de dados inicial, recorreu-se também a uma API pública relativa a dados meteorológicos, a Open Weather Map [31], e também a concretizar conhecimento empírico, tudo isto extraído com base na informação temporal (data e hora) de cada *check- in* registado no conjunto de dados original.

A qualidade de um sistema de recomendação depende da qualidade e quantidade dos seus dados e, por norma, estes fatores influenciam a qualidade das recomendações. Introduzindo o contexto e associando-o aos dados existentes é importante que informação relativa às condições de contexto consideradas esteja presente para cada *item*, que no caso específico deste trabalho são os pontos de interesse.

3.8.4. API's Externas

Este componente é externo a este trabalho, pois diz respeito a um conjunto de serviços já existentes na *Web* e pertencentes a entidades terceiras a este trabalho, sobre os quais apenas serão feitos pedidos de leitura de informação fornecidas por estas. Essa informação é aquela que será usada para preencher inicialmente a base de dados, e conseqüentemente alimentar a aplicação de testes.

Capítulo 4

Implementação do Sistema

Neste capítulo irão ser abrangidos os diferentes processos de implementação deste trabalho, sendo que estes englobam o processo de implementação do lado do servidor, o componente relativo ao “Módulo Recomendações Pontos de Interesse com base em contexto” (secção 3.8.2) e o processo relativo à implementação do cliente, a aplicação *desktop* (secção 3.8.1).

4.1. Servidor

As tarefas principais do servidor remetem à realização de recomendações com base em contexto de pontos de interesse no domínio do turismo aos utilizadores da aplicação de testes, bem como à comunicação com fontes de dados externa, e com o cliente de testes, a aplicação *web*.

Para a realização deste tipo de recomendações é necessária a existência de dados inseridos no domínio mencionado. Para tal é necessário criar uma base de dados para armazenamento dos dados de forma persistente, definindo um modelo de dados capaz de suportar toda a informação obtida a partir das fontes de dados externas, que vão desde um conjunto de dados contendo dados relativos a *check-ins* de utilizadores em pontos de interesse associados ao *Foursquare*[30], a dados complementares extraídos de fontes de dados externas como o *Foursquare*[30] e o *OpenWeatherMap* [31] (procedimento já mencionado na secção 3.7.3).

4.1.1. Modelo de dados

O modelo de dados projetado foca-se sobretudo em duas entidades chave, tendo em conta a realização de recomendações no domínio do turismo e os dados que foram possíveis angariar para a realização deste estudo. Estas entidades chave são os pontos de interesse e o histórico de visitas de utilizadores nestes mesmos pontos de interesse. Aliado à entidade pontos de interesse temos dados relativos às informações de perfil destes. Quanto à entidade relativa ao histórico de visitas dos utilizadores nos diferentes pontos de interesses temos associados dados relativos às diferentes condições contextuais consideradas.

Os pontos de interesse armazenados na base dados registam dados como nome, endereço, telefone, coordenadas geográficas, categoria a que pertencem, entre outras informações. Alguns destes dados podem funcionar também como condições contextuais como é o caso da geolocalização do ponto de interesse bem como a categoria a que este pertence.

Quanto ao histórico de visitas de utilizadores nos diferentes pontos de interesse, a entidade responsável por esta tarefa regista dados como o ponto de interesse onde se realiza a visita, a data e hora bem como o valor definido para aquele momento para cada condição de contexto considerado neste trabalho.

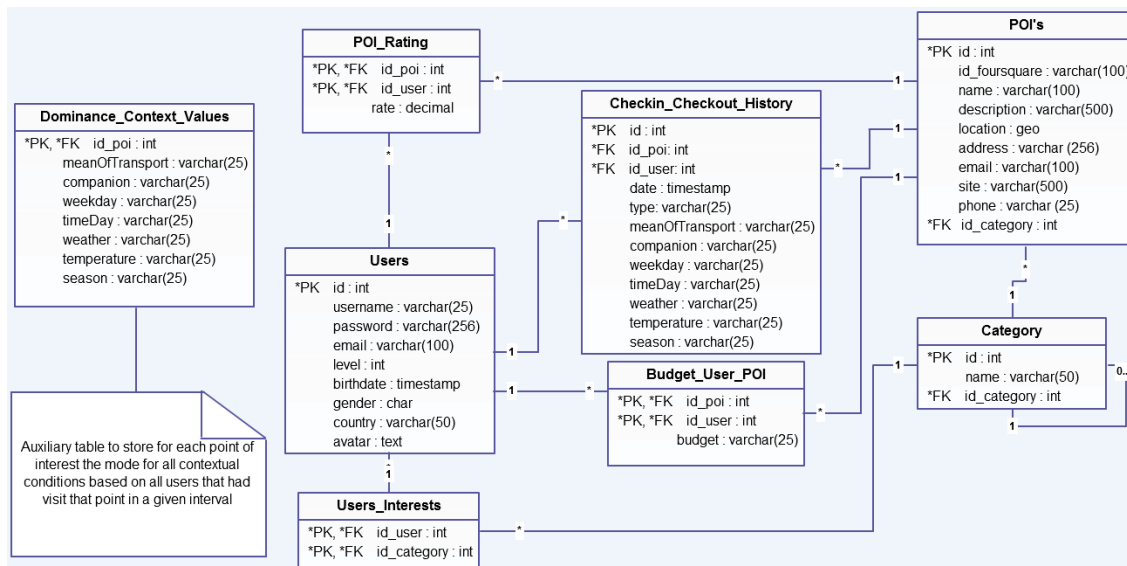


Figura 4.1: Modelo de dados

Observando o diagrama relativo ao modelo de dados projetado (Figura 4.1) é possível constatar todo o tipo de dados que as duas entidades acima descritas permitem armazenar. No entanto é também possível visualizar a existência de mais entidades neste modelo de dados.

Isto deve-se ao facto de o módulo de recomendações a ser implementado neste trabalho ter sido pensado tendo em vista a implementação futura de um sistema de filtragem colaborativa. São considerados *check-ins* de utilizadores em pontos de interesse, mas cujo perfil de utilizador fosse conhecido, seguindo uma abordagem de filtragem colaborativa onde o feedback dos utilizadores é uma forte componente presente nas recomendações.

Para que tal seja possível devem ser consideradas entidades. A entidade do utilizador (“Users”) que diz respeito a toda a informação de perfil do mesmo, onde parâmetros como a idade e o género poderiam funcionar como condições contextuais.

A entidade relativa aos interesses dos utilizadores (“Users_Interests”) que permite armazenar dados relativos aos interesses dos utilizadores nas diferentes categorias relativas a pontos de interesse presentes na base de dados.

A entidade relativa à avaliação orçamental dos pontos de interesse (“Budget_User_POI”) onde os utilizadores podem avaliar se um determinado ponto de interesse é caro, aceitável ou barato (adequado por exemplo no caso da restauração).

E ainda a entidade relativa à avaliação dos utilizadores sobre os pontos de interesse (“POI_Rating”) onde seria possível registar dados relativos à avaliação global de cada utilizador sobre cada um dos pontos de interesse que visitasse.

O que tornou impossível a utilização dos dados relativos a estas entidades (mais ligadas ao lado do utilizador que realiza os *check-ins*) na realização de recomendação prende-se com o facto de o conjunto de dados obtido neste trabalho não incluir dados relativos aos utilizadores. O que acontece é que o acesso a dados onde existe a relação utilizador-ponto de interesse, é considerado privado, visto que as fontes de dados externas a partir das quais foram angariados os dados relativos aos *check-ins* de utilizador não permitem que a informação pessoal dos utilizadores seja partilhada para o público devido às suas políticas de privacidade, e neste caso concreto as políticas de privacidade do *Foursquare*.

Muito sucintamente os únicos dados que são possíveis ter acesso são os de que um utilizador (cuja identidade é desconhecida) realizou um *check-in* num determinado ponto de interesse numa data e hora específica.

A entidade “Dominance_Context_Values” encontra-se isolada das outras, pois esta trata-se de uma entidade auxiliar. Esta entidade é responsável por armazenar os dados que são obtidos quando o conceito de dominância (explicado em 3.8.2) é aplicado. Esta serve apenas para armazenar os dados relativos aos valores considerados ideais para cada condição de contexto associada aos pontos de interesse turísticos existentes na base de dados.

Para a base de dados que irá suportar este modelo de dados foram consideradas várias opções possíveis de Sistema de Gestão de Base de Dados, sendo o sistema escolhido o PostgreSQL [32]. A escolha deveu-se a diversos fatores, dos quais o facto de ser um software open-source, ser compatível com diversas linguagens (Java ou Python, por exemplo), poder ser usada em conjunto com a plataforma Django [33], suportar instruções em SQL, suportar o armazenamento de dados georreferenciados e ser o SGBD mais usado pela empresa Ubiwhere.

