

C •

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

HappyHour Emotion

Sistema de caracterização de pontos de interesse

Dissertação para a obtenção do Grau de Mestre em Engenharia Informática

Autor: Pedro Carmona (pcarmona@student.dei.uc.pt)

Júri

Orientador: Jorge Miguel Sá Silva (sasilva@dei.uc.pt)
Orientador: David Nunes (dsnunes@dei.uc.pt)
Júri Vogal: Amílcar Cardoso (amilcar@dei.uc.pt)
Júri Arguente: Luís Filipe Vieira Cordeiro (cordeiro@dei.uc.pt)

2 de Setembro de 2014

RESUMO

Ao longo desta dissertação é exposto o estudo realizado para o desenvolvimento do projeto *HappyHour*, que engloba três componentes físicas: um servidor Web, um servidor REST e uma aplicação móvel. Hoje em dia, existe uma pluralidade de sensores que tornam os *smartphones* propícios ao desenvolvimento de aplicações que interpretam o contexto do utilizador. O *HappyHour* era um sistema que caracterizava pontos de interesse. Com o trabalho realizado nesta tese foi possível estender as funcionalidades para caracterizar o utilizador com base na sua emoção.

É documentado um estudo comparativo com aplicações concorrentes onde se destaca o *HappyHour* por enveredar por um caminho próprio. Esta aplicação procura ligar o contexto da localização do utilizador ao seu contexto emocional.

O modelo utilizado para desenvolver a aplicação assenta na contextualização emocional do utilizador. Em particular, procura-se identificar a euforia, o aborrecimento, a ansiedade e o estado de calma. Desenvolveu-se uma prova de conceito para identificar as emoções através de um mecanismo de *machine learning* supervisionado. Após ser identificada a emoção, o utilizador é notificado para corrigir a informação. De seguida, utiliza-se o contexto emocional para implementar um determinado mecanismo que possa influenciar positivamente o seu estado emocional. Pretende-se que o utilizador se distancie de estados negativos. Para além disto, são utilizadas tecnologias inovadoras como o *MultipathTCP*.

A aplicação Web é direcionada para os gestores de pontos de interesse e permite notificar pessoas com base na sua localização e emoção.

A utilização do contexto emocional aliado à sua localização poderá impulsionar as aplicações do contexto móvel a entregar informações mais relevantes para o utilizador, e adaptarem-se para melhorarem o seu estado emocional, como se pretende com este projeto.

AGRADECIMENTOS

Ao longo da minha formação acadêmica tive a oportunidade de desenvolver as minhas competências pessoais e profissionais, bem como de conhecer pessoas que rapidamente conquistaram a minha admiração e me transmitiram os seus valiosos conhecimentos. Assim, os meus mais sinceros agradecimentos:

- Aos meus pais e irmãos pela força e pelo apoio incondicional ao longo destes 5 anos de formação universitária.
- À minha namorada Lénia que sempre me apoiou em todas as situações.
- Ao Professor Jorge Sá Silva, meu orientador, e ao Eng. David Nunes, co-orientador, pela disponibilidade e compreensão e pelos conhecimentos essenciais que me transmitiram.
- A todos os meus amigos e colegas, que me brindaram ao longo destes anos com imensa alegria e amizade.

ÍNDICE

RESUMO	v
AGRADECIMENTOS	vii
LISTA DE FIGURAS	xi
LISTA DE TABELAS	xiv
LISTA DE ANEXOS	xv
LISTA DE ABREVIATURAS	xvi
CAPÍTULO	
I. Introdução	1
1.1 Contextualização	1
1.2 Objetivos	4
1.3 Estrutura do relatório	6
1.4 Método e planeamento do trabalho	7
II. Estado da arte	12
2.1 Tecnologias e protocolos	13
2.2 Plataformas semelhantes ao <i>HappyHour</i>	19
2.3 Identificação de música, APIs e bibliotecas	33
2.4 Redes neuronais e bibliotecas	36
2.5 Serviços Web de meteorologia	41

2.6	Sensores de batimento cardíaco	43
2.7	<i>Frameworks</i> Web	44
III. Requisitos		47
3.1	<i>Stakeholders</i>	47
3.2	Atores	48
3.3	Requisitos funcionais	49
3.4	Diagramas de Casos de Uso	65
3.5	Requisitos não funcionais	67
3.6	Restrições técnicas	68
IV. Arquitetura		69
4.1	Visão de produto	69
4.2	Visão de alto nível	71
4.3	Aplicação móvel	73
4.4	Aplicação Web	76
4.5	Servidor REST	77
4.6	Modelo de dados	78
V. Desenvolvimento e testes		82
5.1	Desenvolvimento	82
5.2	Testes	102
VI. Conclusão		109
6.1	<i>HappyHour</i>	109
6.2	Trabalho Futuro	111
6.3	Considerações finais	111
ANEXOS		113
BIBLIOGRAFIA		139

LISTA DE FIGURAS

1.1	Logótipo da aplicação <i>HappyHour</i>	2
1.2	Aplicação móvel <i>HappyHour</i>	3
1.3	Aplicação Web - interface existente	3
1.4	Processo de <i>Human in the Loop</i> do modelo do Eng. David Nunes . .	5
1.5	<i>Backlog</i> do projeto <i>HappyHour</i> com recurso à ferramenta Trello. . .	8
2.1	Processo de controlo no Human-in-the-Loop	17
2.2	O papel do ser humano no processo de controlo nos <i>Human-in-the-Loop Cyber-Physical Systems</i>	18
2.3	Logótipo da aplicação <i>EmotionSense</i>	20
2.4	Ecrã de introdução do <i>EmotionSense</i>	20
2.5	Grelha de emoção do <i>EmotionSense</i>	20
2.6	Formulário do <i>EmotionSense</i>	20
2.7	Logótipo da aplicação NikeRunning	21
2.8	<i>NikeRunning - Heatmap</i> da corrida	22
2.9	Logótipo do <i>DressCode</i>	23
2.10	Ecrã de escolha da cidade na aplicação <i>DressCode</i>	24
2.11	Lista de estabelecimentos noturnos com classificação e descrição na aplicação <i>DressCode</i>	24
2.12	Logótipo do <i>Foursquare</i>	25
2.13	Logótipo do <i>Swarm</i>	25
2.14	<i>Foursquare - Encontrar os melhores locais nas imediações</i>	26
2.15	<i>Swarm - Encontrar amigos nas imediações</i>	26
2.16	Logótipo do <i>Facebook</i>	27
2.17	Aplicação <i>Facebook - Local Search</i>	28
2.18	Logótipo do <i>Waze</i>	29

2.19	<i>Waze</i> - Aplicação móvel que permite reportar eventos nas vias de condução	30
2.20	<i>Waze</i> - Aplicação móvel que assiste o utilizador na condução	30
2.21	Logótipo do <i>Yelp</i>	31
2.22	<i>Yelp</i> é um repositório de estabelecimentos e negócios locais.	32
2.23	<i>Yelp</i> - Aplicação mobile com diretório de serviços.	32
2.24	Logótipo do <i>Shopkick</i>	32
3.1	Diagrama de casos de uso da aplicação móvel.	65
3.2	Diagrama de casos de uso da aplicação web.	66
4.1	Arquitetura de alto nível do sistema existente	71
4.2	Arquitetura de alto nível do sistema em desenvolvimento	71
4.3	Arquitetura de alto nível do sistema existente com serviços externos	72
4.4	Arquitetura de alto nível do sistema em desenvolvimento com serviços externos	72
4.5	Arquitetura da aplicação móvel	75
4.6	Interfaces de rede do <i>smartphone</i> sem <i>MultipathTCP</i>	75
4.7	Interfaces de rede do <i>smartphone</i> com a utilização do <i>MultipathTCP</i>	75
4.8	Arquitetura da aplicação Web	77
4.9	Arquitetura da aplicação <i>Representational State Transfer</i> (REST)	78
4.10	Diagrama de Entidade-Relacionamento da aplicação móvel	80
5.1	Modelo original do estado de fluxo por Csikszentmihalyi	83
5.2	Modelo atual do estado de fluxo por Csikszentmihalyi	84
5.3	Fluxo do sistema baseado na emoção do <i>HappyHour</i>	85
5.4	Notificação de emoção inferida	87
5.5	Corrigir emoção inferida na aplicação móvel.	87
5.6	Contexto das classes na arquitetura da aplicação móvel <i>HappyHour</i>	90
5.7	<i>HappyHour emotional neural network</i>	92
5.8	Perfil de emoções (calma e ansiedade).	96
5.9	Perfil de emoções (euforia e aborrecimento).	96
5.10	Perfil de emoções (calma e ansiedade) num telemóvel com <i>Multipath TCP</i>	97
5.11	Perfil de emoções (euforia e aborrecimento) num telemóvel com <i>Multipath TCP</i>	97

5.12	Lista de músicas identificadas pelo dispositivo.	99
5.13	Aplicação Web - interface nova	101
B.1	Diagrama de Gantt	126
C.1	Diagrama de Gantt do segundo semestre	128
D.1	Diagrama de Gantt que retrata o trabalho realizado segundo semestre	130

LISTA DE TABELAS

2.1	Preço e limite de APIs de identificação de músicas para fins comerciais.	36
2.2	Comparação de algoritmos de <i>machine learning</i> em <i>smartphones</i> . . .	37
2.3	Serviços Web de meteorologia.	42
2.4	Número de repositórios criados no Github em 2013.	46
3.1	<i>Stakeholders</i> - Descrição e Responsabilidades	48
3.2	Parâmetros dos requisitos	49
5.1	Resultado da integração das redes sociais	100
5.2	Testes à interface de reconhecimento de emoções	103
5.3	Testes à interface de reconhecimento de emoções	104
5.4	Primeira fase de testes à rede neuronal (150 registos)	105
5.5	Segunda fase de testes à rede neuronal com 41 registos	106
5.6	Testes à identificação de músicas	107
5.7	Testes à interface de identificação de músicas	107

LISTA DE ANEXOS

A.	Artigo: <i>HappyHour Emotion</i>	115
B.	Diagrama de Gantt (Primeiro Semestre)	125
C.	Diagrama de Gantt (Planeamento do Segundo Semestre)	127
D.	Diagrama de Gantt (Trabalho no Segundo Semestre)	129
E.	Serviços Web	131
F.	Diagrama de Entidade-Relacionamento	135

LISTA DE ABREVIATURAS

ANN	<i>Artificial Neural Networks</i>
API	<i>Application Programming Interface</i>
BCI	<i>Behavior Change Intervention</i>
BN	<i>Bayesian Networks</i>
CPS	<i>Cyber-Physical System</i>
CSRF	<i>Cross Site Request Forgery</i>
DAO	<i>Data access object</i>
ECG	<i>Eletrocardiograma</i>
GA	<i>Genetic Algorithms</i>
HiLCPS	<i>Human-in-the-loop Cyber-Physical System</i>
HiTL	<i>Human in the Loop</i>
HMM	<i>Hidden Markov Models</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
MPTCP	<i>Multipath TCP</i>
MQTT	<i>Message Queue Telemetry Transport</i>
MVC	<i>Model-View-Controller</i>
ORM	<i>Object-Relational Mapping</i>
REST	<i>Representational State Transfer</i>
SVM	<i>Support Vector Machines</i>

TCP *Transmission Control Protocol*

URL *Uniform resource locator*

XML *eXtensible Markup Language*

XSS *Cross Site Scripting attacks*

CAPÍTULO I

Introdução

1.1 Contextualização

A presente dissertação descreve o trabalho desenvolvido no âmbito do Mestrado em Engenharia Informática. Pretendeu-se melhorar e acrescentar novas funcionalidades a um sistema móvel de *'people-centric sensing'*, denominado de *HappyHour*, atualmente em desenvolvimento.

Este projeto contou com a colaboração estreita entre o grupo de investigação do Laboratório de Comunicações e Telemática da Universidade de Coimbra (LCT UC) com a iniciativa Sapo Labs do Grupo Portugal Telecom.

Orientado para a utilização em eventos sociais, no âmbito do entretenimento, a aplicação permitia identificar locais de interesse com afluência. Numa fase inicial, esta aplicação estaria dedicada ao contexto da vida social noturna e disponível no *smartphone*. A aplicação tirava partido das potencialidades do dispositivo eletrónico, nomeadamente dos sensores que permitem adquirir informação sobre cada utilizador para caracterizar o seu meio envolvente.

O trabalho desenvolvido no primeiro semestre adicionou funcionalidades à parte de gestão dos estabelecimentos noturnos, para além de ter adicionado novos tipos de informação ao sistema adquiridos através de redes de sensores sem fios.

Ao longo do segundo semestre, verificou-se uma necessidade de expandir o domínio da aplicação, para além do contexto de diversão noturna, e direcionando-se para uma aplicação com o objetivo de melhorar o estado de espírito do utilizador. Esta mudança de contexto implicou a adição de novos componentes que permitem inferir a emoção do utilizador. Estes componentes foram adicionados com o intuito de idealizar o modelo

[1] proposto pelo doutorando David Nunes no desenvolvimento de um protótipo com um mecanismo de seleção de rede.

1.1.1 O projeto *HappyHour*



Figura 1.1: Logótipo da aplicação *HappyHour*.

No início do projeto, o sistema *HappyHour* tinha sido desenvolvido com o intuito de ajudar o utilizador na escolha do destino de diversão noturna. Utilizava a informação dos sensores dos *smartphones* para criar um serviço que ajudava o utilizador em tempo real. Era um sistema que permitia saber o estado atual dos estabelecimentos noturnos nas imediações.

A aplicação *HappyHour* era instalada nos dispositivos móveis dos utilizadores, informando sobre a localização e tipo dos estabelecimentos de diversão noturna, bem como os seus eventos, lotação e a agitação do seu público. A localização e a agitação eram determinadas através dos sensores de posicionamento (GPS) e movimento (acelerómetro) dos *smartphones* de cada utilizador. Estas informações individuais eram agregadas por um servidor central, o qual as partilhava com a comunidade através de "mapas de calor", apresentados sobre o mapa da aplicação *Android* (ver exemplo na figura 1.2).

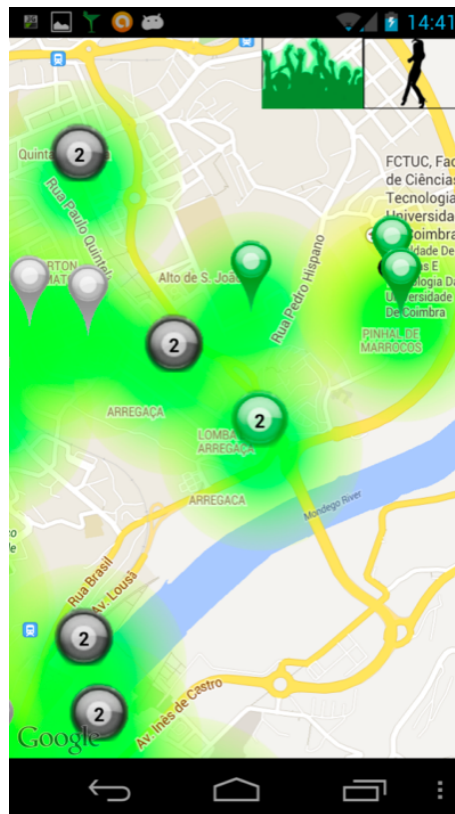


Figura 1.2: Aplicação móvel *HappyHour*.



Figura 1.3: Aplicação Web - interface existente.

O sistema permitia aos estabelecimentos noturnos implementar técnicas de marketing recorrendo a tecnologias inovadoras. Através do agendamento de notificações, os estabelecimentos podiam partilhar informação sobre promoções ou eventos a potenciais clientes que se encontrassem nas proximidades (ver exemplo na figura 1.3).

Era possível definir o conteúdo da notificação (promoções, ofertas, eventos) e o *timing* em que deve ser recepcionada pelo utilizador. Por exemplo, um estabelecimento podia criar uma notificação que informasse todos os utilizadores que se encontrassem num raio de 700 metros em seu redor que as bebidas estavam a 50 por cento de desconto.

O *HappyHour* era constituído por três componentes principais:

- Um sistema central que tratava da gestão dos utilizadores, eventos, notificações e informação sobre estabelecimentos através de uma ligação ao *Foursquare*.
- Uma Aplicação móvel: através da qual o utilizador podia aceder a um mapa para obter informações sobre os pontos de interesse relevantes nas suas imediações.
- Um Interface de gestão Web: utilizada pelos funcionários dos estabelecimentos noturnos para realizar a gestão de eventos e notificações.

O *HappyHour* destacava-se das aplicações sociais móveis orientadas para a localização do utilizador devido a esta dinâmica de partilhar informações de uma forma automática e em tempo real. Esta aproximação contrasta com os típicos *check-ins* colaborativos que requerem a participação ativa e manual do utilizador e que não fornecem a informação atual do ambiente nos estabelecimentos. Por exemplo, a maioria das aplicações deste tipo não permite consultar a afluência a um estabelecimento num determinado momento.

Na próxima secção são apresentados os objetivos do estágio, expondo-se os novos componentes e os desenvolvimentos dos componentes existentes.

1.2 Objetivos

A presente tese de mestrado pretende expor o trabalho desenvolvido no âmbito do projeto *HappyHour*. Em concreto, os objetivos deste trabalho são os seguintes:

- Avaliar a atual solução da aplicação móvel de *people-centric sensing*, *HappyHour*.

- Adquirir conhecimentos neste tema de forma a averiguar as melhores soluções para melhorar os vários componentes.
- Melhorar o sistema *HappyHour*, nomeadamente nos componentes de gestão de pontos de interesse, aquisição de mais informação através de sensores, e melhor caracterização dos ambientes através da identificação automática de música.
- Estender a aplicação do *HappyHour* de uma aplicação focada à diversão noturna, para uma aplicação que permita detetar e melhorar o estado emocional dos seus utilizadores.
- Implementar na aplicação do *HappyHour* um mecanismo de gestão de ligações de rede com recurso à tecnologia *Multipath TCP* [2].
- Escrever um artigo, disponível no anexo A e submetido à conferência EWSN2015, que é uma conferência internacional do tipo A.

A nova aplicação *HappyHour* procura ajudar a pessoa a sentir-se mais pro-ativa a tomar decisões para sair dos estados de ansiedade e aborrecimento. O utilizador pode utilizar o *HappyHour* como um roteiro para melhorar a sua emoção. A deteção de estados emocionais recorrendo a sensores do *smartphone* é uma abordagem que permite desenvolver aplicações que se preocupam com o utilizador, e que processam as informações obtidas deste para melhorar a sua vida quotidiana.

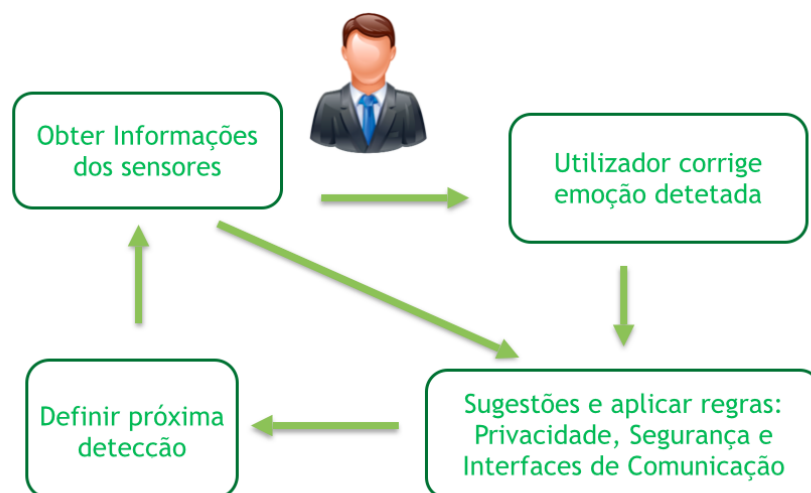


Figura 1.4: Processo de *Human in the Loop* do modelo do Eng. David Nunes

O objetivo científico deste projeto é desenvolver uma aplicação que implemente o modelo proposto pelo Eng. David Nunes (figura 1.4), onde conceitos como ‘*people-centric sensing*’, *Behavior Change Intervention* (BCI) e *Human in the Loop* (HiTL) são postos em prática.

- As aplicações ‘*people-centric sensing*’ procuram conhecer determinadas propriedades do ser humano com base nos sensores;
- As BCI são intervenções e técnicas utilizadas por terapeutas para corrigir comportamentos indesejados tais como fumar, alimentação compulsiva, etc;
- E o conceito HiTL dá relevância ao papel do utilizador no sistema, considerando-o como parte integrante do *loop* de controlo, e consequentemente varia o comportamento do sistema no ambiente.

Neste caso de estudo, pretende-se desenvolver uma aplicação que detete a emoção do utilizador através dos sensores. Este sistema HiTL tem em conta o estado humano (psíquico, físico) no mecanismo de controlo. De seguida, o utilizador deve corrigir a emoção inferida, ao que se segue a implementação de mecanismos que permitam melhorar seu estado emocional. A utilização do estado emocional do utilizador para influenciar a saída do sistema coloca a aplicação como consciente do papel do utilizador no sistema. Os componentes utilizados para a implementação do sistema são os sensores dos telemóveis, a *smart shirt VitalJacket*[®] [3] ou o *BITalino* [4], e ainda um serviço externo denominado *Open Weather Map*.

1.3 Estrutura do relatório

Este relatório encontra-se dividido em 5 capítulos.

No capítulo inicial tem lugar a contextualização do projeto, de forma a enquadrar o projeto *HappyHour* no panorama de tecnologias da atualidade. Neste capítulo, é também apresentado o método de trabalho, assim como os objetivos do projeto.

No segundo capítulo é feita uma introdução acerca das áreas onde o sistema *HappyHour* se enquadra. De seguida, é comparado o *HappyHour* com plataformas semelhantes. Para além disso, analisam-se as diversas opções disponíveis de forma a selecionar as mais adequadas para melhorar o projeto. Nesse sentido, são analisadas bibliotecas

de *machine learning*, serviços Web de reconhecimento de música, serviços Web de informações meteorológicas e ainda *frameworks* Web.

No terceiro capítulo é apresentado o levantamento de requisitos.

No capítulo seguinte caracterizam-se as diversas componentes da aplicação em desenvolvimento.

O quinto capítulo menciona as alterações efetuadas ao projeto. É apresentada uma visão dos diversos módulos implementados e testes que garantem o funcionamento do sistema.

O último capítulo é apresenta um resumo do documento com a conclusão, e apresenta algumas sugestões para trabalho futuro.

1.4 Método e planeamento do trabalho

Tendo em conta o cumprimento dos objetivos estabelecidos, foram definidas metas em conjunto com o Engenheiro David Nunes, no primeiro semestre. Posteriormente, o Professor Jorge Sá Silva reviu e aprovou as tarefas estabelecidas e a respetiva duração. No decorrer do segundo semestre, de forma a enriquecer o estágio, foi alterado o plano de trabalhos. A alteração é exposta na secção 1.4.2. O método de trabalho utilizado ao longo do estágio é descrito na próxima secção.

1.4.1 Metodologia

A metodologia utilizada foi baseada no *Kanban*, onde se organiza o fluxo de trabalho através de estados. Estes estados representam uma fase no ciclo de vida das tarefas. Assim, cada tarefa tem um estado. As tarefas são dispostas no *Backlog* de acordo com o seu estado, como é possível observar na figura 1.5. Na mesma imagem é possível observar os estados do ciclo de vida de uma tarefa utilizados ao longo deste projeto, designadamente:

- *To Do* e *Could Do* - Inicialmente as tarefas são colocadas na lista de acordo com a sua prioridade no projeto.
- *Doing* - À medida que se avança para o desenvolvimento de uma tarefa, o seu estado é atualizado. Por vezes, é necessário ter mais do que uma tarefa neste estado, o que implica uma gestão de prioridades. A resolução deste conflito passa por terminar a tarefa mais urgente.

- *Done* - Quando uma tarefa é terminada é atualizado o seu estado, através da alteração para a lista de tarefas terminadas.

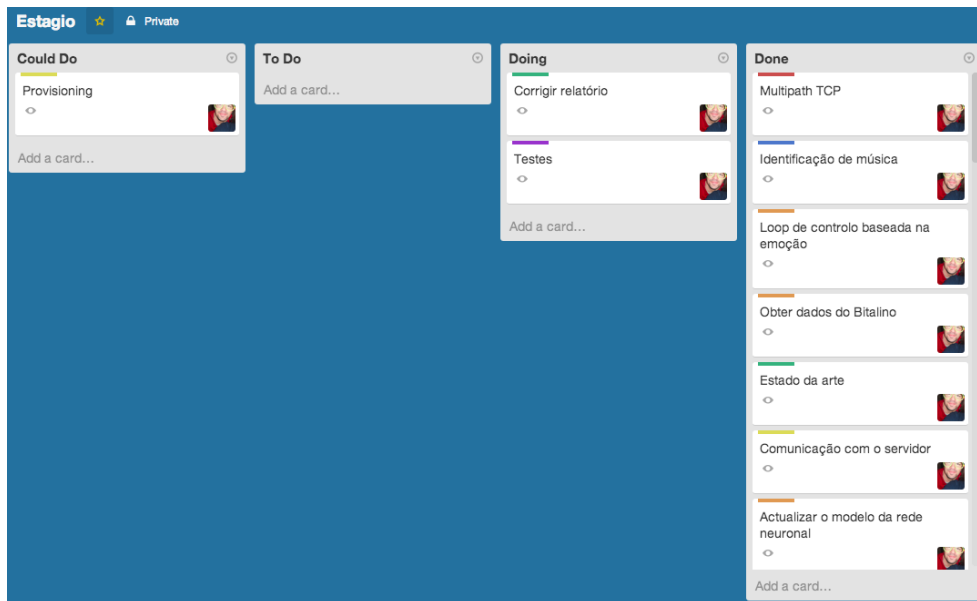


Figura 1.5: *Backlog* do projeto *HappyHour* recorrendo à ferramenta *Trello*[5].

Uma vez que a equipa de desenvolvimento é composta pelo aluno Pedro Carmona quando é atingido um limite *work in progress*, a resolução deste conflito passa por reestruturar o plano de desenvolvimento com o aval dos orientadores, para que as tarefas prioritárias sejam desenvolvidas. Esse limite é de 4 tarefas coincidentes no estado *Doing*.

As tarefas são parte integrante dos artefactos. Recorrendo às tarefas foi possível entregar os artefactos em pequenas iterações com frequência semanal, seguindo a política ágil de particularizar o seu desenvolvimento, com iterações mais curtas. Desta forma, procurou-se manter o orientador Professor Sá Silva e o co-orientador e aluno de Doutoramento David Nunes a par do desenvolvimento do trabalho, permitindo reformulações nos objetivos prioritários a cada semana.

Os seguintes artefactos foram produzidos durante o estágio:

- Código fonte;
- Relatório;
- Artigo.

Para a realização do projeto, foi, portanto, adotada uma política de reuniões semanais com os orientadores. Mais especificamente, na segunda-feira de cada semana eram expostos os progressos, as dificuldades e obstáculos, mas também possíveis soluções. As reuniões tinham uma duração média de 15 minutos, podendo alongar-se, se necessário. No final de cada reunião eram definidos objetivos e estratégias a realizar ao longo da semana. O contacto do aluno com os orientadores foi também realizado de forma assíncrona, como previsto no novo manifesto para o desenvolvimento assíncrono [6], utilizando ferramentas como email e Skype[7] para comunicar à distância, permitindo, ao mesmo tempo, documentar as decisões.

Considerando a colaboração com a Sapo Labs, realizou-se também uma conferência com todos os intervenientes neste projeto que coincidiu com o início deste trabalho. Neste evento, o progresso da aplicação e os objetivos a alcançar foram discutidos.

1.4.2 Planeamento

Nesta secção é abordado o planeamento do trabalho desenvolvido no estágio. O planeamento do trabalho foi estabelecido no início do estágio, em Setembro. Posteriormente, em Fevereiro, decidiu-se rever os objetivos e o plano de trabalhos de forma a incluir o mecanismo de HiTL desenvolvido pelo Eng. David Nunes no projeto *HappyHour*.

No planeamento inicial estavam previstas as tarefas seguintes:

- Estudo da Ferramenta *HappyHour* e discussão da especificação (1 mês).
- Implementação de funcionalidades na aplicação Web de gestão e partilha de notificações e eventos nas redes sociais (1 mês e meio).
- Introdução de um mecanismo de *check-ins* automatizados (2 meses).
- Visualização estatística da afluência e agitação do ponto de interesse na interface Web de gestão desse ponto (1 mês e meio).
- Interface gráfica nos pontos de interesse (1 mês).
- Implementação de um sistema de deteção de música para criação de *playlists* e eventos automatizados (2 meses e meio).
- Testes e Melhorias (1 mês).

No anexo B pode ser consultado o diagrama de *Gantt* construído para o planeamento inicial, considerando o *HappyHour* como um serviço para diversão noturna.

No decorrer do estágio surgiu a possibilidade explorar novas vertentes de investigação o que levou à necessidade de reestruturar o plano de trabalho. Neste sentido, foram acrescentados objetivos para trazer uma componente de inovação científica adicional ao projeto de Mestrado.

O novo plano de trabalhos, definido em Fevereiro de 2014, previa as seguintes tarefas para o segundo semestre:

- Discussão sobre a especificação do sistema de reconhecimento de emoção, planeamento dos novos requisitos, assim como o estudo das plataformas concorrentes do *HappyHour* (1 mês).
- Obter agitação e ruído ambiente através do processamento dos sinais resultantes dos sensores de acelerômetro e do microfone respetivamente (2 semanas).
- Implementação do mecanismo inicial de *machine learning* para detetar a emoção, composto por uma rede neuronal que recebe os valores de agitação e ruído, e devolve a emoção do utilizador (2 semanas).
- Desenvolvimento de uma interface gráfica para que o utilizador possa dar *feedback* da emoção que sente, permitindo ao algoritmo de *machine learning* criar casos de teste (2-3 semanas).
- Implementar a partilha de localização do utilizador numa rede social, para ser utilizado como mecanismo de privacidade no modelo do Eng. David Nunes (1 semana).
- Adquirir dados de meteorologia através de um serviço Web (1 semana).
- Implementar um mecanismo de identificação de música recorrendo ao auscultador do telemóvel (2-3 semanas).
- Adquirir dados de uma *smart shirt* para obter a pulsação do utilizador (1 semana).
- Utilizar um mecanismo de *handoff* do *Multipath TCP* para escolher a(s) interface(s) de rede utilizada(s) para comunicar com a Internet (1-2 semanas).
- Testes e melhorias (2 semanas).

O diagrama de *Gantt* do novo plano de trabalhos encontra-se em anexo C.

Algumas tarefas tomaram mais tempo do que o previsto, como o caso da identificação de música, o dispositivo para obter o batimento cardíaco e ainda a comunicação com o servidor. Com o atraso no desenvolvimento foi decidido adiar a entrega final para Setembro. No anexo D é possível consultar diagrama de *Gantt* que retrata trabalho realizado. A acrescentar ao plano de Fevereiro, foram executadas as seguintes tarefas:

- Criação de um processo para automatizar o lançamento de instâncias de servidos remotos, assim como permitir o teste de alterações ao projeto num ambiente da produção simulado numa máquina virtual (1 semana, não finalizado).
- Escrever um artigo científico (4 semanas).
- Criação de uma base de dados local para registar a emoção do utilizador, assim como as músicas identificadas pelo aparelho (1 semana).
- Implementar e guardar o estado da *loop* de controlo, que pode ser alterada para ir de encontro às definições pretendidas pelo utilizador (1 semana).
- Desenvolver a comunicação com o dispositivo *BITalino*, uma vez que a conectividade com a *smart shirt VitalJacket*[®] era intermitente e não possibilitava o desenvolvimento do projeto. Como o *BITalino* apenas devolve o Eletrocardiograma (ECG), foi necessário converter essa sinal em batimento cardíaco, para utilizar no sistema (1-2 semanas).
- Testes à aplicação (4 semanas).

A formalização dos requisitos encontra-se no capítulo III.

No próximo capítulo são abordadas as tecnologias utilizadas. É também apresentado um estudo comparativo entre o HappyHour e as aplicações com os quais se assemelha já disponíveis no mercado.

CAPÍTULO II

Estado da arte

Na primeira parte deste capítulo são expostos os conceitos sobre os temas abordados e as tecnologias referidas ao longo deste trabalho.

De seguida, são estudadas as plataformas existentes no mercado, com a intenção de verificar funcionalidades e qualidades presentes nestas aplicações concorrentes. Este estudo permite trazer desenvolvimento e inovação ao projeto *HappyHour*.

Outro aspeto importante que será abordado neste capítulo é a identificação da música num determinado local de interesse.

O *HappyHour*, como serviço de utilização no contexto da vida social noturna, poderia fornecer informação sobre a *playlist* dos estabelecimentos noturnos. Assim, cada utilizador poderia facilmente perceber quais os locais que se adaptam aos seus gostos musicais. No entanto, introduzir manualmente a informação da música atual não é viável, pois esta é uma tarefa repetitiva e penosa e iria depender dos funcionários do estabelecimento ou de *crowdsourcing* dos utilizadores. A solução encontrada foi a escolha de um serviço externo de reconhecimento de música que permite o preenchimento destas listas musicais de forma automática. Após um estudo comparativo, foi selecionado o serviço mais apropriado.

Em seguida é abordado o mecanismo que permite inferir a emoção do utilizador. A emoção é inferida através de informação proveniente de sensores e de serviços externos. Estas informações são conjugadas recorrendo a um mecanismo de *machine learning*. Deste modo, é apresentada uma comparação entre bibliotecas de *machine learning*.

Conforme referido acima, a emoção é inferida através de informação proveniente de sensores e de serviços externos.

Os sensores utilizados neste projeto foram os seguintes:

- Acelerómetro e microfone do telemóvel;
- *BITalino* ou *smart shirt VitalJacket*[®].
- Os sensores do serviço externo de meteorologia.

Relativamente aos serviços externos, pretendia-se adquirir informação sobre as condições meteorológicas na localidade do utilizador, de modo a inferir sobre o seu estado emocional, sabendo que afeta o estado emocional dos humanos [8] [9]. Assim, são comparados alguns serviços de meteorologia existentes no mercado, considerando que estes serviços permitem obter informações meteorológicas com base numa localidade ou com base em coordenadas geográficas.

Por último, para administrar um ponto de interesse, como por exemplo um estabelecimento noturno, o administrador utiliza uma página Web. Com o objetivo de facilitar o desenvolvimento desta plataforma são estudadas algumas *frameworks* Web, que potenciam boas práticas no desenvolvimento de plataformas Web.

2.1 Tecnologias e protocolos

Neste capítulo são introduzidas algumas tecnologias utilizadas no desenvolvimento do sistema.

Serviços Web

Os serviços externos são importantes em várias fases deste projeto.

Numa fase inicial, o *Foursquare* permite obter informação acerca dos pontos de interesse, tornando o sistema funcional. As notificações geradas pelos estabelecimentos noturnos chegam a um maior número de pessoas através da partilha destas notificações em redes sociais do *Facebook*, do *Twitter* e do *Google Plus*. O anexo E enuncia as características das diversas redes sociais utilizadas ao longo do projeto.

Posteriormente, a utilização de serviços externos serviu também para adquirir informação sobre as condições climáticas na localidade do utilizador, de modo a inferir sobre o seu estado emocional, pois podem influenciar esse estado[8][9].

Cyber-Physical Systems

O *smartphone* é um aparelho cada vez mais popular, utilizado para aceder a uma variedade de informação na Web a uma taxa sem precedentes. Hoje em dia, os *smartphones* estão equipados com vários sensores e apresentam uma capacidade de

processamento que os tornam cientificamente atrativos. Devido à sua mobilidade e capacidade de processamento são utilizados em tecnologias como *Cyber-Physical Systems* (CPSs). Estes CPSs monitorizam, controlam e adaptam-se ao ambiente. Podem ser aplicados em vários cenários[10], como, por exemplo:

- na área dos transportes, para resolver problemas de congestionamento de tráfego;
- na área da saúde e bem estar, para monitorizar a atividade física e encorajar os utilizadores a realizarem exercício e melhorar a sua saúde [11].

Affective Computing

Os sensores foram inicialmente aplicados para identificar as emoções num trabalho de Rosalind Picard (1995), que originou a área da *Affective Computing* [12]. Esta área refere-se a sistemas capazes de reconhecer, processar e simular as emoções humanas. Existem várias implementações de reconhecimento de emoções que utilizaram expressões faciais, voz, discurso, temperatura do corpo, e ainda, o ritmo de pressão das teclas de um computador [13]. O reconhecimento em si é realizado com recurso a mecanismos de *machine learning*. No caso do reconhecimento de emoções através da voz e do discurso, Paeschke reportou 77% de precisão no reconhecimento das emoções agitação e calma, num *call-center*, utilizando redes neuronais. Os sinais fisiológicos também foram utilizados para reconhecer emoções, num estudo [14] onde foi possível identificar emoções positivas/negativas com uma taxa de 87% de certeza. Noutro estudo realizado pela *Samsung*, procurou-se identificar a emoção do utilizador analisando o comportamento ao toque no ecrã do *smartphone*, onde utilizaram 12 atributos gerados através de 3 sensores, e com uma taxa de certeza de 82% [15]. Com este estudo, foi possível associar padrões de toque a emoções. Gerou-se uma matriz que caracteriza as aplicações, e que permite perceber as emoções do utilizador antes e depois da utilização de uma dada aplicação.

Os estados emocionais dos humanos afetam a sua performance na aprendizagem [16], nas relações humanas, assim como a sua saúde mental e bem estar, sendo estas situações estudadas no contexto científico da área de *Affective Computing*. Acredita-se que as aplicações que determinam a emoção do utilizador vão ter impacto direto na área dos CPSs, por forma a tornar estes sistemas mais cientes da presença humana.

Behavior Change Interventions

Os sistemas BCI procuram alterar o comportamento humano recorrendo a intervenções. Estas intervenções consistem em conselhos e disponibilização de informação sobre comportamentos prejudiciais (fumar, alimentação compulsiva,

sedentarismo), de modo a motivar o indivíduo a corrigir esse comportamento. Estas intervenções era inicialmente realizadas através de consultas presenciais com terapeutas, mas atualmente são também efetuadas através da Internet e em *smartphones* [17]. A aplicação deste tipo de sistemas permite melhorar a saúde e o bem estar dos seres humanos. É possível explorar a utilização dos sensores dos telemóveis para monitorizar o comportamento e fornecer motivação e sugestões de forma mais crítica e direcionada, no sentido de potenciar a correção de comportamentos.

