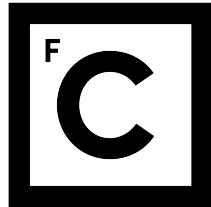


UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Ciências
ULisboa

**Operacionalização de Técnicas de Mudança
Comportamental em Agentes Conversacionais**

Maria Inês Teixeira Bastos

Mestrado em Informática

Trabalho de projeto orientado por:
Prof. Doutor João Carlos Balsa da Silva
e co-orientado pela Prof.^a Doutora Ana Paula Boler Cláudio

2021

*Para o meu avô Eduardo,
que sempre quis ver-me ir para a faculdade e nunca teve essa oportunidade.*

Agradecimentos

A realização deste trabalho não seria possível sem a participação de várias pessoas em diversos aspetos da minha vida.

Antes de mais, agradeço aos meus orientadores, ao Professor Doutor João Balsa da Silva e à Professora Doutora Ana Paula Cláudio, pela oportunidade de trabalhar com eles, e por todo o apoio e disponibilidade constante que me proporcionaram ao longo da concretização deste trabalho.

Agradeço também a toda a equipa do projecto VASelfCare, com a qual tive apenas um breve contacto, mas que me recebeu bastante bem e me pôs à vontade para qualquer dúvida e sugestão que tivesse. Foi o tipo de receção que só espero vir a ter nos meus futuros trabalhos. Um agradecimento especial à Isa Brito Félix, com quem tive oportunidade de reunir mais vezes e que se mostrou sempre disponível para esclarecer dúvidas e rever todo o progresso que foi sendo feito.

Um enorme obrigado aos meus pais, por me proporcionarem a oportunidade de frequentar este mestrado e, juntamente com o meu irmão, serem um constante apoio a nível pessoal, incentivando-me sempre a continuar, mesmo quando as coisas parecem não estar inclinadas a correr tão bem.

Agradeço ao meu namorado, Tiago, por todo o apoio, carinho, e por me incentivar sempre a continuar e a acreditar em mim. Obrigado por todas as vezes que um erro sem aparente sentido cismava em aparecer e me ajudavas a rever tudo até encontrar a razão.

E por último, mas não menos importante, agradeço a todos os meus amigos que, mesmo este tendo sido um ano difícil e mais distante para todos, me apoiaram e me ofereceram sempre uma palavra de força e conforto, quer tenha sido num café ou por videochamada.

Resumo

As intervenções no âmbito de mudança comportamental concretizadas através de aplicações de saúde eletrónica e móvel (*eHealth* e *mHealth*) estão a revolucionar as maneiras pelas quais os indivíduos podem monitorizar e melhorar os seus comportamentos e cuidados de saúde, melhorando os resultados, a experiência do paciente, e reduzindo custos.

Partindo do trabalho realizado no âmbito do projeto VASelfCare, focado no uso de técnicas de mudança comportamental bem estabelecidas numa intervenção *mHealth* baseada num agente conversacional, este trabalho propõe uma nova arquitetura para o *design* de agentes conversacionais no âmbito de uma intervenção de mudança comportamental.

Esta nova abordagem procura ultrapassar algumas das limitações apresentadas pelo agente precedente, combinando o uso de uma plataforma de linguagem natural avançada (Dialogflow) com a representação explícita, numa ontologia, de como as técnicas de mudança comportamental podem ser operacionalizadas. É exposto o modo de conceção e integração destes dois componentes no sistema, bem como o aspeto mais desafiador de utilizar os recursos avançados da plataforma de forma que permita ao agente conduzir o fluxo do diálogo e recorrer ao módulo de conhecimento externo, quando necessário.

Foi construída uma prova de conceito bem-sucedida, que pretende servir de base para o desenvolvimento de agentes conversacionais avançados, combinando ferramentas de linguagem natural com representação de conhecimento baseado numa ontologia.

Palavras-chave: Agentes Conversacionais; Ontologias; Plataformas de Diálogo; Intervenções de Mudança Comportamental.

Abstract

Behavior change interventions delivered through eHealth and mHealth applications are revolutionizing the ways in which individuals can monitor and improve their behaviors and health care, improving outcomes, the overall patient experience, and reducing costs.

Based on the work carried out within the scope of the VASelfCare project, focused on the use of well-established behavior change techniques in an mHealth intervention based on a conversational agent, this work proposes a new architecture for the design of conversational agents within the scope of a behavior change intervention.

This new approach seeks to overcome some of the limitations presented by the previous agent, combining the use of an advanced natural language platform (Dialogflow) with the explicit representation, in an ontology, of how behavior change techniques can be operationalized. It's exposed the process of designing and integrating these two components into the system, as well as the most challenging aspect of using the advanced features of the platform in a way that would allow the agent to conduct the dialogue flow and use the external knowledge module, when necessary.

A successful proof of concept was built, which aims to serve as base for the development of advanced conversational agents, combining natural language tools with ontology-based knowledge representation.

Keywords: Conversational Agents; Ontologies; Dialogue Platforms; Behavior Change Interventions.

Conteúdo

Lista de Figuras	xiv
Lista de Tabelas	xv
Abreviaturas	xvii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Contribuições	3
1.4 Estrutura do documento	4
2 Conceitos e trabalho relacionado	7
2.1 Conceitos de mudança comportamental	7
2.2 <i>eHealth e mHealth</i>	8
2.3 Embodied Conversational Agent (ECA)	9
2.4 Ferramentas de criação e gestão de diálogos	10
2.4.1 Twine	10
2.4.2 Yarn Spinner	11
2.4.3 WOOL	12
2.4.4 Watson Assistant	13
2.4.5 FAtiMA Toolkit	14
2.4.6 Dialogflow	15
2.4.7 Conclusão	17
2.5 Ontologias	18
3 Ontologia de conceitos de mudança comportamental	21
3.1 Identificação dos conceitos	21
3.2 Descrição da ontologia definida	24
3.3 Casos práticos exemplares	27
3.3.1 Operacionalização de BCTs	27
3.3.2 Inferir conhecimento sobre o utilizador	28

3.4	Aplicações noutros contextos	29
3.5	Conclusão	30
4	Desenvolvimento do motor de diálogo	31
4.1	Escolha da ferramenta a utilizar	31
4.2	Definição do agente no Dialogflow	32
4.3	Condução do fluxo de conversação	36
4.4	Processo de treino	37
4.5	Conclusão	37
5	Desenvolvimento da solução	39
5.1	Ferramentas utilizadas	39
5.2	Arquitetura do agente	40
5.3	<i>Integration Core</i>	42
5.3.1	<i>Application Controller</i>	42
5.3.2	<i>Webhook Service</i>	43
5.3.3	<i>User Interface</i>	45
5.4	<i>BCT Knowledge</i>	46
5.5	Conclusão	49
6	Prova de conceito	51
6.1	Interfaces da aplicação	51
6.2	Exemplo 1: Revisão dos objetivos	52
6.3	Exemplo 2: Não adesão ao objetivo	53
6.4	Conclusão	56
7	Conclusões e trabalho futuro	57
7.1	Conclusões	57
7.2	Trabalho futuro	58
	Bibliografia	61
A	Caracterização dos elementos de intervenção da componente de Alimentação	65
B	Caracterização dos elementos de intervenção da componente de Atividade Física	71
C	Questionário - Avaliação de risco de Diabetes Tipo 2	77
D	Integração com o Unity	79
D.1	Watson Assistant	79
D.2	FAtiMA Toolkit	81

Lista de Figuras

1.1	Diagrama simplificado do sistema implementado.	4
2.1	<i>Behavior Change Wheel</i> [Michie et al., 2014].	8
2.2	Grafo exemplificativo do Twine.	11
2.3	Grafo exemplificativo do Yarn Spinner.	12
2.4	Grafo exemplificativo do WOOL.	13
2.5	Árvore lógica que define o fluxo do diálogo, no menu <i>Dialog</i>	14
2.6	Componentes da FATiMA Toolkit [Guimarães et al., 2019].	15
2.7	Árvore lógica que define o fluxo de conversa de um agente no Dialogflow ES.	16
2.8	Fluxo de conversação de um agente no Dialogflow CX.	17
3.1	Distribuição das BCTs (identificadas pelo nome e código numérico da BCTTv1 [Michie et al., 2013]) de acordo com as etapas de diálogo numa interação (extraída de [Balsa et al., 2020]).	22
3.2	Visão geral das principais classes da ontologia.	25
3.3	Entidades relevantes à operacionalização de BCTs.	28
3.4	Inferência do tipo de um indivíduo.	29
4.1	Fluxo de conversação criado para o agente.	33
4.2	Página <i>Counselling</i>	34
4.3	Definição da entidade <i>motivação</i>	35
4.4	Frases de treino da <i>intent</i> “affirmative-answer”.	36
5.1	Arquitetura do sistema.	41
5.2	Variáveis necessárias à autenticação do agente do Dialogflow.	43
5.3	Função “main()”, executada quando o utilizador é redirecionado para a página de conversação.	43
5.4	Função “send_message()”, que recebe a mensagem do utilizador e retorna a resposta do agente.	44
5.5	Processo de geração e definição do URL para o <i>webhook</i>	44
5.6	Função “webhook()”, executada sempre que o agente faz um pedido ao serviço de <i>webhook</i>	45

5.7	Função “submit_message”.	46
5.8	Integração da ontologia através da classe “OntologyConnection”.	47
5.9	Amostra de código onde é feita uma consulta à ontologia.	47
5.10	Amostra do ficheiro counselling.json.	48
6.1	Interfaces da aplicação.	51
6.2	Propriedades associadas a três utilizadores diferentes.	52
6.3	Concretização da BCT <i>Self-monitoring of behavior (2.3)</i> em três utilizadores diferentes.	53
6.4	Demonstração da concretização de diversas BCTs.	54
6.5	Resposta <i>default</i> , retornada pelo <i>event handler</i> da página <i>Get Determinant</i> .	55
D.1	Exemplo de personalização de uma Intenção.	80
D.2	Exemplo de personalização de um nó de diálogo.	80
D.3	Exemplo de <i>input</i> textual, seguido da cena em que o assistente responde, acompanhado de fala.	81
D.4	Costumização do diálogo através da <i>FAtiMA Authoring Tool</i> .	82
D.5	Funcionalidade Text-To-Speech, presente na <i>FAtiMA Authoring Tool</i> .	82
D.6	Interface do utilizador da experiência de integração da FAtiMA Toolkit com o Unity.	83
E.1	O <i>Self-Assessment Manikin (SAM)</i> , usado para classificar as dimensões afetivas de valência (linha do topo), intensidade (linha central), e dominância (linha de fundo) [Bradley and Lang, 1994].	86
E.2	Exemplos de várias expressões faciais a escolher pelo utilizador ao usar o <i>AffectButton</i> [Broekens and Brinkman, 2009].	86
E.3	O <i>Affective Slider</i> , uma ferramenta digital de autorrelato constituída por dois <i>sliders</i> para a medição da intensidade (acima) e da valência (abaixo) numa escala contínua [Betella and Verschure, 2016].	87

Lista de Tabelas

2.1	Características relevantes das ferramentas analisadas.	17
3.1	Classes da ontologia, com identificação da respectiva superclasse e descrição.	26
3.2	Relações da ontologia (OP = <i>Object Property</i> , DP = <i>Data Property</i>), com identificação do seu domínio, contra-domínio e descrição.	27

Abreviaturas

API Application Programming Interface.

BCI Behavior Change Intervention.

BCT Behavior Change Technique.

BCTTv1 Behavior Change Techniques Taxonomy version 1.

BCW Behavior Change Wheel.

CA Conversational Agent.

CSS Cascading Style Sheets.

DT2 Diabetes Tipo 2.

ECA Embodied Conversational Agent.

HTML HyperText Markup Language.

JS JavaScript.

JSON JavaScript Object Notation.

OWL Web Ontology Language.

UI User Interface.

Capítulo 1

Introdução

1.1 Motivação

O uso generalizado de dispositivos móveis por pessoas de todas as idades tem sido responsável pelo aumento do desenvolvimento e uso de aplicações móveis para as mais diversas finalidades. Acompanhando este avanço na tecnologia, é cada vez mais comum vermos a prática da medicina e da saúde pública suportada por dispositivos móveis, atuando na prevenção, monitorização e diagnóstico de doenças.

Embora a utilização de aplicações *mobile health (mHealth)* não seja um substituto para os cuidados profissionais de saúde, estas podem servir de auxílio na prevenção e gestão de diversas patologias, fornecendo indicações aos indivíduos sobre os comportamentos desejáveis à sua prevenção, de forma a que possam tomar decisões bem informadas sobre o seu estilo de vida e comportamentos [IDF, 2014].

As intervenções comportamentais de saúde eletrónica e móvel (*eHealth* e *mHealth*) estão a revolucionar as maneiras pelas quais os indivíduos podem monitorizar e melhorar os seus comportamentos e cuidados de saúde, melhorando os resultados, a experiência do paciente, e reduzindo custos, ao mesmo tempo em que retêm a capacidade de fornecer o suporte abrangente, contínuo, personalizado e interativo necessário para aprimorar a saúde pública [Michie et al., 2017b].