4.1.2. Modelo Contextual

Tendo como base a abordagem mencionada na secção 2.2.1, um sistema de recomendação com base em contexto necessita de ter poder de análise sobre propriedades dos utilizadores e sobre propriedades contextuais, permitindo assim a realização de recomendações mais personalizadas e enriquecidas. Na secção 3.8.2 quando foi explicado o conceito de dominância implementado e que se traduz nos dados armazenados na entidade “Dominance_Context_Values” do modelo de dados ficou-se a perceber como seria considerada a componente relativa aos utilizadores nas recomendações sendo que esta componente já faz menção à associação de condições de contexto ao comportamento dos utilizadores nos diferentes pontos de interesse, e neste caso específico *check-ins* nestes mesmos pontos.

Nesta secção irão ser explicadas as diferentes condições de contexto consideradas bem como os possíveis valores que estas condições podem assumir. No que toca a condições contextuais foram consideradas as seguintes:

- Geolocalização (Geolocation);
- Categorização de pontos de interesse (Points of interest categories);
- Meio de transporte (Mean Of transport);
- Companhia (Companion);
- Dia da semana (Weekday);
- Altura do dia (Time Day);
- Estado do tempo (Weather);
- Temperatura (Temperature);
- Estação do ano (Season);

Para cada uma das condições contextuais associadas aos *check-ins* dos utilizadores nos pontos de interesse, foi definido um conjunto de valores que estas poderiam assumir, sendo que estes valores assumem o espectro de possibilidades que estas condições podem assumir, e que serão considerados sempre que o algoritmo do módulo de recomendações for aplicado, tanto no cálculo da dominância (moda) para cada uma das condições relativamente a cada ponto de interesse, bem como na realização de recomendações em tempo real. Para cada condição de contexto considerada o conjunto de valores associados é auto explicativo como é o caso da geolocalização, categorias, meio de transporte, companhia, dia da semana e estação do ano. No entanto, para as restantes condições: altura do dia, estado do tempo e temperatura, os conjuntos de valores associados são atribuídos mediante determinadas regras estabelecidas. Começando pela altura do dia, para cada valor associado a esta condição existem regras de atribuição, no caso do valor *morning* é para qualquer hora registada para além 6h até às 12h, quanto ao valor *afternoon* é para qualquer hora registada para além das 12h até às 19h e por fim o valor *night* para qualquer hora registada para além das 19h até as 6h. Quanto à condição de contexto relativa ao estado do tempo cada valor do conjunto corresponde ao estado do tempo devolvido na informação retornada pela fonte de dados externa, neste caso a API *Open Weather Map* [31], sendo que os valores considerados correspondem aos códigos utilizados pela própria API para definir o estado do tempo. Por fim para os valores possíveis relativos à condição de contexto temperatura, e à imagem da condição de contexto altura do dia, foram definidos intervalos de temperatura para a atribuição dos valores definidos. No caso do valor *hot* temperaturas superiores a 25 graus *Celcius*, quanto ao valor *warm* temperaturas compreendidas entre 10 e 25 graus *Celcius*, e por fim para o valor *cold* temperaturas inferiores a 10 graus *Celcius*.

Geolocalização	Categorias	Meio de transporte	Companhia	Dia da Semana
Latitude	shops	car	alone	working day
Longitude	travel	bicycle	friends	Weekend
	building	pedestrian	family	
	food	public transport	girlfriend/boyfriend	
	nightlife			
	park outdoors			
	arts entertainment			
	education			

Altura do dia	Estado do tempo	Temperatura	Estação do ano
morning	sunny	hot	spring
afternoon	cloudy	warm	summer
night	clear sky	cold	autumn
	rainy		winter
	snowing		
	thunderstorming		
	drizzling		
	mist		
	extreme weather		

Figura 4.2: Valores possíveis para cada condição de contexto

De referir que existem duas condições de contexto consideradas que tiveram que ser postas de parte devido ao facto de não existirem dados que possibilitassem o seu uso. São elas a condição “Meio de transporte” e “Companhia”. Como tal estas duas foram desconsideradas na aplicação do algoritmo de recomendação e na análise dos testes efetuados.

4.1.3. Obtenção de dados

No que diz respeito à obtenção de dados, e como já explicado na secção 3.8.3, o ponto de partida para popular a base de dados, foi um conjunto de dados disponibilizado contendo dados relativos a *check-ins* de utilizadores em pontos de interesse nas cidades de Nova Iorque e São Francisco nos Estados Unidos da América, recorrendo à aplicação Foursquare [30]. O conjunto de dados apresentava-se num formato CSV, sendo que apenas era dotado de dados relativos às coordenadas geográficas (latitude e longitude) do local e a hora e data onde o *check-in* tinha sido efetuado.

Tendo estes dados como ponto de partida recorreu-se a fontes de dados externas de forma a enriquecer os dados relativos a *check-ins* e aos pontos de interesse registados nos *check-ins*. Para tal recorreu-se a duas API's que permitissem concretizar os requisitos pretendidos.

Recorreu-se à API pública do Foursquare [30] de forma a recolher, com base nas coordenadas geográficas dos *check-ins* registados, informações de perfil relativas aos pontos de interesse onde os utilizadores deram entrada. Esta tarefa tornou possível recolher dados como o nome do ponto de interesse, a morada, a categoria a que este pertence entre

outros (figura 4.1), permitindo que para cada ponto de interesse distinto, onde existem *check-ins* no conjunto de dados inicial, existisse informação de perfil relevante sobre cada um destes pontos de interesse.

Mas existem outros dados considerados, que necessitam do auxílio de uma fonte de dados externa para poderem ser determinados, o estado do tempo e a temperatura. Cruzando, a partir do conjunto de dados inicial, os dados relativos às coordenadas geográficas com a data e a hora, é possível recorrendo a uma API pública relacionada com dados meteorológicos, neste caso o *Open Weather Map* [31], coletar os dados necessários de forma atribuir valores para essas condições de contexto.

Todo este processo de obtenção de dados recorrendo a estas API's foi realizado recorrendo a metodologias de *scripting*, onde para cada *check-in* existente no conjunto de dados inicial estas API's eram acedidas e através do consumo de *web services* estes dados eram recolhidos. Dado o tamanho do data set inicial ser consideravelmente grande (2073740 *check-ins*) e o facto de o consumo destas API's ser um pouco limitado em termos de acesso aos dados de forma individualizada para cada um dos *check-ins* (cerca de um segundo por pedido) foi necessário definir uma amostra deste conjunto de dados, e com base nessa amostra realizar a obtenção dos dados necessários. A amostra considerada remete apenas a *check-ins* realizados numa das cidades, neste caso São Francisco, o que reduziu drasticamente o número de *check-ins* (para 62564 *check-ins*) e tornou viável a coleta de dados necessário ao enriquecimento do conjunto de dados.

Após a coleta de dados complementares ao conjunto de dados inicial, e recorrendo ao modelo de dados projetado criou-se a base de dados e os dados coletados foram inseridos, gerando assim um repositório de dados adequado, para que o módulo a desenvolver pudesse realizar as recomendações com base em contexto pretendidas.

4.1.4. Atualização de dados

No que toca à atualização de dados na base dados, esta é realizada apenas para os valores registados na entidade auxiliar “Dominance_Context_Values”.

Esta atualização existe visto que os valores dominantes (moda) para cada condição de contexto associada aos pontos de interesse podem variar ao longo do tempo, pois os utilizadores variam as suas tendências e os seus hábitos mediante a sua envolvimento com o meio que o rodeia e mutação dos seus hábitos sociais.

Para permitir esta atualização foi criada uma função ao nível do sistema de gestão de base de dados que permitisse atualizar os valores dominantes para cada condição de contexto associada a cada um dos pontos de interesse, numa periodicidade a definir através do agendamento de tarefas recorrendo a uma ferramenta existente no sistema de gestão de base de dados, o *postgres pgagent* [35].

4.1.5. Desenvolvimento

O servidor foi desenvolvido recorrendo à *framework* Django [33]. Esta é uma *framework* usada para auxiliar o desenvolvimento de aplicações *web* tornando este processo mais rápido e fácil, sendo escrita em Python.

Esta *framework* utiliza o paradigma MVC (Model-View-Controller), permitindo assim uma fácil integração entre a base de dados (Model) e a camada de negócio (View) responsável por interagir com a base de dados e tratar e processar os dados nela presente. Toda a comunicação entre estas diferentes camadas é realizada de forma invisível para os seus utilizadores, sendo muito fácil manipularem e interagirem com a base de dados apenas através de código na linguagem Python. Esta suporta também diferentes bases dados, tais como SQLite [36], PostgreSQL [32] ou MySQL [37]. Neste trabalho a base de dados usada foi a PostgreSQL [32].

Inicialmente foi definido o modelo de dados correspondentes ao já apresentado anteriormente. Esse passo foi efetuado num ficheiro com um nome próprio, o `models.py` (ver figura 4.3), no qual são definidos as diferentes tabelas do modelo de dados e os respetivos atributos e ligações entre as tabelas.