Multipath TCP

Prevê-se que o acesso à Internet ocorra cada vez mais através de dispositivos móveis, do que por meio de dispositivos por cabo. Antecipa-se também um aumento no tráfego móvel de 44%, em 2013, para 61%, em 2018, passando este a ser o principal meio de transmissão de dados[18]. Ao mesmo tempo, a utilização de várias interfaces de rede contribui para uma repartição do tráfego de Internet, aumentando a conectividade e a cobertura[19].

O *Multipath TCP* aborda a troca de interfaces de rede para garantir conexão e qualidade de serviço em ambientes onde a mobilidade está presente. É compatível com o TCP/IP existente nos aparelhos convencionais, e permite adicionar as capacidades do *multipath*. Com esta expansão podem ser utilizadas várias ligações de *Transmission Control Protocol* (TCP) de forma transparente para a aplicação[20]. Para além disto, o protocolo mantém compatibilidade com as *Application Programming Interfaces* (APIs) de TCP existentes, sendo as alterações desta expansão aplicadas ao nível do sistema operativo, e, ao mesmo tempo, compatíveis com os equipamentos que suportam a Internet nos dias que correm.

Este protocolo encontra-se numa fase de avaliação, assinalando-se as seguintes implementações [1]:

- Uma implementação de *Multipath TCP* (MPTCP) no *Kernel* do *Linux*, mantida pelo laboratório de redes da UCLouvain, assim como implementações para *Kernel* do *Android*[21].
- A Universidade de Swinburne possui também uma implementação em *FreeBSD*[22].
- O *NetScaler* com MPTCP desenvolvido pela *Citrix Systems*[23].
- A aplicação *SIRI* aplica o protocolo desde a versão do sistema operativo *iOS 7*, da *Apple*[24].

Existem vários estudos que procuram validar este protocolo. As conclusões apontam para que venha a ser parte integrante dos sistemas operativos no futuro, uma vez que beneficia os *datacenters*[25][1]. Ao mesmo tempo, este protocolo apresenta maior eficiência energética do que o protocolo TCP[26][27][1], comparando cenários onde o utilizador perde a ligação WI-FI, sendo trocada a ligação para o interface 3G. Por outras palavras, pode-se estabelecer conexões através de dois ou mais interfaces, mantendo o interface mais eficiente como prioritário. Caso este falhe, a transmissão dos pacotes é realizada através do interface de *backup*. Porém, comparando com o caso em que só é utilizada uma ligação e nunca se troca de ligação, o MPTCP, por ter ambos os interfaces ligados, é energeticamente menos eficiente.

Assim, a troca de interfaces suavizada apresenta vantagens para os sistemas ciber-físicos, especialmente nas aplicações centradas no humano, onde a mobilidade é um requisito destes sistemas. Esta troca suavizada permite uma maior conectividade e reduz intermitência.

O projeto *HappyHour* procura nesta iteração implementar um sistema que permita controlar a escolha de interfaces de rede com base na emoção do utilizador. Com a utilização desta tecnologia é possível implementar um sistema que disponibiliza os dois interfaces de rede, quando o utilizador se sentir ansioso.

Human in the loop

As aplicações que seguem o paradigma dos *Human-in-the-loop Cyber-Physical Systems* (HiLCPSs) [28] dividem-se em três componentes:

- o ser humano;
- o sistema;
- o ambiente.

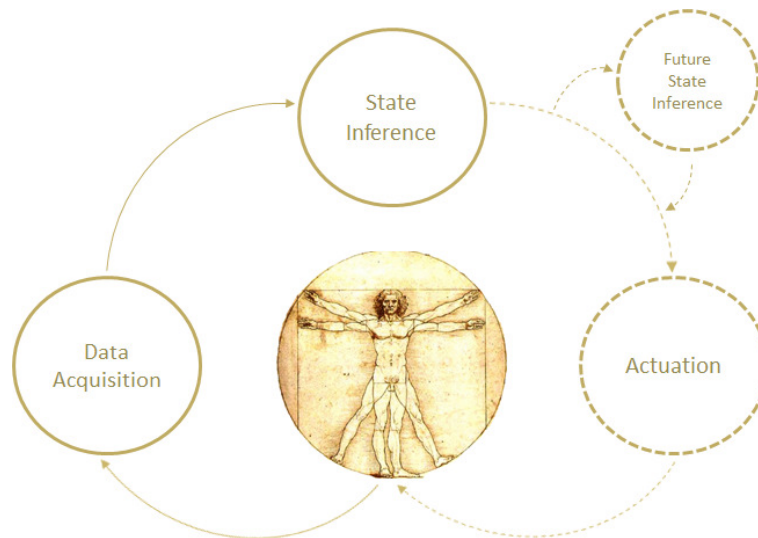


Figura 2.1: Processo de controlo no HiTL[29].

Esta abordagem considera o humano com parte integrante do *control loop*, influenciando a saída do sistema. Por outras palavras, influencia a atuação do sistema no ambiente, como se observa na imagem 2.1.

Existem alguns estudos nesta área, nomeadamente:

- uma cadeira de rodas com condução assistida, que permite evitar obstáculos e dispõe também de um sistema de emergência para evitar precipícios e escadas [30];
- um projeto de HiLCPSs que utiliza sensores relativamente baratos (aproximadamente 25 dólares) para determinar se o utilizador está em casa. Se o utilizador não estiver em casa, é desligado o ar condicionado. O nível de conforto do utilizador é também tido em conta, definindo-se limites de temperatura a manter, caso o utilizador volte para casa [31];
- um sistema de *loop* fechado na área da medicina, com o objetivo de reduzir a probabilidade de ocorrer erro humano. À medida que é injetado o medicamento no utilizador, o sistema obtém os dados do batimento cardíaco e da quantidade de oxigénio, e, através de um algoritmo, calcula-se a taxa de medicamento injetada no paciente em cada iteração do *loop*. Caso seja detetado algum problema através dos sensores fisiológicos o sistema desliga o processo de injeção. Este sistema procura melhorar a segurança do procedimento e o seu efeito no utilizador [32].

Human in the loop Over Networks

O modelo proposto pelo Eng. David Nunes procura seguir uma metodologia HiLCPS, aplicada às ligações de rede dos sistemas ciber-físicos. Este modelo, na imagem 2.2, prevê a utilização de mecanismos de inferência do estado humano para configurar dinamicamente parâmetros tais como interfaces de rede, segurança e privacidade.

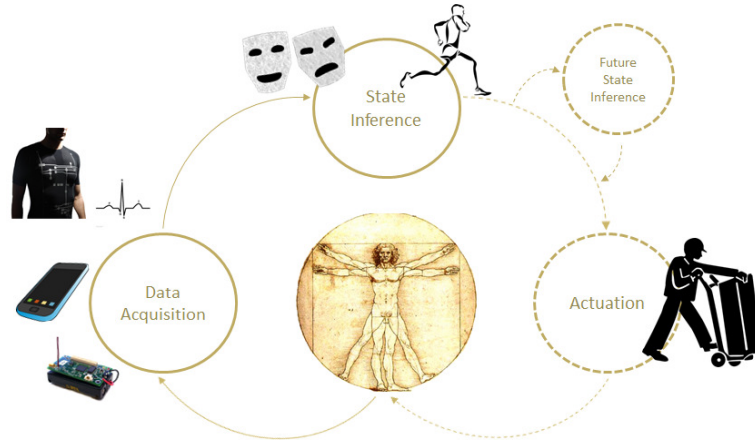


Figura 2.2: O papel do ser humano no processo de controlo no HiLCPS[29].

Este modelo é implementado no sistema *HappyHour*, o qual altera dinamicamente a interface de ligação à Internet recorrendo ao MPTCP, bem como os parâmetros de privacidade da aplicação de modo a melhorar o estado emocional do utilizador. A privacidade no âmbito das redes sociais é *trade off* que se verifica através nível de partilha de informação, uma vez que para que os utilizadores podem usufruir da partilha de localização, têm de ceder essa mesma informação. Assim o nível de partilha de informação poderá permitir encontrar amigos ou conhecer novas pessoas, e possivelmente alterando o estado emocional das pessoas.

Esta aplicação aplica ainda conceitos das BCI para intervir quando o utilizador está num estado emocional negativo, tendo em conta que a mudança de ambiente e o exercício físico moderado (através de caminhadas) melhoram a saúde mental e o bem estar dos seres humanos [33][34][35][36]. Na verdade, o exercício físico beneficia também as capacidades cognitivas, a memória e o humor [37]. Assim, procura-se detetar o humor do utilizador, aplicando-se medidas sugestivas e motivacionais que induzam o utilizador a ir passear, por forma a melhorar o seu estado emocional.

Ao mesmo tempo, configurações são adaptadas dinamicamente ao estado emocional humano, tais como:

- interfaces a serem utilizadas;

- privacidade;
- segurança.

Por exemplo, estados de ansiedade podem levar à utilização de várias interfaces de rede em simultâneo, por forma a se evitarem possíveis frustrações derivadas de quebras de ligação. Estas quebras de ligação são evitadas através de *make-before-break handoff*. Por exemplo, durante o passeio, mesmo que o WI-FI fique desconetado, a ligação é mantida. Mantendo uma mesma ligação de rede em duas interfaces simultaneamente (WI-FI e 3G), permite ao sistema continuar ligado a um serviço através da rede móvel, mesmo que o utilizador se afaste de um *hotspot* WI-FI.

Relativamente à segurança e à privacidade, sentindo-se frustrado, utilizador poderá não querer partilhar a sua localização em redes sociais. Pelo contrário, poderá disponibilizar a sua localização no caso de se sentir eufórico.

Assim, o *HappyHour* utiliza todos os conceitos anteriormente introduzidos para se tornar num sistema altamente configurável que adapta as suas configurações de privacidade e de interface de rede à emoção do utilizador.

2.2 Plataformas semelhantes ao *HappyHour*

O projeto *HappyHour* pode aplicar-se a diversas áreas, como por exemplo:

- na área das aplicações que procuram melhorar o bem estar do utilizador;
- na área das aplicações móveis que se baseiam na localização do utilizador e na caracterização de pontos de interesse.

Ao longo desta secção é exposto um estudo comparativo do projeto *HappyHour* com algumas aplicações enquadradas nos domínios referidos acima.

2.2.1 Aplicações que procuram melhorar o bem estar o utilizador

O bem estar do ser humano é uma área bastante abrangente. Existem aplicações que se focam no bem estar emocional, e aplicações que se focam no bem estar físico.

1. *EmotionSense*



Figura 2.3: Logótipo da aplicação *EmotionSense*[38].

Desenvolvida por uma equipa de investigação da Universidade de Cambridge para o sistema *Android*, a aplicação *EmotionSense* [38] tem como objetivo monitorizar a emoção dos utilizadores. Tendo em conta que os *smartphone* estão presentes no dia-a-dia dos utilizadores, é possível a monitorização contínua da emoção. A técnica utilizada consiste em recolher informações através de:

- sensores (GPS, o acelerómetro e o microfone);
- padrões dos registos de chamadas e SMS's;
- questionários ao utilizador sobre a sua condição emocional [38]. É necessário o preenchimento de uma grelha de emoção, onde se encontram representadas várias emoções (figura 2.5), e de um formulário (figura 2.6). O formulário foi estruturado por psicólogos da equipa de investigação [39].

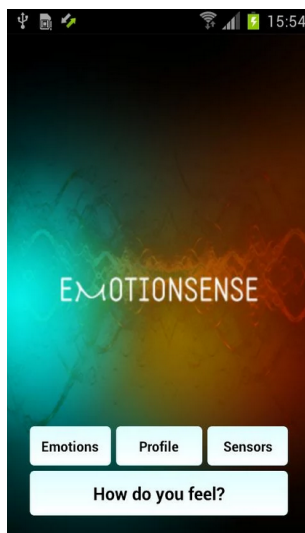


Figura 2.4: Ecrã de introdução do *EmotionSense*[38].

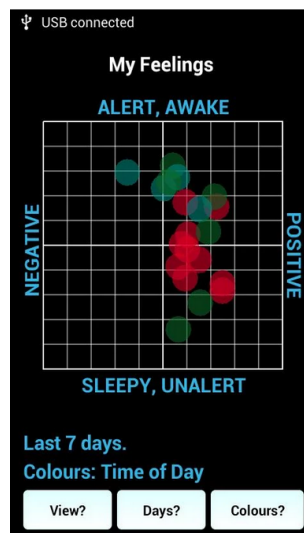
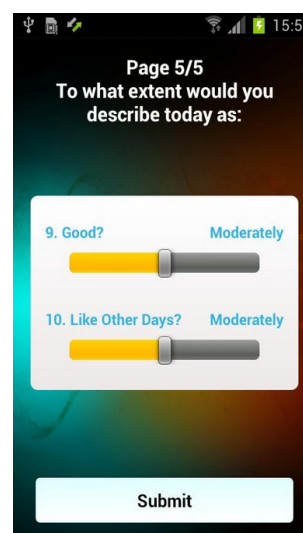


Figura 2.5: Grelha de emoção do *EmotionSense*[38].



Page 5/5

To what extent would you describe today as:

9. Good? Moderately

10. Like Other Days? Moderately

Submit

Figura 2.6: Formulário do *EmotionSense*[38].

A componente emotiva do *HappyHour* é composta por um sistema de *feedback* inspirado na grelha de emoção da *EmotionSense*. Conforme o nome indica, este sistema de *feedback* permite ao utilizador corrigir a emoção detetada pela aplicação. Em comparação, as duas ferramentas permitem monitorizar a emoção do utilizador, porém, no caso do *HappyHour*, esta deteção é realizada com o objetivo de seguir a teoria do HiTL. Assim, a emoção detetada é o veículo que permite introduzir variação ao sistema. Dependendo da emoção detetada, é ativado ou desativado um conjunto de definições que são responsáveis pela configuração automática de vários parâmetros. Enquanto que o *EmotionSense* apenas procura detetar a emoção do utilizador, o *HappyHour* assume uma atitude pró-ativa. Por outras palavras, o *HappyHour* procura, simultaneamente, melhorar a emoção do utilizador, sugerindo um passeio a locais de interesse nas imediações, uma vez que este tipo de actividade física permite melhorar:

- a saúde mental;
- a aprendizagem;
- a memória;
- a atenção;
- o humor [33] [34] [35] [36] [37].

2. *NikeRunning*



Figura 2.7: Logótipo da aplicação *NikeRunning*[40].

A aplicação *NikeRunning* [40] permite registar os dados de uma corrida desportiva. É possível adicionar amigos, comparar resultados e visualizar os amigos com maior número de corridas e mais quilómetros percorridos, incentivando o utilizador a correr com mais frequência [41], mantendo um estilo de vida saudável. O GPS e o acelerómetro do *Smartphone* registam a distância, a velocidade e o tempo do trajeto com precisão.

Genericamente, durante a corrida, a aplicação *NikeRunning* consiste em:

- receber *feedbacks*, em áudio, sobre a distância percorrida;
- consultar o *heatmap* de velocidade disponibilizado [42];
- receber incentivos dos seus amigos do *Facebook*, sempre que publicar o início de uma corrida;
- convidar os seus amigos para se juntarem à corrida;
- compartilhar o mapa do seu trajeto com amigos e familiares;
- configurar a *playlist* a reproduzir durante a corrida.



Figura 2.8: *NikeRunning* - *Heatmap* da corrida [42]

Conforme representado na figura 2.8, o *Heatmap* apresenta o percurso da corrida utilizando uma escala de cor verde a vermelho. As zonas a laranja e vermelho representam zonas do percurso em que a corrida foi mais lenta, enquanto que as zonas a verde representam a parte do percurso em que a corrida foi mais rápida.

Comparando o *HappyHour* e o *NikeRunning*, destaca-se o seguinte:

- ambas as aplicações incentivam os utilizadores a realizarem atividades físicas;
- ambas as aplicações utilizam os *heatmaps*, no entanto, no *NikeRunning* estes mostram a velocidade de corrida do utilizador, enquanto que no *HappyHour* os *heatmaps* caracterizam a afluência aos pontos de interesse;

- o *HappyHour* foca-se no bem estar emocional, enquanto que o *NikeRunning* incide mais no bem estar físico;
- a *NikeRunning* inclui uma componente social que seria interessante implementar em versões posteriores do *HappyHour*;
- a *NikeRunning* é uma aplicação que é apenas informativa em relação ao esforço físico do utilizador, não se adapta. Pelo contrário, o *HappyHour* depende da emoção do utilizador. Para além disso, o *HappyHour* permite definir o contexto do sistema através da emoção do utilizador, respondendo de maneira diferente de caso para caso. São ativados conjuntos de regras configuráveis para contrariar certas emoções, como, por exemplo, ativar dois interfaces de rede recorrendo à tecnologia *Multipath TCP*.

2.2.2 Aplicações que se baseiam em pontos de interesse

1. *DressCode*



Figura 2.9: Logótipo do *DressCode*[43].

A aplicação *DressCode*[44] incide na vida social noturna, informando o utilizador do estado atual da noite nas imediações da sua localização [45]. Especializa-se em guiar o utilizador pelos pontos de interesse, procurando oferecer a melhor experiência possível [44]. A pesquisa pode ser feita por:

- tipo de evento (bar aberto, eventos especiais, VIP);
- localização (escolha por cidade (figura 2.10) ou na interface com mapa os pontos de interesse perto do utilizador);
- palavras-chave (nome do bar e cidade).

É possível ainda filtrar os resultados pelo vestuário necessário para o evento, evitando situações desagradáveis como ser impedido de entrar num determinado espaço público, após uma longa espera [44]. A aplicação disponibiliza também informações

detalhadas sobre estes estabelecimentos e garante precisão nos requisitos de vestuário recomendado num determinado evento. O utilizador pode comentar e classificar o estabelecimento (figura 2.11), sendo estes comentários atualizados mensalmente [45]. A aplicação *DressCode* está disponível em Atlanta, Atlantic City, Boston, Las Vegas, Los Angeles, Nova Iorque, Orlando, Filadélfia, Providence, San Francisco e Washington DC[44].



Figura 2.10: Ecrã de escolha da cidade na aplicação *DressCode*[38].

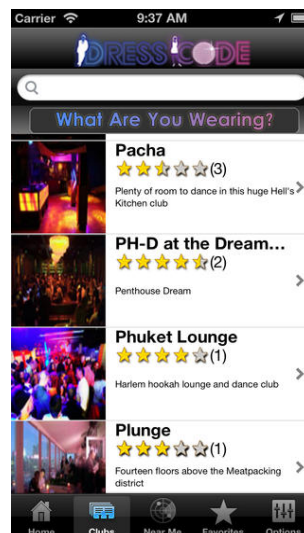


Figura 2.11: Lista de estabelecimentos noturnos com classificação e descrição na aplicação *DressCode*[45]

Comparando o *HappyHour* e o *DressCode*, destaca-se o seguinte:

- ambas as aplicações disponibilizam informações sobre os eventos num determinado estabelecimento noturno;
- enquanto que o *HappyHour* apresenta detalhes mais dinâmicos de um determinado estabelecimento, o *DressCode* foca-se mais em informações estáticas. Como o nome indica, o *DressCode* informa sobre o vestuário permitido num determinado estabelecimento que seria interessante implementar em versões posteriores do *HappyHour*. Por outro lado, o *HappyHour* descreve a afluência e a agitação das pessoas nesse estabelecimento num dado momento, assim como o tipo de música e a emoção geral. Ainda assim, caso o gestor do estabelecimento o pretender pode disponibilizar a informação do vestuário inserindo na descrição do evento através da plataforma Web *HappyHour*. Se

assim for, então o *HappyHour* torna-se uma ferramenta mais abrangente e ao mesmo tempo específica, além de que estas configurações podem influenciar a escolha do utilizador comparativamente à aplicação *DressCode*.

2. *Foursquare* e *Swarm*



Figura 2.12: Logótipo do *Foursquare*[46].



Figura 2.13: Logótipo do *Swarm*[47].

O *Foursquare*[46] é uma rede social bastante popular lançada em 2008 que incentiva os seus utilizadores a divulgar a sua localização a amigos, familiares e terceiros. Permite encontrar outros utilizadores que estejam próximos de uma determinada localização. A aplicação deteta a posição através de GPS, WI-FI ou redes móveis, e mostra os locais mais próximos. Em Janeiro de 2014, contava com cerca de 45 milhões de utilizadores em todo o mundo [48], mais 20 milhões do que em 2012 [49]. O *Foursquare* pode ser utilizado a partir de várias plataformas móveis como *iOS*[50], *Android*[51], *WindowsPhone*[52], *Blackberry*[53] e *Symbian*[54]. Inspira-se na Web 2.0, onde são os utilizadores a contribuir na criação de conteúdos. O *Foursquare* incentiva a participação dos utilizadores da seguinte forma:

- Os utilizadores podem fazer *check-ins* em locais de interesse que se encontrem nas imediações.
- Com cada *check-in* são atribuídos “pontos” e, por vezes, o utilizador é recompensado com “crachás”.
- Os “crachás” podem ser ganhos por realizar certas acções num ponto de interesse, por efectuar *check-ins* em determinadas alturas, ou pela frequência destes.
- Quando um utilizador faz mais *check-ins* num certo ponto de interesse do que qualquer pessoa nos últimos 60 dias, torna-se o “*Mayor*” desse local.

Os pontos acima mencionados promovem a fidelização dos utilizadores aos pontos de interesse e a sua colaboração com conteúdo para a aplicação. De acordo com os dados disponibilizados pela empresa, em Janeiro de 2014, o *Foursquare* contava já com mais de 5 mil milhões de *check-ins*, um número que tende a aumentar [48].

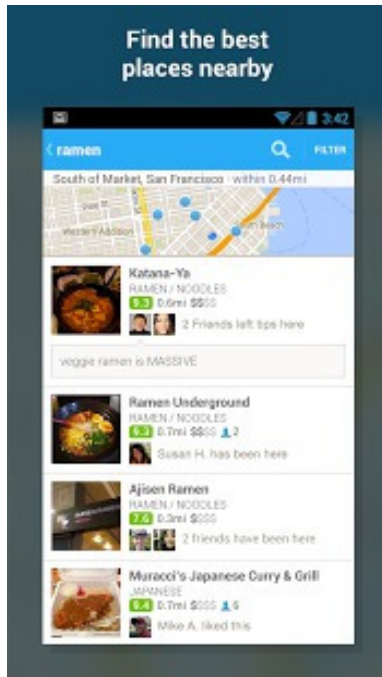


Figura 2.14: Encontrar os melhores locais nas imediações[51].

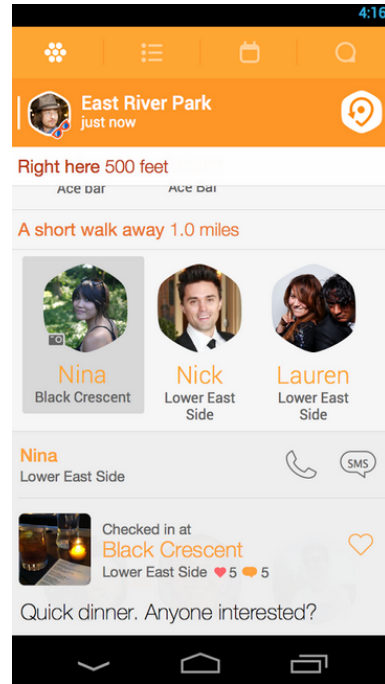


Figura 2.15: *Swarm* - Encontrar amigos nas imediações[47].

Em Maio de 2014, o *Foursquare* anunciou mudanças na sua aplicação, e apresentou a aplicação *Swarm*[55] para procurar amigos nas imediações e combinar um encontro pessoal [56].

Comparando o *HappyHour* com o *Swarm* e o *Foursquare*, destaca-se o seguinte:

- no *Swarm* e *Foursquare* os conteúdos são gerados pelos utilizadores, enquanto que o *HappyHour* associa quer as informações provenientes do *Foursquare* quer as geradas automaticamente (afluência, agitação, *playlist* e emoção geral). Convém referir que no projeto *HappyHour* a API do *Foursquare* foi utilizada para obter informações sobre os pontos de interesse;
- no *HappyHour* não é necessário que os utilizadores definam manualmente o local onde se encontram num determinado momento;
- no *Swarm* e *Foursquare* o número de *check-ins* determina a afluência do local de interesse, enquanto que no *HappyHour* os *check-ins* não são necessários. Por outras palavras, todos os indivíduos que tenham a aplicação vão ser contabilizados na afluência;

- no *HappyHour* o utilizador sabe o que está a acontecer a cada momento e de forma dinâmica, enquanto que no *Swarm* e *Foursquare* a necessidade de *check-ins* torna a aplicação mais estática;
- ao contrário do *Swarm* e *Foursquare*, o *HappyHour* apresenta uma componente de intervenção (promove a melhoria de humor através do exercício físico e a descoberta de novos locais) e uma componente inovadora de HiTL (utiliza a emoção do utilizador para decidir se publica, ou não, nas redes sociais (como o *Facebook*[43]) a sua localização). Assim, o *HappyHour* destaca-se por ser uma aplicação que toma decisões baseando-se na emoção do utilizador e, simultaneamente, promove o seu bem estar e incentiva-o a descobrir novos pontos de interesse.

3. *Facebook Local Search*



Figura 2.16: Logótipo do *Facebook*[43].

O *Facebook* [43] é a maior rede social a nível mundial, contando com 1,23 mil milhões de utilizadores mensalmente ativos [57]. No final de 2012, esta rede social decidiu incluir na sua aplicação uma componente semelhante ao *Foursquare*. Nesta componente, é possível fazer *check-in* nos pontos de interesse, avaliar e recomendar, assim como complementar informações sobre estes locais de interesse. É possível saber a localização dos amigos, perceber quais os locais de interesse mais próximos de uma determinada localização, e verificar a avaliação dos amigos desses locais. O separador *Local Search* está presente nas aplicações móveis do *Facebook* (iOS[58], *Blackberry*[59], *Android* [60] e *Windows Phone* [61]). Uma vez que o *Facebook* já inclui uma componente para gerir eventos, em conjunto com esta componente de pontos de interesse, e aliado à quantidade de utilizadores da rede social, torna-se numa ferramenta influente no domínio dos pontos de interesse.

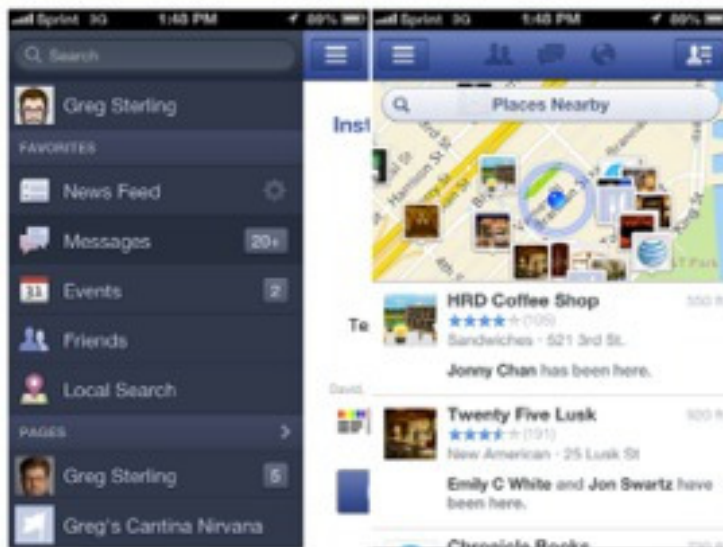


Figura 2.17: Aplicação *Facebook* - *Local Search* [62].

Comparando o *HappyHour* com o *Facebook Local Search*, destaca-se o seguinte:

- Ao contrário do *Facebook Local Search*, o *HappyHour* informa os utilizadores acerca do estado atual dos locais de interesse (afluência, agitação, *playlist* e emoção geral). O *Facebook Local Search* apenas apresenta dados sobre o número de pessoas que já frequentou um determinado local e as avaliações atribuídas;
- O *Facebook* encontra-se amplamente adoptado a nível mundial, tendo, por isso, bastante impacto na sociedade.
- O *Facebook* pode ser utilizado para partilhar o estado emocional através de um *post*, enquanto que o *HappyHour* deteta a emoção sem qualquer intervenção do utilizador, por meio de sensores. Por isso, pensou-se associar este serviço ao *HappyHour* para a partilha de estados emocionais. No entanto, como a API deste serviço ainda não disponibiliza métodos, optou-se, nesta fase, por partilhar apenas a localização do utilizador na rede social.
- O *HappyHour* 'reage' à emoção detetada, sugerindo ações ao utilizador de forma a melhorar as suas emoções, enquanto que o *Facebook* não utiliza a informação sobre a emoção para se adaptar ao utilizador.
- Relativamente ao design, a aplicação do *Facebook* é superior, pois utiliza fotografias para representar o local de interesse. Contudo, a interface do *HappyHour* permite visualizar no mapa parâmetros como o nível de afluência dos

pontos de interesse, o nível de agitação, e os estados emocionais. Apesar de já permitir bastantes funcionalidades na escolha do destino, o projeto *HappyHour* poderá no futuro adotar a representação de pontos de interesse através de fotografias como no *Facebook*, porque torna a interface mais interessante.

O *Facebook* é claramente uma das maiores aplicações sociais do mundo. Considerando a influência que tem sobre um número tão grande de utilizadores, optou-se por utilizar este serviço para:

- extrair pontos de interesse;
- desenvolver o requisito de privacidade do modelo de HiTL do Eng. David Nunes, de forma a permitir aos utilizadores divulgarem a sua localização.

4. *Waze*



Figura 2.18: Logótipo do *Waze*[63].

A aplicação *Waze*[63] disponibiliza um sistema de navegação com informação colaborativa, reunindo informações da comunidade local [64]. Aconselha um trajeto ao utilizador consoante o volume de tráfego automóvel e possíveis acidentes de trânsito na via. Esta aplicação, adquirida pela *Google* por mil milhões de dólares em Junho de 2013 [65], utiliza os dados dos sensores para avaliar os tempos de condução dos utilizadores com o objetivo de fornecer informações de tráfego em tempo real. A colaboração por parte dos utilizadores pode ser feita durante o percurso, introduzindo informações sobre o trânsito. O utilizador pode definir o percurso através dos dados disponibilizados por outros utilizadores, o que contrasta com os sistemas de navegação tradicionais, onde não existia este tipo de colaboração. É possível ainda coordenar encontros para partilhar o veículo com os amigos, através da divulgação da localização e tempos estimados de viagens [66]. Para além disto, de forma a incentivar as pessoas a contribuir para a aplicação, os utilizadores ganham pontos à medida que atualizam as informações de trânsito e são classificados num ranking da comunidade [67].

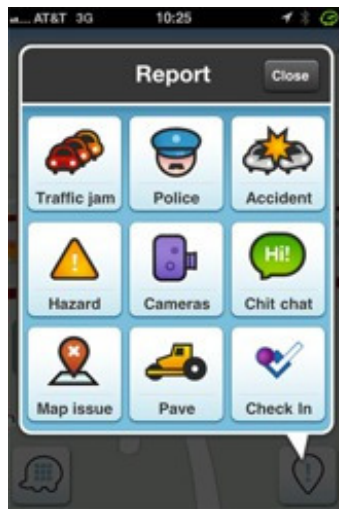


Figura 2.19: *Waze* - Aplicação móvel que permite reportar ocorrências nas vias de trânsito [68].

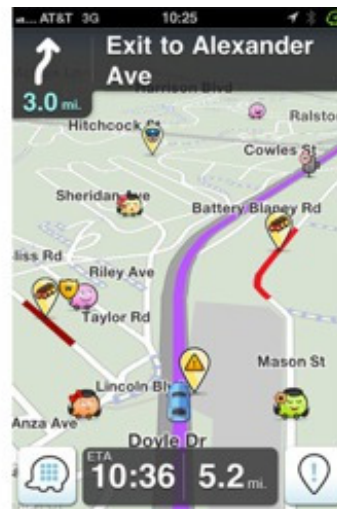


Figura 2.20: *Waze* - Aplicação móvel que assiste o utilizador na condução [68].

Comparando o *HappyHour* com o *Waze*, destaca-se o seguinte:

- ambas as aplicações reúnem dados através de sensores de forma a caracterizar o mundo real, mas em contextos diferentes;
- ambas as aplicações reúnem dados através de sensores que são, posteriormente, disponibilizados a todos os utilizadores, permitindo que cada um escolha o seu destino de uma forma mais informada;
- apesar de o *HappyHour* e o *Waze* não apresentarem objetivos comuns, a forma como o *Waze* dá aos utilizadores a possibilidade de reportar informações sobre o trânsito (figura 2.19) poderia ser aplicada ao *HappyHour* para reportar ocorrências ou mesmo os preços dos combustíveis em determinados locais num certo momento;
- ao contrário do *Waze*, o *HappyHour* permite detetar a emoção dos seus utilizadores;
- seria interessante implementar em versões posteriores do *HappyHour* a criação de comunidades em locais onde não existe um gestor de estabelecimentos, como, por exemplo, um parque ou uma praia. Os utilizadores poderiam colaborar na

atualização de dados destes locais e preços de espetáculos, bebidas, atividades físicas ou outros eventos.

5. *Yelp*



Figura 2.21: Logótipo do *Yelp*[69].

O *Yelp* é uma plataforma que permite localizar serviços. A pesquisa de serviços permite aceder a detalhes de funcionamento e à descrição dos estabelecimentos numa cidade [70]. Os utilizadores podem contribuir para a classificação de um estabelecimento e divulgar as suas opiniões sobre o mesmo, podendo também fazer *check-ins*. Existem 53 milhões de análises críticas escritas pelos utilizadores [70]. Podem ainda adicionar detalhes ao perfil do estabelecimento, sujeitos à aprovação dos moderadores. A estratégia utilizada para motivar os utilizadores a contribuírem para a aplicação é colocar em destaque as descrições mais detalhadas e a primeira a ser escrita, criando uma competição pelas posições de destaque [71]. Outra estratégia com o mesmo fim é atribuir “crachás” quando os utilizadores concluem os objetivos propostos [71]. Ao adicionar amigos é possível visualizar as suas opiniões sobre os locais e também trocar mensagens [71]. Os gostos de cada utilizador condicionam as sugestões que irá receber do sistema, procurando ir de encontro aos interesses de cada um. Para além disso, quando um utilizador gosta do mesmo local que um segundo utilizador, recebe, a partir daí, sugestões de visita dos locais visitados por este segundo utilizador, e vice-versa. Os proprietários dos estabelecimentos podem reagir aos comentários deixados pelos utilizadores, respondendo através da secção de comentários ou por mensagem privada. O *Yelp* conta com cerca de 120 milhões de visitantes mensais [70]. Para além da plataforma Web, é possível interagir com o serviço através das aplicações *iPhone*, *Android* e *BlackBerry*[70]. A aplicação móvel permite localizar serviços com a câmara fotográfica do telemóvel e o sensor bússola.



Figura 2.22: *Yelp* é um repositório de estabelecimentos e negócios locais[72].

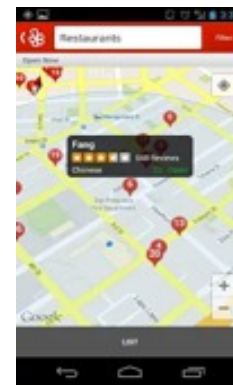


Figura 2.23: *Yelp* - Aplicação *mobile* com diretório de serviços[73].

Comparando o *HappyHour* com o *Yelp*, destaca-se o seguinte:

- ao contrário do *HappyHour*, o *Yelp* apenas disponibiliza informação sobre o número de 'gostos'. O *HappyHour* procura inovar nesta área centrando-se na componente emocional do utilizador, incentivando-o a fazer atividade física, ou apenas passeios, para melhorar a sua disposição;
- o *Yelp* diferencia-se da concorrência pela divulgação de análises críticas que os seus utilizadores inserem na aplicação.

6. *Shopkick*



Figura 2.24: Logótipo do *Shopkick*[74].

Shopkick[75] é uma aplicação que se baseia na fidelidade do cliente a um determinado estabelecimento comercial. Os utilizadores recebem pontos (*Kicks*) sempre que passam nas imediações dos estabelecimentos, ou perto de determinadas secções dos estabelecimentos. A aplicação capta um sinal *shopkick*, que é um som inaudível emitido em cada loja (parceira da aplicação) e que verifica que um utilizador está na loja. Em seguida, o utilizador é recompensado com um *kick*. Os utilizadores podem trocar

kicks por cartões-oferta, músicas do *iTunes*, bilhetes de cinema, e outros prêmios. Esta aplicação conta já com cerca de 6 milhões de pessoas [76]. Constitui a aplicação de domínio shopping que os utilizadores, em média, dedicam mais tempo por mês, enquanto visitam as grandes superfícies comerciais [77].

Esta abordagem do *Shopkick*, em recompensar os utilizadores pela sua fidelidade, poderia ser implementada no *HappyHour*. No entanto, este requisito depende bastante de estabelecimentos parceiros, sendo necessário um estudo de mercado para verificar a sua viabilidade.

2.3 Identificação de música, APIs e bibliotecas

Inicialmente, a funcionalidade de identificação de música no sistema *HappyHour* teria como fim a identificação automática de *playlists* de um determinado estabelecimento noturno. Com a evolução do *HappyHour* para uma aplicação de BCI que pretendia incentivar o exercício físico, a identificação de música poderia fornecer informação adicional sobre os locais de interesse para os utilizadores. As bibliotecas de identificação de música foram criadas para automatizar o processo, em que é possível associar os dados *metadata* a um determinado ficheiro. A *metadata* é a informação associada a uma música, incluindo o nome do artista, o nome do álbum e o ano de edição da música. Com estas bibliotecas, a identificação das músicas é realizada utilizando algoritmos, isto é, através da representação espectral de uma fração da música é possível comparar com representações espectrais existentes na base de dados, correlacionando a música com a *metadata*. Existem várias bibliotecas *open source*:

- *Echoprint*[78].
- *Freetantrum*[79].
- *MusicURI*[80].
- *Jhears*[81].
- *Libfooid*[82].
- *Audioscout*[83].
- *pHash*[84].