Uma pesquisa realizada por Bhuyan et al. concluiu que 60% dos adultos que utilizam aplicações *mHealth* as consideram úteis para atingir os objetivos em relação à sua saúde [Bhuyan et al., 2016]. No entanto, foi observado por Zapata et al. que as aplicações *mHealth* são frequentemente usadas por pessoas com pouca experiência em tecnologia [Zapata et al., 2015]. Isto evidencia que, apesar do grande potencial para melhorar a saúde da população e a eficiência e o alcance da prestação dos cuidados de saúde, a conceção destas aplicações acarreta grande responsabilidade e o processo deve ser fundamentado em estudos científicos validados, uma vez que estas intervenções também têm potencial para causar danos (fornecerem conselhos inadequados, envolverem interações que prejudicam os comportamentos desejados, ou compartilharem dados inadequadamente, por

exemplo) [Michie et al., 2017b].

Avanços em reconhecimento de voz, processamento de linguagem natural e inteligência artificial levaram ao aumento da disponibilidade e uso de agentes conversacionais (CA, do inglês *Conversational Agent*) - sistemas que imitam a conversa humana usando texto ou fala [Laranjo et al., 2018]. Embora os CAs já venham a ser estudados e desenvolvidos há algumas décadas, nomeadamente desde o aparecimento de ELIZA em 1966 [Weizenbaum, 1966], a transição para a mudança comportamental é mais recente (por exemplo, para promover um estilo de vida mais saudável - normalmente em relação à atividade física e ao tipo de dieta).

Nas últimas duas décadas, têm sido evidenciados os potenciais benefícios do uso de agentes conversacionais corporizados (ECA, do inglês *Embodied Conversational Agent*) para fins relacionados à saúde, tendo-se mostrado também bem aceites pelos utilizadores em diversas aplicações [Baptista et al., 2020, Gong et al., 2020]. Vários ensaios clínicos de intervenções envolvendo ECAs mostraram melhorias significativas na atividade física, consumo de frutas e vegetais, entre outros resultados. No entanto, a maioria desses agentes apenas permitia um *input* restrito por parte do utilizador (por exemplo, escolha múltipla de opções de resposta), não tendo a capacidade de entender *input* de linguagem natural [Laranjo et al., 2018].

À luz das suas capacidades em expansão, os CAs têm o potencial de desempenhar um papel cada vez mais importante na saúde e nos cuidados médicos, no entanto, o uso de CAs dotados de capacidades de *input* de linguagem natural sem restrições para fins relacionados à saúde é ainda um campo de pesquisa emergente [Laranjo et al., 2018].

1.2 Objetivos

Tendo em conta a motivação acima, este trabalho tem como principal objetivo expandir o trabalho desenvolvido no âmbito do projeto VASelfCare¹, que tem vindo a ser desenvolvido com o objetivo de auxiliar a gestão e monitorização da Diabetes tipo 2 (DT2) em pessoas com mais de 65 anos [Félix et al., 2019, Balsa et al., 2020]. A aplicação VASelfCare consiste numa intervenção para a mudança comportamental dos utilizadores ao longo do tempo, nas três componentes essenciais para a gestão do tratamento da Diabetes tipo 2: adesão à medicação, à prática de actividade física e à adoção de hábitos alimentares saudáveis. No cerne deste tipo de intervenções encontram-se as técnicas de mudança comportamental (BCT, do inglês *Behavior Change Technique*), definidas como sendo um componente observável, replicável e irreduzível de uma intervenção projetada para mudar o comportamento humano [Michie et al., 2013].

A intervenção do VASelfCare tem a duração máxima de 90 dias, estando dividida nas três componentes mencionadas: Medicação, Atividade Física e Alimentação. Ao longo

¹<https://vaselfcare.esel.pt>

da intervenção, as BCTs são operacionalizadas em diversas fases do diálogo, através de explicações, sugestões ou outras interações direcionadas ao utilizador por parte de um assistente virtual. Por exemplo, para a BCT denominada *Social support (unspecified)*, o agente sugere recorrer a um amigo, familiar ou profissional de saúde, dependendo do determinante (i.e., um fator que influencia um comportamento de forma positiva ou negativa) de não adesão a um determinado comportamento; um exemplo da sua operacionalização é quando o agente diz “Considere convidar amigos ou familiares para comerem consigo ou visite-os ocasionalmente.”.

A interação com a aplicação VASelfCare é feita através de botões, que aparecem após a fala do assistente e que contêm várias opções de resposta, a partir das quais o utilizador escolhe a mais adequada ao seu caso, de forma a dar continuidade à conversa.

Com base no agente desenvolvido no VASelfCare, pretende-se construir um agente conversacional mais avançado e versátil, de forma a que possa vir a ser usado num maior número de situações ligadas ao contexto da saúde (por utilizadores com outras doenças crónicas, com multimorbilidade ou num diferente intervalo de idades, por exemplo).

Para que isso seja possível, é necessário ultrapassar duas limitações: (1) o *input* do utilizador ser escolhido a partir de um conjunto limitado de opções, e (2) a operacionalização das BCTs ser totalmente delineada aquando da criação prévia dos diálogos.

Assim, um dos principais objetivos a que este trabalho se propõe, é criar um agente conversacional que permita receber o *input* do utilizador através de escrita em linguagem natural, tornando a conversa mais apelativa e relacionável para o utilizador, que pode assim exprimir-se sem restrições (resposta real não estar presente nas opções, por exemplo). Outro dos objetivos deste trabalho é dotar o agente conversacional de conhecimento acerca da operacionalização de BCTs, sem que esteja embutido na definição do diálogo, tornando a sua execução mais dinâmica.

Ou seja, neste trabalho pretende-se apresentar uma nova abordagem para o desenvolvimento de agentes conversacionais que combine os recursos de *design* e desenvolvimento oferecidos por ferramentas avançadas de linguagem natural com um módulo de conhecimento necessário para apoiar o agente nos seus objetivos. Na Figura 1.1 encontra-se um esquema simplificado do agente implementado neste trabalho, onde é possível constatar a integração dos dois módulos que se pretendiam desenvolver.

1.3 Contribuições

Ao longo destes meses de trabalho, foi concebido um conjunto de ferramentas que permitem a operacionalização de técnicas de mudança comportamental em agentes conversacionais.

Inicialmente, foi feita uma análise ao trabalho desenvolvido previamente no âmbito do projeto VASelfCare. A análise focou-se na tentativa de abstração dos componentes en-

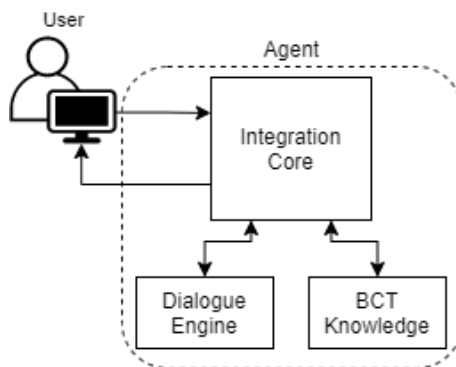


Figura 1.1: Diagrama simplificado do sistema implementado.

volvidos na execução das várias técnicas de mudança comportamental através do diálogo, de forma a tornar essa execução menos rígida, mais dinâmica e mais facilmente adaptável a diferentes tipos de situações ou problemas. Apesar de não estar no planeamento inicial deste trabalho, esta abstração levou à conceção e definição de uma ontologia focada nos conceitos de mudança comportamental, necessários à caracterização da operacionalização de BCTs e à estruturação de uma intervenção de mudança comportamental.

Para alcançar outro dos objetivos deste trabalho, foi criado um agente conversacional que permite a operacionalização de BCTs de forma mais dinâmica que o agente criado no âmbito do projeto VASelfCare. Para criar o agente conversacional, optou-se por adotar uma ferramenta existente, pelo que, diversas ferramentas de diálogo foram analisadas e testadas antes de se escolher a mais adequada e proceder à criação do agente. O agente resultante permite que o utilizador interaja com ele através de linguagem natural e em tempo real, diminuindo bastante a quantidade de diálogo que era necessário definir previamente, não só para as falas do utilizador como para as do agente.

Foi então projetada e implementada uma prova de conceito, uma aplicação *web*, que demonstra o resultado da integração dos componentes descritos acima e permite validar a sua usabilidade no contexto de uma intervenção de mudança comportamental.

Por fim, surgiu ainda a oportunidade de escrever um artigo para a conferência *International Conference on Agents and Artificial Intelligence (ICAART 2022)*. O artigo foi aceite e será apresentado nesta conferência, que se realiza no início de 2022. O artigo, com título homónimo ao do presente documento, é o seguinte:

Bastos MI, Cláudio AP, Carmo MB, Félix I, Guerreiro MP and Balsa J. Operationalizing Behavior Change Techniques in Conversational Agents. In Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022)

1.4 Estrutura do documento

O presente documento encontra-se dividido em sete capítulos, cada um deles constituído por várias secções, organizados de forma a facilitar a compreensão do trabalho desenvol-

vido, nomeadamente:

- O Capítulo 2 compreende uma análise da pesquisa realizada sobre aspetos relevantes para o trabalho desenvolvido, começando por descrever alguns conceitos que motivaram este trabalho (conceitos de mudança comportamental, aplicações *eHealth* e *mHealth*, e ECAs), seguido do levantamento de algumas ferramentas usadas para a gestão de diálogos. Por fim, é feita uma introdução acerca do uso de ontologias e das suas vantagens, evidenciando uma ontologia que está a ser desenvolvida na área em que este trabalho se foca.
- No Capítulo 3 é descrito o processo que levou à definição de uma ontologia de conceitos de mudança comportamental, a identificação dos conceitos a incluir, a ontologia resultante e alguns casos de uso.
- O Capítulo 4 contém todo o procedimento de conceção e definição do motor de diálogo do agente. Apresenta-se a ferramenta escolhida e descreve-se o processo de criação do agente conversacional.
- No Capítulo 5 é descrito o processo de implementação da solução proposta, que faz a integração dos componentes descritos nos capítulos 3 e 4. É feito um levantamento das ferramentas utilizadas e apresentada a arquitetura do agente desenvolvido, evidenciando a comunicação entre as várias ferramentas e componentes do sistema.
- O Capítulo 6 apresenta uma prova de conceito, que demonstra o funcionamento da solução implementada. As suas funcionalidades são evidenciadas através de exemplos práticos.
- No sétimo e último capítulo são feitas algumas considerações finais acerca do trabalho e justificações acerca das decisões tomadas, juntamente com algumas perspectivas de trabalho futuro.

Apesar deste documento estar escrito em português, optou-se por manter certos termos em inglês, como é o caso das entidades da ontologia, das etapas do diálogo e das componentes da arquitetura do agente desenvolvido. As entidades da ontologia e as etapas do diálogo são mencionadas várias vezes ao longo deste documento, não só em texto como também em figuras cruciais para entender a solução implementada. Assim, de forma a manter apenas uma versão dos termos e ser mais fácil relacionar as figuras com o que está a ser descrito, mantiveram-se os termos em inglês.

Capítulo 2

Conceitos e trabalho relacionado

Neste capítulo é apresentada uma introdução teórica de aspetos relevantes para melhor compreensão do trabalho desenvolvido. Inicia-se o capítulo com o esclarecimento de alguns conceitos relacionados com mudança comportamental (Secção 2.1), e com o aprofundamento dos termos *eHealth* e *mHealth* (Secção 2.2). Segue-se uma descrição dos *Embodied Conversational Agents* e quais as principais características a ter em conta na sua projecção (Secção 2.3), passando depois para o levantamento e descrição de algumas ferramentas de gestão de diálogos (Secção 2.4). No fim, é feito um levantamento sobre a definição e o uso de ontologias (Secção 2.5).

2.1 Conceitos de mudança comportamental

Uma intervenção de mudança comportamental (BCI, do inglês *Behavior Change Intervention*) é uma intervenção que envolve o uso simultâneo de um conjunto de técnicas, e que tem como objetivo influenciar o comportamento humano.

A mudança comportamental pode ser um objetivo difícil de alcançar. Assim, para ajudar a criar e melhorar o processo, a avaliação e o desenvolvimento teórico deste tipo de intervenções, foi desenvolvida a *Behavior Change Wheel* (BCW) [Michie et al., 2014]. Esta ferramenta é uma síntese de 19 modelos de mudança comportamental presentes na literatura, e funciona como um guia para escolher as técnicas de mudança comportamental (BCT, do inglês *Behavior Change Technique*) mais adequadas nos diversos contextos. A BCW baseia-se na hipótese de que a interação entre capacidade (C), oportunidade (O) e motivação (M) explicam porque é que um determinado comportamento (*Behavior*, B) é ou não realizado (COM-B). A BCW vincula os três domínios COM-B com nove funções de intervenção e, subsequentemente, a sete políticas com maior probabilidade de alcançar mudanças comportamentais (Figura 2.1) [Michie et al., 2014].