Tendo o modelo implementado e as tabelas criadas, tornou-se possível realizar a obtenção de dados através das fontes externas e guardar os resultados obtidos na base de dados. Foram criados vários scripts, escritos em Python, contendo o código responsável por comunicar com as fontes externas. Após a coleta dos dados foi gerado um projeto referente ao módulo de recomendações a implementar e a testar. Esse projeto contém diversos ficheiros organizados em vários pacotes, de forma a organizar melhor o trabalho e separar claramente o que cada um faz. A Figura 4.3 mostra uma visão geral da forma como esses ficheiros se organizam.

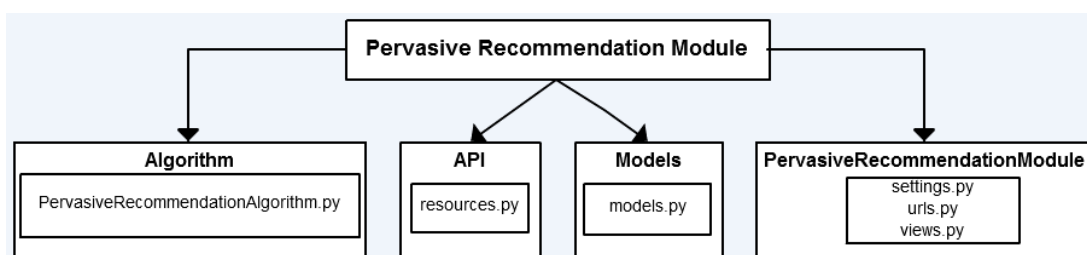


Figura 4.3: Organização dos ficheiros no servidor

O “Pervasive Recommendation Module” é a pasta raiz de todo o projeto do lado do Servidor. Nessa pasta, estão presentes outras pastas, tais como a pasta “API”, “Algorithm”, “models” e “PervasiveRecommendationModule_BackEnd”, sendo esta última automaticamente criada pelo Django aquando da criação de um novo projeto. Essa pasta contém as configurações do servidor (`settings.py`), no qual é por exemplo definido qual o tipo de base dados a ser usado e qual o seu nome e conexão à mesma. Para além deste ficheiro, contém também o ficheiro “`urls.py`” que permite fazer a interligação entre endereços URL e funções presentes em ficheiros Python de forma a processar dados e mostrar estes em páginas *web*. No ficheiro “`views.py`” é onde se encontram algumas

funções em Python para processar dados, que podem por exemplo serem chamadas a partir do ficheiro “url.py”.

A pasta “Algorithm” contém o ficheiro “PervasiveRecomendationAlgorithm.py” onde estará a implementação do algoritmo relativo à realização de recomendações com base em contexto. Este algoritmo será responsável com base na recomendação requisitada de analisar todos os parâmetros de entrada e com base nos mesmos realizar uma recomendação de pontos de interesse.

Os parâmetros de entrada que este algoritmo recebe são a localização do utilizador que requisitou a recomendação e o raio de ação em torno do mesmo (raio a considerar na realização da recomendação), a categoria relativa ao tipo de ponto de interesse que o utilizador pretende obter como recomendação e quais as condições de contexto que pretende considerar na realização da recomendação bem como o peso que cada uma dessas condições consideradas terá na mesma recomendação. A partir destes parâmetros o algoritmo irá filtrar quais os pontos de interesse que se encontram no raio de ação definido pelo utilizador em torno do mesmo, quais desses pontos pertencem à categoria desejada, e calcular uma pontuação para esses pontos de interesse tendo em conta os valores das condições de contexto consideradas no momento do pedido de realização de recomendação face aos valores das condições de contexto dominantes para esses mesmos pontos de interesse e os pesos atribuídos pelo utilizador a cada uma dessas condições consideradas.

Este procedimento permitirá assim gerar um conjunto de pontos de interesse ordenados por uma pontuação contextual, conjunto este que representará a recomendação a apresentar ao utilizador.

4.1.6. Comunicação entre o servidor e o cliente

A comunicação entre o servidor e o cliente é feita recorrendo a um *web service*. Para tal, foi necessário desenvolver uma API que fornecesse esse tipo de serviços, podendo esta ser implementada na forma de uma API REST ou API SOAP. A escolha recaiu sobre a API REST, por ser consideravelmente mais simples a sua implementação e por utilizar o próprio protocolo HTTP na diferenciação das operações, o que torna mais leve as comunicações entre cliente e servidor.

O *web service* está presente no módulo “resources” do pacote “API” (Figura 4.3). Para o seu desenvolvimento foi usado uma *framework*, a *Tastypie* [38], que é facilmente integrada num projeto Django [33] e fornece um modo fácil de desenvolver uma API-REST.

Esta ferramenta possibilita por si só o uso dos métodos GET, POST, PUT e DELETE. No caso deste trabalho é necessária a utilização do método GET para obter informação no cliente, a aplicação *web*, e neste caso concreto o resultado das recomendações requisitadas.

Para o uso do *Tastypie* [38], foi necessário criar um ficheiro onde estivesse explícito qual o serviço que se devia encontrar disponível para a aplicação *desktop*. Esse ficheiro é o “resources.py”, no qual se encontra definido o serviço ao qual a aplicação (cliente) irá ter acesso, na realização de recomendações com base em contexto.

Para usar estes serviços, apenas é necessário usar o *host* público do servidor, seguido de “/api/v1/” e o nome do serviço, definido pelo “resource_name” presente no ficheiro “resources.py”. É preciso ter em atenção que é necessário concatenar no fim da *string* o formato da resposta obtida, sendo o formato usado neste trabalho o json (“?format=json”). O nome do *host* é um nome de domínio criado especificamente para ser possível aceder publicamente ao serviço fornecido por esta API sem necessidade de se encontrar na mesma rede em que o servidor se encontra.

É possível no serviço definir quais os campos a serem obtidos como resposta ou definir que campos podem ser usados para filtragem de resultados.

4.2. Cliente

Esta seção visa abordar o processo de desenvolvimento do cliente de testes que permite testar as funcionalidades do módulo de recomendações com base em contexto, explicando qual o intuito da implementação do mesmo, bem como as suas limitações e a forma como estas foram contornadas na realização dos testes necessários à validação.

A ideia da implementação deste cliente surge como uma maneira de possibilitar a utilização das funcionalidades do módulo desenvolvido do lado servidor de uma forma mais *user-friendly*. O foco deste trabalho encontra-se todo do lado do servidor, e o módulo poderia ser testado com base em técnicas de *scripting* de forma a realizar uma quantidade de testes consideráveis sem necessidade de estar constantemente a interagir com o módulo, o que acontece. No entanto por opção foi decido paralelamente criar um formulário *web* que permitisse introduzir os *inputs* necessários à realização de recomendações por parte do módulo desenvolvido. Em suma trata-se de uma pequena aplicação *web* desenvolvida na linguagem Python recorrendo à *framework* Django [33] que serve como prova de conceito do trabalho realizado e que qualquer utilizador comum poderia usar para experimentar o módulo implementado.

O utilizador ao aceder à aplicação *web* visualiza um formulário onde pode requisitar a sua recomendação (Figura 4.4).

Pervasive Tourism POI's Recommendation

User Location:

 Distance weight

POI Category:

Choose POI category

Radius:

Choose radius

Manual context variables:

Mean of Transport OFF Companion OFF

Automatic context variables:

Weekday ON Time Day ON Season OFF Weather ON Temperature ONWeekday weight Time Day weight Season weight Weather weight Temperature weight

Reset

Get Recommendation

Figura 4.4: Formulário web de requisição de recomendação com base em contexto

Através deste formulário o utilizador introduz os *inputs* necessários à realização da recomendação podendo depois submeter o pedido. O utilizador começa por introduzir uma localização geográfica (que deverá estar dentro das fronteiras consideradas para este trabalho, pois o conjunto de dados é referente à cidade de São Francisco nos E.U.A.), tendo em seguida que definir a categoria relativa aos pontos de interesse turístico que pretende considerar, e o raio de ação a considerar para a realização da recomendação. Estes parâmetros são de carácter obrigatório. O utilizador pode também optar por considerar associar contexto à recomendação que pretende obter, sendo esta opção altamente configurável. Para cada condição de contexto contemplada pelo modelo implementado o utilizador pode optar por a considerar ou não, na elaboração da recomendação através da ativação ou desativação dessas mesmas condições. No caso de o utilizador considerar associar contexto na sua recomendação deverá colocar as condições com o estado “ON” e em seguida definir um peso a atribuir a cada uma dessas condições de contexto ativas. Este peso deverá assumir valores entre 0 e 1 sendo que a soma de todos os pesos atribuídos nunca deverá ser superior a 1.