Inicialmente, ponderou-se implementar uma biblioteca de identificação de música no servidor da aplicação *HappyHour*. No entanto, esta implementação não seria viável, pois iria gastar recursos de memória, assim como de armazenamento e processamento, necessários para as outras componentes do sistema. Assim, a alternativa seria utilizar um cliente de um serviço Web de reconhecimento de música. Em seguida, abordam-se as características de diversas APIs de reconhecimento de música.

2.3.1 Análise a APIs

Nesta secção, são analisadas algumas das alternativas existentes no mercado de serviços Web de reconhecimento de música. Estes comparam a representação espectral, calculada localmente na aplicação *Android*, com *fingerprints* existentes na base de dados do serviço Web.

1. *The Echo Nest*

O sistema *The Echo Nest*[85] é implementado recorrendo à biblioteca *Echoprint*[78]. É um sistema desenvolvido para ajudar os programadores a criarem aplicações baseadas em música. Um dos requisitos do serviço é que a fração da música a ser enviada tenha tempo superior a 20 segundos.

As vantagens do sistema *The Echo Nest* são as seguintes:

- a criação e manutenção da sua base de dados conta com a colaboração das pessoas e das organizações parceiras (Musicbrainz)[86];
- adquirido pelo *Spotify*[87] em Março de 2014, continua de utilização gratuita;
- pode continuar a ser utilizado comercialmente [88] com o limite de 120 pedidos por minuto[89];
- o algoritmo é robusto contra o ruído dos ficheiros de entrada[78];
- cada uma das instâncias do serviço Web *The Echo Nest* é capaz de armazenar informação de cerca de 60 milhões de músicas e cada máquina suporta 50 pedidos por minuto.

Atualmente, este sistema apenas disponibiliza o acesso a perto de 150 mil músicas, uma vez que o mecanismo de sincronização ainda está em desenvolvimento.

2. *Last.fm API*

Lançada em 2007, a API de *fingerprint* da *Last.fm*[90] fornece informações musicais recorrendo ao serviço colaborativo *Musicbrainz*[91]. A utilização gratuita deste serviço está reservada a organizações sem fins lucrativos. Assim, para uso comercial, é negociado o preço com a organização[92].

Para identificar uma música, envia-se uma mensagem com a representação (*fingerprint*) do som gravado, à qual se pode adicionar informação extra (tamanho da música ou *metadata*). Como nos outros serviços aqui apresentados, é necessária uma chave da aplicação para utilizar o serviço Web, caso contrário o pedido é deferido.

3. *AcoustID*

O *AcoustID*[93] é uma ferramenta *open source* para identificar músicas. Conta com uma média de 3 milhões de pedidos por dia e mais de 26 mil pessoas já contribuíram para este serviço Web[94].

Como desvantagens do *AcoustID* destacam-se as seguintes:

- a API livre tem um limite de 3 pedidos por segundo;
- não é permitida a utilização deste serviço para uso comercial.

O algoritmo utilizado para obter a representação espectral é *open source* e denomina-se *Chromaprint* [95].

4. *Gracenote* A *Gracenote*[96] é uma empresa subsidiária da Sony Corporation of America[97].

Uma das vantagens do serviço *Gracenote* é a sua base de dados, uma vez que permite aceder a cerca de 130 milhões de músicas de dois mil géneros musicais diferentes[98] [99]. Outra vantagem é o *Gracenote SDK (GNSDK)*[100] disponibilizado aos programadores e que converte o som gravado na representação espectral. Este SDK é mais simples de implementar do que o algoritmo do *The Echo Nest*. No entanto, apenas está disponível gratuitamente para fins não comerciais.

Escolha final

Nome da API	Preço	Limite de pedidos
<i>The Echo Nest</i>	0€	120 pedidos por minuto
<i>Last.fm</i> API	Negociável	Negociável
<i>AcoustID</i> API	Negociável	Negociável
<i>Gracenote</i> API	Negociável	Negociável

Tabela 2.1: Preço e limite de APIs de identificação de músicas para fins comerciais.

Conforme apresentado na tabela 2.1, o *The Echo Nest* é o único serviço que disponibiliza a utilização para fins comerciais. No caso dos restantes serviços, é necessário obter essa informação por contato e a seguir a uma avaliação cuidada da aplicação. O *The Echo Nest* e o *AcoustID* destacam-se do *Last.fm API* pelo facto de serem construídos com base em ferramentas *open source*, que, por sua vez, oferece a possibilidade de implementar o próprio servidor quando o limite de pedidos for elevado, caso o valor do contrato negociado com as diversas alternativas não for viável. Sendo que no caso do *The Echo Nest* é possível obter bases de dados de *fingerprints* [78]. Por estas razões, para o caso de estudo do *HappyHour* a opção oferecida pela *The Echo Nest* é a mais viável, uma vez que permite o desenvolvimento e implementação do mecanismo de identificação musical sem custos. No entanto, com esta escolha é necessário compilar o código fornecido para gerar a representação espectral, e de seguida implementar um mecanismo que permita abstrair as chamadas ao código compilado através do Java recorrendo ao JNI[101], uma vez que não existe nenhuma versão desta abstração atualizada à menos de 2 anos.

2.4 Redes neuronais e bibliotecas

Conforme descrito na secção de planeamento 1.4, uma das etapas deste projeto consiste na implementação de um mecanismo que permita caracterizar a emoção do utilizador.

Pretende-se que este mecanismo tenha a capacidade de informar o utilizador quando uma emoção seja detetada, com posterior correção e/ou confirmação da emoção pelo utilizador. Com esta abordagem é possível utilizar um mecanismo de *machine learning* que aprende com o utilizador, associando um conjunto de *inputs* a uma emoção. Pretende-se um mecanismo de fácil utilização a partir de dispositivos móveis (*smartphones*), utilizando os dados obtidos dos sensores. De forma a selecionar o

mecanismo de *machine learning* mais adequado, foi utilizado um estudo que compara diferentes técnicas em termos de:

- taxa de certeza;
- tempo de processamento.

<i>Machine learning techniques</i>	<i>Correct classification rate</i>	<i>Ranking of less CPU time needed for classification</i>
<i>Decision trees</i>	96.5%	5°
<i>Support vector machines</i>	80.2%	1°
<i>Naive Bayes classifiers</i>	81.5%	6°
<i>Bayesian networks</i>	90.9%	4°
<i>Logistic regression</i>	83.4%	3°
<i>Artificial neural networks</i>	87.2%	2°
<i>Instance-based classifiers</i> (KStar, LWL, and IBk).	96.6%, 95.6% and 80.	9°, 8°, 7°

Tabela 2.2: Comparação de algoritmos de *machine learning* em *smartphones*[102].

Considerando os resultados apresentados na tabela 2.2, selecionou-se uma rede neuronal. Este mecanismo de *machine learning* oferece uma percentagem de classificações corretas razoável, ao mesmo tempo que é uma das ferramentas que menos tempo de processamento necessita para classificar um teste.

Apesar desta abordagem, seria interessante, em trabalho futuro, estudar o algoritmo mais adequado para o caso de estudo do *HappyHour*. Para este estudo, poder-se-ia recorrer a amostras de vários utilizadores.

Existem muitas implementações de redes neuronais[103]. Para desenvolver este mecanismo são apresentadas ao longo desta secção as bibliotecas escritas na linguagem *Java*[104] que permitem implementar mecanismos de *machine learning* em ambiente *Android*.

1. *Encog – Heaton Research*

A biblioteca *Encog*[105] é um projeto *open source* desenvolvido por Jeff Heaton, dedicado às linguagens *Java* e *C#*. Constitui uma coleção de algoritmos que permite criar rapidamente algoritmos de *machine learning* de vários tipos, como, por exemplo:

- *Support Vector Machines* (SVM);
- *Artificial Neural Networks* (ANN);
- *Bayesian Networks* (BN);
- *Hidden Markov Models* (HMM);
- *Genetic Algorithms* (GA).

É possível exportar a informação de cada algoritmo, como, por exemplo, exportar a estrutura de uma rede neuronal para JSON, permitindo guardar o estado da rede, ou partilhá-lo entre diferentes sistemas.

2. *Java-ML*

A *Java Machine Learning Library* (Java-ML)[106] oferece uma interface comum a todos os algoritmos de *machine learning* disponibilizados. A interface comum a todos os algoritmos contém:

- ***Instance***: é o conjunto de parâmetros de entrada do algoritmo, e representa um exemplo do mundo real, podendo ser uma cadeia com valores dos sensores.
- ***Dataset***: são conjuntos de *instances*.
- ***Algorithm***: é a técnica aplicada aos *datasets* e *instances*.

Outra componente disponibilizada é o classificador, que permite treinar o algoritmo e avaliar o seu desempenho. Nesta biblioteca é ainda possível utilizar os classificadores da biblioteca *Weka*[107], o que a torna numa biblioteca bastante prática e adaptável a exemplos da biblioteca *Weka*.

3. *JOONE*

Joone[108] é uma *framework* orientada a objetos e construída em *Java*[104]. Com uma interface gráfica, permite o treino da rede num ambiente distribuído[108]. A *Joone*

não é atualizada desde 2013, sendo a sua estrutura composta por vários módulos, programados separadamente. Esta estrutura permite separar a interface gráfica da espinha dorsal da biblioteca, e simultaneamente estender os vários módulos. Assim, é possível implementar ferramentas e algoritmos partindo de um componente base e utilizando os módulos. É possível treinar a rede em *multithreading* [109].

4. *Neuroph*

A *framework open source Neuroph* [110] fornece uma implementação de redes neuronais na linguagem *Java*. Apresenta uma implementação simples de redes neuronais, com uma estrutura composta por quatro módulos:

- Um módulo com as classes que formam a espinha dorsal;
- Um módulo com implementações de redes neuronais;
- Um módulo com funções de utilidade;
- Um módulo com uma interface gráfica.

Esta biblioteca pode ser utilizada em conjunto com a biblioteca *Encog*. Esta simbiose permite melhorar o desempenho da *Neuroph* e melhorar o interface gráfico da biblioteca *Encog*[111]. Como em outras implementações, as classes de redes neuronais podem ser adaptadas e expandidas baseando-se na implementação fornecida[110].

5. *Weka*

Waikato Environment for Knowledge Analysis framework (Weka)[112] é uma coleção de algoritmos de *machine learning* que tem o objetivo de ajudar a resolver tarefas de *data mining*. Os algoritmos podem ser aplicados de duas formas:

- diretamente a um conjunto de dados, utilizando a ferramenta;
- programaticamente através da chamada de funções da biblioteca.

Esta coleção contém soluções que permitem:

- fazer pré-processamento de informação;
- aplicar regressão e classificação;

- visualizar os dados.

As seguintes bibliotecas são descendentes do *Weka*:

- *RapidMiner*: é uma ferramenta de *data mining* que disponibiliza uma grande variedade de técnicas. É construída a partir da biblioteca *Weka*, e destaca-se por adicionar métodos de visualização úteis.
- *CougarSquared*: é uma ferramenta de *data mining* construída recorrendo à biblioteca *Weka* e *RapidMiner*.

Escolha final

No artigo [103] são analisadas algumas bibliotecas, nomeadamente à *Encog*, *Neuroph* e *JOONE*. A análise permitiu concluir que a biblioteca *Encog* é mais eficaz. Além disso, é também mencionado que esta biblioteca é fácil de utilizar.

Num estudo [113] que avaliou o desempenho de *Encog*, *Neuroph* e *JOONE*, a biblioteca *Encog*[105] foi a que apresentou melhor desempenho, ao mesmo tempo que disponibiliza uma API simples de utilizar. Registou apenas 18 iterações para treinar a sua rede neuronal do exemplo XOR, contra as 5000 iterações da biblioteca *JOONE*, e as 613 iterações da biblioteca *Neuroph*.

Noutro estudo [114], um *benchmark* com 10 neurónios de *input*, 20 na única camada escondida, 10 neurónios de *output*, e com 100,000 casos de teste e 50 iterações de treino, mostrou que a *Encog* superou as bibliotecas concorrentes. A implementação da *Encog* com uma *thread* demorou 3,1 segundos, contra os 17,8 e 39,8 das bibliotecas *Joone* e *Neuroph*, igualmente suportadas por uma única *thread*, tendo sido utilizado o mesmo computador para correr o *Benchmark*.

A biblioteca *Weka* permite construir a rede através de uma interface gráfica, mas no contexto da aplicação *Android* tornava-se mais complexo o desenvolvimento. Não foi possível encontrar um estudo comparativo entre a biblioteca *Weka* e a *Encog*. Além disso, para a biblioteca *Weka* é distribuída com a licença GPL, o que significa que não pode ser utilizada em contexto comercial, ao contrário da biblioteca *Encog* que é distribuída com a licença *Apache*. Este tipo de licença permite a utilização da ferramenta em aplicações do contexto comercial.

Pretendendo-se implementar a rede neuronal num dispositivo móvel, foi escolhida a biblioteca *Encog* para implementar o mecanismo, não só pelo seu desempenho, mas também pela sua simplicidade.

2.5 Serviços Web de meteorologia

O estado meteorológico condiciona a emoção das pessoas [115]. No entanto, esta relação é difícil de delinear por ser dependente das características de cada pessoa [115]. Tendo em conta esta dificuldade em relacionar as condições meteorológicas com o estado emocional, a estratégia utilizada para inferir a emoção passa por utilizar um mecanismo de *machine learning*. Os componentes do estado da atmosfera são adquiridos recorrendo a Serviços Web que fornecem a informação meteorológica. Nesta secção, são comparados os vários serviços meteorológicos disponíveis, de forma a escolher uma API para adquirir essas informações.

1. *Open Weather Map*

O *Open Weather Map* [116] é um serviço gratuito que disponibiliza informações sobre meteorologia. Procura seguir a ideologia dos serviços *Wikipédia* e *OpenStreetMap* que defendem que a informação pode ser livre e disponível a todos. Neste serviço, são disponibilizadas informações acerca da meteorologia atual, a previsão semanal, força do vento e nebulosidade. A fonte desta informação são 40 000 estações de meteorologia espalhadas pelo mundo. O meio de comunicação com este serviço Web é através de *JavaScript Object Notation* (JSON) ou *eXtensible Markup Language* (XML). O *Open Weather Map* apresenta um limite de pedidos até 3000 por hora, utilizando a versão gratuita.

2. *AccuWeather*

A *AccuWeather*[117] fornece as informações de meteorologia através de uma API *RESTful*, com base na localização. É necessário contactar a entidade responsável por email para obter uma API *key* para comunicar com o serviço. As informações estão disponíveis em 40 línguas e a informação do serviço é recebida através do formato de texto JSON. A *AccuWeather*[117] é um serviço Web pago.

3. *The Weather Channel and Weather Underground API*

O canal de satélite americano *The Weather Channel*, em conjunto com o site *Weather Underground*, disponibilizam a informação meteorológica correspondente aos últimos 30 anos. A API tem cobertura global, e está disponível em 80 línguas.

É possível receber a informação do serviço através dos formatos de texto JSON ou XML, e ainda através de vários formatos de imagem. Podem ser feitos 500 pedidos

ao serviço por dia e 10 pedidos por minuto em ambiente de desenvolvimento. O canal de satélite americano *The Weather Channel*, em conjunto com o site *Weather Underground*, é um serviço Web pago.

4. *WeatherBug's Pulse API*

A *WeatherBug's Pulse API* é um serviço que disponibiliza informação meteorológica em tempo real, sendo os pedidos realizados através de uma API REST. Este serviço não disponibiliza nenhum plano gratuito, sendo os três planos disponíveis pagos, todos eles com um limite de 150 mil pedidos ao serviço por mês.

5. *Forecast.io*

A principal característica da *Forecast.io API* é disponibilizar informação de meteorologia de forma gratuita até 1000 pedidos ao serviço por dia. Por cada 10 mil pedidos extra é cobrado 1 dólar. A interface é baseada em JSON, e existem vários *wrappers* disponibilizados pela comunidade. É possível utilizar o serviço para fins comerciais, desde que seja dado crédito à fonte de informação, sendo requeridos links para o site do serviço sempre que é visualizada a informação.

Escolha final

Nome da API	Preço	Limite de pedidos
<i>Open Weather Map</i> (Plano gratuito)	0€	3 000 por minuto, até 4 000 por dia
<i>AccuWeather</i>	Apenas negociável	Apenas negociável
<i>The Weather Channel and Weather Underground</i>	0€	500 por dia e 10 por minuto
<i>Pulse API</i>	20\$ (14,65€) por mês	150 000 por mês
<i>Forecast.io</i>	preço unitário 1\$ (0,73€)	10 000 pedidos

Tabela 2.3: Serviços Web de meteorologia.

A escolha para adquirir informações de meteorologia recaiu sobre o serviço Web *Open Weather Map*, uma vez que a sua oferta permite até 3000 pedidos ao serviço por minuto, com um máximo diário de 4000 pedidos, sendo este um serviço gratuito. Conforme os resultados apresentados na tabela 2.3, os serviços Web concorrentes disponibilizam um número inferior de pedidos na sua oferta gratuita, ou na sua oferta mais barata.

Para comunicar com a API do *Open Weather Map* existem duas bibliotecas cliente que permitem abstrair o processo da ligação nomeadamente a *OWM Client* [118] e a *OWM Japis*[119]. Optou-se pela *OWM Japis* porque está atualizada em relação à última versão API do *Open Weather Map*.

2.6 Sensores de batimento cardíaco

A utilização de biossensores é um dos métodos de reconhecimento de emoção[120]. As emoções podem ser definidas como construções complexas com limites difusos e variações individuais significativas no que se refere à expressão e experiência[121].

Neste trabalho, descreve-se um procedimento de treino de reconhecimento de emoções para dispositivos móveis utilizando sinais provenientes de biossensores.

Em diferentes fases do desenvolvimento foram utilizados os seguintes biossensores:

- o *BITalino* cujo *output* é o ECG e outros biossinais;
- a *smart shirt VitalJacket*[®] cujos *outputs* são o ECG e o batimento cardíaco.

Estes biossensores permitem neste trabalho obter o batimento cardíaco. Segundo vários autores, o aumento do batimento cardíaco pode estar relacionado com emoções negativas[121] [122].

Existem vários biossensores de batimento cardíaco no mercado que transmitem essa informação através de Bluetooth[®], como o *Mio Alpha*[123] (relógio, preço:€151.49), o *Polar H7* [124] (bracelete, preço€60) ou o *Zephyr*[125] (bracelete, preço:€56). Porém neste trabalho pretendeu-se utilizar sensores disponibilizados no grupo de trabalho.

A *smart shirt VitalJacket*[®] é, como o nome indica, uma *t-shirt* inteligente que monitoriza de forma contínua o ECG e o batimento cardíaco. Como vantagens deste biossensor destaca-se:

- fácil utilização;
- permite liberdade de movimentos.

O *BITalino* é uma plataforma de *hardware* fácil de usar, versátil e escalável para aquisição e transmissão de sinais de ECG. Além disso, é bastante modular, uma vez que permite adicionar vários biossensores[126] para além do sensor de ECG.

Estes dois biossensores apresentam as seguintes semelhanças:

- transmitem as informações através de Bluetooth[®];
- ambos se apresentam como dispositivos não medicinais, informando os utilizadores do risco de utilizar este *hardware* para fins medicinais;
- consistem em *hardware* de baixo custo.

Comparando o *BITalino* com a *smart shirt VitalJacket*[®], destacam-se as seguintes diferenças:

- o *BITalino* não transmite o batimento cardíaco, pelo que é necessário calcular essa informação no *smartphone*.
- por outro lado, o *BITalino* permite obter vários tipos de bio-sinais (Electromiografia, temperatura, actividade electrodermal, respiração, entre outros)[126] que podem complementar o mecanismo de deteção de emoção em versões posteriores do *HappyHour*.

Inicialmente foi utilizada a *smart shirt VitalJacket*[®] por ser a única disponível. Mais tarde foi possível obter o *BITalino* para substituir a *smart shirt VitalJacket*[®].

2.7 Frameworks Web

Uma dos componentes do *HappyHour* é a plataforma Web que permite a um funcionário do estabelecimento noturno interagir com o serviço. A partir desta plataforma Web, um administrador pode:

- introduzir e editar eventos;
- enviar notificações, em tempo real, para os utilizadores que se encontrem nas imediações do estabelecimento noturno.

No início deste projeto, considerou-se que a plataforma Web existente apresentava as seguintes lacunas:

- não assentava em cima de nenhuma *framework*;
- apresentava-se numa fase inicial de protótipo.

Assim, surgiu a necessidade de desenvolver uma nova interface, mais escalável e baseada numa *framework* Web.

Uma *framework* Web permite adicionar código base e camadas de abstração para um número considerável de serviços e pré-definições. Assim, o código base permite:

- simplificar a maioria das operações no desenvolvimento Web;
- uma maior facilidade em estender o sistema com novas funcionalidades.

Existem bastantes implementações de *frameworks* Web, como, por exemplo:

- *Ruby on Rails* [127];
- *Laravel* [128];
- *Symfony* [129].

Enquanto que o *Ruby on Rails* é desenvolvido na linguagem *Ruby*, *Laravel* e *Symfony* são desenvolvidos na linguagem PHP.

As *frameworks* acima referidas são alternativas viáveis porque aplicam os seguintes conceitos e tecnologias:

- Implementam *Model-View-Controller* (MVC), que é um modelo de arquitetura que separa a informação (*Model*) da representação da informação que o utilizador recebe (*View*) e da interação com a página (*Controller*).
- Utilizam *Object-Relational Mapping* (ORM), que é uma camada de abstração da base de dados, onde uma tabela é abstraída como uma classe, e uma linha da tabela da base de dados é um objeto (da classe da tabela). Apresenta as seguintes vantagens:
 - é possível estabelecer relações entre as classes (tabelas);
 - permite realizar validações, que testam, por exemplo, se um dado utilizador preencheu o seu nome num formulário de uma página; caso contrário, é possível inviabilizar o registo no sistema, de uma forma fácil.
- A segurança está incutida por defeito nas *frameworks*, onde são seguidas convenções para a proteção contra ataques de injeção SQL, *Cross Site Request Forgery* (CSRF) e *Cross Site Scripting attacks* (XSS).

- Possibilidade de realizar migrações nas bases de dados. As migrações permitem controlar o estado de uma base de dados num dado momento, assim, é possível determinar as alterações que foram realizadas à base de dados ao longo do tempo.
- Incluem ainda a possibilidade de integrar *frameworks* de *testing*.

Apesar das *frameworks* acima referidas aplicarem estes conceitos, o *Ruby on Rails* destaca-se por ser construído com base numa linguagem altamente colaborativa. A linguagem *Ruby* é apoiada por uma vasta comunidade de programadores, sendo a segunda linguagem mais utilizada para realizar projetos, conforme os dados de número de repositórios criados em 2013 no *Github*[130] por linguagem de programação (tabela 2.4).

Posição	Linguagem	Número de repositórios criados
1	<i>JavaScript</i>	264131
2	<i>Ruby</i>	218812
3	<i>Java</i>	157618
4	PHP	114384
5	<i>Python</i>	95002

Tabela 2.4: Número de repositórios criados no *Github* [130] em 2013[131].

A verdade é que as *frameworks* PHP apresentam maior compatibilidade com o código da plataforma previamente desenvolvido. No entanto, decidiu-se adotar o *Ruby on Rails* tendo em conta as seguintes vantagens:

- é altamente colaborativa, uma vez que os programadores colaboram seguindo as melhores práticas do desenvolvimento Web;
- é fonte de inspiração para implementações de *frameworks* Web em outras linguagens;
- esta *framework* segue os princípios da linguagem *Ruby*, permitindo utilizar uma vasta quantidade de bibliotecas (*gems*) que tornam esta *framework* bastante prática, uma vez que facilitam o desenvolvimento de tarefas repetitivas, como, por exemplo, o processo de registo e *login*.

Para além do acima referido, considerou-se também a experiência em *Ruby on Rails* já demonstrada em trabalhos anteriores do mestrando Pedro Carmona.

No próximo capítulo são abordados os requisitos do projeto e do sistema *HappyHour*.

CAPÍTULO III

Requisitos

O capítulo que se segue aborda os requisitos do sistema e as aplicações desenvolvidas no âmbito do projeto *HappyHour*, com o objetivo de estender a aplicação a uma BCI, capaz de melhorar o estado emocional do utilizador. São também apresentados os requisitos funcionais e os requisitos não funcionais, seguindo o formato especificado no início das secções correspondentes.

Ao longo deste capítulo, identificam-se também as principais funcionalidades que o sistema pretende oferecer aos *stakeholders*. Pretende-se expor uma visão do sistema onde são identificados:

- *Stakeholders*;
- Atores;
- Requisitos funcionais;
- Requisitos não funcionais;
- Diagramas de Casos de Uso;
- Requisitos de qualidade.

3.1 *Stakeholders*

Os *stakeholders* são os elementos que interagem de forma direta ou indireta com o sistema, em particular, engloba:

- as pessoas responsáveis pelo desenvolvimento;

- as pessoas ou entidades que financiam ou suportam o produto;
- as pessoas que utilizam o produto[132].

<i>Stakeholder</i>	Descrição	Responsabilidade
Equipa que desenvolve o <i>HappyHour</i>	Professor Jorge Sá Silva e Engenheiro David Nunes	Monitorizar os processos de desenvolvimento e utilização.
Sapo.Labs	Empresa que colabora no projeto.	Comunicar necessidades e validar o sistema.
Utilizador Comum	Melhora o seu estado emocional e conhece novos pontos de interesse	Comunicar necessidades, permitir enviar os dados dos sensores do <i>smartphone</i> e da <i>smart shirt</i> , assim como validar o sistema.
Gestor do Ponto de Interesse	Responsável pela gestão do local (estabelecimento noturno, museu, parque, entre outros)	Fornecer dados sobre o local, sobre os eventos que organizam e promoções, além de validar o produto.

Tabela 3.1: *Stakeholders* - Descrição e Responsabilidades.

3.2 Atores

Os atores são os elementos que interagem com a aplicação.

O utilizador comum pode:

- melhorar o seu humor através de várias ações como passeios ou frequentar museus, espaços de diversão noturna;
- conhecer os locais mais indicados a visitar através do acesso a informação como afluência, agitação, emoção geral e música ambiente obtida por meio do sistema e da aplicação *Android*;
- partilhar de forma invisível os dados dos sensores do telemóvel com o sistema.

Os gerentes dos locais são responsáveis por participar na geração de conteúdos que dinamizam o sistema e incentivam o utilizador comum a utilizá-lo.

3.3 Requisitos funcionais

Esta secção procura identificar e classificar os requisitos funcionais do sistema.

As *user stories* são definições de alto nível dos requisitos, escritas em linguagem não técnica, que permitem caracterizar o fluxo do sistema à medida que o ator interage com a aplicação. Contém a informação necessária para que o analista consiga estimar o tempo necessário para desenvolver esse requisito [133].

O estudo dos requisitos realizou-se segundo os parâmetros apresentados na tabela 3.2.

Prioridade	<ul style="list-style-type: none">• <i>Must</i> Requisitos obrigatoriamente implementados• <i>Should</i> - Requisitos que devem ser implementados, mas caso não sejam implementados, o sistema deve funcionar.• <i>Could</i> - Requisitos apenas implementados se não afetarem a implementação dos requisitos mais importantes.• <i>Won't</i> - Requisitos não implementados durante o projeto, considerados como trabalho futuro.
Descrição	Exposição geral do requisito.
Atores	Sistemas que intervêm com o produto.
Pré-condições	Estado do sistema antes da realização da ação.
Fluxo dos eventos	Descrição das ações que ator necessita de cumprir para chegar ao resultado esperado.
Resultado esperado	Resultado esperado com a implementação do requisito.

Tabela 3.2: Parâmetros dos requisitos.

Em seguida, é apresentada a lista de requisitos para a Plataforma Web e para a Aplicação Móvel.

3.3.1 Lista de requisitos para a Plataforma Web:

- Associar um ponto de interesse à conta do utilizador da plataforma Web *HappyHour*.
- Associar uma conta de uma rede social à conta do utilizador da plataforma Web *HappyHour*.
- Partilha de eventos e notificações em redes sociais.
- Remover uma conta de uma rede social associada à conta do utilizador da plataforma Web *HappyHour*.
- Partilha de eventos e notificações direcionadas a utilizadores com uma determinada emoção nas imediações do ponto de interesse.
- Gerir a lista de reprodução de um determinado ponto de interesse.

3.3.1.1 Associar um ponto de interesse à conta do utilizador da plataforma Web *HappyHour*

Prioridade: *Must*

Descrição: Ao registar um ponto de interesse na plataforma, o utilizador passa por vários passos.

Atores: Gestor do ponto de interesse.

Pré-condições:

- Fazer *login* na aplicação com o serviço externo *Facebook*.
- Deve existir uma página do ponto de interesse no serviço externo *Facebook*.
- Deve existir o registo do ponto de interesse no serviço externo *Foursquare*.

Fluxo de eventos:

- O utilizador utiliza o botão criar ponto de interesse para associar um ponto de interesse.

- Nesse momento, são apresentadas as páginas que o utilizador possui na rede social *Facebook*, pelo que o utilizador deve escolher a página do ponto de interesse no *Facebook*. Este passo controla o acesso à gestão do ponto de interesse na plataforma *HappyHour*:
 - Se já existir um registo do ponto de interesse por outro utilizador então desde que o atual utilizador tenha permissões de gestor da página do *Facebook*, é cedido o acesso à gestão do ponto. Neste caso os passos posteriores deste fluxo de eventos não são realizados.
- Mais à frente, são apresentados os pontos de interesse existentes no *Foursquare* nas imediações da localização devolvida com os dados devolvidos pelo *Facebook*. Alternativamente o utilizador pode inserir o identificador deste ponto de interesse.
 - Apenas pode existir uma página de *Facebook* associada a um ponto de interesse, caso o utilizador tente associar uma nova página a um ponto com uma página de *Facebook* previamente associada, deve contactar a gerência do *HappyHour*.
- Para finalizar o utilizador pode corrigir os dados obtidos através das duas plataformas, nomeadamente o nome e a descrição.

Resultado esperado:

- O ponto de interesse é associado e pode ser consultado através do menu lateral da aplicação Web.

3.3.1.2 Associar uma conta de uma rede social à conta do utilizador da plataforma Web *HappyHour*

Prioridade: *Must*

Descrição: Pretende-se adicionar uma componente social ao sistema, de modo a que seja possível ao gestor de um local de interesse publicar eventos e enviar notificações aos utentes dos seus locais através das redes sociais. Para isso é necessário associar uma conta da rede social (*Facebook*, *Twitter*, *Google+*) à conta do existente na plataforma *HappyHour*.

Atores: Gestor do ponto de interesse.

Pré-condições:

- Fazer *login* na aplicação.
- Se tiver vários pontos de interesse, escolher o local pretendido, caso contrário é selecionado automaticamente.

Fluxo de eventos:

- O utilizador acede ao seu perfil.
- Associar uma conta da rede social à sua conta.

Resultado esperado:

- A conta é associada ao perfil e essa ligação pode ser consultada na página de perfil do utilizador.

3.3.1.3 Partilha de eventos e notificações em redes sociais**Prioridade:** *Must*

Descrição: Ao publicar eventos e notificações na plataforma *HappyHour* o gestor do local pode selecionar uma opção para também publicar o evento/notificação nas redes sociais (*Facebook, Twitter, Google+*).

Atores: Gestor do ponto de interesse.

Pré-condições:

- Fazer *login* na aplicação.
- Se tiver vários pontos de interesse, escolher o local pretendido, caso contrário é selecionado automaticamente.
- Ter associado uma conta da rede social à sua conta, associada no momento do registo.

Fluxo de eventos:

- Criar um evento/notificação.

- Escolher as redes sociais onde vai ser publicado.
- Enviar a notificação/Criar o evento.

Resultado esperado:

- A notificação é partilhada na rede social.

3.3.1.4 Partilha de eventos e notificações direcionadas a utilizadores com uma determinada emoção nas imediações do ponto de interesse

Prioridade: *Could*

Descrição: A plataforma existente já notificava os utilizadores que se encontram num dado momento nas imediações do local de interesse. Este requisito possibilita aos gestores dos pontos de interesse partilhar notificações e eventos direcionadas a utilizadores que estão a sentir uma determinada emoção (ansiedade, aborrecimento, calma e euforia).

Atores: Gestor do ponto de interesse.

Pré-condições:

- Fazer *login* na aplicação.

Fluxo de eventos:

- Criar um evento/notificação.
- Escolher os estados emocionais associados aos utilizadores que pretende notificar.
- Enviar a notificação/Criar o evento.

Resultado esperado:

- A notificação é partilhada com os utilizadores que se encontram nas imediações e estão a sentir aquela emoção.

3.3.1.5 Remover uma conta de uma rede social associada à conta do utilizador da plataforma Web *HappyHour*

Prioridade: *Should*

Descrição: Caso o utilizador pretenda remover uma conta de uma rede social associada (*Facebook, Twitter, Google+*), deve ser possível remover essa ligação.

Atores: Gestor do ponto de interesse.

Pré-condições:

- Fazer *login* na aplicação.
- Escolher o local pretendido.
- O utilizador acede ao seu perfil.

Fluxo de eventos:

- Remover uma conta da rede social associada à sua conta.

Resultado esperado:

- A conta deixa de estar associada ao perfil e essa ligação deixa de estar na página de perfil do utilizador.

3.3.1.6 Gerir a lista de reprodução de um determinado ponto de interesse

Prioridade: *Could*

Descrição: O utilizador e gestor de um ponto de interesse pode remover uma música da lista de reprodução do ponto de interesse.

Atores: Gestor do ponto de interesse.

Pré-condições:

- Fazer *login* na aplicação.
- Escolher o ponto de interesse pretendido.

Fluxo de eventos:

- O utilizador acede ao seu perfil.
- Remover uma música da lista de músicas identificadas pela aplicação móvel.

Resultado esperado:

- A conta deixa de estar associada ao perfil e essa ligação deixa de estar na página de perfil do utilizador.

3.3.2 Lista de requisitos para a Aplicação Móvel:

- Identificar a emoção do utilizador.
- Corrigir a emoção detetada.
- Definir uma política de privacidade para cada emoção (ansiedade, aborrecimento, calma ou euforia).
- Aplicar a partilha da localização no *Facebook* consoante a emoção.
- Definir uma política de definições de rede para cada emoção.
- Aplicar a utilização dos interfaces do *smartphone* consoante a emoção.
- Consultar as emoções detetadas.
- Ativar/desativar a identificação de músicas pelo *smartphone* do utilizador.
- Identificar automaticamente a música reproduzida num ponto de interesse.
- Consultar músicas reproduzidas num estabelecimento.
- Consultar músicas identificadas pelo telemóvel.
- Receber notificações de pontos de interesse com base na emoção.

3.3.2.1 Identificar a emoção do utilizador

Prioridade: *Must*

Descrição: Pretende-se adicionar uma componente de identificação da emoção do utilizador. Esta componente permite implementar o modelo de HiTL desenvolvido pelo aluno de Doutoramento David Nunes.

Para isso é necessário implementar um mecanismo de identificação.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- A aplicação móvel está iniciada.
- O utilizador tem sessão iniciada.

Fluxo de eventos:

- A aplicação recebe os valores dos sensores do telemóvel (microfone, acelerómetro) e da *smart shirt VitalJacket*[®] ou *BITalino* (batimento cardíaco).
- A aplicação recebe os valores da nebulosidade, probabilidade de precipitação e temperatura, através de um serviço Web de meteorologia.
- A aplicação processa e normaliza os valores dos sensores.
- O valores dos sensores processados são combinados com base num mecanismo de *machine learning*.

Resultado esperado:

- A aplicação identifica a emoção do utilizador (Ansioso, Calmo, Eufórico ou Aborrecido)
- A aplicação alerta o utilizador que foi inferida uma emoção, através do sistema de notificações da plataforma móvel.

3.3.2.2 Corrigir a emoção detetada

Prioridade: *Must*

Descrição: O utilizador é notificado que uma emoção foi inferida, devendo, em seguida, proceder à correção da emoção, para que o mecanismo de *machine learning* aprenda com o utilizador.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- A aplicação móvel está iniciada.
- O utilizador tem sessão iniciada.
- O utilizador recebe uma notificação de que foi inferida uma emoção, através do sistema de notificações da plataforma móvel.

Fluxo de eventos:

- O utilizador abre a notificação do *HappyHour* presente no sistema de notificações.
- A aplicação móvel mostra uma interface gráfica que permite ao utilizador corrigir a emoção detetada pela aplicação.
- O utilizador corrige a emoção, arrastando o círculo desde a emoção detetada pela aplicação até à emoção sentida pelo utilizador, e regista a emoção através de um botão de confirmação.

Resultado esperado:

- A correção da emoção deve alterar definições da aplicação e dos interfaces de rede do telemóvel.
- No caso de o estado do utilizador ter sido alterado, devem ser revertidas as definições associadas ao estado emocional anterior, e aplicadas as novas definições.
- Quando uma emoção negativa (Aborrecido ou Ansioso) é detetada:
 - É proposto ao utilizador que mude de ambiente.
 - Apresenta-se uma lista de possíveis locais de interesse que pode visitar.
 - A identificação da emoção é realizada mais frequentemente (em intervalos de 10-15 minutos), demonstrando preocupação pelo estado do utilizador.
- Quando a emoção é neutra ou positiva (Calmo ou Eufórico):
 - A identificação da emoção é realizada com uma frequência normal (em intervalos de 30-50 minutos).

3.3.2.3 Definir uma política de privacidade para cada emoção (ansiedade, aborrecimento, calma ou euforia)

Prioridade: *Must*

Descrição: Este requisito visa trazer para a aplicação *HappyHour* uma componente social, sendo possível definir uma política de privacidade que determina a partilha da localização atual do utilizador consoante a emoção que está a sentir. Assim, este requisito permite ao utilizador especificar a política de privacidade que está associada a uma emoção, podendo determinar em quais as emoções é que é partilhada a localização.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- Iniciar a aplicação.
- Utilizar o menu de opções das emoções.

Fluxo de eventos:

- Atribuir a uma emoção uma política de privacidade.

Resultado esperado:

- Deve ser perceptível a política de privacidade associadas a uma dada emoção, ou seja se a partilha de localização está ativada ou desativada para essa emoção.