Por BCT, quer-se dizer um componente observável, replicável e irreduzível de uma intervenção projetada com o objetivo de alterar ou redirecionar processos causais que regulam o comportamento; isto é, uma técnica é proposta para ser um “ingrediente ativo”

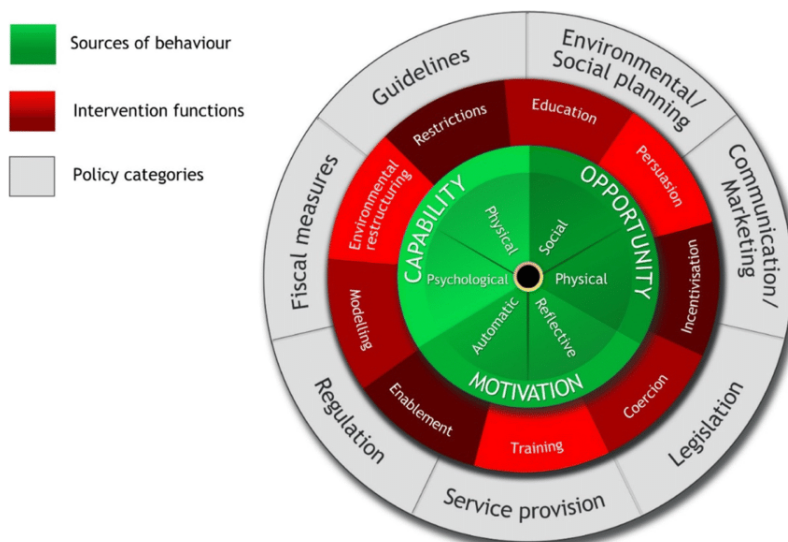


Figura 2.1: *Behavior Change Wheel* [Michie et al., 2014].

na intervenção. As BCTs podem ser usadas sozinhas ou em combinação e numa variedade de formatos [Michie et al., 2013].

Com o aumento de trabalhos no campo das BCIs, surgiu a necessidade de uniformizar as definições de cada BCT, uma vez que muitas vezes surgiam diferentes técnicas com o mesmo nome e vice-versa. Esta necessidade resultou no desenvolvimento de uma extensa taxonomia, organizada hierarquicamente, de 93 BCTs claramente definidas e não redundantes. Esta taxonomia foi designada por “*Behavior Change Technique Taxonomy (v1)*” (BCTTv1) [Michie et al., 2013].

Uma das vantagens da BCTTv1 é o facto de ser relevante para uma vasta gama de comportamentos, ao invés de ser restrito a um único domínio comportamental. A BCTTv1 fornece exemplos de como as BCTs podem ser implementadas e fornece aos utilizadores detalhes suficientes para operacionalizar as BCTs [Michie et al., 2013].

No VASelfCare e no atual trabalho, a referência base para a aplicação das BCTs é o trabalho que tem vindo a ser desenvolvido por Michie et al., sendo o desenho da intervenção desenvolvida sustentado pela BCW [Michie et al., 2014]. Durante o desenvolvimento do VASelfCare, a BCW permitiu a identificação de onze BCTs consideradas as mais adequadas à intervenção para promover a mudança comportamental em idosos por meio de um assistente virtual antropomórfico [Félix et al., 2019]. As BCTs em causa serão detalhadas mais à frente, no Capítulo 3.

2.2 *eHealth e mHealth*

A Organização Mundial de Saúde (OMS) define *eHealth* como a utilização de tecnologias de informação e comunicação no âmbito dos sistemas de saúde, sendo reconhecida como

uma das áreas que apresenta um crescimento mais rápido no setor da saúde¹.

O termo *eHealth* é amplamente usado quando nos queremos referir à utilização dos sistemas de informação de saúde, sendo estes não só ferramentas para as autoridades e profissionais de saúde, mas também sistemas de saúde personalizados para os utentes, tais como, o registo eletrónico dos utentes, a telemedicina, e todo um conjunto de instrumentos de base tecnológica desenhadas para a prevenção, diagnóstico, tratamento, monitorização e gestão da saúde do utente [Espanha and Fonseca, 2010].

Reconhecendo que a área da *eHealth* está a transformar rapidamente a prestação de serviços e sistemas de saúde em todo o mundo, a OMS tem desempenhando um papel central na definição e monitorização do seu futuro, tendo em 2005 inaugurado o Observatório Global para *eHealth*, uma iniciativa dedicada ao estudo da evolução da *eHealth* e do seu impacto na saúde dos países¹.

mHealth, ou *mobile health*, é um caso particular de *eHealth*, definido pelo Observatório Global para *eHealth* como sendo a prática médica e de saúde pública suportada por dispositivos móveis, dispositivos de monitorização de pacientes, assistentes pessoais digitais (PDA) ou outros dispositivos sem fio [Global Observatory for eHealth, 2011].

2.3 Embodied Conversational Agent (ECA)

Muitas das aplicações *eHealth* existentes fornecem conselhos na forma de texto simples ou por meio de um módulo de perguntas e respostas baseado em texto, no entanto, a interação face a face continua a ser uma das melhores maneiras de transmitir a informação no que diz respeito à área da saúde, suscitando maior confiança e melhor comunicação [ter Stal et al., 2020]. Posto isto, introduzir uma forma de interação face a face parece ser a melhor aposta de forma a obter um maior envolvimento por parte do utilizador e estimular o uso das aplicações *eHealth*. Para esse efeito, o uso de *Embodied Conversational Agents* (ECAs) tem vindo a ganhar bastante visibilidade [ter Stal et al., 2020].

Um *Embodied Conversational Agent* é descrito como sendo um agente conversacional que esteja incorporado num qualquer sistema, agrupando uma interface de fala natural de um sistema de diálogo com uma incorporação visual ou avatar [Cassell et al., 2000]. O facto de dar um corpo ao agente virtual permite-o assumir um papel mais social durante a interação, uma vez que os utilizadores podem comunicar com uma entidade com a qual se possam relacionar. Graças a esta capacidade de socializar a um nível mais humano, os ECAs têm sido empregues num vasto número de aplicações que atuam em várias áreas, desde o ensino até à saúde [Wang et al., 2020].

Além das conversas que os ECAs são capazes de manter com os utilizadores, torna-se também necessário incorporar aspetos sociais, emocionais e relacionais da interação face a face num agente computacional, sendo especialmente crucial para agentes proje-

¹<https://www.who.int/observatories/global-observatory-for-ehealth>

tados para interagir com o utilizador durante um longo período de tempo. Assim sendo, não basta apenas projetar um agente que seja um conversador competente, é também necessário projetar o agente para o uso de comportamentos de construção e manutenção de relacionamentos [Bickmore et al., 2005].

A aplicação VASelfCare centra-se num agente que desempenha o papel de uma assistente virtual inteligente, chamada Vitória, que responde às necessidades das pessoas mais velhas com diabetes tipo 2 relativamente à prática de atividade física, toma de medicação e planeamento de hábitos alimentares equilibrados. O avatar da Vitória foi desenvolvido como sendo um ECA em 3D com uma representação antropomórfica do sexo feminino, que interage verbalmente com o utilizador, demonstrando emoções, e algumas expressões faciais e movimentos corporais.

2.4 Ferramentas de criação e gestão de diálogos

No VASelfCare foi utilizado o Unity² para criar os aspetos visuais da aplicação, como a vista de uma sala de estar com objectos 3D, o modelo antropomórfico 3D da assistente virtual, e os menus e botões para a interação do utilizador com a aplicação. Os ficheiros de diálogos das interações entre o utilizador e o agente foram criados *a priori*, usando a ferramenta Yarn Spinner³.

Para este trabalho, optou-se novamente por adotar uma ferramenta já existente, dada a vasta disponibilidade de excelentes ferramentas para este propósito. As duas características principais pretendidas eram: a possibilidade de incorporar interação de linguagem natural e manter o controlo do fluxo do diálogo, de forma a abordar situações em que o agente é o responsável por conduzir a conversa (e não o contrário, como acontece nos contextos de pergunta e resposta típicos de *chatbots*, por exemplo).

Posto isto, diversas ferramentas de diálogo foram consideradas, analisadas e testadas antes de se escolher a mais adequada, estando estas descritas nas secções seguintes. Na Secção 2.4.7 encontra-se uma tabela sumário com as características mais relevantes e que foram tidas em conta para a escolha da ferramenta a utilizar.

2.4.1 Twine

O Twine⁴ é uma ferramenta *open source* gratuita que permite criar histórias interativas e não lineares através da criação de grafos. As histórias criadas podem ser simples diálogos ou podem ser enriquecidas com a adição de variáveis, condições, imagens, vídeos, CSS e JavaScript.

²<https://unity.com>

³<https://yarnspinner.dev/>

⁴<https://twinery.org/>

Nos grafos, cada nó é constituído por uma fala e pelas várias opções de resposta, sendo os vários rumos que a conversa pode tomar representados por setas. A Figura 2.2 mostra um pequeno exemplo de como se parecem os nós no editor *online* do Twine. A fala que o personagem virtual diria é escrita normalmente, e abaixo, as opções de resposta por parte do utilizador são escritas com o formato seguinte: `[[opção de resposta | nó de destino]]`

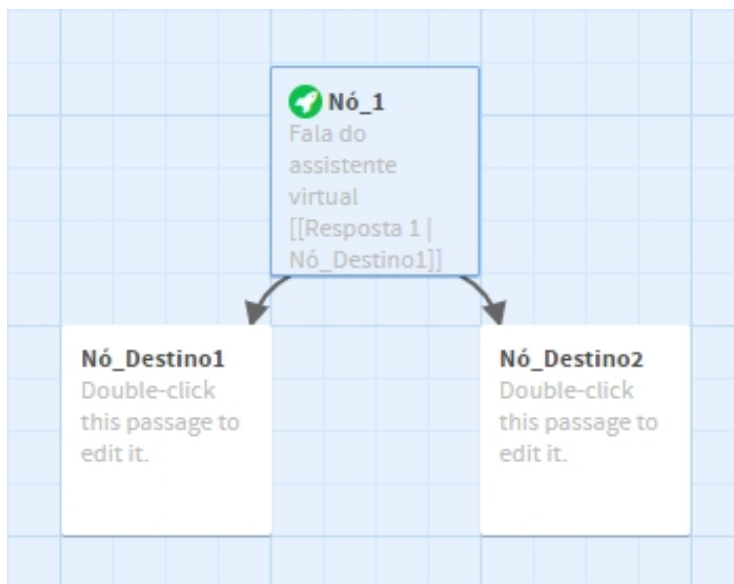


Figura 2.2: Grafo exemplificativo do Twine.

Para controlar a aparência e o comportamento da história durante a interação, o Twine tem três formatos de história disponíveis, sendo por omissão o Harlowe 3.1.0, cujo objetivo é facilitar a adição de interação básica às histórias de uma forma concisa e legível. É também possível escolher o formato Snowman, projetado para ser usado por pessoas familiarizadas com a escrita de JavaScript e CSS, e o formato SugarCube, que veio da primeira versão do Twine. Além destes três formatos, é também possível adicionar um formato personalizado.

2.4.2 Yarn Spinner

O Yarn Spinner é uma ferramenta *open source* gratuita que permite a criação de diálogos interativos, inspirada no Twine. Os diálogos são escritos em Yarn, uma linguagem de programação bastante simples, mas poderosa o suficiente para tornar o diálogo o mais completo possível. Tal como no Twine, é possível controlar o diálogo através de variáveis e condições.

Todos os diálogos são armazenados em arquivos `.yarn`, que podem ser criados através do Yarn Editor ou com qualquer outro editor de texto. Tal como no Twine, os ficheiros de diálogo contêm nós que estão interligados conforme o seu conteúdo. A Figura 2.3 mostra um pequeno exemplo de como se parecem os nós no Yarn Editor. O conteúdo base de

cada nó (fala do personagem virtual e opções de resposta) segue o mesmo formato que no Twine.



Figura 2.3: Grafo exemplificativo do Yarn Spinner.

Uma vantagem que o Yarn Spinner tem em relação ao Twine é que este fornece um conjunto de componentes e *scripts* que facilitam a integração com o Unity.

2.4.3 WOOL

O WOOL⁵ é uma plataforma de diálogo simples e poderosa, totalmente *open source*, para a criação de conversas com agentes virtuais. O WOOL é uma linguagem de especificação de diálogo acompanhado de um conjunto de ferramentas que permitem criar diálogos de forma rápida e dinâmica entre personagens e incorporá-los numa aplicação com facilidade.

A linguagem WOOL é baseada nas duas *frameworks* mencionadas anteriormente. Enquanto o Twine foi projetado para contar histórias interativas, e o Yarn tem as suas raízes no desenvolvimento de videogames, o WOOL foi desenvolvido no contexto de concepção de agentes conversacionais para promoção da saúde.

A definição de diálogo WOOL é essencialmente a definição de uma série de etapas de diálogo (nós) ligadas entre si através das respostas do utilizador. Tal como no Twine e no Yarn Spinner, no seu estado mais básico, a declaração pode ser uma simples linha de texto, no entanto, é possível adicionar variáveis e condições para uma maior personalização. Um exemplo de grafo de diálogo do WOOL está visível na Figura 2.4, sendo bastante semelhante aos grafos mostrados nas duas secções anteriores.