Posto isto bastará ao utilizador submeter o pedido de recomendação através do uso do botão “Get Recommendation”, e o cliente comunicará com o servidor no intuito de requisitar a recomendação pretendida, aguardando pela resposta por parte do servidor. Após este passo o utilizador será reencaminhado para uma página *web* onde será apresentado um documento em formato JSON e que corresponderá aos *itens* recomendados pelo módulo de recomendação de acordo com o modelo implementado. Estes *itens*

encontram-se organizados por relevância (de acordo com o algoritmo implementado) sendo que os apresentados primeiramente terão uma relevância superior aos que se encontram mais abaixo na lista de *itens* devolvidos. Estes *itens* corresponderão aos pontos de interesse turísticos recomendados para um determinado cenário de *inputs* introduzidos, e será possível visualizar alguma informação relativa aos mesmos como demonstra a figura 4.5.

```
{
  - meta: {
    limit: 1000,
    next: null,
    offset: 0,
    previous: null,
    total_count: 6
  },
  - objects: [
    - {
      poi_address: "609 Market St San Francisco 94105 CA United States US",
      poi_category: "food",
      poi_distance: "0",
      poi_id: "10",
      poi_location: "(37.7890988345,-122.401857376)",
      poi_name: "McDonald's",
      poi_phone: "4155432595",
      poi_score: "1.4",
      poi_subcategory: "Fast Food"
    },
    - {
      poi_address: "598 Market St San Francisco 94104 CA United States US",
      poi_category: "food",
      poi_distance: "12",
      poi_id: "1861",
      poi_location: "(37.7891905057,-122.401842423)",
      poi_name: "Peet's Coffee & Tea",
      poi_phone: "4159706615",
      poi_score: "1.34",
      poi_subcategory: "Coffee Shop"
    },
    - {
      poi_address: "1 Post St San Francisco 94104 CA United States US",
      poi_category: "food",
      poi_distance: "35",
      poi_id: "371",
      poi_location: "(37.78903,-122.402169)",
      poi_name: "Specialty's Cafe & Bakery",
      poi_phone: "8775022837",
      poi_score: "0.89",
      poi_subcategory: "Bakery"
    }
  ]
}
```

Figura 4.5: Resposta a pedido de recomendação em formato JSON

Para a validação do modelo de recomendações implementado é necessário realizar um conjunto de testes bastante abrangente, e assim sendo esta aplicação não se adequa às necessidades para a realização dos mesmos. Por esse motivo optou-se por uma solução de *scripting* que permitisse testar o modelo implementado de uma forma exaustiva em grande escala no que toca ao número de recomendações efetuadas. O *script* foi desenvolvido de forma a realizar um conjunto de testes pré-estabelecidos ao modelo de recomendações implementado, recorrendo ao consumo do *web service* de recomendações que se encontra do lado do servidor. Toda a metodologia relativa à implementação deste *script*, e aos cenários de teste que o mesmo contempla será abordado de forma introdutória no capítulo 5, que trata a análise de resultados obtidos.

Capítulo 5

Discussão dos Resultados

Neste capítulo irão ser abrangidos os diferentes processos levados a cabo na obtenção de resultados, de forma a validar e afinar o modelo implementado. Numa primeira fase será explicada a metodologia seguida na obtenção dos resultados e em seguida serão apresentados os resultados obtidos.

5.1. Metodologia

O objetivo para a validação e afinação do modelo implementado passa por identificar qual a relevância do contexto associado a cada um dos *itens* (pontos de interesse turístico) devolvidos na aquisição de uma recomendação. Os *itens* devolvidos pelo módulo de recomendação não são mais do que uma lista ordenada por relevância tendo em conta o utilizador que requisitou a recomendação. A relevância para cada *item* é definida pelo módulo de recomendações através de um algoritmo implementado que permite realizar uma avaliação, com base nas condições contextuais consideradas, sobre os *itens* pertencentes à lista de resultados a apresentar ao utilizador que requisita a recomendação. Cada *item* terá associado um *score*, elemento que permite definir qual a posição do *item* na lista ordenada de resultados, sendo que quanto maior este *score* maior será a sua relevância em termos contextuais para o utilizador.

Dado que o *score* de cada *item* é calculado com base nas condições contextuais associadas a cada *item* interessa analisar quais destas são mais ou menos relevantes na realização de recomendações de forma individual e combinadas entre si. No modelo implementado e tendo em conta os dados disponíveis foi possível considerar as seguintes condições de contextos, já mencionadas na secção 4.1.2:

- Geolocalização;
- Categorização de pontos de interesse;
- Dia da semana;
- Altura do dia;
- Estado do tempo;
- Temperatura;
- Estação do ano.

Para cada ponto de interesse existente na base de dados do sistema, existem estas condições de contexto associadas, sendo que os valores atribuídos a cada uma delas (excluindo a geolocalização e a categoria ao qual o ponto pertence) são calculados com base no padrão de visita dos utilizadores que nos pontos de interesse, ou seja quando um

utilizador realiza um *check-in* num determinado ponto de interesse este *check-in* é registado juntamente com os valores de cada condição de contexto calculados pelo algoritmo no momento do *check-in*. Este histórico de *check-ins* dos utilizadores nos pontos de interesse possibilita que seja calculado para cada ponto de interesse os valores dominantes de contexto, ou seja cada ponto de interesse terá definido os valores dominantes para cada condição de contexto, tendo em consideração todas as visitas registadas pelos utilizadores em cada um desses pontos num dado intervalo de tempo.

De forma a avaliar a relevância das condições de contexto na realização de recomendações de forma individual e combinada, irá ser utilizado um método que permitirá concluir que condições contextuais terão mais influência numa dada recomendação. Pretende-se com base nos diferentes pontos de interesse existentes, considerar um *check-in* aleatório para cada um desses pontos e tendo em conta os parâmetros registados nesse *check-in* realizar uma recomendação. Os parâmetros a considerar em cada um dos *check-ins* são: coordenadas do ponto (latitude e longitude), categoria do ponto e os valores registados para as condições de contexto associadas a esse ponto.

5.1.1. Cenários de teste

Para a realização de uma recomendação é necessário um conjunto de *inputs*. Estes *inputs* vão desde a localização do utilizador que pretende realizar a recomendação, o raio de ação geográfico em torno da posição do utilizador, a categoria relativa ao tipo de ponto de interesse turístico que pretende considerar e os valores relativos às condições de contexto registados na altura da realização da recomendação, bem como o peso a atribuir a cada uma destas condições na recomendação pretendida.

Partindo da metodologia descrita no ponto anterior, irão ser realizadas recomendações com base na informação de um *check-in* escolhido aleatoriamente para cada ponto de interesse existente na base de dados do sistema. O sistema contém 4193 pontos de interesse turístico. No entanto por si só a informação dos *check-ins* considerados não será suficiente para preencher todos os requisitos ao nível dos *inputs* necessários para a realização de recomendações, sendo necessário definir qual o raio em torno da suposta posição do utilizador, bem como o peso a atribuir a cada uma das variáveis de contexto.

Ficou definido que para o raio em torno da suposta posição do utilizador seriam considerados três casos: 300 metros, 500 metros e 1000 metros. Para cada um destes casos foi realizado o mesmo conjunto de testes visando, a análise da relevância do contexto associado aos pontos de interesse nas recomendações. Quanto ao peso a atribuir a cada uma das condições de contexto e quais destas considerar numa recomendação foram gerados os seguintes cenários de teste:

- **Cenário 1:** considerar apenas uma condição de contexto com um peso de 100%. Perante este cenário, e tendo em conta as condições de contexto consideradas, é possível realizar cinco testes distintos, sendo que estes correspondem a todas as combinações sem repetição possíveis. Existem cinco condições de contexto para testar, o que dentro deste cenário oferece 5 possibilidades de teste.

- **Cenário 2:** considerar duas condições de contexto com um peso de 50% cada. Perante este cenário e seguindo a lógica do cenário anterior é possível realizar dez testes distintos, sendo que estes testes correspondem a todas as combinações sem repetição possíveis.
- **Cenário 3:** considerar três condições de contexto com um peso de 33.(3)% cada. Perante este cenário e seguindo a lógica dos cenários anterior é possível realizar dez testes distintos, sendo que estes testes correspondem a todas as combinações sem repetição possíveis.
- **Cenário 4:** considerar quatro condições de contexto com um peso de 25% cada. Perante este cenário e seguindo a lógica dos cenários anterior é possível realizar cinco testes distintos, sendo que estes testes correspondem a todas as combinações sem repetição possíveis.
- **Cenário 5:** considerar cinco condições de contexto com um peso de 20% cada, o que corresponde a um teste (uma combinação possível sem repetição) e não considerar nenhuma das condições de contexto (0% de influência para cada uma das condições de contexto) que corresponde a um teste (uma combinação possível sem repetição).