3.3.2.4 Aplicar a partilha da localização no *Facebook* consoante a emoção

Prioridade: *Must*

Descrição: Uma vez que uma política de privacidade está associada a uma emoção, quando o utilizador estiver a sentir uma emoção deve ser aplicada essa política de privacidade.

Pré-condições:

- Iniciar a aplicação.
- Ter uma política de privacidade associada a uma emoção.
- Ter sido detetada e corrigida uma emoção.

Fluxo de eventos:

- Aplicar a política de privacidade.

Resultado esperado:

- Se a partilha de localização estiver atribuída à emoção atual, deve ser publicada a localização do utilizador no *Facebook*.
- Se a partilha de localização não estiver atribuída à emoção atual, não deve ser publicada a localização do utilizador no *Facebook*.

3.3.2.5 Definir uma política de definições de rede para cada emoção

Prioridade: *Must*

Descrição: Na aplicação *HappyHour* deve ser possível atribuir a uma emoção identificada vários interfaces de rede. Assim, quando for detetada uma emoção negativa (Aborrecido ou Ansioso) poderão ser utilizados vários interfaces de rede, caso o utilizador assim o pretenda.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- Iniciar a aplicação.
- Utilizar o menu de opções das emoções.

Fluxo de eventos:

- Definir se o interface WI-FI está ativo para uma emoção.
- Definir se o interface 3G/4G está ativo para uma emoção.
- Se ambos os interfaces estiverem ativos para uma dada emoção, definir o interface prioritário.

Resultado esperado:

- Deve ser perceptível quais as configurações associadas a uma dada emoção.

3.3.2.6 Aplicar a utilização dos interfaces do *smartphone* consoante a emoção

Prioridade: *Must*

Descrição: Quando a emoção do utilizador for atualizada deve ser aplicada a política de rede associada a essa emoção, sendo utilizados os interfaces de rede atribuídos a essa política.

Pré-condições:

- Iniciar a aplicação.
- Ter uma política de definições de rede associada a uma emoção.
- Ter sido detetada e corrigida uma emoção.

Fluxo de eventos:

- Se existir alguma configuração de interfaces ativada, essa configuração é substituída pela configuração associada à emoção atual. Caso contrario deve ser utilizada a configuração dos interfaces de rede associados à emoção atual.

Resultado esperado:

- Se dois interfaces estiverem atribuídos à emoção atual, devem ser utilizados os dois interfaces para fazer uma ligação à Internet.
- Se apenas um dos interfaces estiver atribuído à emoção atual, apenas deve ser utilizado esse interface para fazer uma ligação à Internet.

3.3.2.7 Consultar as emoções detetadas

Prioridade: *Must*

Descrição: Na aplicação *HappyHour* deve ser possível consultar as emoções identificadas, assim como a emoção que o utilizador introduziu através da interface gráfica de correção. Sendo o *HappyHour* uma aplicação da área da BCI, procura-se mostrar ao utilizador a alteração das suas emoções ao longo do tempo.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- Iniciar a aplicação.

Fluxo de eventos:

- Clicar na tecla de menu.
- Clicar no item que representa a lista de emoções.

Resultado esperado:

- Deve ser apresentada a lista de emoções detetadas com as respetivas respostas do utilizador, assim como o tempo em que foi utilizada essa emoção para definir parâmetros de privacidade e configurações de rede.

3.3.2.8 Ativar/desativar a identificação de músicas pelo *smartphone* do utilizador

Prioridade: *Should*

Descrição: O utilizador deve poder ativar e desativar o reconhecimento automático de músicas.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- A aplicação está ligada e o utilizador está autenticado.

Fluxo de eventos:

- Clicar na tecla de menu.
- Clicar no item que representa a lista de músicas identificadas.
- Selecionar o item que permite ativar ou desativar a identificação musical automática.

Resultado esperado:

- O botão da definição deve apresentar o estado (ativo/desativo) que o utilizador pretende configurar.

3.3.2.9 Identificar automaticamente a música reproduzida num ponto de interesse

Prioridade: *Should*

Descrição: Pretende-se desenvolver um mecanismo que detete a música reproduzida num ponto de interesse. Para isso, deve ser utilizada a aplicação móvel para detetar a música reproduzida

Atores: Utilizador da aplicação móvel.

Pré-condições:

- A aplicação está ligada e o utilizador está autenticado.
- O utilizador deve ativar a deteção de músicas num determinado estabelecimento.

Fluxo de eventos:

- A aplicação deteta o som e envia para à API o som registado.
- A resposta da API é guardada numa base de dados local, e também enviada para o servidor.

Resultado esperado:

- Deve ser possível consultar a informação da música detetada através do perfil do estabelecimento e também através da lista de músicas identificadas.

3.3.2.10 Consultar músicas reproduzidas num estabelecimento

Prioridade: *Should*

Descrição: O utilizador deve poder consultar a *playlist* musical de um ponto de interesse através do perfil do ponto de interesse.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- A aplicação está ligada e o utilizador está autenticado.

Fluxo de eventos:

- O utilizador entra no perfil do ponto de interesse através do ecrã principal, que mostra os pontos de interesse nas imediações.

Resultado esperado:

- Deve ser apresentada a lista musical reproduzida no ponto de interesse.

3.3.2.11 Consultar músicas identificadas pelo telemóvel

Prioridade: *Should*

Descrição: O utilizador deve poder consultar a *playlist* musical que foi reconhecida pelo seu telemóvel.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- A aplicação está ligada e o utilizador está autenticado.

Fluxo de eventos:

- Clicar na tecla de menu.
- Clicar no item que representa a lista de músicas identificadas.

Resultado esperado:

- Deve ser apresentada a lista musical identificada pelo *smartphone* do utilizador.

3.3.2.12 Receber notificações de pontos de interesse com base na emoção

Prioridade: *Could*

Descrição: O utilizador recebe notificações de pontos de interesse nas imediações de acordo com a emoção que sente num determinado momento.

Atores: Utilizador da aplicação móvel.

Pré-condições:

- A aplicação está ligada e o utilizador está autenticado.

Fluxo de eventos:

- Consultar notificações através do menu notificações da sistema *Android* ou através do menu de notificações da aplicação *HappyHour*.

Resultado esperado:

- Deve ser apresentada a lista de notificações de locais de interesse nas imediações. Esta lista apenas contém notificações direcionadas para os estados emocionais (ansiedade, aborrecimento, calma e euforia).

3.4 Diagramas de Casos de Uso

Os diagramas de caso de uso são utilizados para explicar as funcionalidades do produto. Foram criados dois diagramas para representar os casos de uso da aplicação móvel e da aplicação Web.

3.4.1 Diagramas de casos de uso da aplicação móvel

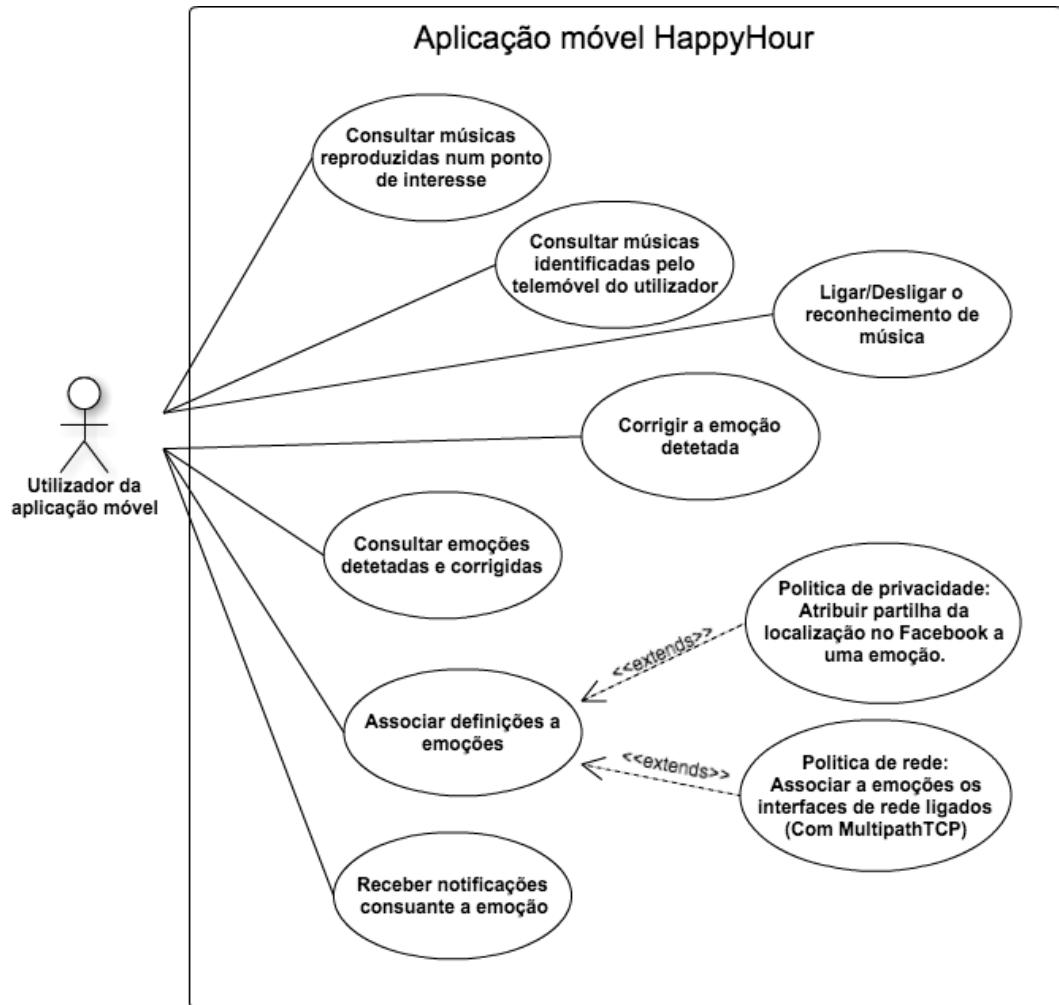


Figura 3.1: Diagrama de casos de uso da aplicação móvel.

3.4.2 Diagramas de casos de uso da aplicação Web

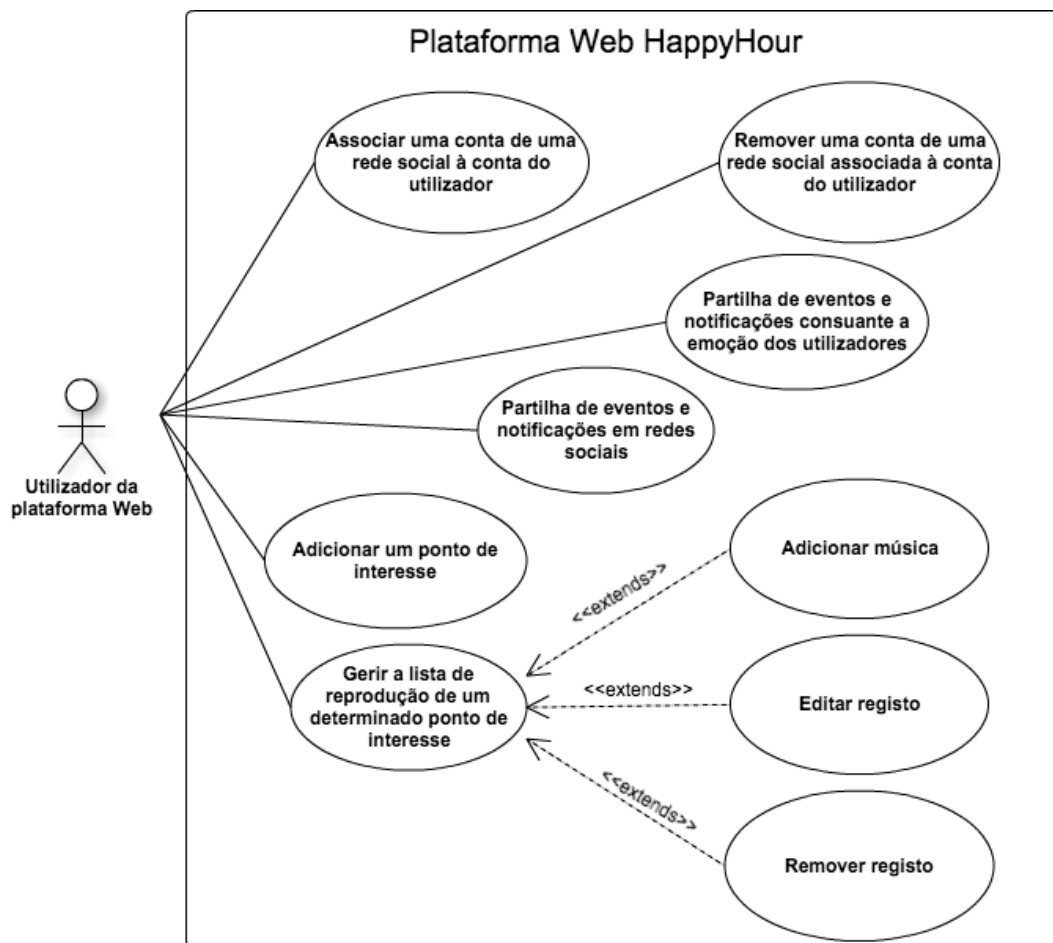


Figura 3.2: Diagrama de casos de uso da aplicação Web.

3.5 Requisitos não funcionais

Os requisitos não funcionais são identificados e categorizados seguindo as seguintes categorias:

- Usabilidade - Determina a estética das interfaces do ponto de vista do utilizador.
- Confiança - Disponibilidade das aplicações e capacidade de recuperação a falhas.
- Desempenho - Tempo de resposta das interfaces na aplicação móvel e Web.
- Suporte - Determina características como compatibilidade, capacidade de teste e configuração, durabilidade e outros.

3.5.1 *User Friendly*

Descrição: Os conteúdos das aplicações (móvel e Web) deverão ser disponibilizados ao utilizador de forma prática e intuitiva.

Prioridade: *Must*

Tipo: Usabilidade

3.5.2 Compatibilidade

Prioridade: *Must*

Tipo: Suporte

Descrição: A aplicação móvel *HappyHour* deverá ser compatível com o sistema operativo *Android* 2.3 e com versões superiores do mesmo sistema, garantindo assim uma compatibilidade de 99,2% com os dispositivos com acesso à loja de aplicações da *Google*[134].

3.5.3 Manutenção

Prioridade: *Must*

Tipo: Suporte

Descrição: O código fonte, as configurações dos sistemas e as configurações

de desenvolvimento devem estar documentadas para que, no futuro, outros programadores possam continuar o projeto.

3.6 Restrições técnicas

As restrições técnicas do projeto incluem parâmetros que limitam o desenvolvimento das várias plataformas.

A aplicação móvel deve ser escrita para plataformas *Android*, e o *smartphone* deve possuir as seguintes características:

- (*Must*) GPS, por meio do qual a aplicação obtém informação acerca da localização do utilizador.
- (*Must*) Sensor acelerómetro e microfone, através dos quais, a aplicação pode classificar a agitação do utilizador e a sua emoção.
- (*Should*) Ligação de dados, em ambientes sem possibilidade de ligar a redes WI-FI; para classificar a emoção do utilizador deve ser possível obter a informação meteorológica, através de uma ligação de dados.
- (*Could*) Para a gestão de vários interfaces de rede, deve ser possível utilizar o módulo *Multipath TCP* que ainda não está presente em versões oficiais do *kernel* do *Android*.

CAPÍTULO IV

Arquitetura

Neste capítulo são apresentadas as alterações à plataforma *HappyHour* resultantes do trabalho efetuado no estágio. Na primeira secção, são apresentadas as alterações ao sistema do ponto de vista do utilizador. De seguida, é exposta a arquitetura de alto nível, que permite caracterizar o sistema incluindo a aplicação móvel, o servidor central e a aplicação Web.

4.1 Visão de produto

O *HappyHour* era um sistema que permitia aos utilizadores visualizar o estado de um estabelecimento num dado momento. Com o desenvolvimento do modelo de HiTL do Eng. David Nunes, procurou-se neste trabalho utilizar o caso de estudo do *HappyHour* para implementar um mecanismo de controlo em que o ser humano é parte integrante e influencia a reação do sistema. Neste novo sistema é proposto que a aplicação móvel *HappyHour* siga uma abordagem que permita influenciar positivamente o estado emocional dos humanos, funcionando como uma aplicação de BCI, que procura corrigir comportamentos negativos. Assim, a aplicação desenvolvida procura caracterizar a emoção do utilizador, de forma a conseguir motivá-lo a neste caso combater o sedentarismo.

A aplicação móvel segue os paradigmas das aplicações *people-centric-sensing*, sendo utilizados sensores para caracterizar o contexto emocional dos humanos. Simultaneamente, é recolhida informação colaborativa para caracterizar o contexto dos pontos de interesse nas imediações e locais que o humano poderá estar interessado em visitar.

A informação que caracteriza o contexto emocional do humano é recolhida periodicamente, por meio de várias fontes, nomeadamente:

- os sensores de acelerómetro e microfone do dispositivo móvel;
- o batimento cardíaco recolhido através de uma *smart shirt VitalJacket[®]* e/ou *BITalino*;
- a informação de meteorologia através do serviço Web *Open Weather Map*.

Posteriormente, esta informação é combinada recorrendo a uma rede neuronal que infere o estado emocional do ser humano.

Quando são detetados estados emocionais negativos, após a confirmação do utilizador, é apresentada uma sugestão de visita de um ponto de interesse, sendo, de seguida, apresentado um mapa ao utilizador com pontos de interesse. Cada ponto de interesse pode ser consultado, sendo disponibilizada uma descrição, a afluência e agitação, assim como o estado emocional mais frequente e música reproduzida, caso seja um ponto de interesse com o reconhecimento de música ativo.

Ao mesmo tempo, este trabalho também adiciona uma componente que permite adquirir a música reproduzida num ponto de interesse automaticamente. Para isso, é utilizado o serviço *Echoprint* e o *The Echo Nest*.

A aplicação existente permitia consultar a afluência e a agitação. Para além disso, o projeto *HappyHour* já contava com um módulo de notificações e eventos dentro do sistema *HappyHour*. No entanto foi necessário fazer alterações em alguns dos módulos existentes para integrar as novas funcionalidades.

Este trabalho foca-se também no desenvolvimento da componente emocional. Conjugando o trabalho desenvolvido neste estágio com o trabalho existente, é oferecida ao utilizador uma escolha de locais que pode visitar e a motivação necessária para se deslocar para um novo local e melhorar o seu estado de espírito durante a caminhada. É ainda implementada uma componente social que permite aos utilizadores partilharem a sua localização através do *Facebook*.

Caso o utilizador assim o entenda, podem ser alterados os perfis associados às emoções e que determinam as ações que tornam a aplicação "consciente" da emoção do utilizador. A aplicação controla dinamicamente aspetos relacionados com a privacidade, sendo que a sua componente social de partilha de localização é adaptada ao estado emocional do utilizador. Por outro lado, as interfaces de rede utilizadas pelo dispositivo móvel

são também alteradas consoante a emoção do utilizador, para adaptar a desempenho e a redundância do sistema, conforme as suas necessidades e por forma a minimizar frustrações derivadas da conectividade intermitente.

4.2 Visão de alto nível

A arquitetura existente é equivalente a uma plataforma *mobile* comum, com um servidor *backend* e uma base de dados. Procurando melhorar a sua emoção, o utilizador interage com o sistema *HappyHour* através da aplicação *Android*. Existe ainda uma componente de *frontend* acedida através da Web, apenas disponível para gestores de pontos de interesse (por exemplo, funcionários de museus, de parques naturais ou de estabelecimentos noturnos).

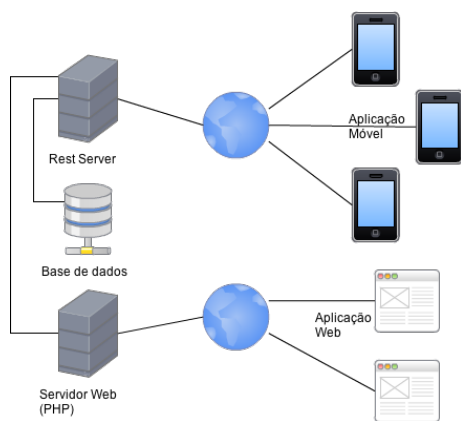


Figura 4.1: Arquitetura de alto nível do sistema existente.

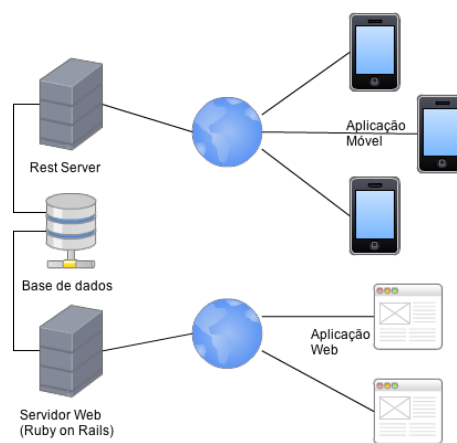


Figura 4.2: Arquitetura de alto nível do sistema em desenvolvimento.

Nas figuras 4.1 e 4.2 é possível identificar as diferenças de alto nível na arquitetura, onde foi utilizada a *framework Ruby on Rails* para implementar o novo interface Web.

A antiga plataforma Web (PHP) recorria ao servidor REST para obter os dados dos utilizadores e pontos de interesse. Por um lado, teria sido possível ligar a plataforma Web mais recente ao servidor REST, no entanto, optou-se por utilizar a utilização a ligação direta à base de dados uma vez que apresenta as seguintes vantagens:

- Permite simplificar o desenvolvimento de novos módulos na plataforma Web, tendo em conta que o projeto *HappyHour* é, na sua essência, uma prova de conceito do LCT-WSN. No modelo anterior era necessário desenvolver métodos

do lado do servidor REST, que tornavam o processo de desenvolvimento pouco ágil;

- Ao mesmo tempo, permite especializar o módulo do servidor REST para receber apenas os pedidos dos *smartphones*, simplificando a lógica neste servidor.

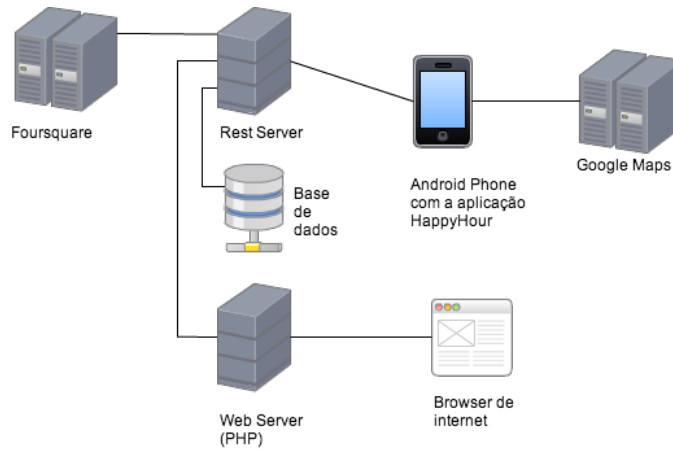


Figura 4.3: Arquitetura de alto nível do sistema existente com serviços externos.

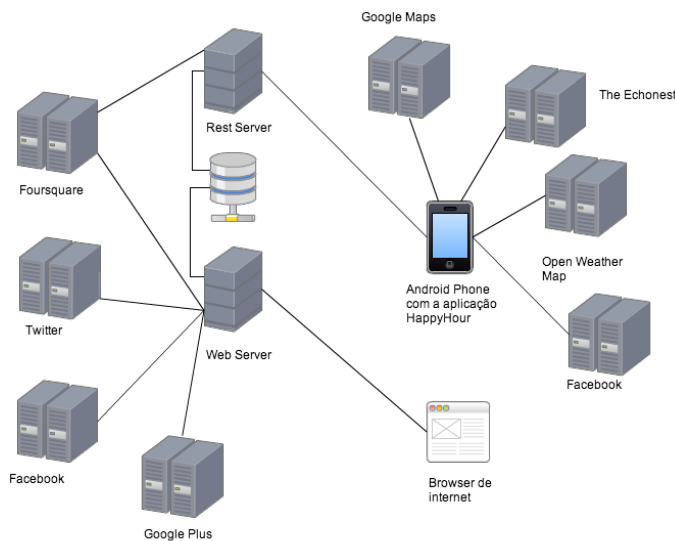


Figura 4.4: Arquitetura de alto nível do sistema em desenvolvimento com serviços externos.

As figuras 4.3 e 4.4 permitem ilustrar o desenvolvimento do sistema. No sistema previamente existente eram utilizados os seguintes serviços externos:

- O *Foursquare* para obter pontos de interesse.
- O serviço *Google Maps* utilizado na aplicação móvel para mostrar a localização dos pontos de interesse.

Na plataforma mais recente são adicionados os seguintes serviços externos:

- Redes sociais (*Facebook*, *Twitter* e *Google+*) para partilhar eventos e notificações, onde são publicadas estas informações nas diversas páginas de cada ponto de interesse. As ligações a estes serviços foram desenvolvidas no primeiro semestre.
- O *Open Weather Map* que disponibiliza informação meteorológica utilizada na aplicação móvel para inferir a emoção do utilizador. Corresponde a trabalho realizado no segundo semestre.
- O serviço de identificação de músicas denominado *The Echo Nest*, utilizado na aplicação móvel para determinar a música reproduzida num determinado local. Implementado no segundo semestre.
- Uma componente de partilha de localização na aplicação móvel, onde foi utilizada a rede social *Facebook* para partilhar a localização. Corresponde a trabalho realizado no segundo semestre.

4.3 Aplicação móvel

A anterior versão da aplicação móvel existente englobava a interação do utente de estabelecimentos noturnos e a fonte de dados do sistema *HappyHour*. Os dados dos sensores eram enviados para o servidor em pedidos REST.

A arquitetura da aplicação móvel contém um processo e um serviço. O processo é responsável por mostrar interfaces aos utilizadores, assim como por responder às ações do utilizador. O serviço é uma componente que processa informações em *background*.

O processo permite mostrar a informação colaborativa sobre os vários pontos de interesse ao utilizador através de uma interface. Esta interface é construída com recurso à biblioteca do *GoogleMaps*, e já estava presente na aplicação existente. Na nova aplicação são adicionados interfaces ligados à componente emocional, e alterados alguns dos interfaces existentes para disponibilizar mais informações aos utilizadores.

Na aplicação existente, o serviço era utilizado para determinar a posição e a agitação do utilizador através dos sensores e enviar essa informação para o servidor REST. Para além disso, é também utilizado para receber informações agregadas do servidor, como a agitação e a afluência aos pontos de interesse.

Na nova arquitetura, este serviço tem mais funcionalidades.

- O serviço recolhe informações de vários sensores (acelerómetro, microfone, *smart shirt* ou *BITalino*) e de um serviço Web externo de informação meteorológica.
- O serviço também processa as informações dos sensores, e, recorrendo a um mecanismo de *machine learning*, combina as informações e infere uma emoção.
- Após o utilizador corrigir a emoção inferida, é processado em *background* o treino do mecanismo de *machine learning*, e guardado um registo desta emoção, localmente.
- Para além disso, este serviço também é responsável por "escutar" a música que está a ser reproduzida num determinado local, processar a informação adquirida pelo microfone e obter uma *fingerprint* da informação. Esta *fingerprint* é enviada para o serviço Web *The Echo Nest* que identifica a música. Finalmente, o serviço é também responsável por enviar esta informação para o servidor REST.
- Os dados adquiridos com os novos componentes são guardados numa base de dados local, sendo as emoções associadas a um utilizador específico.

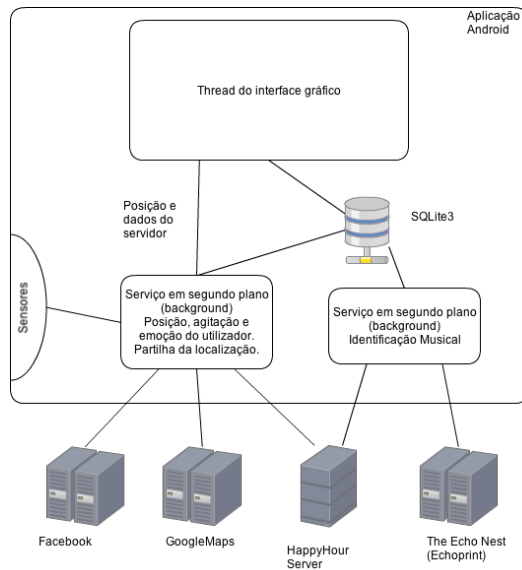


Figura 4.5: Arquitetura da aplicação móvel.

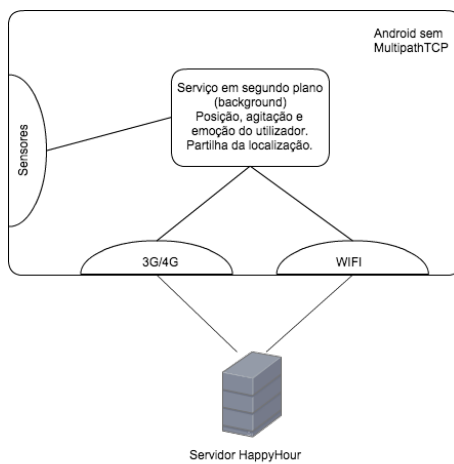


Figura 4.6: Interfaces de rede do *smartphone* sem *MultipathTCP*.

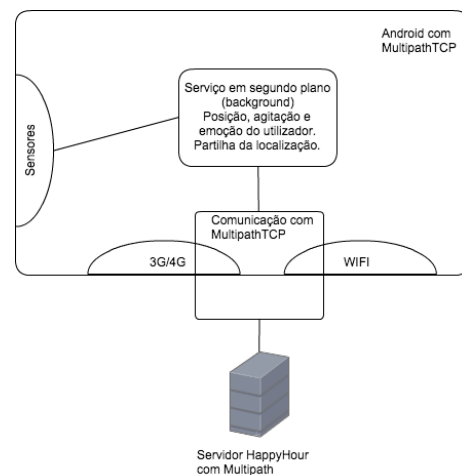


Figura 4.7: Interfaces de rede do *smartphone* com a utilização do *MultipathTCP*.

Em suma, a nova arquitetura segue o padrão do sistema já existente, mas adicionando componentes na aplicação móvel que procuram identificar a emoção do utilizador através de *machine learning*, assim como o mecanismo automático de identificação de música. Além disso é utilizada a tecnologia *MultipathTCP* para abstrair a ligação ao servidor, permitindo realizar uma ligação contínua através dos dois interfaces, como é possível verificar na imagem 4.7.

4.4 Aplicação Web

A aplicação Web é a componente da arquitetura do projeto que permite a um responsável por um ponto de interesse interagir com o serviço. As principais ações que um administrador pode executar são as seguintes:

- introdução e edição de eventos;
- envio de notificações, em tempo real, para os utilizadores que se encontrem nas imediações do estabelecimento e em função do seu estado emocional.

Com o trabalho desenvolvido no primeiro semestre do estágio, o sistema *HappyHour* adquiriu funcionalidades que exigem a ligação a APIs de redes sociais. A introdução da *framework Ruby on Rails* teve como objetivo facilitar a manutenção do projeto. Nos nossos dias, as APIs dos serviços externos utilizados pelo *HappyHour* sofrem atualizações frequentemente. Com o intuito de manter o sistema viável e competitivo, é fundamental a adopção deste tipo de *frameworks* altamente colaborativas, como o *Ruby on Rails*, suportado por uma vasta comunidade de programadores que disponibilizam bibliotecas e *wrappers* de APIs em *open source*.

A aplicação Web foi desenvolvida seguindo o modelo MVC. Quando é recebido um pedido pelo servidor Web, a *framework* interpreta o *Uniform resource locator* (URL) para decidir qual o *Controller* responsável por processar o pedido. O método do *Controller* responsável é de igual modo adquirido através do URL. Este método comunica com a base de dados através de uma abstração, chamada *Model*.

Este *Model* pode ainda conter métodos que processam a informação presente na base de dados. Para além disso, permitem validar dados introduzidos pelos utilizadores, seja pelo tipo de dados, ou pela quantidade de informação.

Após obter a informação da base de dados, o *Controller* envia as informações para a interface correspondente. Esta interface denomina-se *View*.

A ligação às APIs do *Facebook*, *Twitter* e *Google+* é realizada recorrendo a *wrappers*. Os *wrappers* encapsulam os pedidos e as respostas aos serviços Web, o que permite abstrair os pedidos REST em objetos na linguagem *Ruby*. Esta notação facilita a aquisição e a transmissão de informações para estes serviços Web, sendo estes *wrappers* utilizados na camada do *Model*.

A componente acima referida foi desenvolvida maioritariamente durante o primeiro semestre.

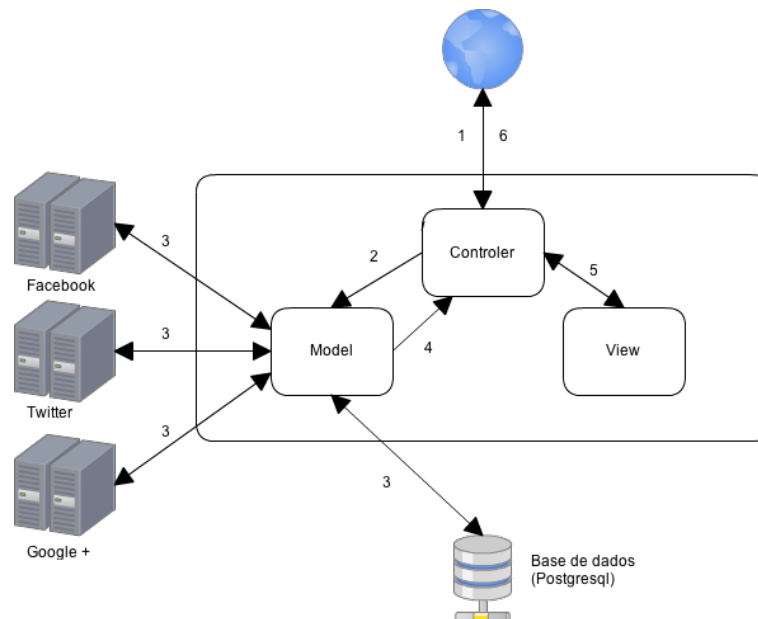


Figura 4.8: Arquitetura da aplicação Web.

4.5 Servidor REST

O servidor REST é o *Backend* da aplicação móvel *HappyHour* e utiliza a *Foursquare* API para prover a base de dados com pontos de interesse. Nesta subsecção é exposta esta componente que já existia previamente no projeto. Esta componente foi desenvolvida em Java, num servidor Apache Tomcat, e pode ser analisada a partir de três partes distintas:

- **A camada de acesso a dados:** é a componente do servidor que utiliza objetos *Data access object* (DAO) para abstrair a ligação a cada uma das tabelas da base de dados. Nesta camada são também utilizadas classes geradas pelo Hibernate, denominadas no projeto de Hibernate Maps, que permitem abstrair as instâncias de cada tabela.
- *A camada de serviços Web* interage com a aplicação móvel. A informação dos pedidos enviados pela aplicação móvel que chegam ao serviço é convertida num objeto *Request Data*. Este objeto é enviado para a camada de implementação, onde o pedido é processado. Consoante o pedido, são adquiridos dados através da camada DAO. De seguida, a camada de implementação elabora a resposta. A resposta é convertida num objeto do tipo *ResponseData* e é enviado para a aplicação móvel. É nesta camada que são atualizados os pontos de interesse através da API do *Foursquare*.

- A **Pooler**: é o serviço responsável por tratar dos dados de notificações, agitação e afluência. É executado com uma periodicidade de cinco minutos. Esta componente interage apenas com os objetos DAO correspondentes às tabelas que contêm a informação manipulada por esta.

Foi utilizada a estrutura existente para desenvolver os métodos que permitem registar as emoções e as músicas reproduzidas, e fazer a distribuição destas informações pelos dispositivos.

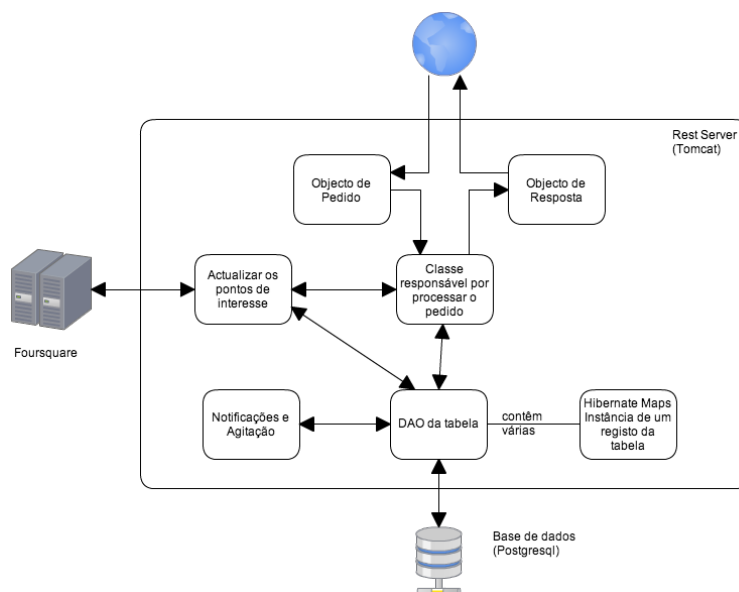


Figura 4.9: Arquitetura da aplicação REST.

4.6 Modelo de dados

4.6.1 Servidor

Os dados registados pelo servidor REST são armazenados na base de dados. No anexo F é apresentado o diagrama de Entidade-Relacionamento que descreve a base de dados. As alterações à base de dados foram realizadas recorrendo migrações. Sendo um recurso disponibilizado no *Ruby on Rails*, as migrações especificam uma alteração na base de dados. As vantagens da sua utilização são as seguintes:

- permitem arquivar o estado da base de dados num dado momento;
- permitem a propagação destes estados pelas instâncias de base de dados.