De momento, o WOOL está adaptado para a criação e execução de diálogos entre agentes virtuais e humanos apenas em aplicações Java ou JavaScript.

⁵<https://www.woolplatform.eu/>

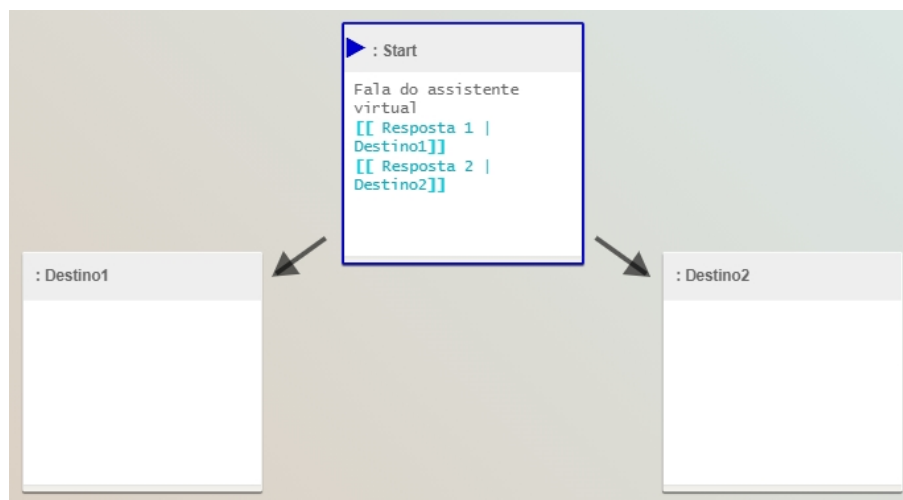


Figura 2.4: Grafo exemplificativo do WOOL.

2.4.4 Watson Assistant

O Watson Assistant⁶ é um serviço na IBM Cloud que permite construir interfaces de conversação com assistentes virtuais em qualquer aplicação e dispositivo. Apesar de não ser uma ferramenta gratuita, a IBM fornece um plano Lite do serviço, que permite fazer 10.000 chamadas à API (do inglês *Application Programming Interface*) por mês sem qualquer custo.

O Watson Assistant é uma plataforma completa, que possui um sistema de inteligência artificial e *machine learning* associado à função do assistente virtual, o que permite um treino para entender diferentes contextos. Este serviço é normalmente utilizado para a criação de *chatbots*, no entanto, é possível tirar partido do mesmo neste contexto.

Os três principais componentes do Watson Assistant são os seguintes:

- **Intents**: consideradas como um propósito ou objetivo;
- **Entities**: consideradas como complemento da informação;
- **Dialog**: onde se cria toda a árvore de decisão do assistente, de forma a criar o fluxo de diálogo.

O menu *Intents* é feito com o propósito de analisar a frase dos utilizadores com o objetivo de entender “Qual é o propósito dele ao fazer esta pergunta?”, sendo este o sítio onde se agrupam exemplos de frases a fim de se analisar uma frase, não pelas palavras chaves, mas pelo contexto.

No Watson Assistant, o diálogo define o fluxo da conversa na forma de uma árvore lógica, e usa a *intent* e a *entity* que forem identificadas para interagir com o utilizador e fornecer uma resposta, que pode ser a resposta a uma pergunta ou a execução de um

⁶<https://www.ibm.com/cloud/watson-assistant>

comando. Na Figura 2.5 encontra-se uma árvore lógica simples, que é gerada por omissão ao criar uma *skill* de diálogo (um *container* para todos os artefatos que definem um fluxo de conversa).

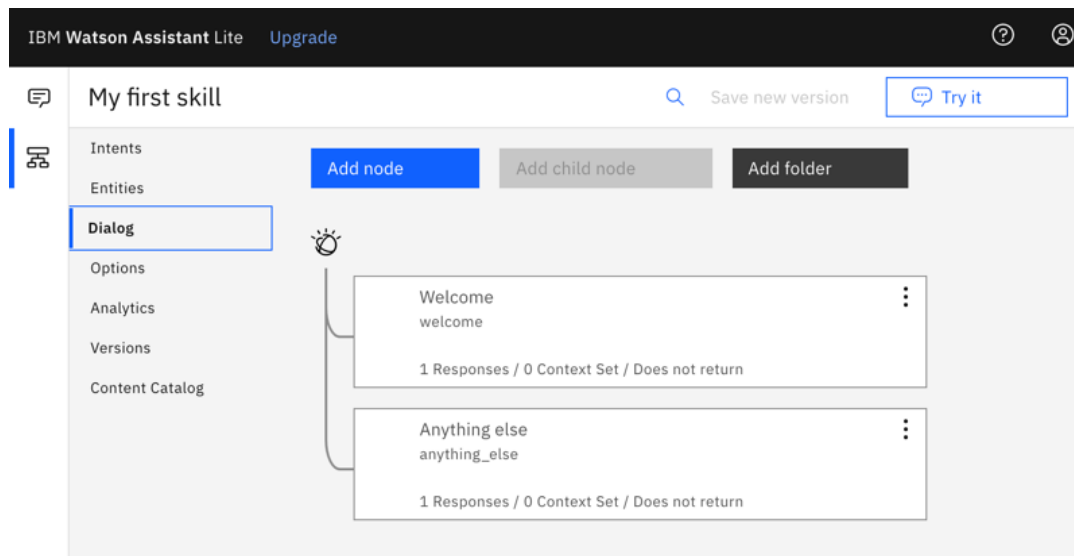


Figura 2.5: Árvore lógica que define o fluxo do diálogo, no menu *Dialog*.

Em cada nó de diálogo é possível introduzir várias respostas que o assistente irá dar de forma sequencial ou aleatória, ou seja, cada vez que o utilizador diz algo, o assistente pode responder de forma diferente, o que contribui para a existência de um diálogo mais natural.

À semelhança do Yarn Spinner, também o Watson Assistant facilita a integração com o Unity, graças ao kit de desenvolvimento de *software* (SDK) suportado pela IBM.

2.4.5 FATiMA Toolkit

FATiMA Toolkit⁷ é um conjunto de ferramentas projetadas para a criação de agentes com inteligência social e emocional, e que podem ser facilmente integrados com o Unity. A *toolkit* é composta por um conjunto de *runtime C# libraries*, a FATiMA Engine, e uma ferramenta de autoria integrada, a FATiMA Toolkit propriamente dita, que é uma aplicação que lida com todos os módulos visíveis na Figura 2.6 [Guimarães et al., 2019].

As bibliotecas C# podem ser importadas diretamente para o Unity, sem impor quaisquer restrições tecnológicas adicionais, e a ferramenta de autoria integrada fornece uma interface gráfica que não só facilita a projeção e criação de cenários e personagens dentro deles, como também permite que o cenário seja testado sem que haja a necessidade de uma implementação real do mesmo [Guimarães et al., 2019].

A *toolkit* permite que se adicionem diferentes níveis de inteligência aos personagens, como capacidades de tomada de decisão, respostas emocionais, avaliação emocional de

⁷<https://fatima-toolkit.eu/>

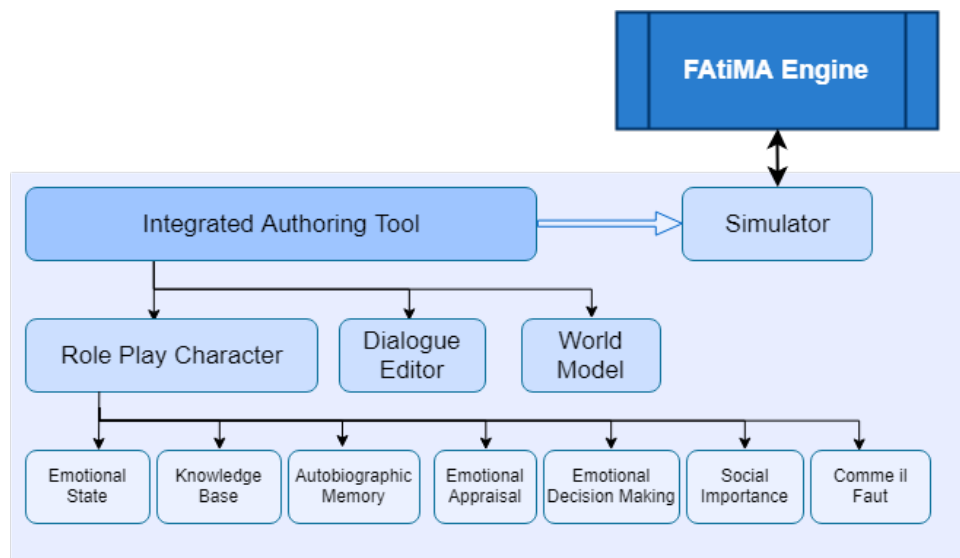


Figura 2.6: Componentes da FATiMA Toolkit [Guimarães et al., 2019].

eventos, memória autobiográfica e capacidades de diálogo. Tem integrado um modelo computacional de emoções e uma estrutura de diálogo explícita. A *toolkit* é centrada no personagem, o que significa que a narrativa emerge conforme o utilizador e os agentes interagem entre si de acordo com os seus objetivos narrativos [Guimarães et al., 2019].

No que diz respeito à gestão do diálogo, a abordagem da FATiMA é considerada pelos autores como sendo uma solução híbrida, não seguindo propriamente a solução dos grafos, usada por muitas das ferramentas mencionadas anteriormente. Cada enunciado é codificado como uma ação de diálogo com as seguintes propriedades: $\langle \text{CurrentState}, \text{NextState}, \text{Meaning}, \text{Style}, \text{UtteranceText} \rangle$. O *Current State* define o estado atual da conversa, o *Next State* define o estado para onde a conversa será movida, os campos *Meaning* e *Style* são usados para *tags* auxiliares, e o *Utterance Text* define a fala do personagem.

2.4.6 Dialogflow

O Dialogflow⁸ é uma plataforma de compreensão de linguagem natural, na qual é possível projetar e integrar uma interface de conversação com o utilizador em qualquer plataforma e dispositivo, como uma aplicação móvel ou uma aplicação *web*. Ao usar o Dialogflow, é possível fornecer novas formas mais envolventes para os utilizadores interagirem com um produto, uma vez que o Dialogflow consegue analisar vários tipos de *input* dos utilizadores, incluindo texto ou áudio, e também responder em ambas as formas (texto ou fala sintética).

O Dialogflow oferece dois serviços de agente virtual, cada um com o seu próprio tipo de agente, interface de utilizador, API, bibliotecas de cliente, e documentação. Um dos serviços, denominado Dialogflow ES, oferece um tipo de agente padrão, adequado para

⁸<https://cloud.google.com/dialogflow/>

agentes mais simples. O outro serviço, denominado Dialogflow CX, fornece um tipo de agente mais avançado, adequado para agentes maiores ou mais complexos.

Tal como o Watson Assistant, o Dialogflow contém *intents* e *entities*. Uma *intent* categoriza a intenção do utilizador final em cada conversa. Em comparação com as *intents* da versão ES, as *intents* do Dialogflow CX foram simplificadas de forma a torná-las um recurso reutilizável e independente da resposta do agente. Uma *entity*, ou entidade, define um tipo de informação que é possível reconhecer e extrair a partir do *input* do utilizador.

O Dialogflow ES, de forma semelhante ao Watson Assistant, define o fluxo da conversa na forma de uma árvore lógica baseada na identificação de *intents* (Figura 2.7). O Dialogflow CX oferece uma nova forma de projetar agentes, usando uma abordagem de máquina de estados (Figura 2.8). As figuras 2.7 e 2.8 mostram parte de um agente pré-criado no Dialogflow ES e CX, respetivamente, concebido para lidar com uma pequena conversa de circunstância com o utilizador (*small-talk*). Enquanto que na versão ES as *intents* contêm uma resposta agregada, definida aquando da sua criação, e enviada ao utilizador assim que o agente deteta a *intent*, na versão CX as *intents* são definidas apenas com o intuito de perceber a intenção do utilizador, estando a resposta do agente definida nos diversos caminhos (o que está definido como *Routes* na Figura 2.8) que a conversa pode seguir a partir de um determinado “estado”.