Os testes realizados incluem cada um destes cenários de *inputs* descritos, sendo que em suma se traduzem no seguinte:

- Para cada ponto de interesse armazenado na base de dados escolher aleatoriamente um *check-in* em cada um desses pontos e utilizar a informação como *input* da recomendação, e em seguida realizar recomendação:
 - Para um raio de 300 metros em torno da localização:
 - De acordo com Cenário 1
 - De acordo com Cenário 2
 - De acordo com Cenário 3
 - De acordo com Cenário 4
 - De acordo com Cenário 5
 - Para um raio de 500 metros em torno da localização:
 - De acordo com Cenário 1
 - De acordo com Cenário 2
 - De acordo com Cenário 3
 - De acordo com Cenário 4
 - De acordo com Cenário 5
 - Para um raio de 1000 metros em torno da localização:
 - De acordo com Cenário 1
 - De acordo com Cenário 2
 - De acordo com Cenário 3
 - De acordo com Cenário 4
 - De acordo com Cenário 5

O conjunto de testes realizado tem em consideração todos os pontos de interesse existentes na base de dados do sistema, sendo que para cada um dos cenários descritos são realizadas recomendações sucessivas para cada um dos casos a analisar. Por exemplo para um raio de 300 metros irão ser realizadas as mesmas recomendações tantas vezes quantos os casos tidos em conta em cada cenário considerado. Para um raio de 300 metros e considerando o cenário 1, os mesmos testes irão ser realizado 5 vezes pois esta quantidade corresponde ao número de casos existentes naquele cenário, ou seja irão ser realizadas 4193 recomendações (numero de pontos de interesse existente) 5 vezes onde apenas varia a configuração relativa ao peso das condições de contexto consideradas.

5.1.2. Resultado dos testes

O *output* relativo ao conjunto de testes realizado terá uma base estatística para que permita perceber se a associação de contexto nas recomendações traz melhorias na realização de recomendações, e quais das condições de contexto consideradas mostram ser mais relevantes na realização das recomendações.

Na realização destes testes o que se sabe é que os *inputs* utilizados nas recomendações correspondem de facto a *check-ins* de utilizadores nos pontos de interesse. A partir desta situação pretende-se supor que o utilizador na altura em que tomou a decisão de visitar aquele ponto teria recorrido a este sistema (daí a consideração dos valores de contexto registados na altura do *check-in* do utilizador nesse ponto), e que na base da sua decisão estaria o facto de aquele mesmo ponto ter sido sugerido na recomendação com uma avaliação bastante superior aos restantes da mesma categoria que se encontram por perto, ou seja teria aparecido na lista ordenada como um dos primeiros *itens*.

Sendo esta suposição a base para a realização dos testes, pretende-se que para cada um dos raios e cenários considerados seja possível saber em que posição da lista ordenada da recomendação surge o ponto de interesse para o qual o utilizador em condições de contexto similares realizou uma visita a esse ponto. Sabendo a posição do *item* na lista de *itens* da recomendação é possível calcular um erro que dirá respeito à posição que o *item* ocupa na lista.

Para o cálculo deste erro, assume-se que se o *item* (ponto de interesse) estiver em primeiro lugar na lista de recomendações será de 0% enquanto se este se encontrar em último lugar da lista o erro será de 100%. Assim sendo o cálculo do erro para posições da lista facultada na recomendação será dado pela seguinte fórmula:

$$\text{Erro} = \left(\frac{1}{(n - 1)} \times \text{posição} \right) \times 100\% , \text{posição} \in [0, n - 1],$$

onde n corresponde ao número de erros possíveis e *posição* corresponde ao índice do ponto de interesse na lista de itens devolvidos na recomendação.

Este erro será o output essencial para a avaliação da relevância do contexto associado às recomendações. Este é calculado em todos os testes realizados. Para cada caso de teste em

cada cenário considerado é calculada uma média deste erro onde todos os erros calculados para cada ponto de interesse são considerados. Será com base nesta média de erro que irão ser tiradas as conclusões sobre os testes efetuados.

5.2. Resultados

Considerando o conjunto de testes efetuados e tendo em conta as condições e cenários estipulados para a realização dos mesmos obtiveram-se os seguintes resultados:

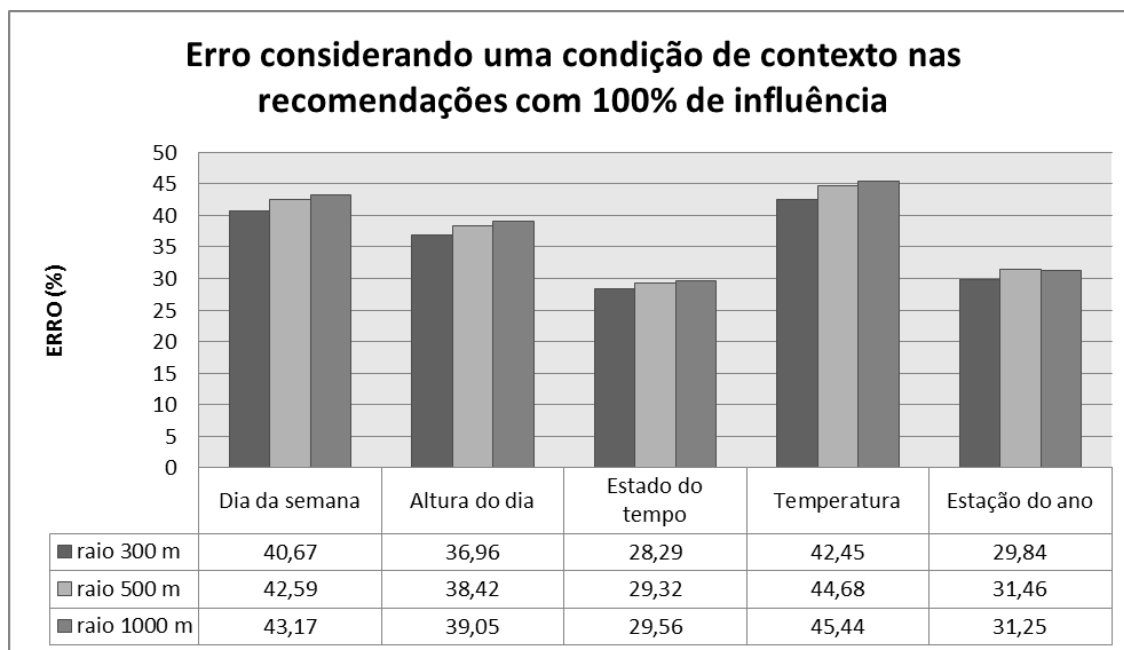


Figura 5.1: Gráfico relativo à média de erro calculada para testes com uma condição de contexto

Através da análise do gráfico da figura 5.1 é possível constatar que num cenário em que apenas uma condição contextual é tida em conta na obtenção de recomendações, existem diferenças no impacto que estas têm na realização de recomendações. Logo existem condições que apresentam mais relevância do que outras para a realização de recomendações. Através da análise direta do gráfico 5.1 é possível constatar que a condição contextual que garante um menor erro médio na obtenção de recomendações é o “Estado do tempo”, seguida da “Estação do ano”, “Altura do dia”, “Dia da semana” e “Temperatura”.

Logo será lógico no que toca ao algoritmo de recomendações implementado atribuir pesos diferentes a cada uma das condições de contexto consideradas, pois assim garante-se que seja dada mais importância às condições que realmente tem mais impacto nos resultados das recomendações apresentados. Assim sendo serve esta análise para diferenciar a importância das condições contextuais dentro do conjunto considerado, validando assim a questão relativa ao peso a atribuir a cada uma das condições contextuais.

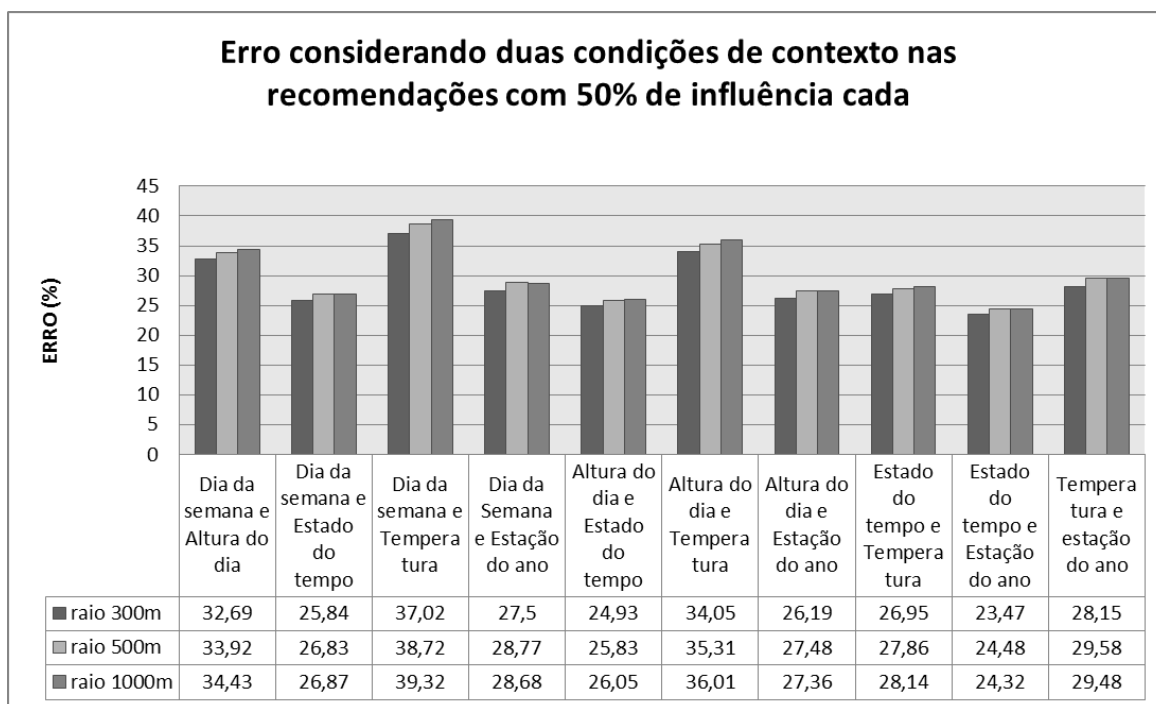


Figura 5.2: Gráfico relativo à média de erro calculada para testes com duas condições de contexto