As principais alterações à base de dados são:

- A inclusão da tabela de emoções, que relaciona as emoções sentidas pelos utilizadores com um ponto de interesse. As emoções são associadas aos pontos de interesse. Ao desenhar esta tabela procurou-se não associar a emoção com o utilizador. Com esta decisão, a privacidade dos utilizadores é salvaguardada, e não é mantida qualquer ligação entre o utilizador e a sua emoção. Assumiu-se também que podem ser registadas emoções sem um ponto de interesse associado, porém essa informação apenas fica registada na base de dados e não é difundida na aplicação móvel.
- A adição da tabela de músicas, associadas a um ponto de interesse, em que cada registo inclui o nome do artista, o nome da música, e uma cadeia de caracteres que identifica a música na plataforma *The Echo Nest*.
- A adição de tabelas autorização e de gestores que permitem ao gestor do ponto de interesse iniciar a sessão com contas de serviços externos, e publicar informações nas redes sociais.

4.6.2 Aplicação móvel

A base de dados da aplicação móvel foi criada no decorrer do estágio, uma vez que era necessário registar as informações dos sensores e os resultados do mecanismo de *machine learning* que infere a emoção do utilizador.

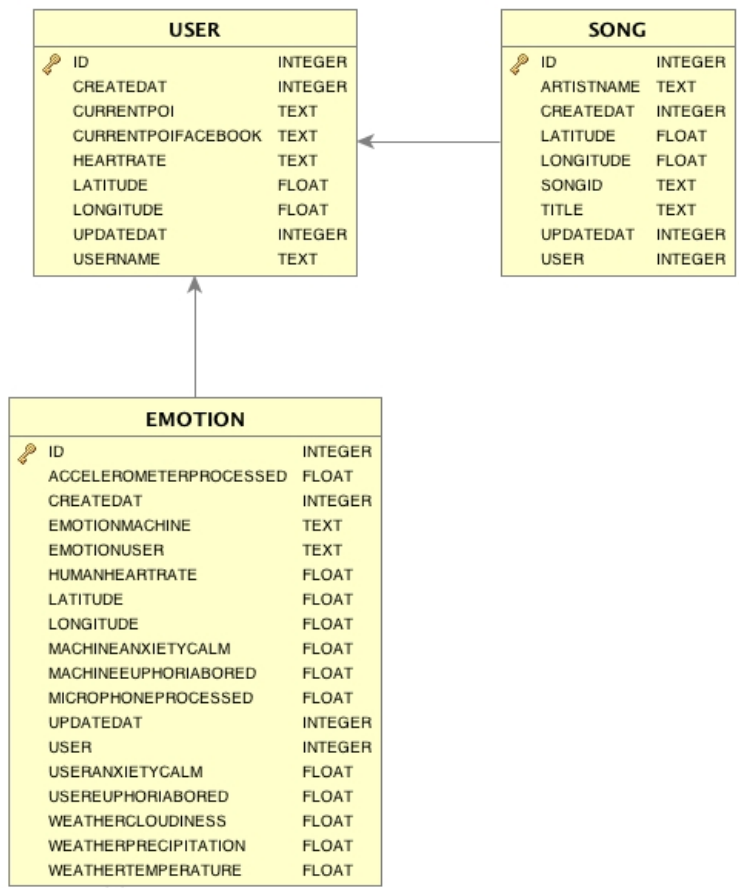


Figura 4.10: Diagrama de Entidade-Relacionamento da aplicação móvel.

Na figura 4.10 é possível observar os campos associados a uma emoção, nomeadamente os valores normalizados dos sensores e do serviço externo:

- O valor da ritmo cardíaco do utilizador;
- O valor processado do acelerómetro;
- O valor obtido através do microfone;
- O valor meteorológico da nebulosidade;
- O valor meteorológico da precipitação;
- O valor meteorológico da temperatura.

Na mesma figura também é possível observar os valores normalizados da camada de saída do mecanismo de *machine learning*:

- O valor de ansiedade e calma resultante;
- O valor de euforia e aborrecimento resultante;
- O valor de ansiedade e calma corrigido pelo utilizador;
- O valor de euforia e aborrecimento corrigido pelo utilizador.

Uma vez que o código anterior da aplicação móvel não recorria a bases de dados para guardar pontos de interesse, decidiu-se manter a estrutura de dados da aplicação, e registar apenas dados essenciais à nova componente. Assim, a tabela denominada *user* descreve alguns dados do utilizador, como o ponto de interesse onde se encontra, o identificador desse ponto na rede social *Facebook*, e o seu ritmo cardíaco atual.

A tabela de músicas regista a informação adquirida pelo *smartphone* e recorrendo ao serviço Web *The Echo Nest*.

CAPÍTULO V

Desenvolvimento e testes

Neste capítulo são abordados os desenvolvimentos e as alterações realizadas na aplicação móvel, na aplicação Web e no servidor REST. Posteriormente são apresentados os testes realizados e os seus resultados.

5.1 Desenvolvimento

5.1.1 Aplicação móvel

O público alvo da aplicação móvel engloba:

- pessoas que estão recetivas a aceitar sugestões da aplicação *HappyHour* de acordo com as emoções detetadas;
- pessoas que gostam de conhecer novos locais.

O utilizador pode escolher o seu próximo destino com base nas características dos pontos de interesse. Esta aplicação oferece uma vasta quantidade de informações sobre locais de interesse, incluindo:

- a afluência das pessoas a esse local;
- a agitação;
- as emoções das pessoas;
- as músicas reproduzidas.

Assim, estas informações colaborativas caracterizam os pontos de interesse. Com o desenvolvimento deste trabalho, o caso de estudo do *HappyHour* tornou-se numa ferramenta que também permite monitorizar o estado emocional do utilizador, e que intervém no sentido de melhorar a sua saúde e bem-estar. Como resultado, esta nova vertente da aplicação móvel *HappyHour* aborda o campo da psicologia positiva.

A psicologia positiva é um campo que se concentra nas experiências positivas do ser humano[135]:

- As experiências passadas: o bem-estar e a satisfação;
- As experiências presentes: o estado de fluxo e a alegria;
- As experiências futuras: o otimismo e a esperança.

No modelo proposto por Mihaly Csikszentmihalyi, o estado de fluxo é uma experiência intrinsecamente recompensante, e fora dos parâmetros da preocupação e do aborrecimento [136]. O Professor de Psicologia da Universidade de Pós-Graduação de Claremont, California, EUA, sugere que um individuo consegue encontrar o estado de fluxo em qualquer atividade, mesmo nas profissões que exigem tarefas repetitivas. Mihaly Csikszentmihalyi defende que o estado de fluxo é um estado subjetivo porque depende da história do indivíduo. Qualquer experiência pode criar sentimentos de ansiedade e aborrecimento, como uma visita a um museu, ou assistir a concertos de um estilo musical concreto, mas estes sentimentos são sempre dependentes de pessoa para pessoa.

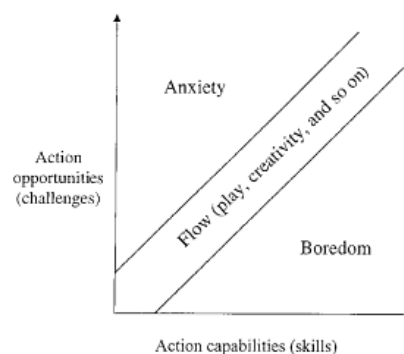


Figura 5.1: Modelo original do estado de fluxo por Csikszentmihalyi[135]

O modelo proposto por Mihaly Csikszentmihalyi procura explorar a relação da pessoa com o ambiente, ao invés de se focar apenas na pessoa. Quando uma pessoa sente

ansiedade ou aborrecimento, ela pressiona-se a adaptar o seu nível de habilidade ou o seu desafio para escapar a esse estado de adversidade e reentrar no estado de fluxo.

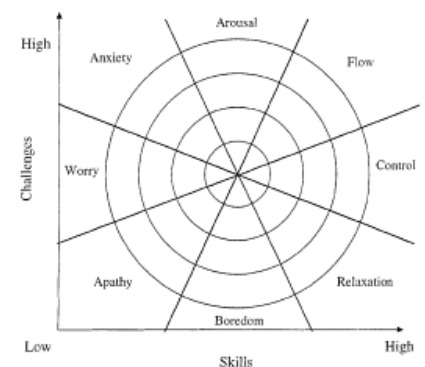


Figura 5.2: Modelo atual do estado de fluxo por Csikszentmihalyi[135]

Por outras palavras, o estado de fluxo é a experiência positiva que o ser humano sente quando o desafio e a capacidade requerida para uma determinada tarefa está acima da sua própria capacidade. Neste novo modelo são apresentados mais estados emocionais e a intensidade do estado é medida pela sua distância ao centro (figura 5.2).

Recorrendo a este modelo foi desenvolvida uma aplicação que procura caracterizar os estados emocionais do utilizador com base em sensores. Finalmente, a informação obtida permite criar um fluxo de eventos que têm como objetivo melhorar o estado emocional do utilizador.

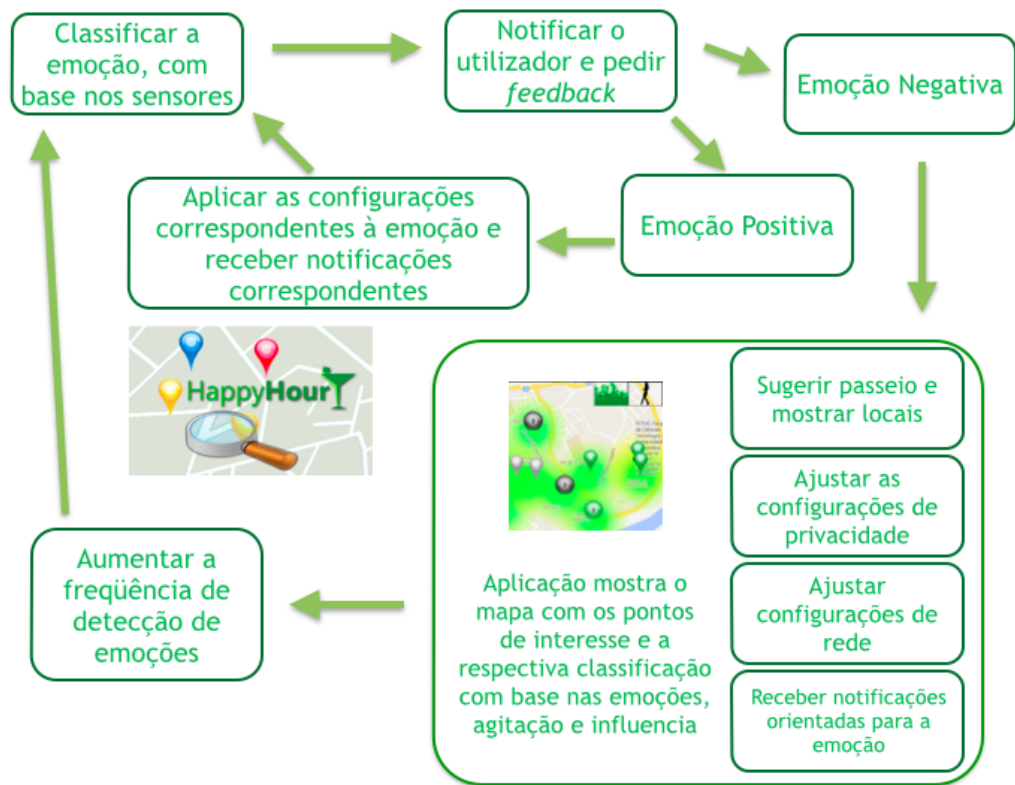


Figura 5.3: Fluxo do sistema baseado na emoção do *HappyHour*

A figura 5.3 representa o mecanismo de *loop* proposto pelo Eng. David Nunes e implementado neste trabalho. O objetivo da aplicação é influenciar o comportamento e o estado emocional dos utilizadores. Neste mecanismo, as emoções do ser humano são inferidas periodicamente. Consoante o resultado corrigido pelo ser humano, a aplicação motiva o utilizador a realizar exercício físico e a visitar locais diferentes com o objetivo de estimular um humor positivo.

A abordagem seguida para implementar este mecanismo de inferência é uma prova de conceito, com o objetivo de identificar 4 estados emocionais, que são interpretados pelo sistema como 2 pares de estados emocionais opostos:

- O estado calmo, oposto do estado ansioso;
- O estado eufórico, antagónico do estado aborrecido.

Assim, são identificadas duas emoções como negativas (aborrecido e ansioso) e duas emoções como positivas (euforia e calma). A aplicação acompanha as transições para estado de fluxo e procura ajudar o utilizador a sentir-se melhor.

A implementação deste sistema encontra-se dividida em duas partes:

- a primeira monitoriza a emoção do utilizador;
- a segunda aplica regras e medidas que permitem ao utilizador melhorar a sua emoção.

Por exemplo, as emoções negativas levam o sistema a sugerir um passeio ao utilizador. O destino do passeio pode ser escolhido através do mapa da aplicação. Simultaneamente, é ativado um sistema que se adapta dinamicamente, modificando definições de privacidade e definições de interface de ligação à Internet. A possibilidade de utilizar várias ligações à Internet é conseguida recorrendo ao *Multipath TCP*, que permite:

- reduzir a conectividade intermitente;
- aumentar a sua performance.

A verdade é que a utilização de diversas interfaces aumenta o consumo de bateria do dispositivo e os possíveis custos adicionais da ligação de dados móvel (3G ou 4G). O utilizador pode, assim, personalizar o conjunto de regras que são aplicadas a cada emoção, denominados de "perfis de emoções".

Quando o sistema está consciente de emoções positivas por parte do utilizador, são aplicadas as regras dos perfis de emoção positiva e revertidas as regras aplicadas na iteração anterior.

Monitorizar a emoção do utilizador

As figuras 5.4 e 5.5 representam a interface gráfica do sistema que permite monitorizar a emoção do utilizador.

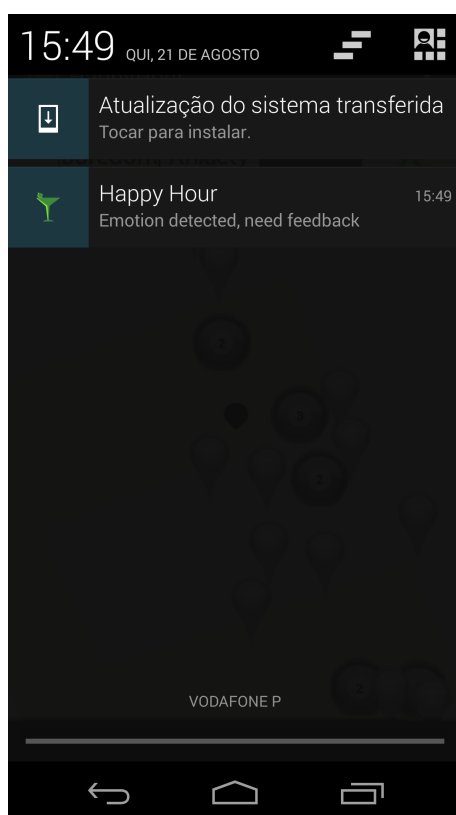


Figura 5.4: Notificação de emoção inferida.

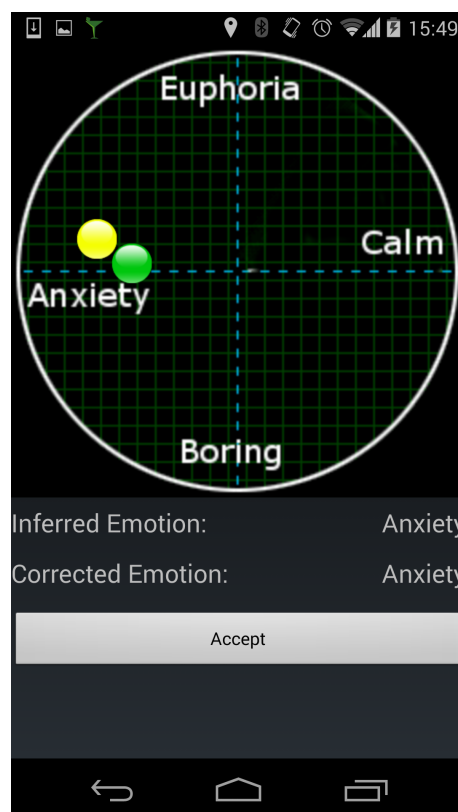


Figura 5.5: Corrigir emoção inferida na aplicação móvel.

Esta aplicação utiliza os sensores do *smartphone*, os sensores de uma *smart shirt VitalJacket*[®] ou o *BITalino* e serviços externos, com o objetivo de associar a agitação, o ruído ambiente, o batimento cardíaco do ser humano e a meteorologia à sua emoção. Por outras palavras, as informações do estado do ambiente e do utilizador são utilizadas como *input* de um mecanismo de *machine learning* e a saída deste mecanismo representa a emoção do utilizador.

Os utilizadores são notificados após ser detetada uma nova emoção (figura 5.4). Quando selecionam a notificação, é apresentada a interface de *feedback* que permite ao utilizador corrigir a emoção inferida, como é apresentado na figura 5.5. Nesta interface o utilizador arrasta a bola da emoção desde a emoção inferida até à emoção que realmente está a sentir. Quando submete a alteração, é ativado o treino do mecanismo de *machine learning* que irá refletir esta correção em futuros processos de inferência.

Perfis e regras das emoções

Quando o utilizador torna o sistema consciente da sua emoção, é iniciado um processo que altera definições e toma decisões em função da emoção do utilizador.

Se for detetada uma emoção positiva, são tomadas decisões que adaptam o sistema a esta emoção e o sistema de inferência de emoção aguarda pela próxima iteração.

Por outro lado, se for detetada uma emoção negativa, o sistema toma medidas para melhorar o estado emocional do utilizador. Esta reação permite classificar o sistema como consciente do estado emocional do utilizador, procurando influenciar o seu humor de forma positiva. Ao associar perfis com regras é possível automatizar vários processos e moldar o sistema ao utilizador. As regras podem incluir configurações de privacidade e rede.

As configurações de privacidade, no caso do *HappyHour*, permitem decidir a partilha da localização do utilizador nas redes sociais. Uma vez que cada indivíduo interage de maneira diferente com a sociedade, faz sentido que as regras de privacidade possam ser subscritas e personalizáveis. A personalização permite distinguir, por exemplo, os utilizadores que, face a um estado emocional negativo, preferem a companhia dos amigos, dos que, na mesma situação, preferem estar sozinhos. Assim, quando um utilizador estiver num estado emocional negativo, pode ser partilhada a sua localização automaticamente, ou não, conforme a sua preferência. Por outro lado, quando um indivíduo está eufórico pode também decidir se pretende partilhar a sua localização.

Com este modelo é também possível definir a utilização das interfaces de rede WI-FI e 3G/4G, permitindo a automatização através de regras para definir configurações de rede. As regras de rede dos perfis de emoção permitem configurar um sistema que utilize mais do que uma interface de rede numa única comunicação TCP. Assim, em telemóveis com um *kernel* que suporte MPTCP, é possível definir a prioridade dos interfaces de rede, assim como utilizar vários interfaces simultaneamente na mesma comunicação com um servidor. O utilizador pode então associar diferentes prioridades a dois interfaces de rede (3G/4G e WI-FI) ao perfil da emoção. Pode definir, por exemplo, a utilização de WI-FI como interface prioritária quando se sente calmo, evitando os custos associados à utilização de tráfego móvel. No entanto, se se sentir ansioso, o utilizador pode pretender utilizar simultaneamente as interfaces 3G e WI-FI, por forma a ter uma melhor performance e a evitar quebras na ligação, mesmo considerando o aumento de custos e o maior consumo de bateria das duas interfaces.

A utilização de várias interfaces simultaneamente é benéfica ao sistema *HappyHour* no caso de emoções negativas. A maior largura de banda da ligação permite um envio de uma maior quantidade de dados para o servidor, o que leva a um acompanhamento mais eficiente do estado emocional do utilizador.

Para implementar estas funcionalidades, a aplicação utiliza um processo e dois serviços:

- No processo estão implementadas as atividades responsáveis por apresentar a informação ao utilizador e por reagir às interações;
- Um dos serviços de *background* recolhe informações dos sensores, infere a emoção e troca dados com o servidor;
- O outro serviço é responsável por identificar a música no ponto de interesse onde o utilizador se encontra.

Na figura 5.6 é possível visualizar o contexto das classes desenvolvidas.

A informação colaborativa é atualizada a cada 5 minutos. Existem três serviços externos implementados nesta iteração:

- o *Open Weather Map* [116] que permite obter informações meteorológicas;
- o *The Echo Nest* que permite identificar músicas;
- o *Facebook* que permite partilhar a localização do utilizador.

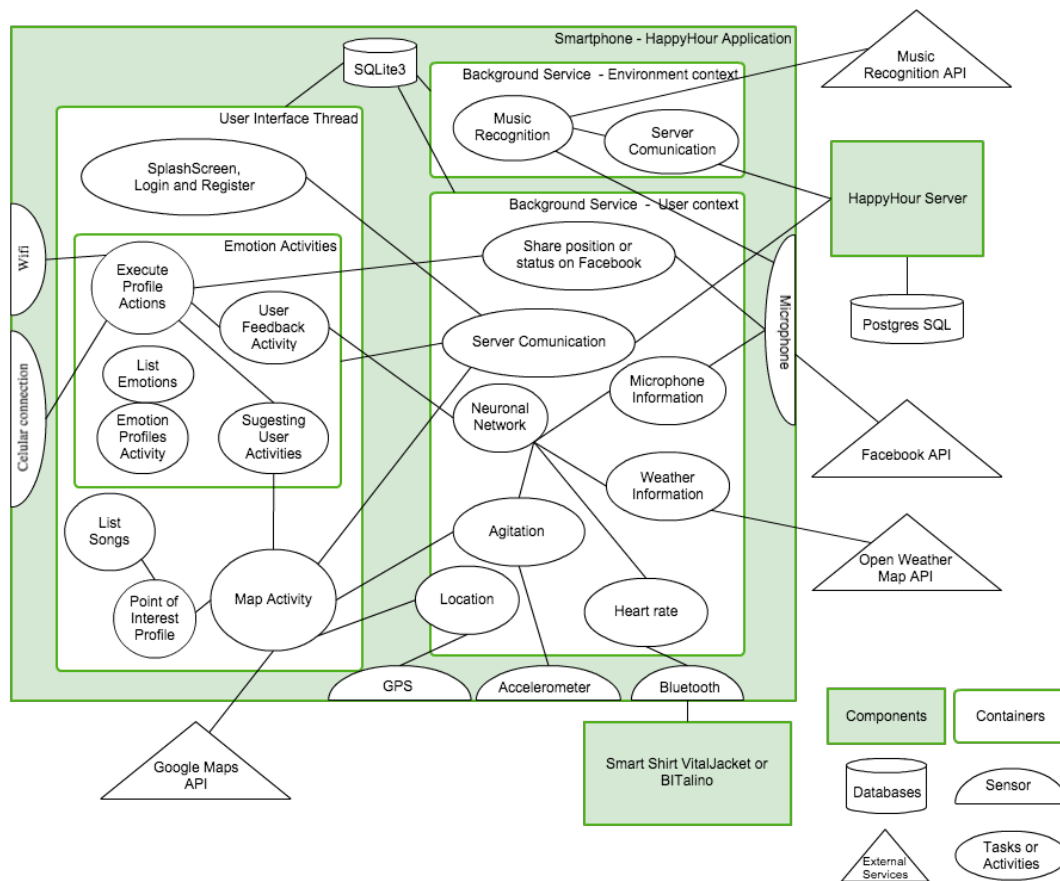


Figura 5.6: Contexto das classes na arquitetura da aplicação móvel *HappyHour*.

A aplicação previamente existente apresentava uma interface de Mapa com informações colaborativas de agitação e afluência através de *heatmaps*, e também a ligação ao servidor do *HappyHour*.

Nesta nova versão encontra-se implementado o seguinte:

- um sistema de reconhecimento de emoção;
- um sistema de identificação de músicas;
- uma base de dados (que permite guardar as informações sobre as emoções detetadas e sobre as músicas identificadas).

Como é possível observar na figura 5.6 existem várias classes pertencentes a um contexto, nomeadamente:

- As classes que processam a informação recebida pelos sensores pertencem ao contexto do serviço de *background*, e adquirem a informação da agitação, do

batimento cardíaco, da quantidade de ruído. No mesmo contexto, é também recolhida a informação meteorológica, obtida através do *Open Weather Map*.

- A informação recolhida e processada é conciliada recorrendo à classe da rede neuronal que pertence ao contexto *background*. Quando é obtida a emoção, o utilizador é notificado através do sistema de notificações nativo da plataforma *Android*.
- A atividade *User feedback* é iniciada através do sistema de notificações. Esta classe recebe a confirmação/correção da emoção efetuada pelo utilizador.
- Após a confirmação da emoção, é sugerido ao utilizador um passeio, para descobrir novos locais e promover o exercício físico. Nesta fase inicial não são propostas sugestões às emoções positivas, porém o utilizador pode beneficiar da aplicação para procurar outros locais. Para além disto, existe a possibilidade de adicionar mais casos de utilização no futuro recorrendo à classe *SugestUserActivities*.
- A classe *ExecuteProfileActions* ativa o mecanismo de partilha da localização do utilizador nas redes sociais consoante a sua emoção. Previamente, o utilizador deve aceder à *EmotionProfilesActivity* para atribuir a partilha da localização automática a determinadas emoções, de acordo com a sua preferência. Como pré-definição, a partilha da localização é atribuída à euforia.
- Os perfis de redes podem ser selecionados recorrendo à mesma *EmotionProfilesActivity* através do menu, onde o utilizador atribui a utilização de um ou mais interfaces de rede a uma emoção. Como pré-definição, são utilizados os dois interfaces de rede do telemóvel (WI-FI e 4G) para comunicar com a Internet, quando o utilizador se sente ansioso. É também na classe *ExecuteProfileActions* que são alteradas estas definições.
- A deteção de música é realizada no contexto do serviço *background*, onde a melodia é registada e processada recorrendo à biblioteca de *Echoprint*. Esta biblioteca converte o som numa *fingerprint* que o caracteriza. A *fingerprint* é comparada com as existentes no serviço *The Echo Nest*. O serviço externo devolve, posteriormente, a informação da música identificada, sendo esta classe responsável por enviar esta informação para o servidor do *HappyHour*.

5.1.1.1 Inferência da emoção através de *machine learning*

Na secção 2.4 foi apresentado um estudo que comparou diversos algoritmos de *machine learning* e permitiu classificar as redes neuronais como um bom algoritmo para utilizar em contextos *mobile*. Neste contexto de aplicações *mobile* é necessário ter em conta o tempo de processamento dos algoritmos, sendo as redes neuronais uma escolha razoável porque oferecem uma taxa de classificações certas elevada em menos tempo.

O esquema da figura 5.7 representa os *inputs* da rede neuronal e o número de camadas escondidas. Foram utilizadas 2 saídas, onde cada uma das saídas representa duas emoções: eufórico-aborrecido e agitado-calmo. Esta representação permite converter cada um dos eixos da grelha de emoções numa saída da rede neuronal. A grelha de emoções consiste numa interface que possui dois eixos e que permite ao utilizador confirmar as emoções movimentando uma bola para o sentido da que melhor caracteriza o seu estado emocional.

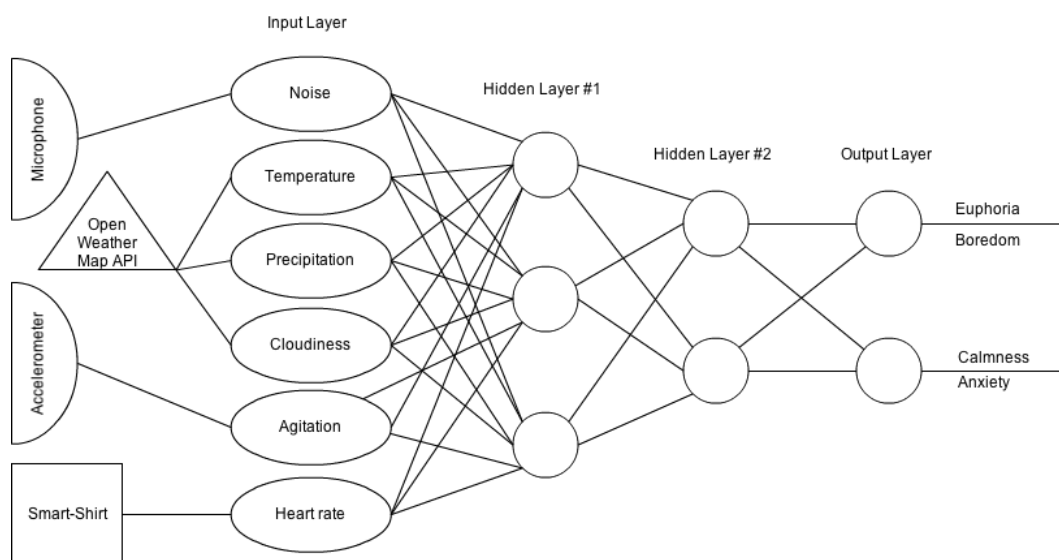


Figura 5.7: *HappyHour emotional neural network*.

A rede neuronal foi desenvolvida com uma biblioteca de algoritmos de *machine learning* denominada de *Encog*[105]. Na secção 2.4 foi realizada uma comparação entre bibliotecas similares, e foram apresentadas as razões para a escolha desta implementação.

No livro [137], Adam Blum defende que o número de neurónios da camada escondida deve estar compreendido entre o tamanho da camada de *input* e o tamanho da camada de *output*. Jeff Heaton, autor da biblioteca, descreve que apenas uma camada serve

para a maioria dos casos, e que duas camadas ocultas permitem ter em conta decisões arbitrárias[138].

Pretende-se nesta prova de conceito verificar se, através dos sensores e com mecanismos de *machine learning*, é possível inferir a emoção do utilizador. Sabendo que existem seis neurónios de entrada e dois neurónios de saída foram feitos testes para duas configurações na secção 5.2.2, com as seguintes configurações:

- uma configuração com duas camadas escondidas, a primeira com três neurónios e a segunda com dois, num total de cinco neurónios;
- outra configuração com uma camada escondida com quatro neurónios.

O método de propagação de erro utilizado foi a propagação resiliente, uma vez que é um dos métodos mais eficientes, para além de utilizar apenas o sinal do gradiente de aprendizagem, e não a sua magnitude. Na propagação original é utilizada a magnitude do gradiente da função de erro para treinar a rede, o que torna a propagação original propicia a aproximar-se para mínimos locais[139]. O erro permitido no treino da rede é 0.03. Se, no final de uma época, a rede apresentar um erro menor do que 0.03, o treino termina. Caso contrário, a rede é treinada até satisfazer essa condição.

Uma vez que esta biblioteca permite adaptar a estrutura da rede neuronal facilmente, durante a fase de testes são comparadas as duas configurações referidas acima, de forma a verificar a melhor opção.

Os atributos que são utilizados para classificar a emoção são:

- O ruído ambiente: utilizado para identificar o tipo de ambiente em que o utilizador sente emoções mais positivas, podendo estes ser ambientes com mais ou menos ruído. É feita uma soma das magnitudes registadas, e depois é normalizada essa soma entre 0 e 1. Sendo zero o valor de um ambiente quase sem ruído e um o valor de um ambiente muito ruidoso.
- Os dados meteorológicos: permitem relacionar as emoções humanas com as condições atmosféricas, que afetam as emoções dos humanos [8] [9]. , pelo que são utilizados os dados da nebulosidade, da probabilidade de precipitação e da temperatura obtidos através da *API* do *Open Weather Map*[116]. Os dois primeiros valores vêm em percentagem, tornando-se fácil convertê-los para o domínio [0.0, 1.0]. O valor da temperatura foi normalizado de -10 graus *celsius* até 40 graus.

- A informação obtida a partir do acelerómetro: analisada recorrendo à transformada de *Fourier*; as componentes são convertidas numa representação da força do movimento do utilizador, de seguida os seus valores foram normalizados. Esta informação pretende verificar se o utilizador está mais agitado ou mais calmo.
- O batimento cardíaco do utilizador: transmite informação fisiológica ao sistema. Este sinal é obtido recorrendo à *smart shirt VitalJacket*[®]. No caso do *BITalino* é necessário processar o ECG para obter o batimento cardíaco. O batimento cardíaco é normalizado a partir dos limite inferior 40 ao limite superior 170 para o domínio [0.0 , 1.0]. Todos os valores abaixo de 40 são considerados 40, e todos os valores acima de 170 são de igual modo considerados 170. Foi considerado este limite de 170 porque pode ocorrer em algumas faixas etárias [140].

5.1.1.2 Processo de inferência da emoção

O serviço de *background* responsável pelo contexto do utilizador é iniciado em simultâneo com a aplicação. Este serviço implementa uma classe denominada de *EmotionControl*, com a responsabilidade de controlar o fluxo da emoção, cabendo-lhe, também, a execução das regras associadas ao perfil emocional num determinado momento. Quando a aplicação é iniciada, esta classe não possui nenhum estado emocional associado, por isso requisita à rede neuronal uma nova inferência.

A classe da rede neuronal implementa vários métodos de inicialização. Apresenta duas classes descendentes:

- a classe *RequestNeuronalNetworkOutput*;
- a classe *NeuronalNetworkTrainManager*.

A classe *RequestNeuronalNetworkOutput* é a classe descendente responsável por solicitar a resposta ao impulso da emoção. As classes que processam e normalizam os valores dos sensores são executadas na classe *RequestNeuronalNetworkOutput* e, de seguida, é utilizada a rede neuronal para determinar a emoção do utilizador. Esta emoção é rapidamente registada na base de dados local SQLite3 [141]. No final, é devolvido um objeto *Emotion* ao serviço responsável por obter a emoção.

O utilizador é alertado através do sistema de notificações do sistema operativo *Android*. Quando o utilizador pretende supervisionar a emoção detetada, acede através desse

sistema e é apresentada a interface que permite corrigir a emoção inferida, observada na figura 5.5. Os passos seguintes do processo realizados num serviço em *background* são:

- Aplicação das regras associadas ao perfil emocional, isto é, a ativação/desativação de interfaces de rede e de partilha de localização.
- Realização do treino da rede neuronal de acordo com a supervisão do utilizador, que a uma dada inferência de emoção corrige para a emoção que o utilizador sentiu, recorrendo à classe *NeuronalNetworkTrainManager*.
- Difusão da emoção sentida, sendo esta informação enviada para o servidor, mas apenas no que se refere ao ponto de interesse, de forma a salvaguardar a privacidade dos utilizadores.

5.1.1.3 Perfis emocionais

Os utilizadores podem adaptar as configurações às suas preferências através do menu da aplicação.

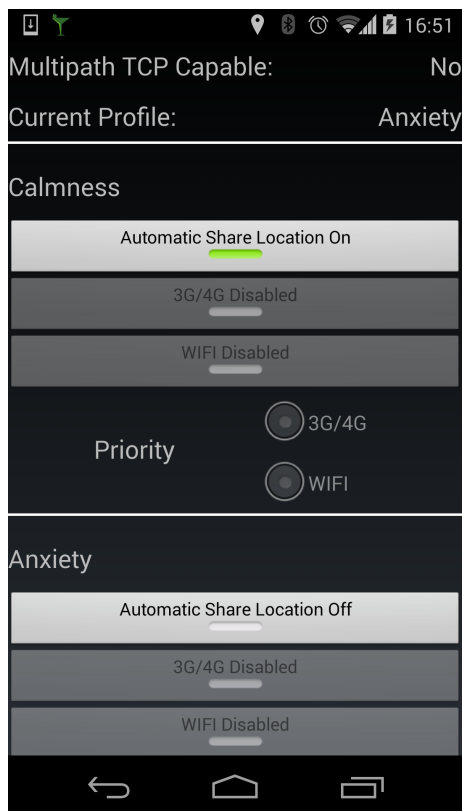


Figura 5.8: Perfil de emoções (calma e ansiedade).

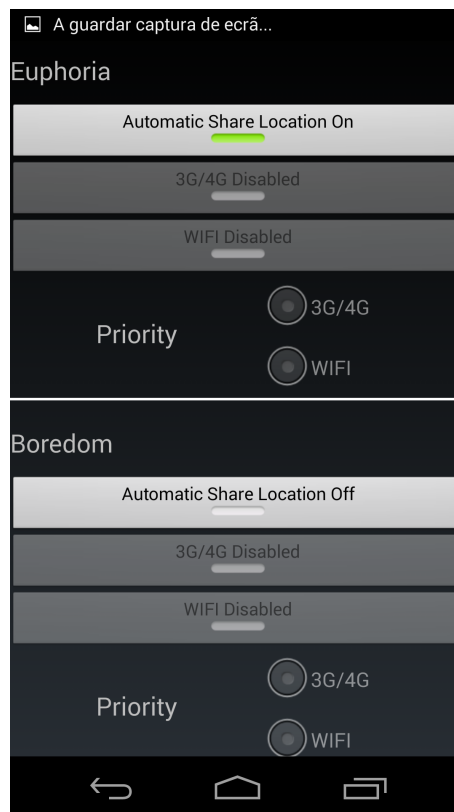


Figura 5.9: Perfil de emoções (euforia e aborrecimento).

Caso o utilizador especifique que pretende partilhar a sua localização automaticamente para uma dada emoção, então quando uma emoção é detetada é partilhada a sua localização no *Facebook*, e a implementação desta mensagem é construída recorrendo à identificação do ponto de interesse onde se encontra.

A definição de interfaces de rede apenas está disponível em telemóveis com *Multipath TCP*. Caso o telemóvel possua esta tecnologia, é mostrada essa informação neste menu. Assim, os botões dos interfaces são desativados para telemóveis sem a tecnologia, como é possível comprovar através das diferenças entre as figuras 5.8 e 5.9 (telemóvel sem *Multipath TCP*) com as figuras 5.10 e 5.11 (telemóvel com *Multipath TCP*).

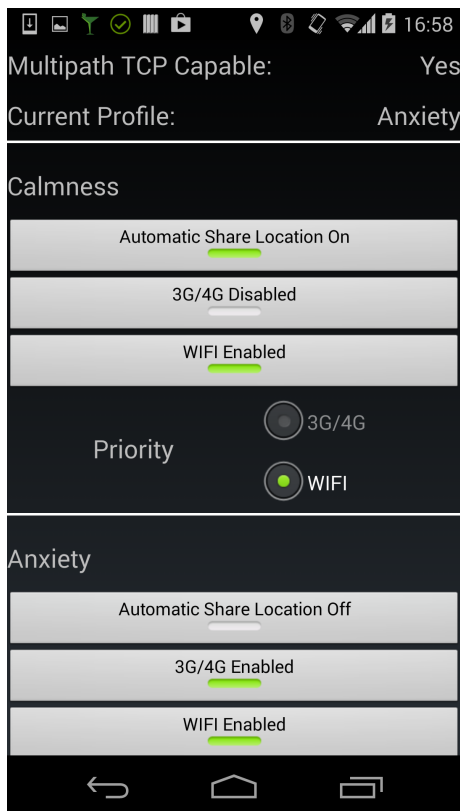


Figura 5.10: Perfil de emoções (calma e ansiedade) num telémovel com *Multipath TCP*.