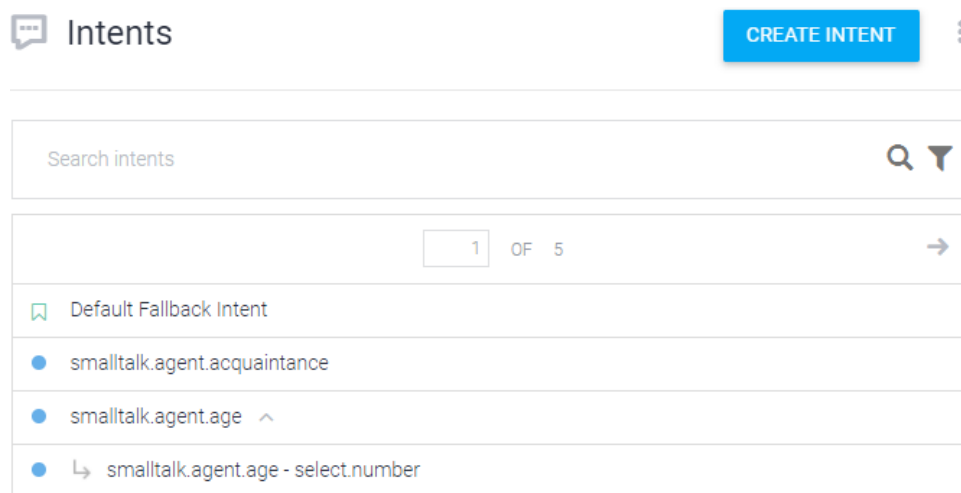


Figura 2.7: Árvore lógica que define o fluxo de conversa de um agente no Dialogflow ES.

Para além de se definirem respostas estáticas para o agente responder ao utilizador, também é possível usar a opção de *fulfillment* caso seja necessário devolver uma resposta mais dinâmica. Quando a opção de *fulfillment* está ativada numa determinada parte do fluxo de conversação, o Dialogflow envia uma solicitação para o serviço de *webhook* (serviço que hospeda a lógica de negócios — regras ou algoritmos personalizados que lidam com a troca de informações entre os dados do sistema e a interface do utilizador) associado ao agente, permitindo o uso dos dados extraídos por parte do sistema, quer seja para gerar respostas dinâmicas, validar dados recolhidos ou ativar ações no back-end.

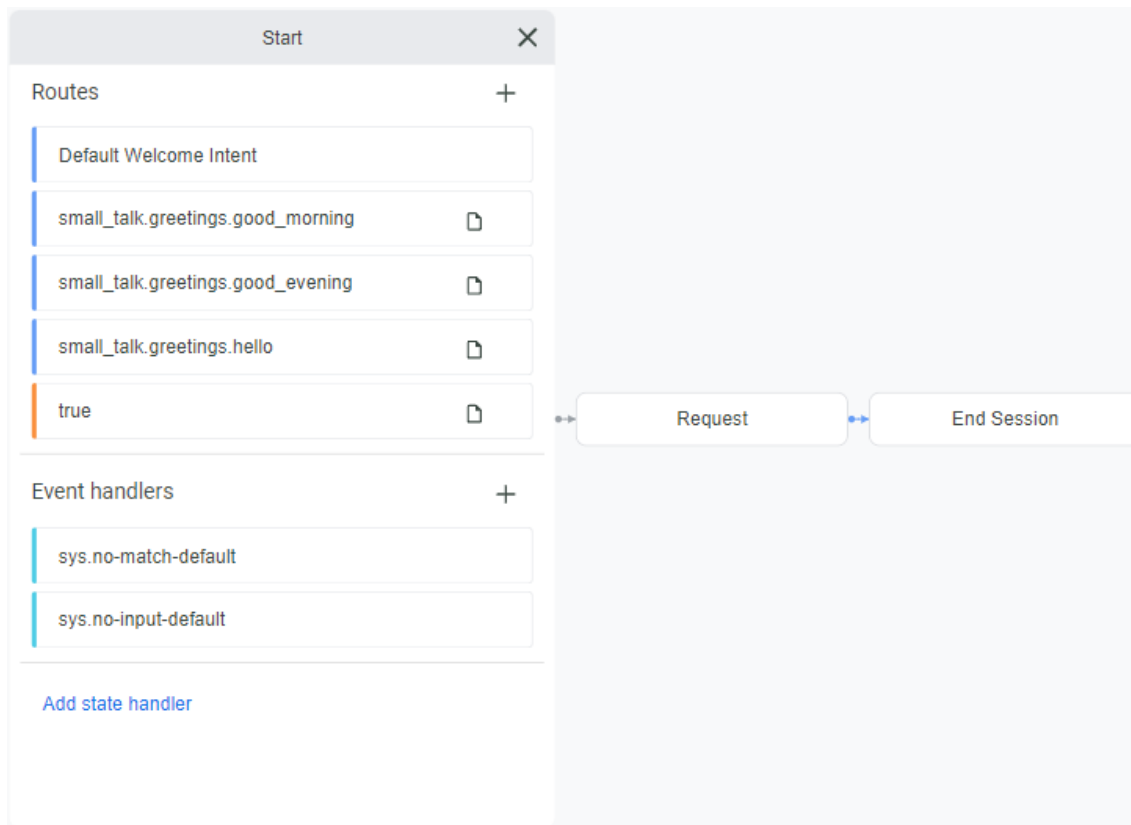


Figura 2.8: Fluxo de conversação de um agente no Dialogflow CX.

Para aceder à API do Dialogflow, este fornece bibliotecas de cliente e guias para as seguintes linguagens de programação: Java, Node.js e Python.

2.4.7 Conclusão

A Tabela 2.1 resume as características mais relevantes presentes nas ferramentas consideradas para este trabalho.

Características	Twine	Yarn Spinner	WOOL	Watson Assistant	FAtiMA Toolkit	Dialogflow
Open source	✓	✓	✓	✗	✓	✗
Gratuito	✓	✓	✓	✗	✓	✗
Controlo do fluxo de diálogo	✓	✓	✓	✓	✓	✓
Processamento de linguagem natural	✗	✗	✗	✓	✗	✓
Fácil integração com o Unity	✓	✓	✗	✓	✓	✗

Tabela 2.1: Características relevantes das ferramentas analisadas.

Tendo em conta que as duas características principais pretendidas eram a possibilidade de incorporar interação de linguagem natural e manter o controlo do fluxo do diálogo,

verifica-se que o Watson Assistant e o Dialogflow são os únicos que apresentam ambas as características. Apesar do Watson Assistant e do Dialogflow serem ferramentas típicas de criação de *chatbots* (em que o utilizador costuma colocar as perguntas e o agente se limita a responder), é possível implementar um agente que possua controlo na condução do diálogo, mesmo não sendo com a mesma facilidade que nas restantes ferramentas.

Com base nesta informação, na Secção 4.1 é apresentado o processo de escolha da ferramenta a utilizar neste trabalho.

2.5 Ontologias

Uma estratégia que tem vindo a ser utilizada em diversos campos de trabalho é a utilização de ontologias, graças ao seu potencial para suportar a acumulação de conhecimento de forma eficiente em “*knowledge bases*” (bases de conhecimento), que são bases de dados que armazenam dados que pertencem a classes de ontologia [Larsen et al., 2017].

No ramo da Ciência da Informação, ontologias são estruturas de informação que definem classes abstratas e as relações entre essas classes. Uma ontologia é a especificação dos conceitos, dos seus atributos e relacionamentos, dentro de um determinado domínio. As ontologias podem incluir diferentes níveis de abstração. Uma ontologia pode ser de mais alto nível e cobrir conceitos mais genéricos, ou de mais baixo nível, especificando mais o seu conteúdo [Larsen et al., 2017].

As ontologias baseiam-se na lógica das taxonomias, mas são mais flexíveis, uma vez que permitem mais do que um tipo de relação. As taxonomias definem unicamente relações pai-filho entre classes, enquanto que as ontologias permitem que cada classe tenha relações com mais do que uma classe, podendo ser não só relações hierárquicas de pai-filho como outros tipos de relações [Larsen et al., 2017]. As classes definidas podem ser caracterizadas com base na identificação de condições que os seus elementos têm que satisfazer relativamente aos valores das suas propriedades ou às relações em que participam.

Idealmente, as ontologias são codificadas num formato legível por computadores, de forma a que os computadores “entendam” a ligação entre as várias classes, bem como a linguagem usada para as definir. Um formato de ontologia bastante usado é o *Web Ontology Language* (OWL), e o sistema normalmente usado para codificar uma ontologia em OWL é o Protégé⁹, um editor de ontologias *open-source* gratuito [Larsen et al., 2017].

Um dos grandes interesses das ontologias é o facto de poderem ser usadas para realizar inferências lógicas para a dedução de novos factos, através de um *reasoner*. Outro aspeto interessante é o facto de se poderem ligar vários pedaços de conhecimento a partir de diferentes ontologias na Web Semântica [Lamy, 2017].

No contexto do *Human Behaviour-Change Project* [Michie et al., 2017a], já está a ser

⁹<https://protege.stanford.edu/>

desenvolvida uma ontologia relacionada com as intervenções de mudança comportamental, a *Behaviour Change Intervention Ontology* (BCIO) [Michie et al., 2021]. Embora a BCIO incida sobre os aspectos mais gerais que caracterizam as BCIs, esta ontologia possibilita uma ligação a este trabalho, através dos conceitos comuns de BCI e BCT. Assim, na sequência deste trabalho, no Capítulo 3, é apresentada uma proposta de ontologia que reúne diversos conceitos de mudança comportamental, com maior foco na representação da operacionalização de BCTs no contexto de uma intervenção de mudança comportamental.

Capítulo 3

Ontologia de conceitos de mudança comportamental

Neste capítulo é descrito o processo de desenvolvimento da ontologia de conceitos de mudança comportamental. O capítulo inicia-se com a análise e identificação dos conceitos relevantes ao tema da ontologia (Secção 3.1). Segue-se uma descrição da ontologia definida, das suas classes e relações (Secção 3.2), passando depois a apresentar e discutir alguns casos práticos de uso da ontologia (Secção 3.3). Por fim, é discutida a possibilidade de aplicar a ontologia definida noutros contextos, diferentes daquele que levou à sua definição (Secção 3.4).

3.1 Identificação dos conceitos

Uma das principais limitações do agente antecedente, chamado Vitória e desenvolvido no contexto do projecto VASelfCare, incide no facto da operacionalização das BCTs ser muito rígida. O fluxo de conversação é bastante dependente da forma como os diálogos correspondentes foram definidos *a priori*, em Yarn.

De forma a tornar os diálogos do agente mais naturais e flexíveis, optou-se por caracterizar a operacionalização das BCTs de uma forma mais geral. Para tal, foi necessário identificar os principais conceitos envolvidos e quais as relações entre eles.

O ponto de partida para a identificação dos conceitos teve por base o esquema da Figura 3.1, que representa a distribuição das várias BCTs de acordo com as etapas de diálogo da intervenção VASelfCare [Balsa et al., 2020]. Para cada etapa do diálogo, a(s) BCT(s) operacionalizadas estão identificadas pelo nome e pelo código numérico de acordo com a BCTTv1 [Michie et al., 2013]. Além do esquema representado na Figura 3.1, foi também tido em conta a arquitetura da intervenção durante as componentes de Alimentação e Atividade Física, que podem ser consultadas na parte inicial dos Apêndices A e B, respetivamente.

Para complementar a análise, foi também feito um levantamento dos diálogos do agente na aplicação VASelfCare, de forma a entender o que caracteriza cada BCT, aquilo

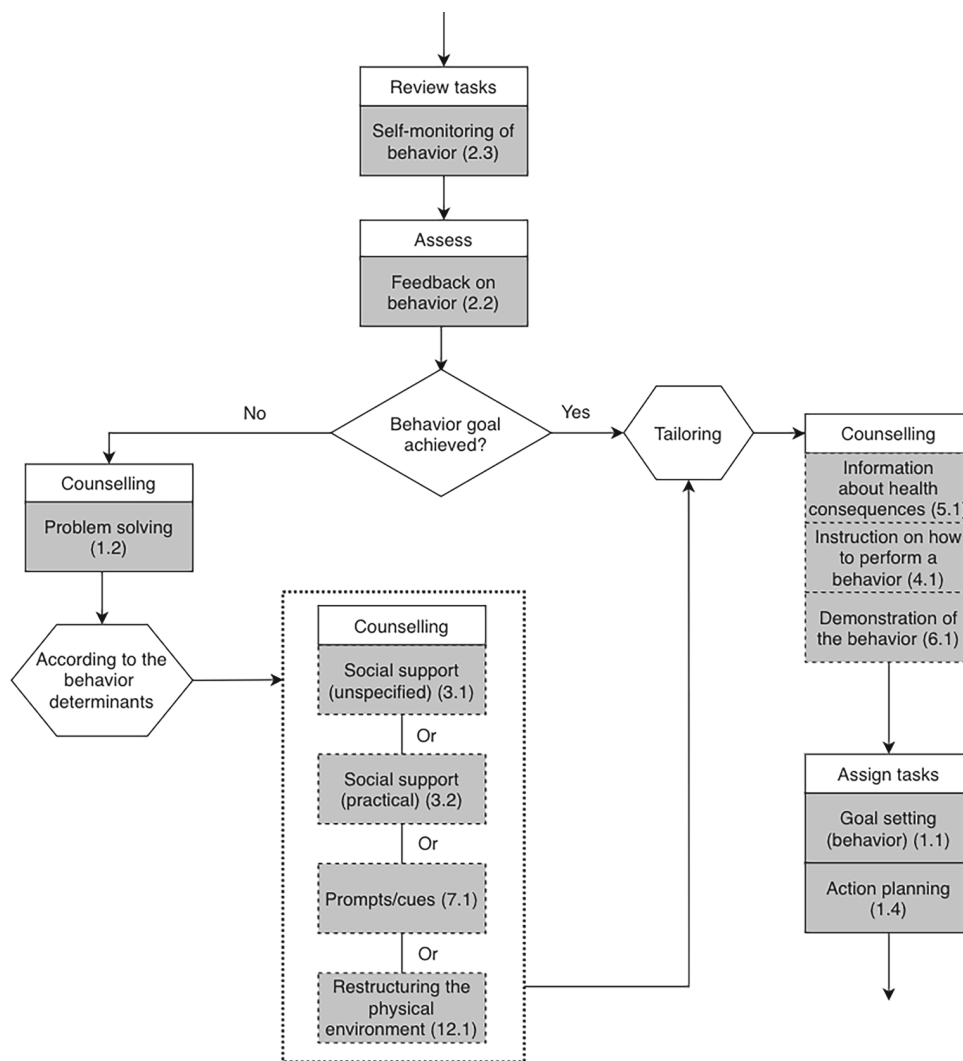


Figura 3.1: Distribuição das BCTs (identificadas pelo nome e código numérico da BCTTv1 [Michie et al., 2013]) de acordo com as etapas de diálogo numa interação (extraída de [Balsa et al., 2020]).