Após verificar a relevância de cada condição de contexto nas recomendações de forma individualizada (figura 5.1), seria interessante verificar se a conjugação de múltiplas condições de contexto na realização de recomendações afetaria os resultados obtidos fazendo com que o erro médio diminuísse. Pela análise do gráfico da figura 5.2 é possível constatar que para duas condições de contexto é possível gerar 10 conjuntos diferentes tendo em conta as condições de contexto consideradas no modelo implementado. Se cruzarmos a informação extraída do gráfico da figura 5.1 com o gráfico da figura 5.2, é fácil concluir que os conjuntos formados pelas condições contextuais mais relevantes apresentam os melhores resultados sendo que o erro médio é inferior aos restantes. Por exemplo, já tinha sido concluído que o “Estado do tempo” e a “Estação do ano” são as condições de contexto mais relevantes do modelo implementado, e se analisarmos o gráfico da figura 5.2 constata-se que o conjunto formado por essas duas condições contextuais é o que apresenta melhores resultados. De notar também que combinando condições contextuais obtém-se uma melhoria percentual ao nível dos valores registados para o erro médio. Por exemplo se repararmos no melhor caso que remete à combinação das condições mais relevantes, “Estado do tempo” e a “Estação do ano”, obteve-se um valor para o erro médio de cerca de 24% (fazendo a média para os testes relativos aos três raios considerados). Analisando os casos relativos ao gráfico da figura 5.1 conclui-se que tanto para o caso relativo à consideração da condição “Estado do tempo” como da condição “Estação do ano” os valores registados para o erro médio são piores, ficando na ordem dos 29/30% (em média para os testes relativo aos três raios considerados).

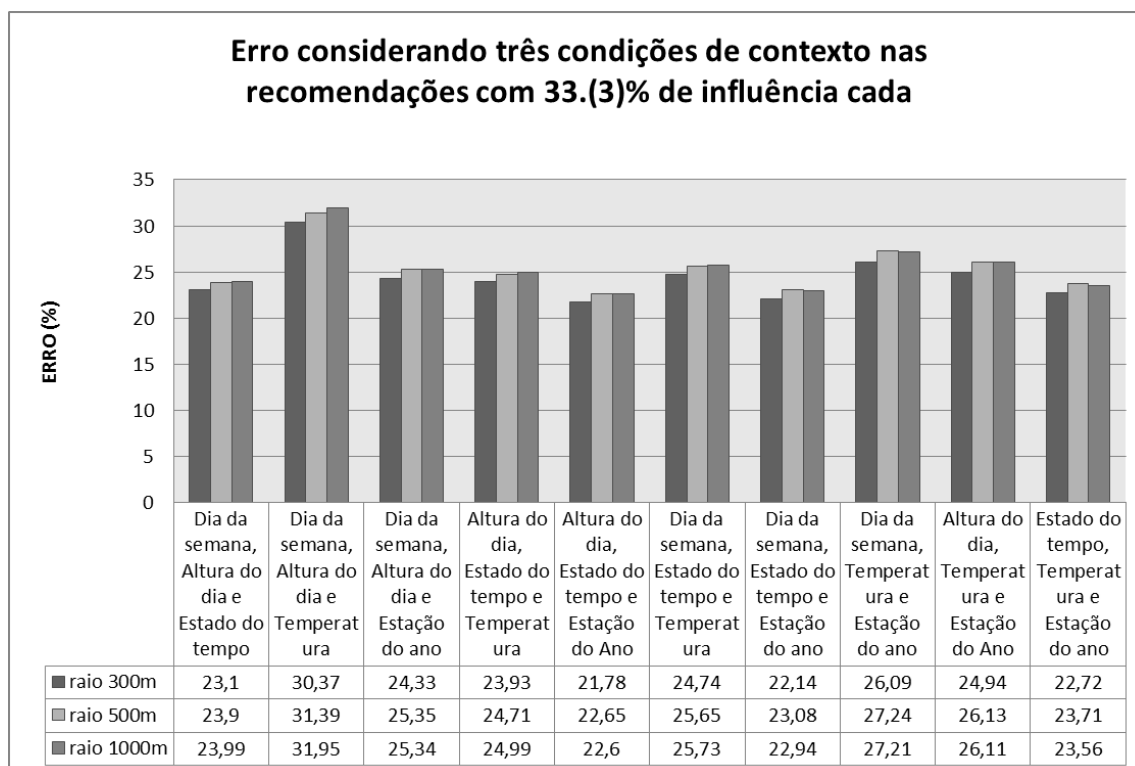


Figura 5.3: Gráfico relativo à média de erro calculada para testes com três condições de contexto

Segundo o modelo implementado, na maioria dos cenários considerados quantas mais condições de contexto forem consideradas na recomendação menor será o erro associado aos resultados obtidos. O gráfico da figura 5.3 vem mostrar isso mesmo, onde é possível constatar uma evolução em termos de resultados para os erros registados, sendo que os conjuntos de condições contextuais que apresentam melhores resultados são os que utilizam as condições mais relevantes como já foi descrito anteriormente. No entanto é também possível constatar através da análise do gráfico da figura 5.2 que em alguns casos a consideração de apenas duas condições de contexto consegue obter melhores resultados que a consideração de três condições. Isso deve-se ao facto de todas as condições terem relevâncias diferentes nas recomendações como foi justificado para o gráfico da figura 5.1, no entanto se forem comparados cenários onde existe pelo menos uma condição em comum, conclui-se que existe uma melhoria nos resultados.

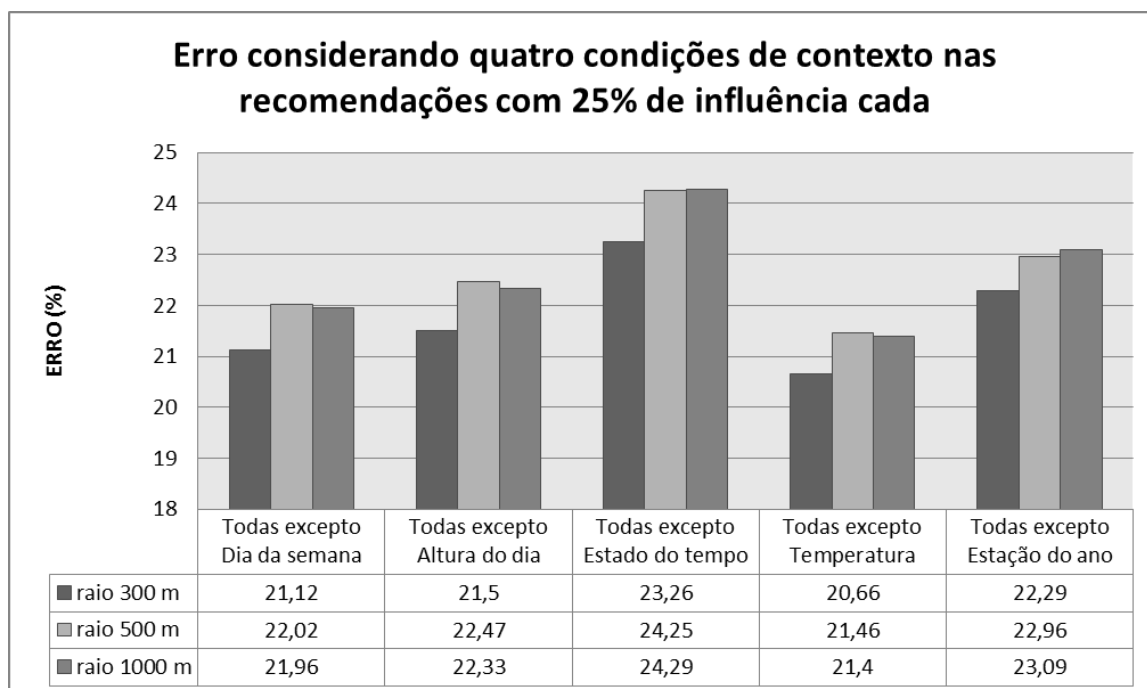


Figura 5.4: Gráfico relativo à média de erro calculada para testes com todas as condições de contexto excepto uma

No gráfico da figura 5.4 é possível retirar a mesma conclusão que do gráfico da figura 5.3, sendo que apenas vem reforçar o facto de que com o acréscimo de mais uma condição de contexto na realização de recomendações, o erro médio desce de novo, e como seria de esperar o conjunto que apresenta piores resultados trata-se do conjunto cuja condição mais relevante não se encontra presente, o “Estado do tempo”, e em contraponto o conjunto que apresenta os melhores resultados é o que exclui a condição de contexto menos relevante no modelo implementado, a “Temperatura”.