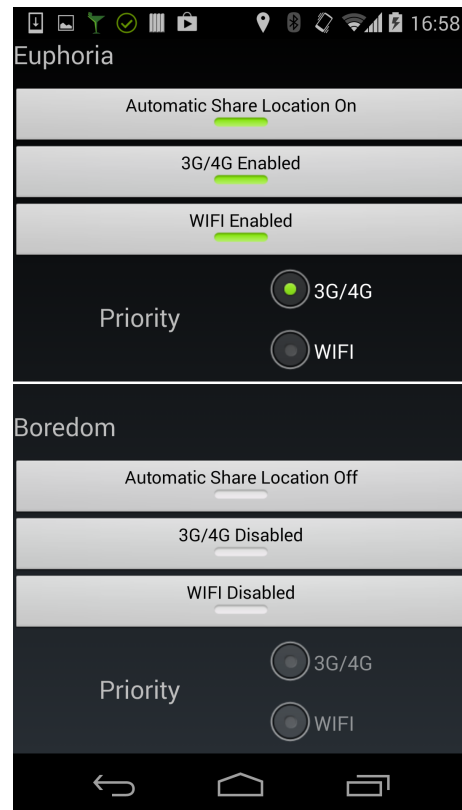


Figura 5.11: Perfil de emoções (euforia e aborrecimento) num telémovel com *Multipath TCP*.

Caso o utilizador determine os interfaces de rede que pretende utilizar quando está a sentir uma emoção, é utilizada a classe *MPTCPControl* para controlar o estado dos interfaces de rede. Esta classe necessita de um *smartphone Android* com acesso *root* e de um *kernel* específico, uma vez que esta ferramenta ainda não é disponibilizada oficialmente no *kernel* do *Android*, nem no *kernel* do *Linux*. Esta classe foi desenvolvida com base no código de um repositório [142], onde são executados comandos no sistema recorrendo à função *Runtime.getRuntime().exec()*. Estes comandos permitem:

- ativar e desativar a tecnologia *Multipath*, assim como verificar o estado atual;
- alterar e consultar o estado de cada uma das interfaces de ligação à Internet, que no caso do *Nexus 5* (telefone utilizado para o desenvolvimento), são "*rmnet0*" e "*wlan0*".

Os estados das interfaces podem ser:

- *Multipath* ativo;
- *Multipath* desligado;
- interface como *backup*;
- interface no estado *handover*.

Quando uma interface está no estado *backup* apenas se a interface principal falhar é que é utilizada esta interface.

Quando uma interface está no estado *handover* está em transição para ser utilizada como interface de comunicação. Assim, é possível definir o estado das interfaces, a partir das configurações.

5.1.1.4 Identificação das músicas de um ponto de interesse

A identificação das músicas reproduzidas num ponto de interesse é realizada no serviço de *background* destinado ao contexto do ambiente. Quando a aplicação é iniciada é utilizado um mecanismo que permite iniciar este serviço de 2 em 2 minutos.

Tendo em conta a ordem cronológica, o serviço tem como responsabilidade:

1. gravar o som ambiente;
2. obter o resultado do processamento do som gravado;
3. fazer um pedido ao Serviço Web de identificação de música;
4. enviar esta informação para o servidor e, simultaneamente, guardá-la na base de dados.

O processamento do som gravado foi implementado com a biblioteca da *echoprint* que gera o *fingerprint*. Esta biblioteca está escrita na linguagem C++[143], pelo que é necessário utilizar a *Java Native Interface* para criar uma camada de abstração que permite ao código desenvolvido em Java fazer chamadas a funções escritas na linguagem C++. O desenvolvimento deste componente foi baseado numa referência disponível *online* [101], sendo necessário ajustar alguns parâmetros, como a versão do *Codegen do Echoprint* e algumas *flags*. O processamento do sinal é realizado recorrendo à camada de abstração que, no final, devolve a *fingerprint* na estrutura de dados *String*.

Para realizar o pedido ao serviço Web *The Echo Nest*, que devolve a identificação da música, é necessário obter uma chave para comunicar com o serviço. Esta chave é enviada juntamente com a *fingerprint* da música para o serviço. O serviço responde ao pedido com nenhuma, uma ou várias músicas identificadas.

A cada uma das músicas identificadas é associado um determinado conjunto de coordenadas GPS. Estas informações são posteriormente enviadas para o servidor e guardadas localmente numa base de dados *SQLite*.



Give It to Me Dj Battle
Give It to Me Dj Battle
Numb/Encore Jay-Z, Linkin Park
In The End Linkin Park
Magdalena Various Artists
Hey Baba Rebob Lionel Hampton
A Ring On Every Finger Liars
Shut Hoover

Figura 5.12: Exemplo de uma lista de músicas identificadas pelo dispositivo.

5.1.2 Aplicação Web

A aplicação Web é utilizada pelo responsável de um ponto de interesse para introduzir/editar eventos e notificações. O utilizador pode adicionar as redes sociais do estabelecimento de forma a que as notificações sejam propagadas para estas plataformas. A nova plataforma Web, desenvolvida durante o primeiro semestre, integra as seguintes redes sociais:

- *Facebook*;
- *Foursquare*;
- *Twitter*;
- *Google+*.

Na tabela abaixo é apresentado o resultado das alterações realizadas pela introdução da nova plataforma Web.

Funcionalidade	Resultado
<i>Login com Facebook</i>	substituído
<i>Login com Foursquare</i>	adicionado
<i>Login com Twitter</i>	adicionado
<i>Login com Google+</i>	adicionado
Adicionar <i>venue</i> do <i>Foursquare</i>	substituído
Adicionar página do <i>Facebook</i>	adicionado
Publicar no <i>Facebook</i>	adicionado
Publicar no <i>Twitter</i>	adicionado
Publicar no <i>Google+</i>	adicionado

Tabela 5.1: Resultado da integração das redes sociais.

A implementação destas funcionalidades pressupôs o registo no site de cada uma das redes sociais. A utilização de bibliotecas (*gems*) que abstraem as APIs das redes sociais foi determinante para que a implementação fosse mais rápida. Foram utilizadas as seguintes *gems*:

- *OmniAuth* para implementar *OAuth*[144]: esta ferramenta abstrai as dificuldades sentidas ao integrar com vários serviços Web (*omniauth-facebook*, *omniauth-foursquare*, *omniauth-google-oauth2*, *omniauth-twitter*);
- *Gem Koala*[145]: biblioteca que abstrai os pedidos à *Graph API*[146];
- *Gem quimby*[147]: pedidos à *Foursquare API*[148];
- *Gem twitter*[149]: pedidos à *Twitter API* [150];
- *Gem google-api-client*: pedidos à *Google API*.

Para escrever os ficheiros da aplicação Web foi utilizado o editor de texto SublimeText.

O servidor foi implementado recorrendo à popular *framework* Web *Ruby on Rails*. Na secção dedicada à nova versão são expostas as razões da escolha desta *framework*. A resposta gerada pelo servidor Web *Ruby on Rails* é HTML que é mostrado no *browser* do responsável pelo ponto de interesse.

A aplicação utiliza a linguagem de *markup* que abstrai o HTML chamada *Haml*[151], que permite escrever menos código. O conteúdo das *tags* HTML deixa de estar rodeado pelas *tags*, e passa a estar na linha inferior, com mais uma tabulação do que a *tag* HTML correspondente.

A biblioteca de *Javascript* utilizada é o *JQuery*[152]. Esta biblioteca suportada em vários *browsers*, foi utilizada com o intuito de simplificar os *scripts* criados do lado do servidor. Para simplificar a escrita em *Javascript*, foi utilizado o *Coffescript*.

Para simplificar o design da aplicação, foi utilizada uma *framework open source* chamada *Foundation* [153] da empresa ZURB. Trata-se de uma *framework* bastante responsiva, isto é, adapta-se facilmente ao tamanho do ecrã do dispositivo (computador/*smartphone*). Esta ferramenta é escrita na linguagem LESS[154], que permite simplificar a escrita do CSS. Esta linguagem também foi utilizada para customizar o design.

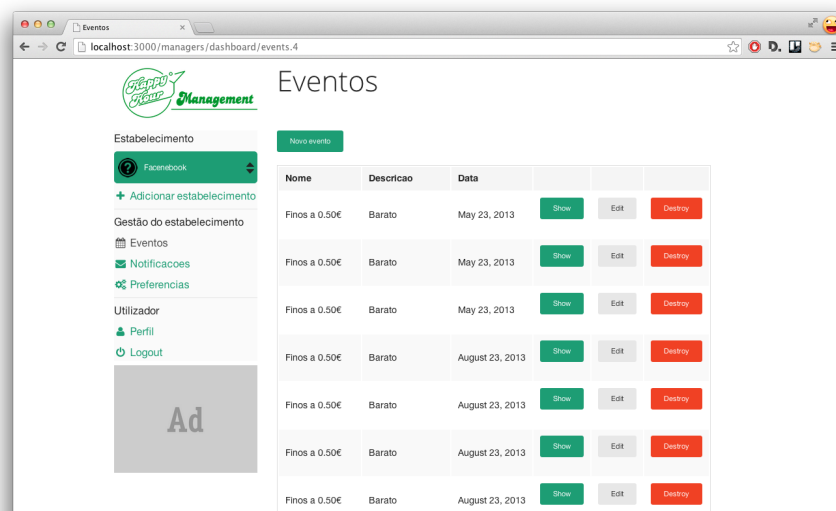


Figura 5.13: Aplicação Web - nova interface.

5.1.3 Servidor REST

Como mencionado na secção 4.5, foram implementados os métodos no servidor REST seguindo a estrutura existente e também descrita nessa secção para receber:

- as informações da rede neuronal;
- as informações da música reproduzida;

Além disso, foram também implementados métodos para difundir essas mesmas informações pelos vários dispositivos móveis que possuem a aplicação *HappyHour*.

5.2 Testes

5.2.1 Testes ao mecanismo de inferência de emoção

Nesta secção são apresentados os resultados dos testes funcionais do mecanismo de inferência da emoção, incluindo:

- a obtenção de informação;
- a implementação das definições de privacidade e de rede;
- a partilha da informação no servidor, para, de forma colaborativa, caracterizar o ponto de interesse ao mesmo tempo que se procura melhorar a emoção do utilizador.

5.2.1.1 Primeira fase

Descrição	Resultado
Ao registar uma emoção é calculada a agitação do utilizador	Passou
Ao registar uma emoção é calculado o batimento cardíaco do utilizador	Não Passou
Ao registar uma emoção é obtida uma informação meteorológica através de uma API	Passou
É utilizado um mecanismo de <i>machine learning</i> para inferir e registar a emoção do utilizador	Passou
A emoção é registada numa base de dados local	Passou
A cada 40 minutos é registada uma nova emoção	Passou
O utilizador é notificado que uma emoção foi registada	Não Passou
Quando o utilizador clica na notificação, deve ser apresentada a interface que permite corrigir a emoção	Não Passou
Após o utilizador corrigir a emoção devem ser atualizados os mecanismos de privacidade e ligação à rede	Não Passou
A emoção é associada ao local mais próximo do utilizador	Não Passou
A emoção é enviada para o servidor, apenas com a informação do local mais próximo do utilizador, mas sem ser associada ao utilizador	Não Passou

Tabela 5.2: Testes à interface de reconhecimento de emoções.

Durante a primeira fase de testes, a *smart shirt VitalJacket*[®] não oferecia garantias de funcionamento devido a uma falha de *hardware*. Individualmente, os vários módulos cumpriam as suas funções, mas no teste funcional, uma vez que não era possível utilizar o batimento cardíaco, estava comprometido o mecanismo no seu todo.

Como forma de ultrapassar esta situação, substituiu-se o *hardware*, sendo o *BITalino* o dispositivo selecionado.

Quer o *BITalino*, quer a *smart shirt VitalJacket*[®], comunicam com o *smartphone* por Bluetooth[®]. No entanto, o primeiro não transmite o batimento cardíaco, sendo necessário implementar um mecanismo de deteção. Durante a pesquisa por um mecanismo de deteção de batimento cardíaco através do ECG apenas foi encontrada uma biblioteca, a *iHealth*[155] onde o batimento é obtido através dos picos R. A performance deste algoritmo é aceitável para o caso de estudo.

5.2.1.2 Segunda fase

Descrição	Resultado
Ao registar uma emoção é calculada a agitação do utilizador	Passou
Ao registar uma emoção é calculado o batimento cardíaco do utilizador	Passou
Ao registar uma emoção é obtida uma informação meteorológica através de uma API	Passou
É utilizado um mecanismo de <i>machine learning</i> para inferir e registar a emoção do utilizador	Passou
A emoção é registada numa base de dados local	Passou
A cada 40 minutos é registada uma nova emoção	Passou
O utilizador é notificado que uma emoção foi registada	Passou
Quando o utilizador clica na notificação, deve ser apresentada a interface que permite corrigir a emoção	Passou
Após o utilizador corrigir a emoção devem ser atualizados os mecanismos de privacidade e ligação à rede	Passou
A emoção é associada ao local mais próximo do utilizador	Passou
A emoção é enviada para o servidor, apenas com a informação do local mais próximo do utilizador, mas sem ser associada ao utilizador	Passou

Tabela 5.3: Testes à interface de reconhecimento de emoções.

5.2.2 Testes à rede neuronal

A validação do mecanismo de *machine learning* envolveu duas etapas:

1. Inicialmente, foram gerados valores de *input* para testar duas configurações de redes neuronais;
2. Na segunda etapa, são utilizados valores de *input* reais, para aferir a capacidade da rede.

5.2.2.1 Teste à rede neuronal com valores gerados

As configurações da rede neuronal propostas foram:

- Uma configuração com 2 camadas escondidas, composta por 3 neurónios na primeira camada e 2 neurónios na segunda camada.

- Uma configuração com uma camada escondida com 4 neurónios, que permite apenas mapear uma relação entre um espaço finito e outro.

O número de neurónios da camada escondida deve ser um número entre a cardinalidade da camada de *input* e a cardinalidade da camada de *output*[138].

O primeiro passo permitiu observar o comportamento da rede neuronal para emoções geradas. A geração das emoções previa valores para os vários componentes. Foram associados valores de probabilidade para gerar as componentes de:

- temperatura;
- nível de nebulosidade;
- se era de dia ou de noite;
- nível de agitação do utilizador;
- nível de ruído ambiente;
- batimento cardíaco.

Assim, um dado intervalo do nível de cada componente tinha uma probabilidade para cada emoção. Foram então geradas 150 emoções. A tabela 5.4 mostra os resultados dos testes iniciais às configurações das redes neuronais. A bateria de instâncias de treino geradas contem todos os *inputs* da rede e a resposta que permite supervisionar a rede. Assim, é possível correr o algoritmo para cada uma das instâncias e verificar o número de épocas necessárias para treinar a rede em cada uma das configurações.

Configuração	Número de épocas necessárias
Uma camada escondida	100
Duas camadas escondidas	3000

Tabela 5.4: Primeira fase de testes à rede neuronal (150 registos).

Para treinar uma rede é utilizado um conjunto de registos de treino. O número de épocas representa o número de vezes que uma bateria de instâncias de treino é utilizada

para treinar a rede neuronal. Como mencionado na página 92, o número de épocas é condicionado pelo erro exigido à rede.

Nesta primeira fase, pretendia-se verificar qual a melhor configuração a treinar.

Conforme os resultados apresentados na tabela 5.4, a configuração com apenas uma camada escondida (4 neurónios) precisou de 100 épocas, enquanto que a configuração com 2 camadas escondidas (5 neurónios) precisou de 3000 épocas.

5.2.2.2 Teste à rede neuronal com valores reais dos sensores

Configuração	Sensibilidade	Especificidade
Uma camada escondida	0,679	0,766
Duas camadas escondidas	0,720	0,830

Tabela 5.5: Segunda fase de testes à rede neuronal com 41 registos.

A figura 5.5 apresenta os testes realizados ao algoritmo com valores obtidos da experimentação com o *BITalino*. É possível observar que a rede neuronal com duas camadas é, em média, melhor do que o algoritmo com 1 camada.

Considerando os resultados das 2 fases de testes faz sentido ao longo do tempo utilizar subconjuntos de emoção menores. Assim, é possível manter o compromisso da solução estatisticamente melhor, mas com menor tempo de processamento.

Tendo em conta os resultados obtidos, apesar de a configuração com apenas uma camada escondida (4 neurónios) apresentar um menor número de épocas necessárias, a configuração com duas camadas escondidas (5 neurónios) apresenta melhor especificidade e sensibilidade.

5.2.3 Testes à identificação de músicas

A tabela 5.6 apresenta os resultados dos testes realizados ao mecanismo de reconhecimento de músicas. Os testes foram realizados num ambiente controlado, onde o volume da música ambiente foi alterado.

Músicas identificadas corretamente	Músicas identificadas incorretamente	Músicas não identificadas
19	31	80

Tabela 5.6: Testes à identificação de músicas.

É possível verificar que este mecanismo não é perfeito, uma vez que depende do microfone do telemóvel. Este tipo de microfones consiste num *hardware* vocacionado para a realização de conversas de voz e não dedicado à deteção de música, de forma a gerar uma *fingerprint*.

Ao mesmo tempo, o serviço utilizado é disponibilizado gratuitamente, o que também poderá ter peso no resultado final deste mecanismo. Ainda assim, continua a ser um mecanismo bastante útil para caracterizar pontos de interesse consoante a *playlist* reproduzida nesse local.

Descrição	Resultado
Se o utilizador ativar a identificação musical, deve ser iniciado o processo de identificação a cada 2 minutos	Passou
O processo de identificação deve registar o som ambiente	Passou
O processo deve gerar uma <i>fingerprint</i> através do som registado	Passou
O processo deve utilizar um serviço Web para identificar a <i>fingerprint</i>	Passou
A música identificada deve ser enviada para o servidor, juntamente com a posição do utilizador, para ser identificado o ponto de interesse onde foi escutada a música.	Passou
Quando é identificada uma música deve ser associada ao perfil do estabelecimento	Passou

Tabela 5.7: Testes à interface de identificação de músicas.

A tabela 5.7 descreve os testes funcionais realizados ao processo de obtenção de música reproduzida num local, e também reflete as alterações à interface que permite consultar a informação gerada por esse processo.

5.2.4 Testes ao *Multipath TCP*

Ao longo deste projeto foi utilizado a tecnologia MPTCP, que permite partilhar uma mesma ligação TCP por vários interfaces de rede e de forma transparente para

as aplicações. Foi realizado um estudo simples para verificar o performance desta tecnologia.

O sistema do teste consiste num *smartphone Android* com Multipath TCP e um servidor com o sistema operativo *Debian*[156] que também utiliza essa tecnologia. Apenas se pode tirar partido desta tecnologia se ambas as partes tiverem implementado no seu *kernel* o *Multipath*. Procurou-se determinar quanto tempo era necessário para continuar a transmissão dos dados.

O caso de teste utilizou uma ligação de TCP com transmissão contínua de dados. O *smartphone* enviava dados para o servidor. No servidor foram capturados os pacotes com recurso ao software Wireshark[157].

Os testes realizados permitiram observar que esta tecnologia oferece aos utilizadores uma conexão melhor para ligações persistentes. Quando um interface era desligado o interface que se encontrava em *backup* continuava a ligação em menos de 3 segundos e por vezes em menos de um segundo, com o mínimo nos 0,7 segundos.

CAPÍTULO VI

Conclusão

Ao longo deste trabalho foram abordados temas como os sistemas ciber-físicos, a computação afetiva e os mecanismos de intervenção no comportamento. O ser humano foi considerado parte integrante do sistema. As aplicações para dispositivos móveis que se inspiram nestes temas são fruto da conjunção dos valores obtidos através dos sensores com mecanismos de processamento e, frequentemente, recorrendo a mecanismos de *machine learning*.

6.1 *HappyHour*

A aplicação desenvolvida e estendida recorreu à utilização dos sensores para classificar os pontos de interesse e utilizadores. Este tipo de aplicações são desafiantes porque inovam na perceção do meio ambiente. Para além disto, na sua implementação é necessário ter preocupações energéticas e de experiência de utilizador.

No módulo emocional procurou-se registar duas vezes por hora a emoção do utilizador porque:

- a energia despendida pelo módulo é tanto maior quanto maior for a frequência de amostragens e o número de sensores;
- foi utilizado um mecanismo de *machine learning* supervisionado que necessita da resposta do utilizador, optando-se por diminuir estas amostragens para não se tornar um tarefa penosa para o utilizador.

Porém, esta frequência é condicionada pelo estado emocional do utilizador num dado momento. Como a aplicação procura acompanhar o seu progresso, é necessário,

em determinadas situações, desprover-se de preocupações energéticas e privilegiar o utilizador.

Os sensores utilizados permitem obter a agitação do telemóvel e o batimento cardíaco do utilizador. Estes dois valores podem ser identificados como a resposta do utilizador ao ambiente. O meio ambiente, por sua vez, é caracterizado com valores de ruído e condições meteorológicas. Segundo a teoria de Mihaly Csikszentmihalyi, o ser humano procura sair dos estados onde se sente aborrecido e ansioso. Este projeto acompanha o progresso do utilizador por meio de sensores e informações processadas.

Atualmente, existem aplicações que se contextualizam com a posição do utilizador, e até reconhecem a atividade que o utilizador está a desenvolver (parado, a andar e a correr), mas ainda não é explorado o seu estado emocional. Os sistemas conscientes da emoção do utilizador são capazes de entregar informações mais relevantes, além de que vão compreender os contextos onde o utilizador se insere num dado momento e adaptar-se a ele, como é o caso do *HappyHour*.

Comparativamente com os projetos concorrentes, esta aplicação é melhor porque procura ajudar o utilizador a sair dos estados emocionais negativos. O utilizador é aconselhado a escolher um ponto de interesse que melhore o seu estado emocional recorrendo às informações disponíveis sobre locais nas imediações. Para além disto, pode receber notificações direcionadas para o seu estado emocional nesse momento. Estas notificações são enviadas por gestores de pontos de interesse localizados nas imediações do utilizador.

Neste trabalho, estendeu-se a caracterização existente dos pontos de pontos de interesse, que incluía a agitação e a afluência. A extensão permite ao utilizador consultar os pontos de interesse onde as pessoas se sentem mais eufóricas ou mais calmas, para além de incluir a informação da lista musical num ponto de interesse. O módulo que caracteriza os pontos de interesse assume o *trade off* de oferecer informação sobre o local num dado instante, mas, por outro lado, é pouco eficiente energeticamente. Este método permite oferecer informações colaborativas mais fidedignas do estado do local comparativamente com os métodos oferecidos pelas aplicações concorrentes, onde apenas é medida a afluência através do *check-in* que representa:

- o número de pessoas que já visitaram um local;
- ou o número de pessoas que vai a um evento durante toda a duração do evento e não o número de pessoas que se encontra atualmente no evento.

6.2 Trabalho Futuro

Em trabalho futuro, poderão ser associadas outras características às entradas do sistema de identificação emocional, nomeadamente a temperatura corporal do indivíduo, que é utilizada em outros sistemas para identificar emoções. Também era relevante caracterizar a emoção do utilizador consoante a hora do dia ou a época do ano.

Para além disto, a recomendação de atividades baseada nos interesses e com base na emoção dos utilizadores é um ponto que deverá ser abordado no futuro. Este desenvolvimento permitirá oferecer sugestões que vão ao encontro das preferências de cada utilizador.

O estado emocional dos utilizadores poderá ser importante no futuro do contexto das aplicações móveis. No estudo [14] foram classificadas as aplicações de acordo com estado emocional do utilizador. Se o estado emocional fosse fornecido como um serviço a todas as aplicações do sistema operativo, então seria possível em tempo real adaptar os seus conteúdos e interfaces da aplicação para melhorar o estado emocional do utilizador, e conseqüentemente, melhorar a classificação das aplicações nas lojas e, ao mesmo tempo, adquirir mais utilizadores.

Sendo o *HappyHour* uma aplicação voltada para dados em tempo real, poderia ser alterado o mecanismo existente de partilha de dados de afluência, agitação e emoções para o protocolo de mensagens *Message Queue Telemetry Transport* (MQTT) [158]. Com esta alteração será possível ao *HappyHour* distribuir as informações num protocolo que necessita de pouca largura de banda e conseqüentemente poupa energia [159][160]. Atualmente o *Facebook Messenger* utiliza o MQTT nas versões mobile da aplicação[160]. Seria interessante estudar o MQTT em conjugação com o MPTCP.

6.3 Considerações finais

Sublinho a importância que teve este trabalho no desenvolvimento académico e pessoal.

Ao longo do segundo semestre, aprofundei os conhecimentos na plataforma *Android*, com o desenvolvimento de uma aplicação que conjuga os sensores e bibliotecas em código nativo (JNI). Conheci tecnologias e áreas da Informática que me entusiasmaram como o *MultipathTCP* e a área da computação afetiva. Tomei também o primeiro contacto com as ferramentas e *frameworks* utilizadas no servidor REST, nomeadamente o *Hibernate*, o *Jersey*.

Considero que a extensão do projeto, assim como as dificuldades ultrapassadas, prepara-me para a nova etapa da minha vida. Um dos exemplos foi o desenvolvimento de competências críticas que serão importantes na tomada de decisões e na análise de sistemas informáticos.

ANEXOS

ANEXO A

Artigo: HappyHour Emotion

Happy Hour Emotion - Improving Mood With Emotionally Aware Application

Pedro Carmona¹, David Nunes¹, Duarte Raposo¹, David Silva¹, Carlos Herrera² and Jorge Sá Silva¹

¹University of Coimbra, Department of Informatics Engineering, 3030 Coimbra, Portugal
pcarmona@student.dei.uc.pt, dsununes@dei.uc.pt, {draposo, djsilva}@student.dei.uc.pt, sasilva@dei.uc.pt

²Escuela Politecnica Nacional Quito, Pichincha, Ecuador
carlos.herrera@epn.edu.ec

Abstract—Mobile sensing in Cyber-Physical Systems has been proportionally evolving with smartphones. Nowadays, we witness a tremendous increase in systems that sense various facets of human beings and their surrounding environments. In particular, the detection of human emotions can lead to emotionally-aware applications that use this information to benefit people's daily lives. This work presents the implementation of a Human-in-the-loop emotionally-aware Cyber-Physical System that attempts to positively impact its user's mood through moderate walking exercise. A smartphone's and smartshirt's raw sensor data is processed to infer current emotional states through a machine learning algorithm. When negative emotions are detected, the application timely suggests users to go for a walk and provides real-time information regarding nearby points of interest, such as their events, music, affluence, agitation and general mood. In addition, the system also dynamically adapts privacy and networking configurations based on the current emotional state. The sharing of the user's location on social networks depends on user-defined rules associated to certain emotions, while the device's networking interfaces are configured in order to provide better connections and reduce frustration resultant from intermittent connectivity.

Keywords—Human-in-the-loop, Emotion Inference, People-Centric Sensing, Smartphones

I. INTRODUCTION

Nowadays, the broad majority of people in developed countries own smartphones. The United States, in particular, showed a smartphone market penetration of up to 66.8 percent, in January 2014 [1]. These mobile devices represent cutting edge technology at reasonable cost, providing Internet connectivity, computing power, as well as several sensors. On a single device, it is possible to enumerate not only several wireless connectivity interfaces, such as 4G/LTE, Wi-Fi, NFC, Bluetooth 4.0 LE, but also various sensors like GPS, accelerometer, gyroscope, proximity/ambient light, compass, pressure and hall effect sensors.

Due to their mobility, flexibility and processing power, smartphone sensors are often made part of larger Cyber-Physical Systems (CPS) that easily sense, control and adapt our environment. CPSs cover many scenarios, from the monitoring of industrial environments, to disaster management deployments and healthcare. In fact, these sensors have already been applied in many different fields [2], for instance in the transportation area, through applications that use localization

in order to solve the problem of congestion in traffic. Likewise, in the health and well-being areas, sensors have also been used to monitor physical activity and encourage users to exercise and improve their health [3]. Sensors have been applied in the affective computing field for emotion recognition, as initially proposed by Rosalind Picard, in 1995 [4]. The term affective computing refers to systems and devices that can recognize, process and, additionally, "have" human affects. Previous proposals for the recognition of emotion have used facial expressions, posture, speech, body temperature, even the strength and pace of keystrokes, together with machine learning algorithms [5]:

- In facial recognition, it is technically difficult to implement real-time emotion detection due to the complexity of image processing algorithms and due to the inherent physiological distinctions between different humans expressing the same feeling [6].
- In speech, Paeschke [7] reported a 77% classification accuracy in recognizing agitation and calmness using neural networks, applied to a call-center scenario.
- Physiological signals were used in [8] to identify negative and positive emotions with 87% accuracy, as well as states of high and low arousal with 95% accuracy [5].
- A recent pilot study by Samsung [9] suggests the motivation of the smartphone manufacturer in detecting a user's emotion through simple touching behavior.

Some research has combined several of these techniques with some success. For example, a research work presented in [10] fused facial recognition and speech analysis, with results showing significant improvements in the performance and the robustness of the emotion recognition system. This suggests that fusing several different types of information can lead to more accurate detection of emotions.

Emotion inference is of extreme importance for future CPSs. The emotional states of humans affect their performance in learning [11], commitment to work, their relationships with friends, family, and co-workers, as well as their mental health and well-being. Therefore, e-learning and therapy are often the context of Affective Computing research.

In this paper, we propose an emotion inference application, HappyHour, in a therapy context. More precisely, we use smartphones in a Behavior Change Intervention (BCI) system to improve human physical and mental well-being. BCIs are therapeutic systems that focus on providing advice, support and relevant information to patients, in order to motivate the correction of prejudicial behaviors. Traditionally associated with presential therapeutic consultations, BCIs have recently begun to be delivered through the Internet and smartphones [12]. Using smartphone sensors to monitor humans in BCIs not only helps in providing more effective feedback to help the user in correcting their prejudicial behavior, but also helps behavioral scientists' research.

Our objective is to improve the user's emotional state, using a machine learning algorithm to infer the user's current emotion and use this information to trigger suggestive feedback that motivates walking exercise and positively affects mood. Previous research has found evidence that moderate walking exercise and the change of environments can trigger positive emotions, since walking is an important physical activity for improvement of mental health [13] [14], much like other outdoor physical activities [15] [16]. In fact, changing from one place to another while walking provides several cognitive benefits such as improved memory, attention and mood [17].

Our HappyHour application follows a Human-in-the-loop Cyber-Physical System (HiLCPS) [18] paradigm, where the human is part of the control-loop and his emotions affect the system outcome. For instance, when a person is stressed our system responds in order to improve their emotional state; thus, a person's feelings are responsible for direct system actuation. Since the human emotions are introduced in the system through sensor data, HappyHour also follows a "people-centric-sensing" paradigm, as proposed by the MetroSense project [19], where not only sensors, but also collaborative data is used to infer the user emotion.

We follow a HiLCPS concept by dynamically adapting the privacy settings and network interfaces based on the human's emotion [20]. Seamless network handoff between interfaces is achieved through a new protocol, Multipath TCP (MPTCP), capable of extending TCP with multipath capabilities. This technology is compatible regular TCP / IP and implements multipath transport by pooling multiple TCP paths within a transport connection, transparently to the application [21].

In the next section, the application developed by our research team will be presented. The third section contains a comparison between HappyHour and other applications. In section IV, we present the technical details of our case-study application, and discuss some of the design decisions made to implement the emotion inference mechanism. We conclude our article on section V, with a consideration on future work.

II. HAPPYHOUR APPLICATION

As stated in section I, walking and changes of environment are important triggers of positive emotions. HappyHour is a BCI application that draws on this premise to positively influence its users' mood. It follows a HiLCPS and "people-centric-sensing" paradigms, where smartphone's sensors acquire data to infer a human's current emotional context. It also

employs collaborative data gathering to determine the real-time context of nearby places, or points-of-interest (POI) that might be of interest to visit.

As a smartphone-based HiLCPS, HappyHour senses a human emotions and presents timely suggestions for walks when negative moods are detected. Data is periodically acquired from different sources, namely the smartphone's microphone and accelerometer, a smartshirt's ECG and weather information from a web API. This information is fed to a neural network, which infers the current emotional state. When negative emotional states are detected, the application provides a map showing the position and a description nearby POIs. The application also displays events associated with each POI, together with information regarding their environmental conditions in real-time. The system groups information from several users to infer a place's affluence, agitation, general mood and music, if any is being played. It is also possible for managers of POIs to schedule relevant notifications regarding cultural events or public notices to be delivered to all HappyHour users in the surrounding area. All of this real-time information allows user's to more easily select places that are worth visiting, providing the necessary motivation for walking towards them and improving mood along the way. HappyHour also encompasses a social networking component by interfacing with Facebook and allowing the automatic sharing of location, depending on the user's mood. This sharing is optional and can be custom-tailored by each user through the definition of emotionally-aware profiles.

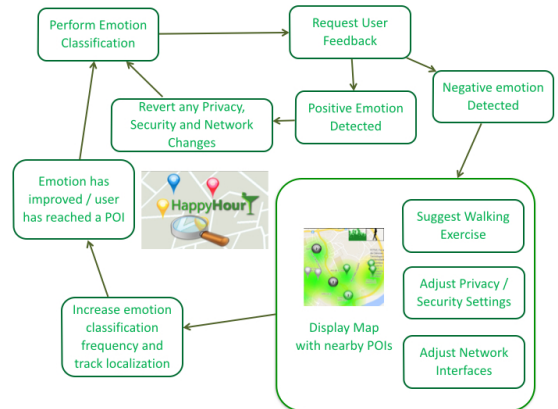


Fig. 1. Happy Hour application flow

Figure 1 represents the loop mechanism proposed and implemented by the HappyHour research team, to positively influence users' behavior and mood. Human' emotions are periodically inferred and, depending on the result, the application motivates the user towards walking exercises in an attempt to improve it. In our current system, we consider four distinct moods: "euphoria", "calmness", "boredom" and "anxiety". "Boredom" and "Anxiety" are considered "negative" emotions, whereas "Euphoria" and "Calmness" are considered their "positive" counterparts.

Negative emotions prompt the system to suggest to the user to go for a walk and, at the same time, a map is shown with possible destinations. Additionally, the system dynamically adapts privacy settings and the network interfaces being used.

Multiple network interfaces may be used in order to provide greater performance and reduce intermittent connectivity, at the cost of greater battery consumption and possible monetary fees for cellular connections. These settings are modified according to rules that are completely customizable by the user. On the other hand, if the user's emotion is positive, the system applies the positive rules and reverts the countermeasures applied in the negative emotional state.

The application can be divided in three parts. The first is responsible to monitor the human emotion, the second applies the rules and countermeasures in order to improve the human emotion, while the last part provides information about points of interest to the user.

A. Monitoring human emotions

The Figures 2 and 3 represent the application's context responsible for emotion inference.

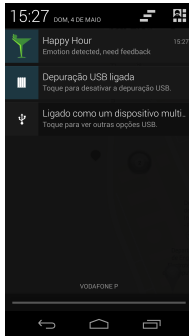


Fig. 2. User is notified that an emotion was detected.

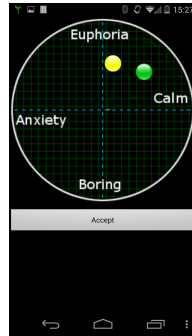


Fig. 3. Emotion detected and user adjustment.

HappyHour's machine learning algorithm receives inputs from sensors and external services. The smartphone's accelerometer sensor gives, after processing, an approximation of the state of agitation of the user. The microphone sensor measures the quantity of noise that the user is exposed to, and helps to understand if the human is sensible to the surrounding noise. The human heart-rate is also acquired through a smartshirt that communicates directly with the smartphone through Bluetooth and provides an ECG signal. Another factor affecting humans emotions is the weather [22] [23], which is taken into consideration by the HappyHour's machine learning algorithm by cross referencing precipitation, cloudiness and temperature. The weather information is acquired and fed to the machine-learning algorithm through the Open Weather Map [24] API.

Users are notified when an emotion is detected, as shown in Figure 2. By selecting the notification, the application opens and displays the feedback screen, as shown in Figure 3. The output representing the inferred emotion is shown as a yellow circle in a two-dimensional circular space containing the four emotions ("Euphoria", "Anxiety", "Calmness", "Boredom"). The user can give feedback on this inference result by dragging the circle to the correct place, now shown as a green circle. Thus, the yellow ball represents the emotion detected by the application and the green ball is the user's correctional feedback. In the example, the application processed the sensor

data and inferred that the user was in a state between euphoria and calmness, as represented by the yellow circle. The user then corrected the application by moving the circle towards a more calm and less euphoric state, represented by the green circle. This correction is taken into consideration by the machine learning algorithm, improving accuracy in future inferences.

B. Emotionally aware rule system

Depending on the emotion classification result and the user's feedback, the HiTL control assumes different decisions. If a positive emotion is detected, no further actions are taken, and the system resumes its standby state until it is time for the next emotion classification procedure. However, if a negative emotional state is detected (e.g. high anxiety or boredom), the system takes direct measures to improve the user's mood.

Part of these measures are responsibility of the "emotionally aware rule system", which is implemented based on sets of actions. Users specify profiles for each emotional state, and actions associated with each profile. These profiles allow the application to automatize and specify privacy, security and networking configurations in an ubiquitous way, in accordance to a HiTLCPS paradigm [20].