que as suas operacionalizações podem ter em comum e as diferentes formas de concretizar uma mesma BCT através de diálogo. Para isso, analisou-se um total de 61 ficheiros Yarn (50 ficheiros relacionados com o diálogo apresentado durante a componente de Alimentação e 11 ficheiros relacionados com a componente de Atividade Física), dos quais resultaram quatro tabelas. Por motivos de confidencialidade, as tabelas resultantes foram reduzidas de forma a poderem ser incluídas neste documento, nos Apêndices A e B. Uma das tabelas relacionadas com a componente de Alimentação foi totalmente removida, por conter bastante diálogo referente à etapa de *Counselling* (na qual o agente partilha grande parte da informação com o utilizador acerca dos vários tópicos abordados ao longo da intervenção). Todas as BCTs presentes na Figura 3.1 foram identificadas, com exceção da *Prompts/cues (7.1)*, que não é operacionalizada em nenhuma das componentes analisadas (apenas na componente de Medicação, não analisada para este trabalho).

A Tabela 1 do Apêndice A representa a caracterização dos elementos de intervenção e respectivas BCTs e determinantes da componente de Alimentação. Para cada um dos tópicos abordados durante a componente de Alimentação (oito no total, dois incluídos na versão reduzida e visíveis na primeira coluna da tabela), foram analisados os diálogos da Vitória em cada uma das etapas que envolvem BCTs (visíveis na segunda coluna), evidenciando a(s) BCT(s) concretizadas nesse elemento de diálogo (quarta e quinta colunas, respetivamente). Nas etapas que envolviam determinantes, dos quais as BCTs concretizadas dependem, estes foram também discriminados (terceira coluna da tabela).

A Tabela 1 do Apêndice B representa a caracterização dos elementos de intervenção e respectivas BCTs e a Tabela 2 contém a mesma caracterização mas de elementos associados a determinantes (ambas relacionadas à componente de Atividade Física). Na Tabela 1, para cada uma das etapas de diálogo (primeira coluna da tabela), foram analisados os diálogos da Vitória em cada uma delas, evidenciando a(s) BCT(s) concretizadas nesse elemento de diálogo (segunda e terceira colunas, respetivamente). Na Tabela 2, a lógica é a mesma, apenas foi adicionada uma coluna para evidenciar os determinantes dos quais dependem as BCTs concretizadas naquela etapa do diálogo.

A última coluna das tabelas de ambos os apêndices representa a condição da qual o elemento de diálogo depende, com base na arquitetura da intervenção durante as duas componentes analisadas (Figura 1 do Apêndice A e Figuras 1 e 2 do Apêndice B). As frases sublinhadas com a mesma cor são iguais ou bastante semelhantes, sendo que algumas delas se encontravam escritas mais do que uma vez nos ficheiros de diálogo analisados, motivando a sua otimização neste trabalho.

Analisando todas estas fontes de informação, tiraram-se as seguintes conclusões:

- Cada dia de interação é estruturado através de uma sequência geral de fases do diálogo, comum a todas as componentes da intervenção e definida de acordo com a literatura em agentes relacionais [Bickmore et al., 2005]. As etapas de diálogo são, ordenadamente: *Opening* (saudação/abertura da conversa), *Social talk* (diálogo social, conversa curta em que se pergunta sobre o estado emocional e físico do utente), *Review tasks* (rever os objetivos estabelecidos anteriormente e averiguar se foram cumpridos), *Assess* (averiguar se o utilizador cumpriu as tarefas atribuídas anteriormente), *Counselling* (falar sobre temas e assuntos e dar aconselhamento motivacional), *Assign tasks* (negociar novos objetivos e tarefas com o utilizador), *Pre-closing* (preparar para encerrar a conversa), e *Closing* (despedida).
- As BCTs são operacionalizadas nas etapas centrais do diálogo: *Review tasks*, *Assess*, *Counselling* e *Assign tasks*. Adicionalmente, percebeu-se que, em algumas dessas etapas - *Review tasks*, *Assess* e *Assign tasks* - as BCTs operacionalizadas são sempre as mesmas.
- Dependendo do dia da intervenção, as etapas *Review tasks* e *Assign tasks* lidam

com os objetivos comportamentais, controlando o seu cumprimento ou atribuindo um novo objetivo.

- Dependendo do cumprimento de um dado objetivo comportamental, a etapa de *Counselling* pode incluir a aplicação de alguma BCT adicional, de acordo com o determinante comportamental (um fator que influencia um comportamento de forma positiva ou negativa) identificado. Este facto é notório na Figura 3.1, quando deparado com o losango central “*Behavior goal achieved?*”, no caso da resposta ser negativa e o fluxo seguir pelo lado esquerdo da figura.
- Alguns determinantes de não adesão desencadeiam as mesmas BCTs, sendo que algumas vezes essas BCTs são concretizadas pelo mesmo diálogo. Um exemplo disso é visível na Tabela 1 do Apêndice A, onde os determinantes “Não gosto de preparar refeições só para mim.” e “Não gosto de fazer refeições sozinho(a).” (do tópico 1. Rotinas Alimentares) desencadeiam exatamente as mesmas BCTs e parte do mesmo diálogo.

Posto isto, ficou claro que existem certos conceitos que são centrais ao estruturamento de uma intervenção de mudança comportamental e à correta operacionalização de BCTs que não podem faltar na definição desta ontologia. Para além dos mais óbvios, como é o caso dos próprios conceitos de *Behavior Change Intervention* e *Behavior Change Technique*, é também importante definir os conceitos de determinante comportamental (*behavior determinant*) e objetivo comportamental (*behavior goal*), uma vez que algumas das BCTs a operacionalizar dependem destes conceitos. Outros conceitos relacionados com a arquitetura e componentes da intervenção foram também definidos, conforme pode ser visto na secção seguinte.

Além dos conceitos relacionados com a arquitetura e componentes da intervenção, a operacionalização de uma BCT depende também de informações específicas do utilizador. Assim sendo, para caracterizar uma forma específica de operacionalização de uma BCT foi necessário incorporar também conceitos relacionados com as características do utilizador.

3.2 Descrição da ontologia definida

A partir dos conceitos identificados e descritos na secção anterior, recorrendo ao editor Protégé, definiu-se uma ontologia de conceitos de mudança comportamental, que pretende reunir os conceitos relevantes à modelação de uma intervenção de mudança comportamental, focando-se na caracterização da operacionalização de BCTs.

A ontologia resultante pode ser dividida em duas classes principais: *Behavior Change Intervention* e *User*. A classe *Behavior Change Intervention* tem 12 subclasses: *BCI*

Component, Behavior Change Technique, Behavior Determinant, Behavior Goal, Behavior Topic, Food Topic, Dialogue Step, Operationalization, Basic, Complex, Grouped, e Sequential. A classe *User* tem 6 subclasses: *Adult, Risk Person, High-Risk Person, Very High-Risk Person, Senior, e Underage.* Estas entidades encontram-se descritas na Tabela 3.1, e a Figura 3.2 mostra uma visão geral da sua organização hierárquica.

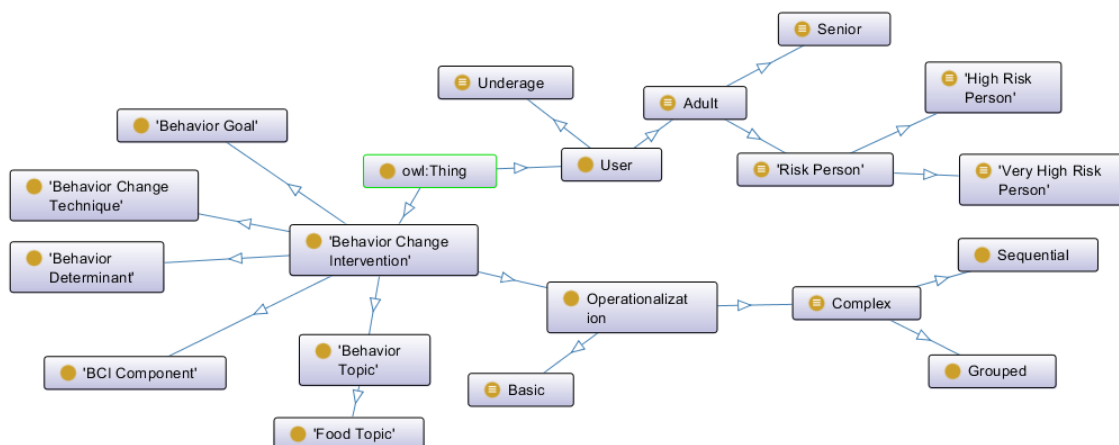


Figura 3.2: Visão geral das principais classes da ontologia.

Numa ontologia, a estrutura hierárquica de classes pode ser estendida através de relacionamentos mais abertos, chamados de *object properties* e *data properties*. As *object properties* fazem uma ligação entre indivíduos (instâncias de classes), enquanto que as *data properties* relacionam indivíduos a algum tipo de atributo (que pode ser um valor inteiro, decimal, *string*, entre outros). Algumas das classes mencionadas acima estão relacionadas através de 6 *object properties*: *has active goals*, *has active topic*, *related to*, *targeted during*, *triggered by*, e *triggers*. Para uma caracterização mais detalhada, existem também 9 *data properties*: *BCT order*, *has age*, *has BMI* (do inglês *Body Mass Index*, Índice de Massa Corporal), *has competence score*, *has genre*, *has height*, *has name*, *has risk level*, e *has weight*. Estas relações são descritas na Tabela 3.2, juntamente com os seus respetivos domínios e contra-domínios.

É ainda possível adicionar informação às várias entidades através de *annotation properties*, que são muito mais simples que as *data* e *object properties*. Um exemplo deste tipo de propriedade é a *rdfs:label*, que é uma propriedade pré-definida, através da qual é possível associar rótulos legíveis por humanos a IRIs (do inglês *Internationalized Resource Identifier*) da ontologia, e usar esses rótulos ao visualizar a ontologia. Assim, em vez de sermos confrontados com algo tão longo e técnico como é, por exemplo, o caso do IRI “http://www.semanticweb.org/asus/ontologies/2021/2/bcco#Behavior_Goal”, podemos facilitar a interpretação adicionando o *label* “Behavior Goal”.

Além de tirar partido da propriedade *rdfs:label*, criou-se também duas outras *annotation properties*: *taxonomyNumber*, e *username*. A propriedade *taxonomyNumber* foi definida para indicar o código numérico, segundo a BCTTv1, de cada BCT (Domínio:

Behavior Change Technique). A propriedade *username* foi definida para indicar o nome que cada utilizador (Domínio: *User*) definiu para se diferenciar dos restantes ao aceder à aplicação (detalhada mais à frente, nos capítulos 5 e 6).