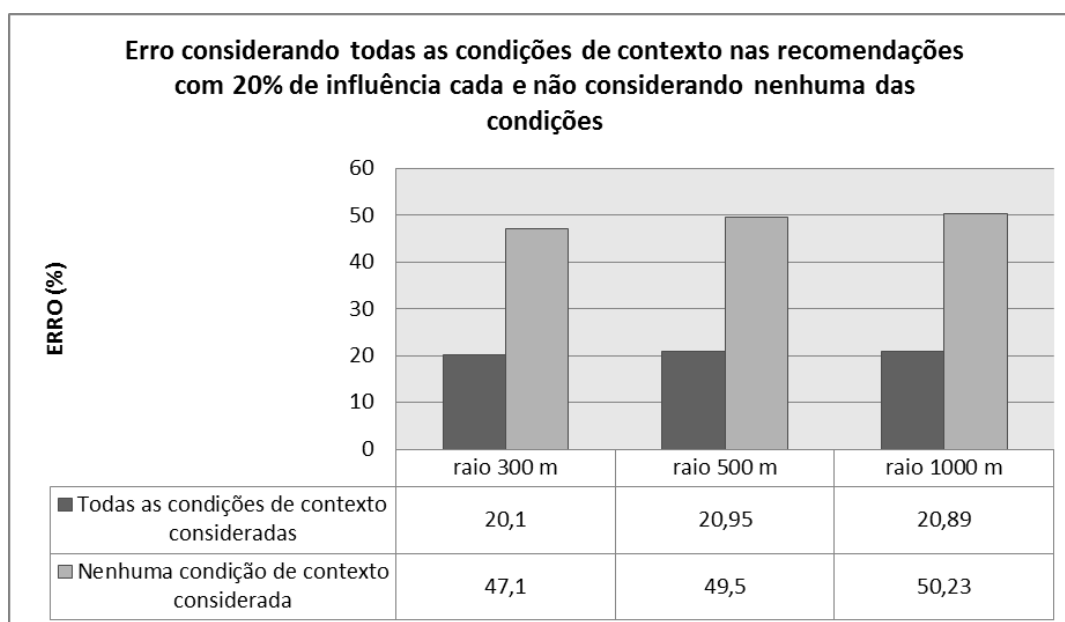


Figura 5.5: Gráfico relativo à média de erro calculada para testes com todas as condições de contexto e sem nenhuma condição de contexto

Por fim temos o gráfico da figura 5.5 onde foi considerada a utilização de todas as condições de contexto na realização de recomendações, e em contraponto a não utilização das mesmas. No caso da não consideração de condições contextuais na realização de recomendações o que se obtém como resultado das recomendações trata-se apenas de um conjunto de pontos não ordenado, que simplesmente preenche os requisitos diretos de filtragem do algoritmo implementado, como o caso da categoria dos pontos considerados e a localização dos mesmos com vista a verificar se estes se encontram no raio de ação pretendido, sendo que não existe contexto associado aos resultados. Neste caso o erro médio verificado regista valores aproximados de 50%. Esta trata-se de uma solução tradicional sem recurso a contexto, e como é possível constatar trata-se do cenário que apresenta piores resultados.

No caso da consideração de todas as variáveis de contexto usadas no modelo implementado na realização de recomendações verifica-se que os resultados obtidos são muito melhores, onde o erro médio ronda os 21%. Em suma os resultados obtidos vêm comprovar que a associação de contexto na realização de recomendações vem acrescentar valor na realização de recomendações para o modelo implementado apresentando um erro médio de 21% quando todas as variáveis de contexto consideradas são usadas. Foi também possível estabelecer a relevância entre as condições de contexto consideradas no que toca a realização de recomendações através do modelo implementado, permitindo assim numa atribuir pesos distintos a cada uma das condições para que as mais relevantes tenham um impacto maior na realização de recomendações. A análise direta do gráfico da imagem 5.1 mostra bem qual a ordem de relevância dentro das condições de contexto consideradas.

De frisar também que existiria mais uma condição de contexto para a qual o sistema se encontra preparado, a distância do utilizador aos pontos recomendados. No entanto torna-se, dada a natureza do conjunto de dados existentes, impossível testar a relevância desta condição no modelo implementado.

Conclusões e trabalho futuro

6.1. Conclusão

Este trabalho consiste no desenvolvimento de um módulo para realizar recomendações com base em contexto de pontos de interesse turístico. Durante a especificação do problema e levantamento de requisito conclui-se que para a implementação do modelo não seria possível seguir uma abordagem de filtragem colaborativa porque estes sistemas baseiam-se em dados históricos (*ratings*) de utilizadores que não se encontravam disponíveis. Por isso optou-se pela possibilidade de os sistemas de recomendação considerarem contexto nos seus modelos.

Este trabalho partiu da solução apresentada em [23], que se baseia numa abordagem de filtragem colaborativa. Nesta solução para o levantamento dos dados históricos foram realizados inquéritos a um conjunto de pessoas para que estas pudessem avaliar visitas a pontos de interesse turístico numa determinada situação contextual, e com base nos resultados recolhidos conseguir realizar recomendações orientadas ao contexto.

Ao contrário da solução apresentada em [23] optou-se por fazer um levantamento estatístico das condições contextuais registadas no momento das visitas dos utilizadores aos pontos de interesse turístico. Este levantamento estatístico permite encontrar um padrão associado ao comportamento dos utilizadores em função das condições de contexto.

Este método permite que o contexto possa estar associado às recomendações de uma forma similar à solução base original, sendo que em alguns aspetos apresenta mais-valias. O facto de informação de contexto ser obtida através de um método estatístico sabe-se que ela é fiel ao comportamento contextual dos utilizadores, enquanto na solução original o preenchimento prévio de um inquérito que visa definir o contexto para os pontos de interesse pode ser ambíguo, isto porque muitas vezes o que as pessoas dizem ser as suas preferências não se concretiza num cenário real. Outra vantagem relativamente ao método escolhido é o facto de se partir de um conjunto de dados, que apesar de ser mais pobre em relação à informação que se pode extrair do inquérito, permite obter uma representatividade superior. O processo de recolha de avaliações contextuais dos pontos de interesse através de inquéritos é um processo mais moroso, enquanto na solução proposta parte-se de conjunto de dados grande para a realização do levantamento estatístico contextual.

A solução original [23] e a implementada neste trabalho visavam a implementação de algo similar, mas seguindo abordagens distintas. Seria interessante poder comparar em termos de resultados obtidos as duas soluções de forma a perceber qual a melhor, no entanto ambas visam objetivos diferentes em termos de resultados. A solução original visa

responder à questão relativa à preferência por parte dos utilizadores perante o cenário de utilização de sistemas de recomendação em que num caso o contexto é considerado e noutro não. Na solução do trabalho desenvolvido o que se pretendia era avaliar a relevância de cada condição contextual considerada na realização de recomendações e como é que estas trazem melhorias na realização das mesmas. Foi possível identificar quais as condições de contexto, das consideradas, mais relevantes na recomendação de pontos de interesse turísticos, assim como quais as diferenças registadas quando existe consideração de contexto e o oposto na realização de recomendações, sendo que os resultados mostram que o contexto refina os resultados obtidos.

Outra questão que a solução desenvolvida mostra poder resolver, tem a ver com a problemática da falta de dados numa fase inicial em que um sistema de recomendação é posto em funcionamento, pois esta prejudica a qualidade das suas recomendações. A solução desenvolvida assenta numa abordagem orientada ao conhecimento, ou seja com base num conjunto de dados que não incluem informação sobre os utilizadores. Os utilizadores poderão assim numa fase inicial usar este sistema de recomendação e o sistema está preparado para recolher informação adicional que possibilitará numa fase posterior implementar uma solução de filtragem colaborativa.

Para além do objetivo principal, ao longo do desenvolvimento deste trabalho foram adquiridos inúmeros conhecimentos que eram totalmente ou parcialmente desconhecidos, como o caso da linguagem Python, que acabou por ser uma mais-valia para este trabalho. O mesmo se aplica à framework Django e outras associadas, como a Tastypie totalmente desconhecidas, mas que permitiram implementar quase todo este sistema. Os conhecimentos já existentes relativamente a *web services* foram também melhorados ao longo deste trabalho.

Por fim de ressaltar todos os conhecimentos adquiridos sobre as temáticas do contexto, orientação ao contexto e sistemas de recomendação orientados ou não a contexto ao longo do desenvolvimento deste trabalho, visto que todas estas temáticas se tratavam de matérias praticamente desconhecidas previamente.

6.2. Trabalho Futuro

Um sistema de recomendação é tanto mais poderoso quanto a diversidade e quantidade de dados que utiliza e relaciona na realização de recomendações. Por este fator é possível concluir que o tipo de dados e a quantidade relativa aos mesmos é um ponto crucial no desenvolvimento, teste e por fim sucesso na realização das funções inerentes a um sistema de recomendação.