- Privacy configurations allow users to create rules that govern the automatic sharing of location in social networks. Some people feel that negative emotion states may be eased by social-interaction and thus, the automatic sharing of location among the user's friends may contribute towards positive socialization. Other people prefer solitude to ease their minds and thus, might want to refrain from publishing their location in a social network. Thus, having defined their privacy profiles, the human can make sure that the HappyHour HiTLCPS will adapt its privacy settings to the current emotional context.
- Network interface configurations are also governed by a specific profile. If the device has an MPTCP-enabled kernel, it is possible to define rules that use more than one interface at the same time. When the user's emotions are positive, the standard network profile defines that WiFi is used as the primary interface while cellular communications are used as a backup, since they usually involve additional monetary costs. However, if the system enters a negative emotion state, the standard network profile defines that cellular connections and WiFi will be used at the same time. For example, the user can set the configuration to use two interfaces (3G/4G and Wifi) if he feels more anxious. This is done for two main reasons. Firstly, to reduce possible sources of frustration that the user might experience, resultant from intermittent connectivity issues. These issues often occur when the user is walking and loses connection to a WiFi hotspot, or enters an area that does not have cellular coverage. Since the established TCP connection is lost, the system needs to reestablish a new session on the remaining interface. UDP connections are less prone to this intermittent connectivity problem, but they are not adequate for the reliable data transfer that

is often required from HiTLCPSs and people-centric-sensing applications. Also, 99.7% of all mobile traffic is currently transmitted through TCP [25]. Therefore, using both interfaces at the same time means that the TCP sessions can continue through the remaining network interface, providing a better Quality of Experience (QoE). Maintaining a single TCP session while using several networking interfaces and performing seamless network handoff between them is achieved through MPTCP [21]. This protocol extends TCP with multipath capabilities by pooling multiple TCP paths within a transport connection, transparently to the application. Users that are emotionally distressed might be willing to pay the additional price of cellular connectivity to reduce frustration and thus, contribute towards the improvement of mood. The second reason is of a more technical nature and is due to the additional data being sent to the server that results from the increased emotional classification frequency that is performed when negative emotions are detected. When the human is on a negative mood, his position and emotional states are monitored more closely, in order to provide better feedback and improve the impact of the positive reinforcement resultant from walking suggestions. Using several network interfaces provides better reliability in data collection during these negative states. Nevertheless, the network profile that governs these changes is completely configurable by the user in case he wishes to disable cellular connections due to their monetary costs, for example.

C. Points of interest informations

The information provided by the application allows users to identify their best place in a local area to visit. Since every person has their own music tastes and emotional responses, providing detailed information about the environment of each POI might have impact on decision-making process. For instance, a person could choose a crowded bar with music if they want to dance, or decide to go visit a more quiet nature park for a relaxing walk instead. Tracking and aggregating users' emotions and their physical activity allows HappyHour to categorize points of interest.

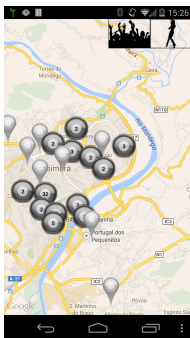


Fig. 4. Points of interest nearby with clusters.

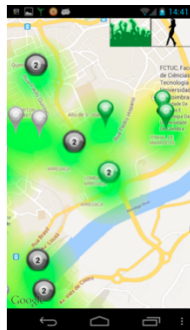


Fig. 5. Heatmaps of affluence.

The Android application allows users to select a destination for their walk through a Map. Each place of interest is

represented though an icon, which is aggregated into a cluster of icons when the map's zoom is low. These clusters are represented as circles with the number of aggregated places. The application renders real-time information about each place, namely:

- Average agitation of other HappyHour users
- Affluence of other users to that place
- Current music being played in bars or nightclubs
- Average mood of the place

The agitation and affluence are represented through heatmaps (fig. 5), colored layers surrounding the point of interest. The colours green, yellow and red classify the amount of affluence or agitation of the associated place. These heatmaps can be toggled through two buttons located in the upper-right side of the screen: the left-side button represents a crowd and activates the affluence heatmap, while the right-side button represents agitation and shows the associated heatmaps. By clicking on a place's icon, the users can access a description, and request navigation to the place through an external navigation application. Users can also browse a list of events associated with that place, which is maintained by the place's managers through a management web-interface. Managers can also schedule notifications that are delivered to all HappyHour user's in the surroundings of the POI. The primary purpose of these notifications is to advertise current cultural activities or initiatives, events, or promotions that might be of interest to the public.

Information about music being played at each point of interest is also presented and is acquired using the smartphone's microphone. The sound sample is processed by the application, generating a fingerprint, and is identified through a free music recognition API (Echoprint [26]), which compares the obtained fingerprint with a fingerprint database. This music recognition feature may be of particular interest to users that enjoy visiting bars or establishments where a certain type of music is played that allows them to vent their emotional stress.

Individual informations is provided to the server in a periodic fashion, every 5 minutes. Information from all users is then aggregated and updated regularly, in an anonymous fashion. Providing real-time, non-identifiable information can help system users in making informed decisions regarding the destinations of their walking exercise, which can lead to improvement of emotional state, without compromising privacy.

Some of the ideas applied in HappyHour are not new, such as the showing of POI. However, we believe that the real-time information provided on each of these POIs, together with the emotional-awareness and BCI purpose of the application make HappyHour a highly innovative HiTLCPS. In the next section, we will be presenting and comparing HappyHour with other people-centric-sensing, social and mobile applications.

III. PLATFORMS AND APPLICATIONS COMPARISON

HappyHour can be compared with applications from many areas, such as point-of-interest applications, sensor driven applications that register users physical activity, and applications

consider the user's feelings and emotions. The main differences between HappyHour and other point-of-interest driven applications currently on the market are its focused objective, which is promoting behavior changes towards the improvement of mood, its intent to deliver real-time information about places, unlike static collaborative contributions such as "check-ins" or reviews, and its ubiquitous collaborative information gathering, that does not require direct user intervention to share data. Another aspect pertaining to other emotionally-aware applications rests on their lack of intervention, that is, they do not attempt to positively influence the users' mood. On the other hand, HappyHour takes on a proactive approach, encouraging users to take walks and to discover different places. Additionally, HappyHour follows a HiTLCPS paradigm and dynamically adapts system parameters, such as privacy and networking, according to configuration profiles adapted to each emotion. We will now consider three applications that are related to HappyHour and discuss how HappyHour's features distinguish it in comparison.

A. Facebook

Facebook [27] is a social networking company. The Facebook mobile application allows users to post messages with their current location, and even with the emotion they are feeling. The posts can be shared with anyone, only friends or be private. Users can also search for services and establishments in their local city, publish and browse reviews.

The difference between HappyHour and Facebook is that the information about the user location and information (affluence and agitation) about the points of interest is not as specific as in HappyHour. Whereas in HappyHour users can see the current affluence of a place in real-time, Facebook only provides an estimate on the number of participants in a certain event, which may not represent the actual number of persons actually present. Secondly, Facebook does not take advantage from the information about how the user is feeling, in contrast to HappyHour, which attempts to proactively and positively influence the user to feel better.

B. Foursquare and Swarm

Until May 2013, Foursquare [28] was an application that allowed users to share their location with friends by "checking-in" on visited establishments and to search for contacts close to them. It also employed rewarding techniques that motivated users to keep using the App, such as by nominating the users that checked-in the most as "Mayors" of a certain point of interest. In May 2014, the Foursquare company announced modifications in Foursquare application [29], and announced a brand new application named Swarm. With these modifications, Foursquare application becomes a platform for discover places, and Swarm is directed to searching for friends close, chat with them and arrange meetings. Discovering of places will be personalized to the users' tastes and with reviews of the people the user trusts. Foursquare platform is available in several platforms, such as Web, Android, iPhone, among others. In January 2014 it had 45 million users worldwide [30].

HappyHour is different from Swarm since the collaborative information is generated by users in an ubiquitous, automatic

and distributed manner, allowing the user to search for the currently more or less crowded and agitated points of interest, instead of simply finding friends. HappyHour also focuses on providing real-time information instead of static information such as "check-ins" or "reviews". Users can know what POI are the most interesting for them at that moment, with information such as the current affluence, agitation, music being played and notifications of interest. Swarm also lacks any sort of emotionally-aware components and does not directly contribute towards the improvement of the user's mood, simply relaying information. HappyHour, on the other hand, takes a much more personal approach to POI discovery, motivating users to visit new places and, at the same time, exercise and improve their moods.

C. EmotionSense

EmotionSense [31] is a mood tracking application for Android, developed by the research team at the University of Cambridge. As Smartphones are carried by users throughout their days, the EmotionSense application collects data about how users feel, turning Smartphones into medical and psychological tools [32]. The acquired data contains information collected with sensors (such as GPS, accelerometer, and microphones), as well as the users' calling and texting patterns, along with questions asked to users. In order to detect the emotions users are asked to fill a form twice a day. In the form, the application first asks the user to mark how he is feeling, using a matrix named "emotion grid". Secondly the user responds to a more specific survey. EmotionSense system has been designed by psychologists within the research team [32]. Then the sets of data are processed and it is possible to show when the user is more relaxed, or more stressed.

HappyHour's emotion inference model was inspired in the emotion grid system from EmotionSense. We differentiate ourselves from EmotionSense because we developed a mechanism that follows the Human-in-the-loop paradigm, where the user emotion adds variation to the system. In HappyHour, the emotion is also automatically inferred and used to set configurations of privacy, security and networking interfaces. HappyHour also takes a proactive approach and encourages the user to take walks and to discover different places in order to improve his mood. Also, we provide a point of interest management platform that allows managers to define events and schedule notifications to be sent to users close to their point of interest, promoting the POIs. On top of that, HappyHour applies an ubiquitous collaborative approach to acquire the real-time context of each of the POIs, providing information on the current affluence, agitation, music and general mood of the place.

IV. TECHNICAL DISCUSSION

This section begins by describing the system's general architecture, then focusing on the Android application, and, finally, describing the machine learning technique used to infer the emotion of the user.

The Happy Hour system scheme contains an Android application, a main Server and a Web Server. There are two types of personas that interact with the system, users and managers of points of interest (such as nightclub owners or museum personnel), as seen in Figure 6.

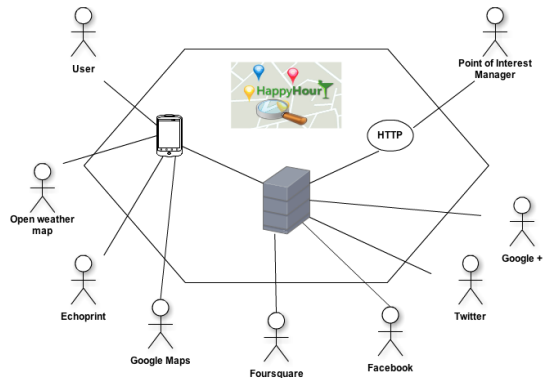


Fig. 6. Happy Hour System

POI managers use a web interface to control events, notifications and other informations about the POI. Managers can also publish the events and notifications to their Facebook [27], Twitter [33] and Google+ [34] Account, in order to reach more audience. The web interface is developed in Ruby On Rails [35].

The main server handles requests of creation, update or deletion of users, events, notifications and points of interest. The POIs are acquired through external services, such as Foursquare [28] and Facebook [27].

Regular users interact with HappyHour through an Android application. The mobile devices running HappyHour update their contextual information to the server every 5 minutes. Figure 6 represents the context of the HappyHour Android application. The connection to HappyHour server provides information about the agitation, affluence and emotions felt close to each POI. Users can navigate through a Map provided by Google Maps [36]. There are also external services used to get information about the current weather close to the user location (Open Weather Map [24]), together with the ability to publish location information on external social networks such as Facebook. A music recognition API [26] is also used to automatically identify the current music being played at each POI.

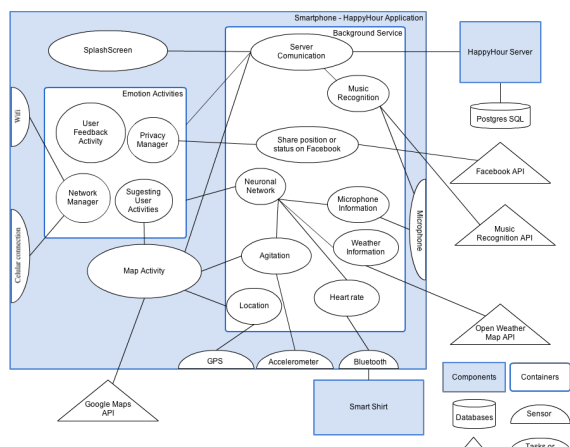


Fig. 7. Happy Hour Android application

The Android application is composed by several "Activities", responsible for interacting with the user, and one background service responsible for syncing information with the server, sensing the user and environment and processing the raw sensor data. Figure 7 shows the Android architecture, and classes such as:

- User feedback activity: responsible for acquiring the user's feedback on the emotion detection result.
- Map activity: displays the map, heatmaps, points of interest, and current position.
- Suggesting user activities: as negative user emotions have been detected, these activities provide suggestions for the user to go for a walk.
- A music recognition background service that listens to the ambient sound, and retrieves information about the current music through the music recognition API Echoprint [26].
- A privacy manager class, that is responsible for applying the privacy profiles associated with emotions.
- A network manager class, that is responsible for applying the network profiles associated with emotions.
- A neural network class, that recognizes the emotion based on processed input information, and is trained by the user input.
- Classes that process data received from sensors (agitation, heart-rate, sound) and external services (weather information from Open Weather Map [24]).

A. Emotion with machine learning

The HappyHour uses the smartphone sensors, a wearable smartshirt's ECG and external services in order to detect the user's emotion. The information gathered from sensors is processed in order to be used as an input of a machine learning algorithm. The approach taken when designing this emotion inference tool was to create a simple proof-of-concept. In order to determine the best machine learning technique for our application we studied previous comparisons between the different possibilities. The next table shows the classifications of each algorithm scored in correct classification rate and in central processing unit (CPU) time needed for classification in a mobile application environment.

Machine learning techniques	Correct classification rate	Ranking of less CPU time needed for classification
Decision trees	96.5%	5 ^o
Support vector machines	80.2%	1 ^o
Naive Bayes classifiers	81.5%	6 ^o
Bayesian networks	90.9%	4 ^o
Logistic regression	83.4%	3 ^o
Artificial neural networks	87.2%	2 ^o
Instance-based classifiers (KStar, LWL, and IBk).	96.6%, 95.6% and 80.	9 ^o , 8 ^o , 7 ^o

TABLE I. MACHINE LEARNING APPROACH FOR SENSING CONTEXTS IN SMARTPHONES [37].

Taking into consideration that the machine learning technique is to be used in a mobile application, it was decided to implement an artificial neural network as the emotion inference tool, since it offers a reasonable correct classification rate while being one of the least time-consuming techniques when processing.

The implemented proof-of-concept uses microphone and accelerometer processed values from the smartphone, weather information from an external API that provides the temperature, cloudiness and precipitation, and the heart-rate from a smartshirt's ECG signal.

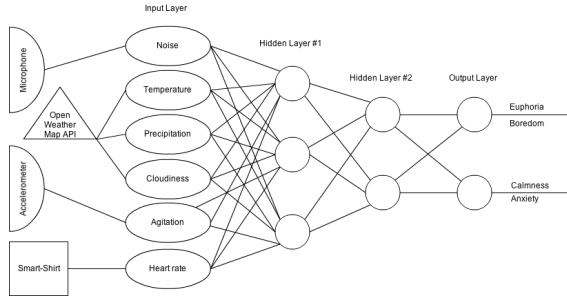


Fig. 8. Happy Hour emotional neural network.

- Noise is an input of the neural network that tries to classify the environmental noise, and is used to identify in which environments, more noisy or less noisy, the user feels more positive emotions. The noise input results from an average of the microphone's sensed data. However, there are some reservations in using the smartphone's microphone sensor to evaluate environmental noise, because generally these microphones are design to be used in phone-calls and using them to listen to the user's surroundings can induce errors in the input.
- Cloudiness, Precipitation and Temperature are inputs retrieved from the weather API and are used to detect a correlation between human emotion and meteorological conditions.
- The accelerometer raw data is processed using a fourier transformation component analysis and the resultant information represents the amount of movement made by the user. This information pretends to characterize the user's movement as "agitated" or "calm".
- The smartshirt provides ECG information, from which the user's heart rate can be extracted. The heart-rate is a signal that brings direct physiological information from the user to the system.

The neural network presented in the figure 8 was develop using the Encog [38] machine learning framework, which is a Java implementation developed by Jeff Heaton.

We followed Jeff Heaton's empirical knowledge to establish the number of nodes in the hidden layer. In [39] he identified that the optimal size for the hidden layer is a number between the size of the input and size of the output layers. Therefore, we selected a configuration with two hidden layers, the first with three nodes and the second with two nodes.

The user's emotion is inferred once or twice an hour. The time between two sensory acquisitions is randomly determined within these constraints in order to avoid user habituation. The output is shown to the user, and the application then uses the user's corrections (by dragging the two-dimensional emotional matrix's yellow circle to the correct place) as feedback to the neural network's training.

The inferred emotion in the machine learning algorithm is then directly used as input to the privacy and networking profiles and sets of rules for each emotion. While default profiles are provided, these are completely customizable by the user. The user is, therefore, responsible for determining how the system adapts to his emotions, not only in terms of privacy settings (sharing of location in social networks), but also the network interfaces being used (such as 4G and Wifi). Through MPTCP-enabled Android kernels, the profile may allow activate the use of multiple interfaces.

V. DISCUSSION AND FUTURE WORK

In this work a HiTLCPS based on smartphones was presented, that takes a BCI approach and follows a people-centric sensing paradigm to infer and positively affect a user's emotion. This emotional context-awareness allows mobile applications to provide better and more accurate information to the user, as well as adapt to his emotional needs. Much like geolocation applications use the user's geographic position to dynamically adapt several settings, inferring the users emotion allows to applications to dynamically adapt to the user's emotions. While the inference of emotions is still a developing area, in today's world where users wear smart watches, smartshirts, smartphones and other smart wearables, the process of gathering the user's contextual data is greatly simplified. With the increase of available data there will be new and promising mechanisms that infer users emotions with increased accuracy to create applications that are aware of the user's emotion state. These applications can be used with the goal of improving the user's mood, interact more socially or even perform better at learning and working tasks.

Our HiTLCPS also shows how a mobile device's system can dynamically adapt to a user's emotional context in terms of privacy and networking connections. Our current implementation for mobile handoff based on MPTCP serves as a proof-of-concept of the advantages of such emotion inference tools.

In the future, it could be interesting to perform an in-depth experimental study of which machine learning techniques are better for classifying emotion based on smartphone sensors. Our HiTLCPS could also be expanded to consider other types of contextual data in the control-loop, such as human actions and position. Additionally, it could also be relevant to understand how much the addition of new types wearable devices to the machine learning algorithm's input can improve the accuracy of the detected context.

ACKNOWLEDGMENTS

The work presented in this paper was partially financed by the iCIS project (grant CENTRO-07-ST24-FEDER-002003)

REFERENCES

- [1] (2014, May) comscore reports january 2014 u.s. smartphone subscriber market share. comScore. [Online]. Available: https://www.comscore.com/Insights/Press_Releases/2014/3/comScore_Reports_January_2014_US_Smartphone_Subscriber_Market_Share/
- [2] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, Sept 2010.
- [3] (2014, May) Health benefits of physical activity. Beverage Institute. [Online]. Available: <http://beverageinstitute.org/us/article/health-benefits-of-physical-activity/>
- [4] R. W. Picard, *Affective Computing*. Cambridge, MA, USA: MIT Press, 1997.
- [5] H. Gunes and M. Pantic, "Automatic, dimensional and continuous emotion recognition," *International Journal of Synthetic Emotions*, vol. 1, no. 1, pp. 68–99, January 2010. [Online]. Available: <http://doc.utwente.nl/75888/>
- [6] G. Piolle, "Affective computing, software agents and online communities," MSc ISO report, Imperial College London, Dpt of Computing (supervisors: Jeremy Pitt and Keith Clark), London, United Kingdom, April 2005, mSc background paper. [Online]. Available: <http://guillaume.piolle.fr/doc/piolle05a.pdf>
- [7] W. F. Paeschke, Astrid / Sendlmeier, "Prosodic characteristics of emotional speech: Measurements of fundamental frequency movements," in *In SpeechEmotion-2000*, 2000, pp. 75–80.
- [8] J. Wagner, J. Kim, and E. Andre, "From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, July 2005, pp. 940–943.
- [9] H.-J. Kim and Y. S. Choi, "Exploring emotional preference for smartphone applications," in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, Jan 2012, pp. 245–249.
- [10] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, "Analysis of emotion recognition using facial expressions, speech and multimodal information," in *Proceedings of the 6th International Conference on Multimodal Interfaces*, ser. ICMI '04. New York, NY, USA: ACM, 2004, pp. 205–211. [Online]. Available: <http://doi.acm.org/10.1145/1027933.1027968>
- [11] C. Akbiyik, "Can affective computing lead to more effective use of ICT in education?"
- [12] N. Lathia, V. Pejovic, K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Smartphones for large-scale behavior change interventions," *IEEE Pervasive Computing*, vol. 12, no. 3, pp. 66–73, 2013.
- [13] M. G. Berman, E. Kross, K. M. Krpan, M. K. Askren, A. Burson, P. J. Deldin, S. Kaplan, L. Sherdell, I. H. Gotlib, and J. Jonides, "Interacting with nature improves cognition and affect for individuals with depression," *Journal of Affective Disorders*, vol. 140, no. 3, pp. 300 – 305, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165032712002005>
- [14] C. C. for Health, "The benefits of regular walking for health, well-being and the environment," September 2012. [Online]. Available: www.mind.org.uk/mindweek
- [15] Mind, "Ecotherapy - the green agenda for mental health," May 2007. [Online]. Available: www.mind.org.uk/mindweek
- [16] M. Atkinson and L. Weigand, "A review of literature: The mental health benefits of walking and bicycling," 2008.
- [17] M. G. Berman, E. Kross, K. M. Krpan, M. K. Askren, A. Burson, P. J. Deldin, S. Kaplan, L. Sherdell, I. H. Gotlib, and J. Jonides, "Interacting with nature improves cognition and affect for individuals with depression," *Journal of affective disorders*, vol. 140, no. 3, pp. 300–305, 2012.
- [18] G. Schirmer, D. Erdogmus, K. Chowdhury, and T. Padir, "The future of human-in-the-loop cyber-physical systems," *Computer*, vol. 99, no. PrePrints, p. 1, 2012.
- [19] (2012, June) Metrosense project. [Online]. Available: <http://metrosense.cs.dartmouth.edu/>
- [20] P. Z. David Nunes and J. S. Silva, "A survey on human-in-the-loop applications towards an internet of all," Julho 2014, submitted to IEEE Communications Surveys & Tutorials.
- [21] A. Ford, C. Raiciu, M. Handley, S. Barré, and J. Iyengar, "Architectural guidelines for multipath tcp development," Internet Engineering Task Force (IETF), Request for Comments 6182, p. 28, March 2011.
- [22] A. Barnston, "The effect of weather on mood, productivity, and frequency of emotional crisis in a temperate continental climate," *International Journal of Biometeorology*, vol. 32, no. 2, pp. 134–143, 1988. [Online]. Available: <http://dx.doi.org/10.1007/BF01044907>
- [23] Z. Spasova, "The effect of weather and its changes on emotional state - individual characteristics that make us vulnerable," *Advances in Science and Research*, vol. 6, pp. 281–290, Mar. 2011.
- [24] Open weather map. Open Weather Map. [Online]. Available: <http://openweathermap.org/>
- [25] A. Rahmati, C. Shepard, C. Tossell, A. Nicoara, L. Zhong, P. T. Kortum, and J. P. Singh, "Seamless flow migration on smartphones without network support," *CoRR*, vol. abs/1012.3071, 2010.
- [26] Echoprint. The Echonest. [Online]. Available: <http://echoprint.me/>
- [27] Facebook. Facebook. [Online]. Available: <https://www.facebook.com/>
- [28] Foursquare. Foursquare. [Online]. Available: <https://www.foursquare.com/>
- [29] A look into the future of foursquare, including a new app called swarm. Foursquare. [Online]. Available: <http://blog.foursquare.com/post/84422758243/a-look-into-the-future-of-foursquare-including-a-new>
- [30] Make the most of where you are with foursquare. Foursquare. [Online]. Available: <https://pt.foursquare.com/downloads>
- [31] emotionsense. emotionsense. [Online]. Available: <http://emotionsense.org/>
- [32] Mood-tracking app paves way for pocket therapy. emotionsense-review. [Online]. Available: <http://www.cam.ac.uk/research/news/mood-tracking-app-paves-way-for-pocket-therapy>
- [33] twitter. twitter. [Online]. Available: <http://twitter.org/>
- [34] Google+. Google. [Online]. Available: <https://plus.google.com>
- [35] D. H. Hansson. (2012) Ruby on rails. [Online]. Available: <http://rubyonrails.org/>
- [36] Google maps. Google. [Online]. Available: <https://maps.google.com>
- [37] R. Guinness. Beyond where to how: A machine learning approach for sensing mobility contexts using smartphone sensors. Tampere University of Technology, Finland. [Online]. Available: <http://www.ion.org/gnss/abstracts.cfm?paperID=856>
- [38] J. Heaton. Encog machine learning framework. Heaton Research, Inc. [Online]. Available: <http://www.heatonresearch.com/encog>
- [39] —, *The Number of Hidden Layers*. [Online]. Available: <http://www.heatonresearch.com/node/707>

ANEXO B

Diagrama de Gantt (Primeiro Semestre)

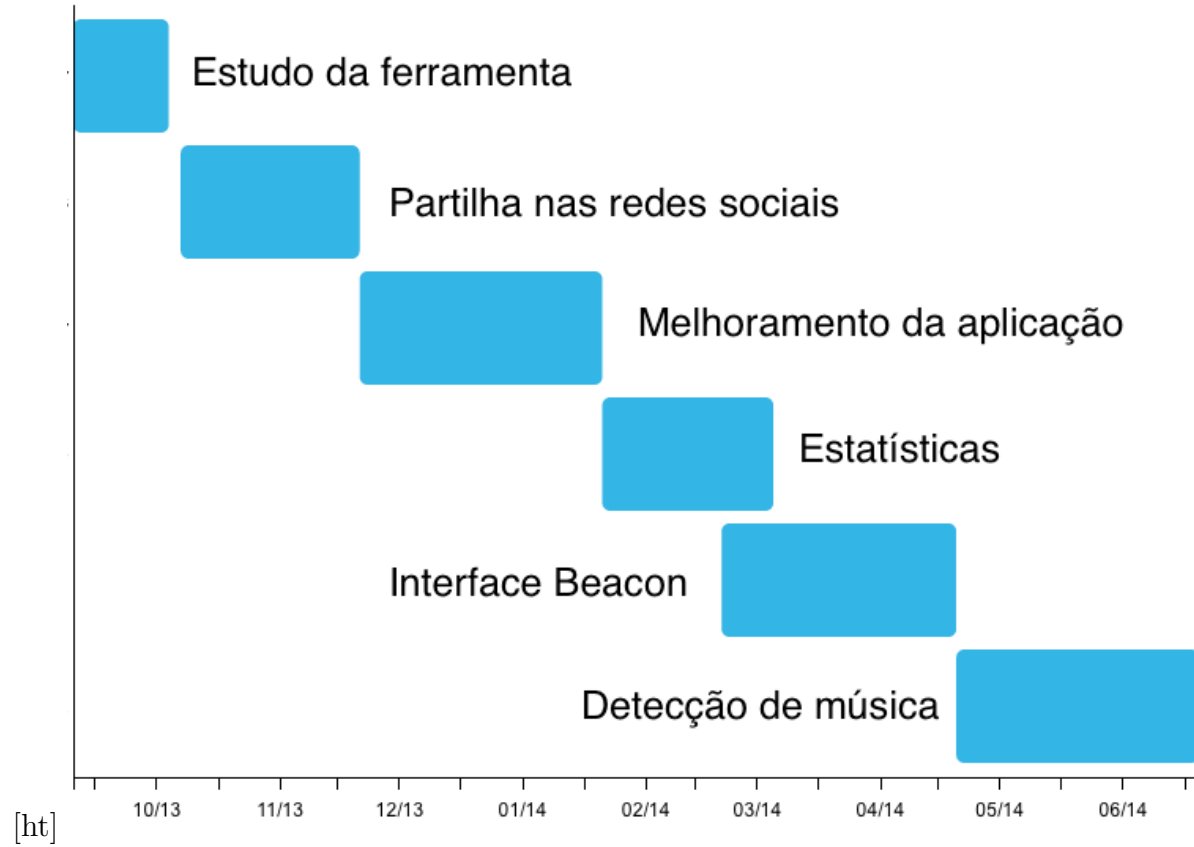


Figura B.1: Diagrama de Gantt.

ANEXO C

Diagrama de Gantt (Planeamento do Segundo Semestre)

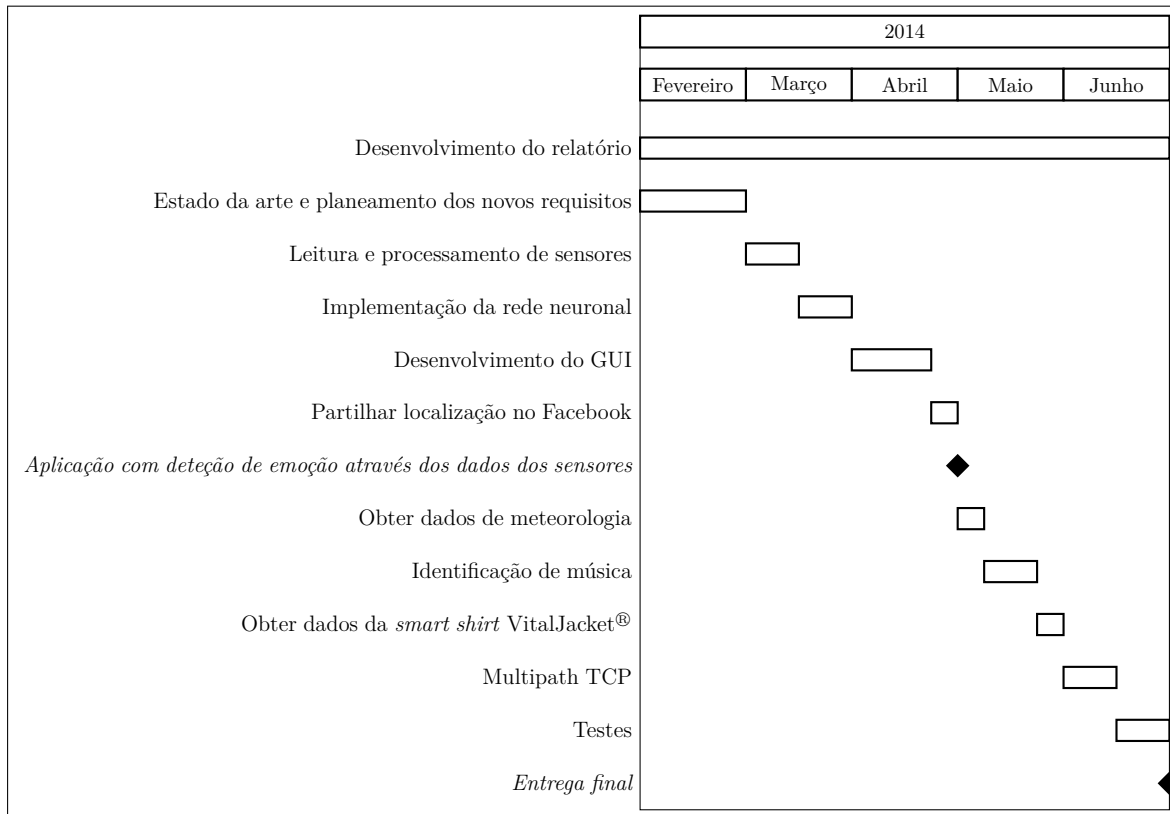


Figura C.1: Diagrama de Gantt do segundo semestre.

ANEXO D

Diagrama de Gantt (Trabalho no Segundo Semestre)

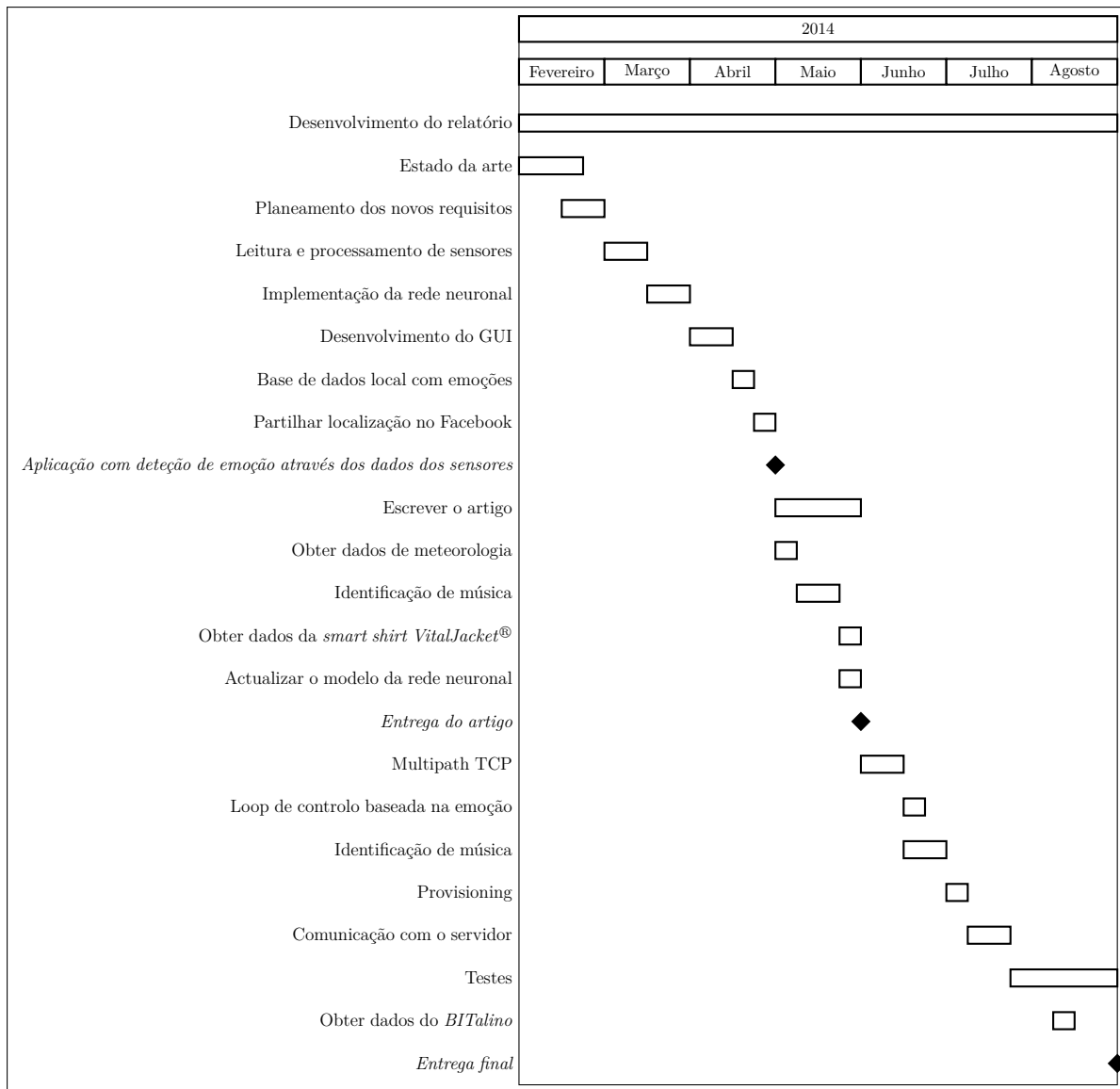


Figura D.1: Diagrama de Gantt que retrata o trabalho realizado segundo semestre.

ANEXO E

Serviços Web

Os Serviços Web têm como objetivo promover a troca de informação entre aplicações através da Internet. A troca de informações entre as aplicações cria a integração entre estas. A integração das aplicações permite a interoperabilidade entre diferentes organizações, onde as informações disponibilizadas pelas várias aplicações é difundida de uma forma simples e sem expor a base de dados. Os serviços web aumentam a eficiência das aplicações básicas, onde através de uma plataforma é possível expandir o uso destas aplicações por parte dos utilizadores. As mensagens dos Serviços Web permitem a chamada de procedimentos à distância.

Representational State Transfer (REST)

O REST é uma arquitectura que utiliza o protocolo HTTP e os seus recursos. As principais características são:

- Todos os recursos disponíveis na web são identificados por um URL.
- A interação com o fornecedor do serviço é feito com recurso aos métodos GET, POST, PUT e DELETE. Estes métodos fazem a ponte com os métodos CRUD da base de dados (Create, Read, Update e Delete).
- As interações são stateless - o fornecedor do serviço conhece o estado dos recursos, porém não mantêm as sessões dos clientes (escalabilidade)
- Utilização dos elementos intermediários para fazer caching e proxies, contribuindo também para a escalabilidade. O tempo de implementação é mais reduzido, pois estes elementos intermediários retiram a necessidade de desenvolver mecanismos de caching.

Interfaces de Programação de Aplicativos (API)

Uma API especifica um conjunto de procedimentos para interagir com uma aplicação. No contexto do desenvolvimento Web, a API é um conjunto de pedidos a um servidor, escritos em *Hypertext Transfer Protocol* (HTTP).

O sistema *HappyHour* trabalha com Serviços Web na comunicação com serviços externos (Facebook, Foursquare, Twitter, Google+), assim como, dentro do sistema para a comunicação da aplicação central com os telemóveis Android. Neste projecto pretende-se ainda implementar a comunicação do sistema com bibliotecas de reconhecimento de música, para automatizar o processo de construção das listas de música reproduzidas num estabelecimento. Neste sentido, dedico uma secção às bibliotecas de reconhecimento de musica e outra secção às APIs das redes sociais implementadas.

Redes Sociais

Recentemente, a Internet era apenas fonte de informação. O advento da Web 2.0 permitiu que a Internet seja hoje, e cada vez mais, um local onde os próprios utilizadores geram e controlam informação.

As redes sociais tornaram-se um fenómeno de grande popularidade. Ao longo do dia, o utilizador interage de diversas formas com outros utilizadores. Grandes quantidades de conteúdo são compartilhadas em tempo real. O ser humano aderiu em massa às redes sociais, utilizando-as como veiculo para partilhar opiniões, pensamentos e intenções. O desejo de comunicar em tempo real e com um maior número de indivíduos denota-se no crescimento exponencial das redes sociais como o Twitter e Facebook.