Entidade (<i>label</i>)	Subclass Of	Descrição
Behavior Change Intervention	owl:Thing	Intervenção que tem como objetivo influenciar o comportamento humano.
BCI Components	Behavior Change Intervention	Os componentes nas quais a intervenção se foca (alimentação, atividade física, medicação, ...).
Behavior Change Technique	Behavior Change Intervention	Um componente observável, replicável e irreduzível de uma intervenção projetada para alterar ou redirecionar processos causais que regulam o comportamento - um “ingrediente ativo” numa intervenção [Michie et al., 2013].
Behavior Determinant	Behavior Change Intervention	Um fator que influencia um comportamento de forma positiva ou negativa. Por exemplo, um motivo de não adesão a um objetivo previamente acordado (falta de motivação, esquecimento, ...).
Behavior Goal	Behavior Change Intervention	Um objetivo definido em função do comportamento desejado.
Behavior Topic	Behavior Change Intervention	Um tópico abordado durante a intervenção, com o objetivo de alcançar/manter o comportamento desejado.
Food Topic	Behavior Topic	Um tópico comportamental que visa ajudar a alcançar/manter um comportamento desejado relacionado com a componente de alimentação.
Operationalization	Behavior Change Intervention	O ato de concretização de uma ou mais BCTs, com base em várias condições.
Basic	Operationalization	Uma operacionalização onde apenas uma BCT é concretizada. Equivalent To: Operationalization and (triggers exactly 1 'Behavior Change Technique')
Complex	Operationalization	Uma operacionalização onde duas ou mais BCTs são concretizadas. Equivalent To: Operationalization and (triggers min 2 'Behavior Change Technique')
Grouped	Complex	Uma operacionalização complexa onde as BCTs são codificadas juntas e concretizadas de uma só vez.
Sequential	Complex	Uma operacionalização complexa onde as BCTs são concretizadas sequencialmente.
User	owl:Thing	Uma pessoa que está a ser submetida à intervenção de mudança comportamental (BCI).
Adult	User	Um utilizador cuja idade é igual ou superior a 18 anos. Equivalent To: User and (hasAge some xsd:integer[>= 18])
Risk Person	Adult	Um adulto cujo nível de risco seja igual ou superior a 15. Equivalent To: Adult and (hasRiskLevel some xsd:integer[>= 15])
High Risk Person	Risk Person	Uma pessoa de risco cujo nível de risco esteja entre 15 e 20 (inclusive). Equivalent To: 'Risk Person' and (hasRiskLevel some xsd:integer[<= 20])
Very High-Risk Person	Risk Person	Uma pessoa de risco cujo nível de risco seja superior a 20. Equivalent To: 'Risk Person' and (hasRiskLevel some xsd:integer[> 20])
Senior	Adult	Um adulto cuja idade seja igual ou superior a 65 anos. Equivalent To: Adult and (hasAge some xsd:integer[> 65])
Underage	User	Um utilizador cuja idade seja inferior a 18 anos. Equivalent To: User and (hasAge some xsd:integer[< 18])

Tabela 3.1: Classes da ontologia, com identificação da respetiva superclasse e descrição.

Relação (<i>label</i>)	Domínio	Contra-Domínio	Descrição
has active goals (OP)	User	Behavior Goal	Relação entre um utilizador e os objetivos comportamentais que este tem ativos num dado momento da intervenção.
has active topic (OP)	User	Behavior Topic	Relação entre um utilizador e o tópico comportamental que está a ser abordado num dado momento da intervenção.
related to (OP)	Operationalization	Behavior Goal	Relação entre uma operacionalização e o objetivo comportamental ao qual está relacionada.
targeted during (OP)	Behavior Goal	Behavior Topic	Relação entre um objetivo comportamental e o tópico comportamental durante o qual é abordado.
triggered by (OP)	Operationalization	Behavior Determinant	Relação entre uma operacionalização e o determinante que a desencadeia.
triggers (OP)	Operationalization	Behavior Change Technique	Relação entre uma operacionalização e a(s) BCT(s) que esta desencadeia.
BCT order (DP)	Sequential	xsd:string	Relação entre uma operacionalização sequencial e a ordem em que as BCTs desencadeadas devem ser concretizadas.
has age (DP)	User	xsd:integer	Define a idade de um utilizador.
has BMI (DP)	User	xsd:decimal	Define o índice de massa corporal de um utilizador.
has competence score (DP)	User	xsd:integer	Define a pontuação de competência obtida por um utilizador na fase de avaliação.
has genre (DP)	User	{"F", "M"}	Define o género de um utilizador.
has height (DP)	User	xsd:decimal	Define a altura de um utilizador.
has name (DP)	User	xsd:string	Define o nome de um utilizador.
has risk level (DP)	User	xsd:integer	Define o nível de risco de um utilizador desenvolver DT2 nos próximos 10 anos.
has weight (DP)	User	xsd:decimal	Define o peso de um utilizador.

Tabela 3.2: Relações da ontologia (OP = *Object Property*, DP = *Data Property*), com identificação do seu domínio, contra-domínio e descrição.

3.3 Casos práticos exemplares

A ontologia definida pode ser usada para diversas finalidades no que diz respeito ao desenvolvimento de uma intervenção de mudança comportamental dinâmica, tornando mais fácil conectar os conceitos envolvidos e moldar a estrutura da intervenção.

Para uma melhor perspectiva, esta secção descreve dois cenários específicos nos quais se pode tirar partido do uso da ontologia durante a interação do agente com o utilizador. O primeiro cenário descreve como usar a ontologia para a operacionalização de BCTs, e o segundo cenário descreve como usar o poder de inferência da ontologia para classificar o utilizador.

3.3.1 Operacionalização de BCTs

Tal como mencionado, este primeiro cenário descreve uma forma de usar a ontologia para a operacionalização de BCTs. A Figura 3.3 mostra as classes e relações que são relevantes

para o cenário descrito.

Durante a etapa *Review tasks*, o agente pergunta ao utilizador se conseguiu cumprir ou não um objetivo previamente acordado. No caso do utilizador responder de forma negativa, o agente questiona-o sobre a razão de não cumprimento do objetivo. A razão dada pelo utilizador é um determinante de não adesão ao comportamento desejado, que pode ser algo tão simples como *falta de motivação*, ou *relutância em alterar os hábitos*. As instâncias da classe *Behavior Determinant* são palavras chave desses motivos. Com base nos diálogos analisados, e para efeitos de prova de conceito, as instâncias criadas até ao momento foram: *motivação*, *hábitos*, *dificuldade*, *apetite*, e *solidão*.

Após receber a resposta do utilizador (com o determinante de não adesão), o agente consulta a ontologia para verificar qual é a operacionalização relacionada com o objetivo ativo (instância da classe *Behavior Goal*) que é acionada por esse determinante. Ao identificar a operacionalização correta, o agente verifica quais as BCTs que devem ser concretizadas. No caso da operacionalização ser sequencial (instância da classe *Sequential*), o agente verifica também qual a ordem pela qual as BCTs devem ser concretizadas no diálogo.

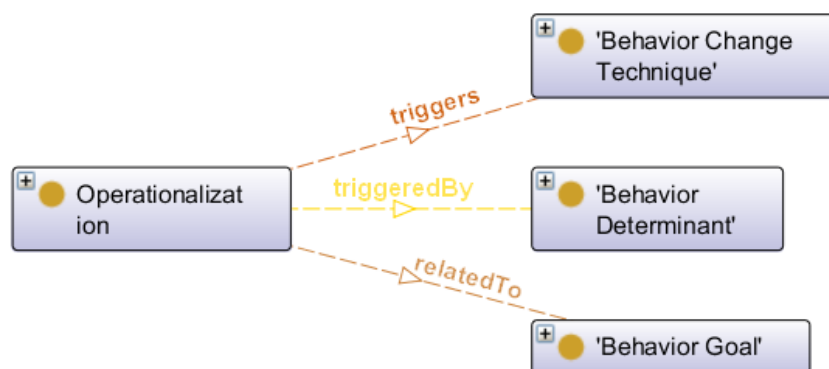


Figura 3.3: Entidades relevantes à operacionalização de BCTs.

3.3.2 Inferir conhecimento sobre o utilizador

A intervenção estruturada para a aplicação VASelfCare consiste em duas fases: uma fase de avaliação, e uma fase de *follow-up*. O principal objetivo da fase de avaliação é recolher dados sobre as características do utilizador relativamente às diferentes componentes da intervenção [Balsa et al., 2020]. Como objetivo para este trabalho, pretendia-se tornar a execução do diálogo menos rígida e mais personalizada, atuando conforme as características e avanços do utilizador, sem se prender a um número predefinido de dias de intervenção. Daí a importância de definir as características do utilizador na ontologia.

Numa perspectiva de trabalho futuro, pretende-se criar uma aplicação que atue na prevenção da Diabetes tipo 2, através de uma intervenção de mudança comportamental. Partiu-se assim do princípio que, de forma a ter uma intervenção mais personalizada ao

utilizador, esta deve atuar conforme o nível de risco associado à patologia sob a qual atua. Assim sendo, definiu-se a classe *Risk Person*, que tem como subclasses *High-Risk Person* e *Very High-Risk Person*.

Uma das vantagens do uso de ontologias encontra-se na possibilidade de fazer inferências a partir do conhecimento existente. Assim sendo, ao definir as classes mencionadas, tirou-se partido da opção *Equivalent To*. Isto permite adicionar expressões que declaram que um indivíduo, mesmo que não esteja declarado como instância daquela classe, pode ser inferido como tal.

Na Tabela 3.1 vemos que *Risk Person* é definido como sendo equivalente à expressão “Adult and (hasRiskLevel some xsd:integer[>=15])”. A entidade *High Risk Person* é equivalente a “Risk Person’ and (hasRiskLevel some xsd:integer[<=20])”. E a entidade *Very High-Risk Person* é equivalente a “Risk Person’ and (hasRiskLevel some xsd:integer[>20])”. Estes valores foram definidos com base num questionário fornecido pela DGS para o Programa Nacional para a Diabetes (Anexo C).

Na Figura 3.4 encontra-se um exemplo de resultados inferidos, sublinhados a amarelo. O indivíduo foi criado como sendo instância da classe *User*, e tem o valor 15 como nível de risco. Assim sendo, ao iniciar o *reasoner* (ferramenta que infere fatos que não se encontram expressos na ontologia), esse indivíduo passa também a ser classificado como *High Risk Person*, uma vez que se enquadra nas condições da expressão de equivalência.

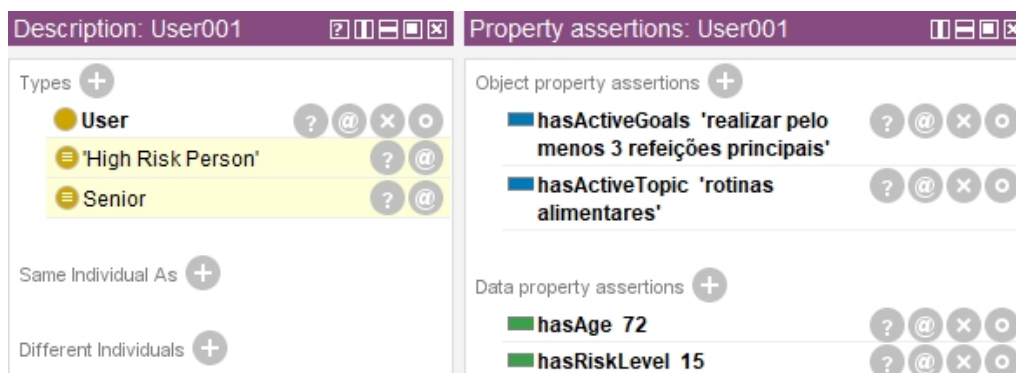


Figura 3.4: Inferência do tipo de um indivíduo.

3.4 Aplicações noutros contextos

Ao criar uma aplicação com o objetivo de influenciar o comportamento humano, o uso de uma ontologia pode ser útil para separar a lógica da aplicação dos outros dados mais específicos (os indivíduos e as suas relações).

O trabalho desenvolvido insere-se no contexto da saúde, partindo de um projeto que tem como foco a Diabetes tipo 2 [Félix et al., 2019, Balsa et al., 2020]. No entanto, apesar da modelação da ontologia ter sido feita com base na intervenção desenvolvida nesse projeto, o uso da ontologia definida não se limita a intervenções centradas nesta patologia.

Há certas entidades que são dependentes do domínio da Diabetes tipo 2, como é o caso da classe *Food Topic*, que especifica tópicos relacionados com a mudança na alimentação do utilizador. No entanto, a classe *Behavior Topic* (super-classe de *Food Topic*) é abrangente a qualquer tópico que se pretenda abordar durante uma intervenção de mudança comportamental.

A classe *Risk Person*, e consequentemente as suas sub-classes, também são dependentes do domínio pela forma como estão definidas. Como mencionado previamente, os valores definidos para os limites na classificação do utilizador tendo por base o nível de risco são baseados num questionário fornecido pela DGS para o Programa Nacional para a Diabetes (Anexo C), pelo que, são apenas válidos quando o risco em causa é desenvolver Diabetes tipo 2 nos próximos 10 anos. Para outras patologias, é possível que também existam ferramentas científicas que usem dados quantitativos sobre os mais variados fatores de risco para a saúde (como idade, peso, níveis de colesterol e glicose, histórico familiar de doenças, etc.) para calcular o risco de uma pessoa adquirir certas doenças ou condições, como por exemplo o QRISK3¹, uma ferramenta que calcula o risco de uma pessoa vir a ter um ataque cardíaco ou um acidente vascular cerebral (AVC) nos próximos 10 anos [Hippisley-Cox et al., 2017]. Posto isto, contornar esta limitação e alargar a ontologia para incluir outros valores de nível de risco e/ou patologias é totalmente viável, sendo inclusive uma das expectativas de trabalho futuro.

As restantes classes são completamente independentes do domínio, e podem servir de auxílio na estruturação de qualquer intervenção de mudança comportamental. No entanto, ao usar esta ontologia para desenvolver aplicações que incorporem BCIs, não é estritamente necessário que se utilizem todas as classes e propriedades aqui definidas. Da mesma forma, se as classes e/ou as propriedades definidas não forem suficientes, a ontologia pode ser importada e completada como for mais conveniente.