O grande desafio enfrentado no desenvolvimento deste trabalho relacionou-se inteiramente com o tipo e quantidade de dados disponibilizados. Seguindo uma abordagem na implementação de um sistema de recomendação, como explicado na secção 2.2.1, é fácil de notar que a realização de uma recomendação baseia-se em três componentes chave, o utilizador, o *item* (que se traduz naquilo que poderá ser recomendado ao utilizador) e o contexto associado. Todos os dados que possam ser associados a estas três componentes de

forma coerente na realização de recomendações, tornam as mesmas mais personalizadas para um determinado utilizador dentro de um determinado contexto.

O que aconteceu ao longo deste trabalho foi que os dados coletados não permitiram que alguns dos objetivos iniciais pudessem ser concretizados, o que por si só limitou as capacidades do módulo de recomendações implementado. Exemplos desta situação são a ausência de dados de perfil relativos aos utilizadores que realizaram *check-ins* nos pontos de interesse. A disponibilização desta informação (que não acontece devido à política de privacidade da entidade que disponibilizou os dados, neste caso específico, o *Foursquare*) poderia enriquecer a objetividade do sistema de recomendação, permitindo que dados relativos aos utilizadores possam ter um peso contextual na realização das recomendações, bem como permitir a comparação entre utilizadores de modo a que se possa perceber a semelhança entre os mesmos e utilizar essa informação de forma significativa na realização de recomendações.

Perante este cenário, e como trabalho futuro seria interessante o desenvolvimento de um trabalho que visasse a criação de um repositório de dados onde existam dados de *check-ins* de utilizadores em pontos de interesse mas que para além disso, estes mesmos *check-ins* estejam associados a utilizadores e que dados sobre os perfis destes, como por exemplo género, interesses no âmbito turístico, idade e avaliação dos diferentes pontos de interesse já visitados, existissem associados aos utilizadores. O enriquecimento ao nível dos dados de utilizador permitirá planear uma melhoria do algoritmo de recomendações proposto.

Referências

- [1] T. P. Moran and P. Dourish, "Introduction to this special issue on context-aware computing," *Human-Computer Interact.*, vol. 16, pp. 87–95, 2001.
- [2] S. Perugini, M. Gonçalves, and E. Fox, "Recommender systems research: A connection-centric survey," *Journal of Intelligent Information Systems*, vol. 23, pp. 107–143, 2004.
- [3] B. N. Schilit, N. York, and M. M. Theimer, "Disseminating Active Map Information to Mobile Hosts 1 Introduction," *IEEE Network*, vol. 8, pp. 22-32, 1994.
- [4] P. J. Brown, J. D. Bovey, and X. C. X. Chen, "Context-aware applications: from the laboratory to the marketplace," *IEEE Personal Communications*, vol. 4, pp. 22 - 32, 1997.
- [5] N. Ryan, J. Pascoe, and D. Morse, "Enhanced Reality Fieldwork : the Context Aware Archaeological Assistant," *Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 25th Anniversary Conference, University of Birmingham*, vol. 4, pp. 269 - 274, 1999.
- [6] A. K. Dey, G. D. Abowd, and A. Wood, "CyberDesk: a framework for providing self-integrating context-aware services," *Knowledge-Based Systems*, vol. 11, no. 1, pp. 3–13, 1998.
- [7] P. J. Brown, "The stick-e document : a framework for creating context-aware applications," *In Proceedings of EP'96, Palo Alto*, vol. 8, pp. 259–272, 1996.
- [8] D. Franklin and J. Flaschbart, "All gadget and no representation makes jack a dull environment," *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environment*, pp. 155–150, 1998.
- [9] B. N. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," *In Proceedings of 1st International Workshop on Mobile Computing Systems and Applications*, pp. 85–90, 1994.
- [10] S. Jang, E. Ko, and W. Woo, "Unified User-Centric Context : Who , Where , When ,," *UbiPCMM: Personalized Context Modelling and Management for Ubicomp Applications*, 2005.
- [11] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," *Technical Report. Dartmouth College*, pp. 1–16, 2000.
- [12] S. R. Hagen P., Manning H., "Smart Personalization," *Cambridge, MA: Forrester Research*, 1999.

- [13] F. Ricci, "Mobile Recommender Systems," *Information Technology & Tourism.*, vol. 12, no. 3, pp. 205–231, 2010.
- [14] R. Burke, "Hybrid Web Recommender Systems," *In The adaptive web Lecture Notes In Computer Science*, vol. 4321, pp. 377 – 408, 2007.
- [15] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.
- [16] D. Billsus and M. J. Pazzani, "Adaptive News Access," *In The adaptive web Lecture Notes In Computer Science*, vol. 4321, pp. 550–570, 2007.
- [17] G. Adomavicius and A. Tuzhilin, "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach University of Minnesota University of Connecticut," vol. 23, no. 1, pp. 103–145, 2005.
- [18] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of Library and Information Systems*, vol. 69, no. 32, pp. 175–186, 2000.
- [19] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.
- [20] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," *In Proceedings of the 2008 ACM conference on Recommender systems RecSys 08*, vol. 16, p. 335-336, 2008.
- [21] S. S. Anand and B. Mobasher, "Contextual Recommendation," *From Web to Social Web: Discovering and Deploying User and Content Profiles*, pp. 142–160, 2007.
- [22] L. Baltrunas, B. Ludwig, and F. Ricci, "Context relevance assessment for recommender systems," *Proc. 15th Int. Conf. Intell. user interfaces*, pp. 287–290, 2011.
- [23] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci, "Context-Aware Places of Interest Recommendations for Mobile Users," *Design, User Experience, and Usability. Theory, Methods, Tools and Practice Lecture Notes in Computer Science*, vol. 6769, pp. 531–540, 2011.
- [24] M. Van Setten, S. Pokraev, and J. Koolwaaij, "Context-aware recommendations in the mobile tourist application COMPASS," *Adaptive hypermedia and adaptive web-based systems*, pp. 515–548, 2004.
- [25] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide : A Mobile Context-Aware Tour Guide," *Baltzer Journals*, vol. 3, no. 5, pp. 421 - 433, 1997.

- [26] K. Cheverst, N. Davies, K. Mitchell, and A. Friday, “Experiences of Developing and Deploying a Context-Aware Tourist Guide : The GUIDE Project,” *In Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00)* pp. 20–31, 2000.
- [27] C. Svizzera, D. Informatica, and U. Torino, “INTRIGUE : Personalized recommendation of tourist attractions for desktop and handset devices,” *Applied Artificial Intelligence, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries*, vol. 17, no. 8-9, pp. 687–714, 2003.
- [28] K. Cheverst, N. Davies, K. Mitchell, a. Friday, C. Efstratiou, I. Amendola, F. Cena, L. Console, a. Crevola, C. Gena, a. Goy, S. Modeo, M. Perrero, I. Torre, and a. Toso, “UbiquiTO: A Multi-device Adaptive Guide,” *Proceedings of the 6th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI'2004)*, vol. 1, no. 2, pp. 538–540, 2004.
- [29] Q. Labs, “Gimbal,” 2012. [Online]. Available: <http://www.gimbal.com/> . [Accessed: 3-November-2012]
- [30] Foursquare, “API Foursquare.” [Online]. Available: <https://developer.foursquare.com/>. [Accessed: 5-January-2013].
- [31] O. W. Map, “API Open Weather Map.” [Online]. Available: <http://openweathermap.org/API>. [Accessed: 7-December-2012].
- [32] P. G. D. Group, “PostgreSQL: The world’s most advanced open source database.” [Online]. Available: <http://www.postgresql.org/>. [Accessed: 14-November-2012].
- [33] D. S. Foundation, “Django.” [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 12-March-2013].
- [34] H. Gao, J. Tang, and H. Liu, “Exploring Social-Historical Ties on Location-Based Social Networks,” *Sixth Int. AAAI Conf. Weblogs Soc. Media*, pp. 114–121, 2012.
- [35] T. P. G. D. Group, “pgAdmin 1.4 online documentation.” [Online]. Available: <http://www.pgadmin.org/docs/1.4/pgagent.html>. [Accessed: 14-November-2012].
- [36] SQLite, “SQLite Home Page.” [Online]. Available: <http://www.sqlite.org/>. [Accessed: 9-July-2013].
- [37] O. Corporation, “MySQL :: The world’s most popular open source database.” [Online]. Available: <http://www.mysql.com/>. Accessed: 09-July-2013].
- [38] D. Lindsley, “Tastypie - RESTful APIs for Django.” [Online]. Available: <http://django-tastypie.readthedocs.org/en/latest/>. [Accessed: 25-July-2013].

- [39] P. G. D. Group, “PostGIS: Spatial and Geographic objects for PostgreSQL” [Online]. Available: <http://postgis.net/>. [Accessed: 14-November-2012].
- [40] GiST, “The GiST Indexing Project” [Online]. Available: <http://gist.cs.berkeley.edu/>. [Accessed: 14-November-2012].