Estes sistemas são desenvolvidos sobretudo com base em software livre e caracterizam-se por:

- Gestão de um enorme volume de informação em tempo real;
- Utilização segura;
- Garantir que as necessidades dos utilizadores ficam satisfeitas com sucesso;
- Constituem fonte de informação importante a aplicações externas que, por sua vez, permitem direccionar o conteúdo certo para o indivíduo certo, consoante as suas preferências;

Hoje em dia as redes sociais fornecem bibliotecas que permitem a aplicações externas interagirem com a informação das redes sociais. As funcionalidades disponibilizadas

permitem autenticação e autorização. A autenticação é o serviço responsável por certificar um cliente, numa aplicação externa, com a utilização das credenciais do utilizador na rede social. Autorização é um serviço que permite às aplicações externas requisitarem acesso autorizado a informações dos utilizadores, e exercer interação com a rede social em nome da pessoa.

Facebook API

Os serviços fornecidos pelo facebook permitem aos websites incorporar funcionalidades sociais, como a utilização do botão Gosto. Uma funcionalidade permitida pela integração com o Facebook é a possibilidade dos utilizadores fazerem o registo e/ou início de sessão na aplicação cliente.

A principal interface de programação do facebook é a Graph API com base em HTTP[43].

Nesta API existe o acesso a informações públicas e o acesso a informações privadas. É necessária a autorização do utilizador para adquirir informações privadas. Com autorização é permitida a recolha de informações acerca da pessoa, como os gostos, os dados pessoais, e galerias de fotografias. Outra informação devolvida pela Graph API são os amigos da pessoa.

A informação recolhida, recorrendo à graph API, pode ser utilizada para determinar as preferências das pessoas.

No sentido da ciência da computação, esta API pode ser vista como um grafo social. Os utilizadores, as páginas e os eventos são objectos que têm uma identificação única, e podem ser acedidos através de um URL. Todos os objectos são ligados através de relações, por exemplo relacionamentos entre amigos, conteúdo partilhado pelos utilizadores, ou fotos onde as pessoas são marcadas.

O facebook utiliza OAuth 2.0 que permite uma autorização dos acessos a objectos segura.

Twitter API

Twitter é uma plataforma de microblogging, onde as mensagens são limitadas a 140 caracteres. Na biblioteca fornecida pelo Twitter é possível uma utilização pública e privada.

Na utilização pública a API permite a possibilidade de pesquisar por hashtags, e ver informações públicas do utilizador[161].

No o acesso privado, através da autorização do utilizador, é permitido aceder às suas

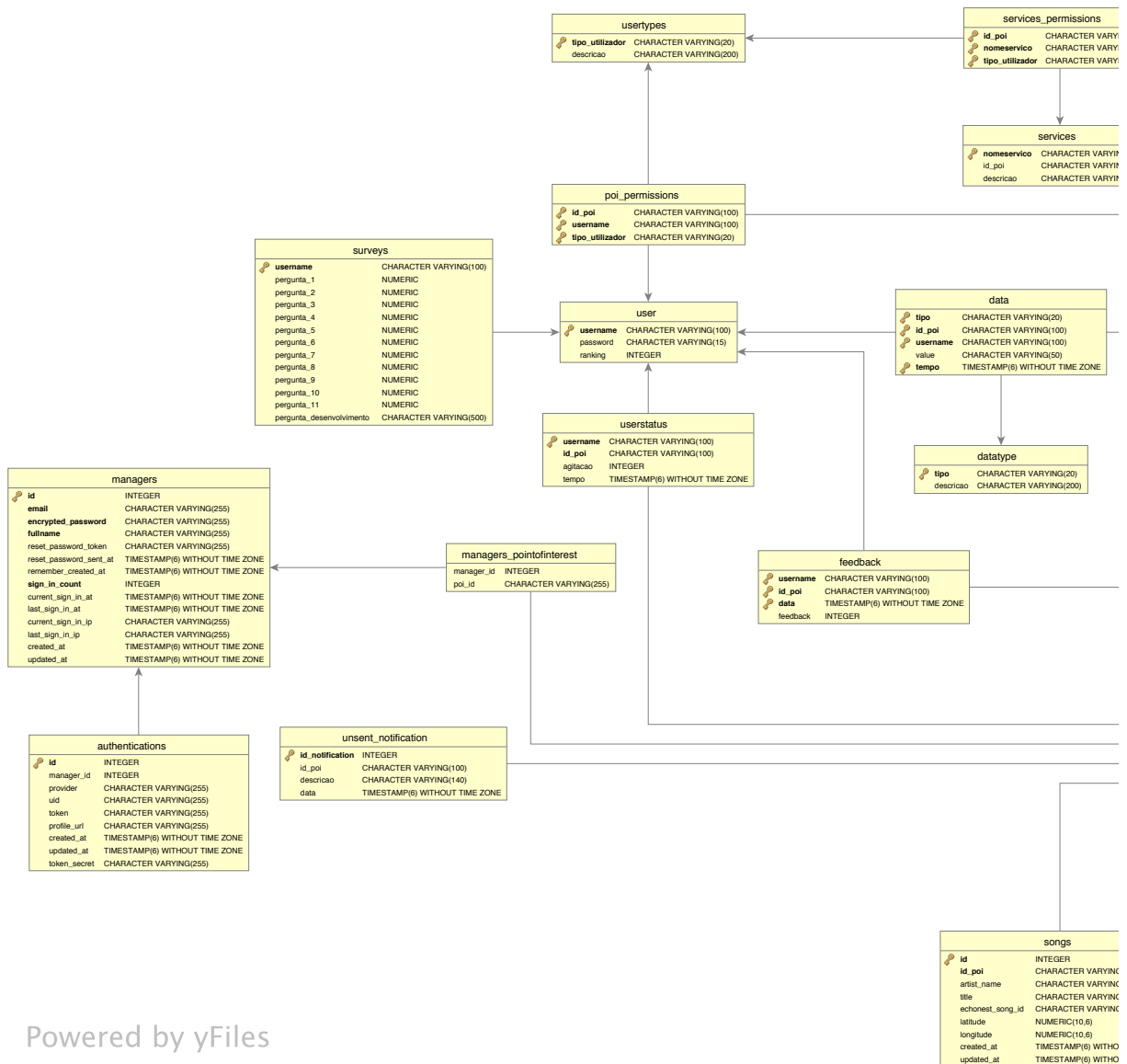
informações privadas. Permite também o início de sessão no site/aplicação cliente. Os utilizadores quando escrevem *tweets* têm a possibilidade identificar as palavras-chave do seu comentário através de hashtags. No caso de o comentário ser efectuado através do site/aplicação cliente, esta pode anexar ao *tweet* uma referência para o conteúdo mencionado (Twitter Cards), explorando a rede social do twitter para promover o seu website/aplicação.

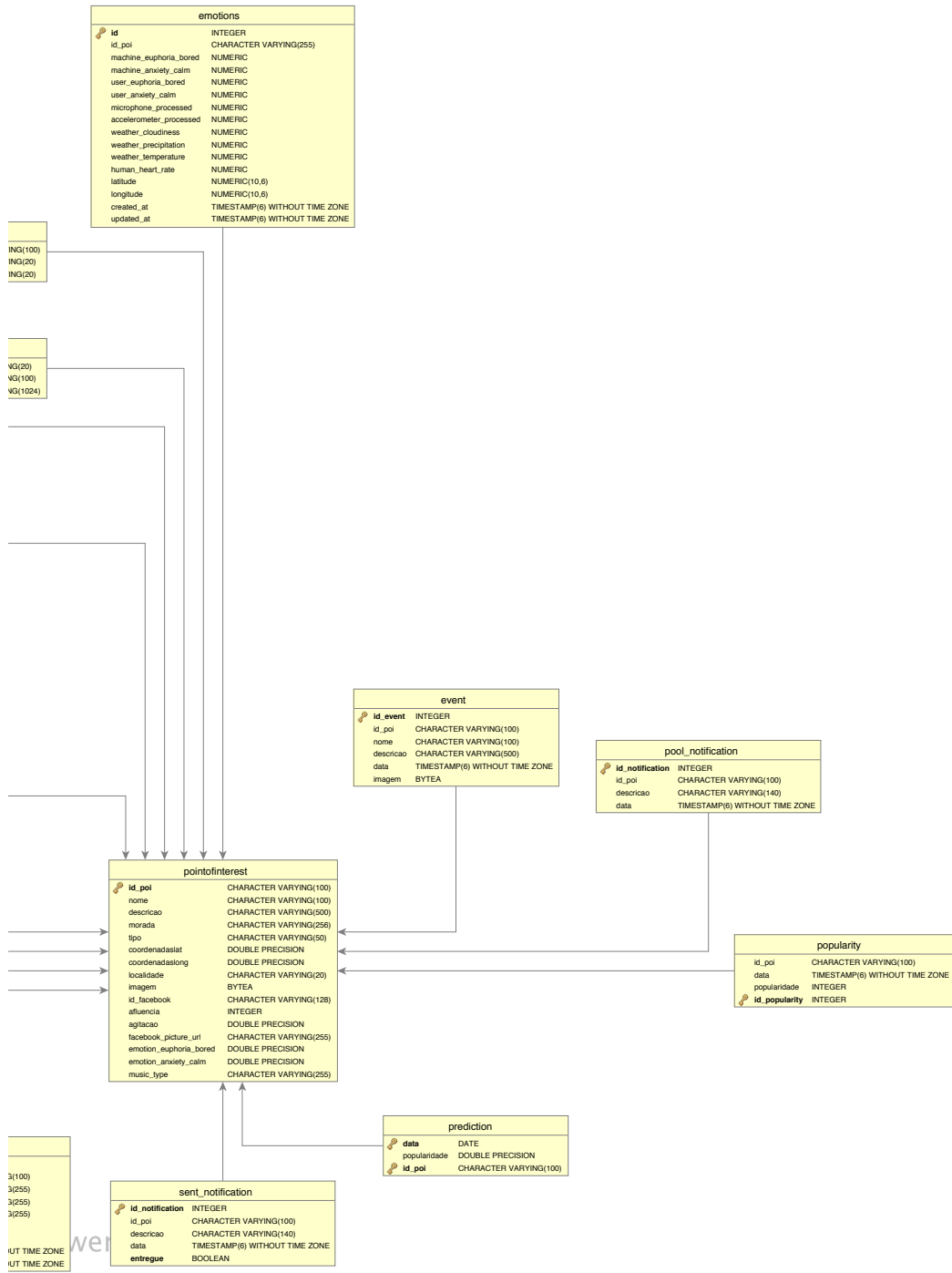
Google Plus API

O Google Plus é uma plataforma que pretende englobar vários produtos da companhia Google, complementado a experiência destes produtos com as funcionalidades de uma rede social. A autenticação em aplicações cliente também está disponível nesta plataforma, com o nome Google+ Sign-In[162]. Esta plataforma é um método de autenticação seguro, que deixa as pessoas iniciarem sessões em aplicações externas, com as credenciais da google, tanto através do browser ou nos dispositivos Android. A API está organizada por recursos, que podem ser pessoas, actividades, comentários ou momentos. Cada recurso tem uma ou mais informações que pode ser obtidas através de vários métodos. No recurso Pessoa, podemos listar as pessoas que existem, pesquisar pessoas, ou por exemplo listar todas as pessoas que *repostaram* uma actividade.

ANEXO F

Diagrama de Entidade-Relacionamento





BIBLIOGRAFIA

BIBLIOGRAFIA

- [1] D. Nunes, D. Raposo, D. Silva, P. Carmona, and J. S. Silva, “Achieving context-oriented seamless handoff through multipath tcp,” Submitted to the 2014 IEEE Global Communications Conference, GLOBECOM 2014, Austin, TX, USA, April 2014.
- [2] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “TCP Extensions for Multipath Operation with Multiple Addresses,” RFC 6824 (Experimental), Internet Engineering Task Force, Jan. 2013. [Online]. Available: <http://www.ietf.org/rfc/rfc6824.txt>
- [3] (2014, Julho) Site da vitaljacket. biodevices. [Online]. Available: <http://www.biodevices.pt/>
- [4] (2014, Julho) Site do bitalino. BITalino. [Online]. Available: www.bitalino.com/
- [5] (2014, Julho) Trello. trello. [Online]. Available: <https://trello.com/>
- [6] (2014, Agosto) Manifesto for async software development. asyncmanifesto. [Online]. Available: <http://asyncmanifesto.org/>
- [7] (2014, Agosto) Skype. Microsoft. [Online]. Available: <http://www.skype.com/>
- [8] A. Barnston, “The effect of weather on mood, productivity, and frequency of emotional crisis in a temperate continental climate,” *International Journal of Biometeorology*, vol. 32, no. 2, pp. 134–143, 1988. [Online]. Available: <http://dx.doi.org/10.1007/BF01044907>
- [9] Z. Spasova, “The effect of weather and its changes on emotional state - individual characteristics that make us vulnerable,” *Advances in Science and Research*, vol. 6, pp. 281–290, Mar. 2011.

- [10] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, “A survey of mobile phone sensing,” *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, Sept 2010.
- [11] (2014, Agosto) Health benefits of physical activity. Beverage Institute. [Online]. Available: <http://beverageinstitute.org/us/article/health-benefits-of-physical-activity/>
- [12] R. W. Picard, *Affective Computing*. Cambridge, MA, USA: MIT Press, 1997.
- [13] H. Gunes and M. Pantic, “Automatic, dimensional and continuous emotion recognition,” *International Journal of Synthetic Emotions*, vol. 1, no. 1, pp. 68–99, January 2010. [Online]. Available: <http://doc.utwente.nl/75888/>
- [14] J. Wagner, J. Kim, and E. Andre, “From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification,” in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, July 2005, pp. 940–943.
- [15] H.-J. Kim and Y. S. Choi, “Exploring emotional preference for smartphone applications,” in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, Jan 2012, pp. 245–249.
- [16] C. Akbiyik, “Can affective computing lead to more effective use of ict in education?”
- [17] N. Lathia, V. Pejovic, K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, “Smartphones for large-scale behavior change interventions,” *IEEE Pervasive Computing*, vol. 12, no. 3, pp. 66–73, 2013.
- [18] “Cisco visual networking index: Forecast and methodology, 2013–2018,” Cisco, 2013.
- [19] A. Balasubramanian, R. Mahajan, and A. Venkataramani, “Augmenting mobile 3g using wi-fi,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys ’10. New York, NY, USA: ACM, 2010, pp. 209–222.
- [20] A. Ford, C. Raiciu, M. Handley, S. Barré, and J. Iyengar, “Architectural guidelines for multipath tcp development,” Internet Engineering Task Force (IETF) , Request for Comments 6182, p. 28, Agosto 2014.

- [21] (2014, Agosto) Multipath tcp - linux kernel implementation. UCLouvain. [Online]. Available: <http://mptcp.info.ucl.ac.be/>
- [22] N. Williams, L. Stewart, and G. Armitage, “Design Overview of Multipath TCP version 0.3 for FreeBSD-10,” Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 130424A, 24 April 2013. [Online]. Available: <http://caia.swin.edu.au/reports/130424A/CAIA-TR-130424A.pdf>
- [23] (2014, Agosto) Maximize mobile user experience with netscaler multipath tcp. citrix. [Online]. Available: <http://blogs.citrix.com/2013/05/28/maximize-mobile-user-experience-with-netscaler-multipath-tcp/>
- [24] (2014, Agosto) ios: Multipath tcp support in ios 7. Apple. [Online]. Available: <http://support.apple.com/kb/HT5977>
- [25] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley, “Opportunistic mobility with multipath tcp,” in *Proceedings of the sixth international workshop on MobiArch*, ser. MobiArch ’11. New York, NY, USA: ACM, 2011, pp. 7–12.
- [26] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, “Exploring mobile/wi-fi handover with multipath tcp,” in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, ser. CellNet ’12. New York, NY, USA: ACM, 2012, pp. 31–36.
- [27] C. Pluntke, L. Eggert, and N. Kiukkonen, “Saving mobile device energy with multipath tcp,” in *Proceedings of the sixth international workshop on MobiArch*, ser. MobiArch ’11. New York, NY, USA: ACM, 2011, pp. 1–6.
- [28] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, “The future of human-in-the-loop cyber-physical systems,” *Computer*, vol. 99, no. PrePrints, p. 1, 2012.
- [29] P. Z. David Nunes and J. S. Silva, “A survey on human-in-the-loop applications towards an internet of all,” Julho 2014, submitted to IEEE Communications Surveys & Tutorials.
- [30] R. S. Desmond, M. F. Dickerman, and J. A. Fleming, “A human-in-the-loop cyber physical system: Modular designs for semi-autonomous wheelchair navigation,” Master’s thesis, Worcester Polytechnic Institute, 2013.

- [31] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, “The smart thermostat: Using occupancy sensors to save energy in homes,” in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’10. New York, NY, USA: ACM, 2010, pp. 211–224. [Online]. Available: <http://doi.acm.org/10.1145/1869983.1870005>
- [32] D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky, “Toward patient safety in closed-loop medical device systems,” in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*. ACM, 2010, pp. 139–148.
- [33] M. G. Berman, E. Kross, K. M. Krpan, M. K. Askren, A. Burson, P. J. Deldin, S. Kaplan, L. Sherdell, I. H. Gotlib, and J. Jonides, “Interacting with nature improves cognition and affect for individuals with depression,” *Journal of Affective Disorders*, vol. 140, no. 3, pp. 300 – 305, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165032712002005>
- [34] C. C. for Health, “The benetits of regular walking for health, well-being and the environment,” September 2012. [Online]. Available: www.mind.org.uk/mindweek
- [35] Mind, “Ecotherapy - the green agenda for mental health,” Agosto 2014. [Online]. Available: www.mind.org.uk/mindweek
- [36] M. Atkinson and L. Weigand, “A review of literature: The mental health benefits of walking and bicycling,” 2008.
- [37] M. G. Berman, E. Kross, K. M. Krpan, M. K. Askren, A. Burson, P. J. Deldin, S. Kaplan, L. Sherdell, I. H. Gotlib, and J. Jonides, “Interacting with nature improves cognition and affect for individuals with depression,” *Journal of affective disorders*, vol. 140, no. 3, pp. 300–305, 2012.
- [38] (2014, Agosto) Aplicação emotion sense. Emotion Sense. [Online]. Available: https://play.google.com/store/apps/details?id=com.ubhave.emotionsense&hl=pt_PT
- [39] (2014, Agosto) Mood-tracking app paves way for pocket therapy. Emotion Sense. [Online]. Available: <http://www.cam.ac.uk/research/news/mood-tracking-app-paves-way-for-pocket-therapy>

- [40] (2014, Agosto) Aplicação nike running. Nike. [Online]. Available: <http://1opinionatedwoman.com/tag/nike-running-app/page/2/>
- [41] (2014, Agosto) Aplicação nike running. Nike. [Online]. Available: http://nikeplus.nike.com/plus/products/gps_app/
- [42] (2014, Agosto) Aplicação nike running. Nike. [Online]. Available: <http://www.technologyguide.com/softwarereview/nike-gps-running-app-review/>
- [43] Facebook, “Facebook website,” The Facebook, Agosto 2014. [Online]. Available: <https://www.facebook.com/>
- [44] “Dresscode website,” DressCode, Agosto 2014. [Online]. Available: <http://www.dresscodeapp.com/>
- [45] “Dresscode facebook page,” Facebook, Agosto 2014. [Online]. Available: <https://www.facebook.com/DressCodeApp/info>
- [46] “Foursquare website,” Foursquare, Agosto 2014. [Online]. Available: <https://pt.foursquare.com>
- [47] (2014, Agosto) Swarm in google play. Google Play. [Online]. Available: https://play.google.com/store/apps/details?id=com.foursquare.robin&referrer=af_tranid%3D3PRJ43HY7Z723KVZ%26c%3Dswarmcid%26pid%3Dfoursquare.com
- [48] “About foursquare,” Foursquare, Agosto 2014. [Online]. Available: <https://pt.foursquare.com/about>
- [49] D. Silva, “Combinação entre mundos reais e mundos virtuais - caso de estudo sapo.labs.”
- [50] “Foursquare application for ios,” Foursquare iTunes Store, Agosto 2014. [Online]. Available: <https://itunes.apple.com/pt/app/foursquare/id306934924?mt=8>
- [51] “Foursquare application for android,” Foursquare Play Store, Agosto 2014. [Online]. Available: https://play.google.com/store/apps/details?id=com.joelapenna.foursquared&hl=pt_PT
- [52] “Foursquare application for windows phone,” Foursquare WindowsPhone Store, Agosto 2014. [Online]. Available: <http://www.windowsphone.com/pt-pt/store/app/foursquare/26cf3302-469f-e011-986b-78e7d1fa76f8>

- [53] “Foursquare application for blackberry,” Foursquare Blackberry store, Agosto 2014. [Online]. Available: <https://appworld.blackberry.com/webstore/content/6921/?countrycode=PT&lang=en>
- [54] “Foursquare application for symbian,” Foursquare Ovi store, Agosto 2014. [Online]. Available: <http://store.ovi.com/publisher/foursquare>
- [55] (2014, Agosto) Swarm. Foursquare. [Online]. Available: <https://www.swarmapp.com/>
- [56] (2014, Agosto) A look into the future of foursquare, including a new app called swarm. Foursquare. [Online]. Available: <http://blog.foursquare.com/post/84422758243/a-look-into-the-future-of-foursquare-including-a-new>
- [57] “Facebook: 10 years of social networking, in numbers,” The Guardian, Agosto 2014. [Online]. Available: <http://www.theguardian.com/news/datablog/2014/feb/04/facebook-in-numbers-statistics>
- [58] “Facebook application for ios,” Apple iTunes Store, Agosto 2014. [Online]. Available: <https://itunes.apple.com/pt/app/facebook/id284882215?mt=8>
- [59] “Facebook application for blackberry,” Blackberry Store, Agosto 2014. [Online]. Available: <http://blogs.blackberry.com/2010/11/facebook-places-blackberry/>
- [60] “Facebook application for android,” Google Play Store, Agosto 2014. [Online]. Available: https://play.google.com/store/apps/details?id=com.facebook.katana&hl=pt_PT
- [61] (2014, Agosto) Facebook application for windows phone. Windows Phone Store. [Online]. Available: <http://www.windowsphone.com/pt-pt/store/app/facebook/82a23635-5bd9-df11-a844-00237de2db9e>
- [62] (2014, Agosto) Facebook nearby is now facebook “local search”. searchengineland. [Online]. Available: <http://searchengineland.com/facebook-nearby-is-now-facebook-local-search-154507>
- [63] (2014, Agosto) Site da aplicação waze. Waze. [Online]. Available: <https://www.waze.com/>
- [64] (2014, Agosto) Sobre o waze. Waze. [Online]. Available: <https://www.waze.com/about>

- [65] (2014, Agosto) Google melhora serviço de mapas com aquisição da waze. Jornal de Noticias. [Online]. Available: http://www.jn.pt/PaginaInicial/Tecnologia/Interior.aspx?content_id=3267318
- [66] (2014, Agosto) Waze goes seriously social: Now you can share driving trips and easily pick up friends. Venture Beat. [Online]. Available: <http://venturebeat.com/2012/10/31/waze-social-app-updates/>
- [67] (2014, Agosto) Waze wiki - your rank and points. Waze Wiki. [Online]. Available: https://www.waze.com/wiki/Your_Rank_and_Points
- [68] (2014, Agosto) Imagens da aplicação waze wiki. Bloomberg. [Online]. Available: http://go.bloomberg.com/tech-blog/files/2012/06/blog_wazesceenshots.jpg
- [69] (2014, Agosto) Aplicação ios yelp. Apple. [Online]. Available: <https://itunes.apple.com/us/app/yelp/id284910350?mt=8>
- [70] (2014, Agosto) Sobre a yelp. Yelp. [Online]. Available: <http://www.yelp.com/about>
- [71] (2014, Agosto) Diretrizes de conteúdo. Yelp. [Online]. Available: <http://www.yelp.com/guidelines>
- [72] (2014, Agosto) Imagem da aplicação web yelp. Search Engine Journal. [Online]. Available: <http://www.searchenginejournal.com/study-reports-90-trust-yelp-reviews/68617/>
- [73] (2014, Março) Imagem da aplicação móvel yelp. Search Engine Journal. [Online]. Available: <http://aieseccp.wordpress.com/2013/06/14/appsintercambio/>
- [74] (2014, Agosto) Logótipo do shopkick. ShopKick. [Online]. Available: <https://itunes.apple.com/us/app/free-rewards-for-shopping/id383298204?mt=8>
- [75] (2014, Agosto) Site do shopkick. ShopKick. [Online]. Available: <https://www.shopkick.com/>
- [76] (2014, Agosto) Sobre do shopkick. ShopKick. [Online]. Available: <https://www.shopkick.com/about>
- [77] (2014, Agosto) Parceiros do shopkick. ShopKick. [Online]. Available: <https://www.shopkick.com/partners>

- [78] (2014, Junho) Echoprint. Echoprint. [Online]. Available: <http://echoprint.me>
- [79] (2014, Junho) Freetantrum is an umbrella codename for the open-source projects of etantrum, inc, including the etantrum songprint music identification library. Freetantrum. [Online]. Available: <http://sourceforge.net/projects/freetantrum/>
- [80] (2014, Junho) A java library for fingerprinting music files, assigning unique uris to them, and matching them to unknown audio clips. MusicURI. [Online]. Available: <https://github.com/dkourtesis/musicuri>
- [81] (2014, Junho) jhears is an acoustic fingerprinting framework. jHears. [Online]. Available: <https://gitorious.org/jhears>
- [82] (2014, Junho) Libfooid free audio fingerprinting library. Libfooid. [Online]. Available: <https://code.google.com/p/libfooid/>
- [83] (2014, Junho) Audioscout distributed audio content indexing system. audioscout. [Online]. Available: <https://code.google.com/p/audioscout/>
- [84] (2014, Junho) phash the open source perceptual hash library. pHash. [Online]. Available: <http://www.phash.org/>
- [85] (2014, Junho) The echo nest. The Echo Nest. [Online]. Available: <http://www.echonest.com/>
- [86] (2014, Junho) Musicbrainz - the open music encyclopedia. MusicBrainz. [Online]. Available: musicbrainz.org
- [87] (2014, Junho) Spotify: Music for everyone. Spotify. [Online]. Available: <http://www.spotify.com/>
- [88] (2014, Junho) Spotify acquires the echo nest, gaining control of the music dna company that powers its rivals. Techcrunch. [Online]. Available: <http://techcrunch.com/2014/03/06/spotify-acquires-the-echo-nest/>
- [89] (2014, Junho) Echo nest api overview. The Echo Nest. [Online]. Available: <http://developer.echonest.com/docs/v4>
- [90] (2014, Junho) Last.fm. Last.fm. [Online]. Available: <http://www.lastfm.com>
- [91] (2014, Junho) Fingerprint api and app updated. Last.fm. [Online]. Available: <http://blog.last.fm/2007/12/07/client-14-released-fingerprinting-begins>

- [92] (2014, Junho) Last.fm. Last.fm. [Online]. Available: <http://www.lastfm.com.br/api/tos>
- [93] (2014, Junho) Acoustid. Acoustid. [Online]. Available: <http://acoustid.org/>
- [94] (2014, Junho) Acoustid statistics. Acoustid. [Online]. Available: <http://acoustid.org/stats>
- [95] (2014, Junho) How does chromaprint work? oxygene. [Online]. Available: <https://oxygene.sk/2011/01/how-does-chromaprint-work/>
- [96] (2014, Agosto) Gracenote. Gracenote. [Online]. Available: <https://www.gracenote.com/>
- [97] (2014, Agosto) Sony corporation of america. Sony. [Online]. Available: www.sony.com/SCA/
- [98] (2014, Agosto) Gracenote opens up its apis and sdks for developers. gigaom. [Online]. Available: <http://gigaom.com/2013/02/15/gracenote-opens-up-its-apis-and-sdks-for-developers/>
- [99] (2014, Agosto) Gracenote. Gracenote. [Online]. Available: <https://www.gracenote.com/press/02/15/2013/>
- [100] (2014, Agosto) Gracenote sdk. Gracenote. [Online]. Available: <https://developer.gracenote.com/gnsdk>
- [101] (2014, Junho) Echoprint for android. The Mobile Applications and Services Laboratory (MASL). [Online]. Available: <http://masl.cis.gvsu.edu/2012/01/25/android-echoprint/>
- [102] R. Guinness. (2014, Agosto) Beyond where to how: A machine learning approach for sensing mobility contexts using smartphone sensors. Tampere University of Technology, Finland. [Online]. Available: <http://www.ion.org/gnss/abstracts.cfm?paperID=856>
- [103] D. Baptista and F. Morgado Dias, “A survey of artificial neural network training tools,” *Neural Computing and Applications [Impact Factor SJR2012: 1.168]*, vol. 23, no. 3-4, pp. 609–615, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s00521-013-1408-9>
- [104] Java. Oracle. [Online]. Available: <http://www.java.com/>

- [105] J. Heaton. (2014, Agosto) Encog machine learning framework. Heaton Research, Inc. [Online]. Available: <http://www.heatonresearch.com/encog>
- [106] Y. V. . S. Y. Abeel, T.; de Peer, “Java-ml: A machine learning library,” *Journal of Machine Learning Research*, pp. 931–934, 10 2009. [Online]. Available: <http://java-ml.sourceforge.net/>
- [107] (2014, Agosto) Java-ml weka classifier. Java-ML. [Online]. Available: <http://java-ml.sourceforge.net/content/weka-classifier>
- [108] (2014, Agosto) an object oriented neural engine. Sourceforge - joone. [Online]. Available: <http://sourceforge.net/projects/joone/>
- [109] (2014, Julho) Joone 1.2.1. nixbit. [Online]. Available: <http://nixbit.com/cat/science-and-engineering/artificial-intelligence/joone/>
- [110] (2014, Agosto) Java neural network framework. Neuroph. [Online]. Available: <http://neuroph.sourceforge.net/>
- [111] (2014, Julho) Encog and neuroph collaboration announcement. neuroph. [Online]. Available: <http://neuroph.sourceforge.net/NeurophEncogCollaboration.html>
- [112] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1656274.1656278>
- [113] (2014, Agosto) Comparing neural networks in neuroph, encog and joone. codeproject. [Online]. Available: <http://www.codeproject.com/Articles/85385/Comparing-Neural-Networks-in-Neuroph-Encog-and-JOO>
- [114] (2014, Agosto) Benchmarking and comparing encog, neuroph and joone neural networks. codeproject. [Online]. Available: <http://www.codeproject.com/Articles/85487/Benchmarking-and-Comparing-Encog-Neuroph-and-JOONE>
- [115] Z. Spasova, “The effect of weather and its changes on emotional state - individual characteristics that make us vulnerable,” *Advances in Science and Research*, vol. 6, pp. 281–290, 2011. [Online]. Available: <http://www.adv-sci-res.net/6/281/2011/>

- [116] (2014, Agosto) Open weather map. Open Weather Map. [Online]. Available: <http://openweathermap.org/>
- [117] (2014, Agosto) Accuweather api. AccuWeather. [Online]. Available: <http://api.accuweather.com/developers/>
- [118] (2014, Julho) Biblioteca owmclient. github. [Online]. Available: <https://github.com/migtavares/owmClient>
- [119] (2014, Julho) Biblioteca owmjapis. aksingh. [Online]. Available: <http://code.aksingh.net/owm-japis/src>
- [120] A. S. Andreas, A. Haag, S. Goronzy, P. Schaich, and J. Williams, “Emotion recognition using bio-sensors: First steps towards an,” in *In André et al (Eds.): Affective Dialogue Systems, Proceedings of the Kloster Irsee Tutorial and Research Workshop on Affective Dialogue Systems, Lecture Notes in Computer Science 3068, Springer-Verlag, 2004*, pp. 36–48.
- [121] H. Gunes, B. Schuller, M. Pantic, and R. Cowie, “Emotion representation, analysis and synthesis in continuous space: A survey,” in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (FG’11), EmoSPACE 2011 - 1st International Workshop on Emotion Synthesis, rePresentation, and Analysis in Continuous spaceE*, Santa Barbara, CA, USA, March 2011, pp. 827–834.
- [122] P. Rainville, A. Bechara, N. Naqvi, and A. R. Damasio, “Basic emotions are associated with distinct patterns of cardiorespiratory activity,” *International Journal of Psychophysiology*, vol. 61, no. 1, pp. 5 – 18, 2006, psychophysiology and Cognitive Neuroscience. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167876005002801>
- [123] (2014, Agosto) Mio alpha. mioglobal. [Online]. Available: <http://www.mioglobal.com/de/Im-Shop/MIO-Alpha/Product.aspx?ProductID=1&DeptID=1>
- [124] (2014, Agosto) H7 heart rate sensor. polar. [Online]. Available: http://www.polar.com/us-en/products/accessories/H7_heart_rate_sensor
- [125] (2014, Agosto) Zephyr hxm bluetooth wireless heart rate sensor for android and windows phone 8.

- Zephyr. [Online]. Available: <http://www.zephyranywherestore.com/Zephyr-Bluetooth-Wireless-Android-Windows/dp/B002PL33AQ>
- [126] (2014, Agosto) Bitalino | diy biosignals. BITalino. [Online]. Available: <http://www.bitalino.com/index.php/buy/product-finder>
- [127] D. H. Hansson. (2014, Agosto) Ruby on rails. [Online]. Available: <http://rubyonrails.org/>
- [128] (2014, Agosto) The php framework for web artisans. Laravel. [Online]. Available: <http://laravel.com/>
- [129] (2014, Agosto) Symphony. an open source cms. Symphony. [Online]. Available: <http://www.getsymphony.com/>
- [130] (2014, Agosto) Github. Github. [Online]. Available: <https://github.com/>
- [131] (2014, Agosto) Top github languages for 2013. Adam Bard. [Online]. Available: <http://adambard.com/blog/top-github-languages-for-2013-so-far/>
- [132] (2014, Agosto) Active stakeholder participation: An agile best practice. agilemodeling. [Online]. Available: <http://www.agilemodeling.com/essays/activeStakeholderParticipation.htm>
- [133] (2014, Agosto) User stories: an introduction. agilemodeling. [Online]. Available: <http://www.agilemodeling.com/artifacts/userStory.htm>
- [134] (2014, Julho) Platform versions. Google. [Online]. Available: <http://developer.android.com/about/dashboards/index.html>
- [135] A. L. Stanton, S. J. Sullivan, and J. L. Austenfeld, *Coping through emotional approach: Emerging evidence for the utility of processing and expressing emotions in responding to stressors*, 2nd ed. New York, NY, US: Oxford University Press, 2009, pp. 225–235, using Smart Source Parsing Oxford handbook of positive psychology (2nd ed.). (pp. 225-235). New York, NY : Oxford University Press. xxxi, 709 pp.
- [136] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY: Harper Perennial, March 1991. [Online]. Available: http://www.amazon.com/gp/product/0060920432/ref=si3_rdr_bb_product/104-4616565-4570345
- [137] A. Blum, *Neural Networks in C++*. John Wiley and Sons, Agosto 2014.

- [138] J. Heaton. (2014, Agosto) The number of hidden layers. Heaton Research, Inc. [Online]. Available: <http://www.heatonresearch.com/node/707encog>
- [139] N. Prasad, R. Singh, and S. Lal, “Comparison of back propagation and resilient propagation algorithm for spam classification,” in *Computational Intelligence, Modelling and Simulation (CIMSIM), 2013 Fifth International Conference on*, Sept 2013, pp. 29–34.
- [140] heart. (2014, Agosto) Target heart rates. [Online]. Available: http://www.heart.org/HEARTORG/GettingHealthy/PhysicalActivity/Target-Heart-Rates_UCM_434341_Article.jsp
- [141] (2014, Agosto) Sqlite home page. SQLite. [Online]. Available: <http://www.sqlite.org/>
- [142] (2014, Agosto) An application that allows to configure multipath tcp on a mptcp-enabled phones. github. [Online]. Available: https://github.com/mptcp-galaxys2/android_packages_apps_MPTCPCControl
- [143] (2014, Junho) Codegen for echoprint. Github. [Online]. Available: <https://github.com/echonest/echoprint-codegen>
- [144] (2014, Julho) Oauth. oauth. [Online]. Available: <http://oauth.net/2/>
- [145] (2014, Junho) A lightweight, flexible library for facebook with support for oauth authentication, the graph and rest apis, realtime updates, and test users. <http://developers.facebook.com/>. Github. [Online]. Available: <https://github.com/arsduo/koala>
- [146] (2014, Junho) The graph api. Facebook. [Online]. Available: <https://developers.facebook.com/docs/graph-api>
- [147] (2014, Junho) Ruby wrapper for the foursquare api. patches welcome. Github. [Online]. Available: <https://github.com/groupme/quimby>
- [148] (2014, Junho) The foursquare api. Foursquare. [Online]. Available: <https://developer.foursquare.com/>
- [149] (2014, Junho) The twitter ruby gem. Rdoc. [Online]. Available: <http://rdoc.info/gems/twitter>

- [150] (2014, Junho) Twitter developers. twitter. [Online]. Available: <https://dev.twitter.com/>
- [151] (2014, Julho) Haml. Haml. [Online]. Available: <http://haml.info/>
- [152] (2014, Julho) jquery. jquery. [Online]. Available: <http://jquery.com/>
- [153] (2014, Julho) foundation. foundation. [Online]. Available: <http://foundation.zurb.com/>
- [154] (2014, Julho) sass. sass. [Online]. Available: <http://sass-lang.com/>
- [155] (2014, Julho) ihealth. github. [Online]. Available: <https://github.com/TangRufus/iHealth/>
- [156] Debian. (2014, Agosto) Debian site. [Online]. Available: <http://www.debian.org>
- [157] wireshark. (2014, Agosto) Wireshark site. [Online]. Available: <http://www.wireshark.org/>
- [158] oasis open. (2014, Agosto) Especificação do mqtt. [Online]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>
- [159] (2014, Agosto)
- [160] Facebook. (2014, Agosto) Building facebook messenger. [Online]. Available: <https://www.facebook.com/notes/facebook-engineering/building-facebook-messenger/10150259350998920>
- [161] *Twitter+ Website*, Twitter, Agosto 2014. [Online]. Available: <https://dev.twitter.com/docs>
- [162] *Google+ Website*, Google, Agosto 2014. [Online]. Available: <https://developers.google.com/+api/>