3.5 Conclusão

Ao longo deste capítulo foram descritos os vários componentes do processo que levou à definição de uma ontologia de conceitos de mudança comportamental.

Foram identificados os conceitos a incluir, e de seguida, descreveu-se a ontologia resultante desses conceitos. Para complementar, descreveram-se dois casos práticos de uso da ontologia definida. Por fim, discutiu-se a possibilidade de aplicar esta ontologia noutros contextos, mencionando algumas das limitações existentes nesse aspeto.

No próximo capítulo é descrito o processo de escolha da ferramenta de criação e gestão de diálogo que foi usada neste trabalho, sendo depois descrito em profundidade quais os seus componentes e como foram definidos em função do contexto do trabalho.

¹<https://www.qrisk.org/three/>

Capítulo 4

Desenvolvimento do motor de diálogo

Neste capítulo descreve-se o processo de escolha da ferramenta de criação e gestão de diálogo que foi usada neste trabalho (Secção 4.1), assim como o processo de criação e a descrição do seu uso (Secção 4.2). É também detalhada a forma como o agente lida com as situações em que deve assumir o controlo do diálogo (Secção 4.3), e por fim, o processo de treino do agente (Secção 4.4).

4.1 Escolha da ferramenta a utilizar

Após recolher informações sobre as ferramentas mencionadas na Secção 2.4, foram realizadas experiências com o Watson Assistant e com o Dialogflow, de forma a ver se era possível usar este tipo de tecnologia num ambiente mais complexo que os *chatbots* básicos onde são tipicamente utilizados.

Tanto no Watson Assistant como no Dialogflow, é possível definir mais do que uma variante da mesma resposta, sendo o próprio sistema a escolher aleatoriamente uma das opções para enviar ao utilizador. Esta característica dota o agente de aspetos sociais importantes para manter uma interação natural prolongada com o utilizador (cuja importância foi evidenciada na Secção 2.3), sem que corra o risco de o cansar com a mesma resposta ao longo do tempo.

Embora o Watson Assistant também apresente características bastante sólidas e interessantes ao contexto do trabalho, a opção de *fulfillment* do Dialogflow provou ser mais adequada à aplicação a desenvolver, aliado com a maior facilidade em construir um agente e controlar o fluxo da conversa.

Um dos principais objetivos deste trabalho era dinamizar a execução do diálogo, aceitando respostas escritas do utilizador em tempo real. O Dialogflow, sendo uma plataforma de compreensão de linguagem natural, provou ser bastante útil nesse aspecto, tornando mais fácil construir e entregar as falas do agente e capturar as do utilizador. No VA-SelfCare, as partes do diálogo correspondentes à operacionalização das BCTs foram definidas em Yarn [Balsa et al., 2020], obrigando a que as respostas do utilizador fossem

pré-definidas. Usar esse tipo de ferramenta para a criação dos diálogos, onde as falas de ambas as partes têm de ser definidas previamente, acaba por ser uma tarefa demorada, além de dificultar a lógica da interação quando comparada com o Dialogflow. Com esta ferramenta, o único diálogo da parte do utilizador que é necessário fornecer são as frases de treino para cada *intent*, depois disso o próprio agente aprende com o tempo e com o uso.

Uma vez que o agente deve seguir oito etapas de diálogo sequenciais, e conseguir manter o controlo do fluxo de conversação, a abordagem de máquina de estados fornecida pelo Dialogflow CX pareceu adequar-se melhor às necessidades deste trabalho. Assim, a ferramenta escolhida para criar o motor de diálogo foi o Dialogflow CX.

4.2 Definição do agente no Dialogflow

O Dialogflow CX tem um construtor visual na sua consola, onde os fluxos de conversação são representados graficamente seguindo um modelo de máquina de estados, facilitando assim o seu *design*, aprimoramento e manutenção. Os estados de conversação e as transições de estado são tipos de primeira classe que fornecem controlo explícito sobre os caminhos de conversação, podendo definir claramente uma série de etapas que desejamos que o utilizador percorra. Aqui, os estados de conversação são representados por *páginas* e as transições são representadas por *state handlers*.

Cada página pode ser configurada para recolher informações do utilizador que sejam relevantes para o estado de conversação representado pela página, que são chamados de *parâmetros*. A cada página podem também ser adicionados *state handlers*, que são usados para controlar a conversa, definindo respostas para enviar aos utilizadores e/ou fazendo a transição da página atual para outra página. Existem dois tipos de *state handlers*: rotas e *event handlers*.

As rotas são chamadas quando um *input* do utilizador corresponde a uma *intent* e/ou a alguma condição. Como foi enunciado na Secção 2.4.6, uma *intent* categoriza a intenção do utilizador em cada intervenção durante uma conversa. Os *event handlers* são chamados quando um evento é invocado. Alguns eventos pré-definidos são acionados quando é recebido um *input* inesperado do utilizador ou quando ocorre um erro do serviço de *webhook*. É também possível definir eventos personalizados que são invocados quando algo acontece fora da conversa.

Com isto, foram cobertos todos os termos importantes à compreensão do funcionamento do agente desenvolvido no Dialogflow CX, sendo de seguida descrito o agente definido para este trabalho.

Como especificado previamente, seguindo o trabalho desenvolvido no âmbito do projeto VASelfCare [Félix et al., 2019, Balsa et al., 2020] e, com base no trabalho do grupo de agentes relacionais de [Bickmore et al., 2005], o diálogo do agente conversacional é

organizado em oito etapas: *Opening*, *Social talk*, *Review tasks*, *Assess*, *Counseling*, *Assign tasks*, *Pre-closing* e *Closing*. Partindo desse princípio, a máquina de estados visual criada para o agente encontra-se representada na Figura 4.1.



Figura 4.1: Fluxo de conversação criado para o agente.

Definiram-se dez páginas diferentes para o agente. Duas destas páginas são definidas automaticamente pelo Dialogflow assim que o agente é criado: a página *Start* e a página *End Session*, que definem, respetivamente, o ponto de partida e o ponto de término de uma sessão, que representa uma conversa entre um agente e um utilizador. As restantes oito páginas foram definidas conforme necessário, de forma a cobrir os tópicos essenciais a cada etapa. Como se pode ver na Figura 4.1, nem todas as páginas correspondem exatamente a uma etapa de diálogo, como é o caso das etapas *Review tasks* e *Assess*, e das etapas *Pre-closing* e *Closing*. Da mesma forma, algumas etapas foram divididas em mais do que uma página, como é o caso das etapas *Social talk* e *Counseling*. O Dialogflow não permite alterar o nome da página *Start*, pelo que, apesar de não estar perceptível, esta diz respeito à etapa *Opening*.

Para etapas de diálogo mais simples, como *Opening*, *Social talk*, *Assess*, *Pre-closing* e *Closing*, definiram-se as respectivas páginas para enviar respostas de texto estáticas ao utilizador, uma vez que não existem parâmetros ou condições externas das quais dependam (relacionadas com o estado do utilizador, como o tópico ativo ou o cumprimento de objetivos comportamentais, entre outros). Para etapas de diálogo mais complexas, como *Review tasks*, *Counselling* e *Assign tasks*, definiram-se as respectivas páginas para habilitarem a solicitação do serviço *webhook*, a fim de fornecer uma resposta dinâmica com base em condições externas.

A Figura 4.2 mostra as rotas definidas na página *Counselling*. As três primeiras rotas (com a margem esquerda azul) são rotas de *intent*, o que significa que qualquer uma dessas rotas é chamada quando o *input* do utilizador corresponde a uma dessas *intents*. As duas últimas rotas (com a margem esquerda laranja) são rotas de condição, que são chamadas se a respectiva condição for satisfeita. No final de uma rota, existe um campo *Transition*, que define a próxima página ativa na conversa. No caso das três primeiras rotas visíveis na imagem, a transição é direcionada para a mesma página de origem. Isto acontece porque a etapa requer mais do que uma intervenção por parte do utilizador e do agente, assim como respostas dinâmicas, tirando partido da opção de *fulfillment* através do serviço de *webhook* (indicado na Figura 4.2 pelos símbolos na coluna *Fulfillment*).

Intent	Condition	Fulfillment	Transition
☰ affirmative-answer		☰	📄 Counselling
☰ negative-answer		☰	📄 Counselling
☰ counselling		☰	📄 Counselling
☰	\$session.params.counselling = "FINISH" AND \$session.params.assignTasks = True		📄 Assign Tasks
☰	\$session.params.counselling = "FINISH" AND \$session.params.assignTasks = False		📄 Pre-closing + Closing

Figura 4.2: Página *Counselling*.

A página *Get Determinant* tem como função recolher informação sobre o determinante de não adesão do utilizador a um objetivo previamente definido durante a intervenção de mudança comportamental. Para tal, tirou-se partido da opção de configuração de parâmetros na página, e do uso de *entities*.

Uma *entity*, ou entidade, define um tipo de informação que é possível reconhecer e

extrair a partir do *input* do utilizador. Desse modo, o tipo de informação que se pretende reconhecer trata-se dos possíveis determinantes de não adesão, pelo que se definiu um tipo de entidade denominada *determinant*. Posto isto, é necessário definir exemplos das entidades que queremos ver reconhecidas pelo agente. Neste caso, adicionaram-se como entidades as instâncias já definidas na ontologia: *motivação*, *hábitos*, *dificuldade*, *apetite* e *solidão* (enunciadas na Secção 3.3.1). Ao adicionar uma entidade, é necessário providenciar exemplos de frases ou expressões nas quais o agente possa reconhecer a entidade em causa. Na Figura 4.3 encontra-se o exemplo da entidade *motivação*, na qual podemos ver os sinónimos adicionados para identificação da mesma.

Entity	Synonyms
motivação	Não tenho motivação ×
	Não tenho motivação para cozinhar ×
	Não me tem apetecido ×
	motivação × Add synonyms

Figura 4.3: Definição da entidade *motivação*.

Assim, na configuração do parâmetro a recolher na página *Get Determinant*, indicou-se a entidade *determinant*. Após *input* do utilizador, o agente irá tentar identificar alguma das entidades especificadas dentro desse tipo. No caso do utilizador indicar outro motivo ainda não definido, impossibilitando o agente de identificar alguma das entidades definidas, foi adicionado um *event handler* para lidar com a situação e não deixar o utilizador sem resposta, enviando uma resposta *default* e fazendo a transição para a página seguinte.

Noutras ferramentas, e na versão ES do Dialogflow, uma *intent* geralmente contém um campo com as frases de treino e um campo com a resposta do agente, que é enviada de volta ao utilizador se o seu *input* corresponder às frases de treino definidas para essa *intent*. Durante uma interação neste contexto de intervenção, é expectável que o utilizador possa dizer “sim” ou “não” várias vezes, o que significa que, com esse tipo de *intent*, todas as vezes que o utilizador pudesse vir a dizer “sim” ou “não” durante a conversa, teria que haver uma *intent* com as mesmas frases de treino, mas com respostas diferentes, dependendo do contexto.

No Dialogflow CX, uma *intent* contém apenas a lógica para detectar o que o utilizador diz (as frases de treino). As *intents* são seleccionadas na definição das rotas, o que significa que uma *intent* pode ser reutilizada em vários momentos da conversa, não sendo necessário definir várias *intents* semelhantes. Assim sendo, para este agente foram definidas dez *intents*: “DEFAULT_WELCOME_INTENT”, “opening”, “social-talk-good”, “social-talk-bad”, “social-talk-question”, “affirmative-answer”, “negative-answer”, “get-

determinant”, “counselling”, e “closing”. Algumas das *intents* definidas são exclusivas para determinadas páginas (cujo nome deriva da etapa/página à qual estão associada), enquanto outras são repetidas em diversas páginas.

Na Figura 4.4 podem ser vistas algumas das frases de treino definidas para a *intent* “affirmative-answer”, que, tal como o nome indica, serve para detetar uma resposta afirmativa por parte do utilizador. Na Figura 4.2, está visível na primeira rota definida que a *intent* selecionada foi precisamente a “affirmative-answer”. No entanto, esta *intent* é também selecionada noutras rotas de outras páginas, como por exemplo numa rota na página *Review Tasks + Assess*, onde o utilizador é questionado sobre o cumprimento de um objetivo previamente definido, e cuja resposta pode claramente ser afirmativa.

Training phrases
exatamente
com certeza
consegui
certo
pode ser
claro
sim

Figura 4.4: Frases de treino da *intent* “affirmative-answer”.

4.3 Condução do fluxo de conversação

Ferramentas como o Dialogflow são geralmente usadas em agentes virtuais destinados a lidar com perguntas do utilizador, os típicos *chatbots*. No contexto deste trabalho, é importante que seja o agente a assumir o controlo da conversa, fazendo as perguntas e entregando a informação num formato que se assemelhe a uma consulta médica, onde o médico (agente conversacional) conduz a interação com o paciente (utilizador).

Na aplicação VASelfCare, após a fala do agente, a interface tinha botões para o utilizador escolher a resposta que mais se assemelhasse à sua, e, durante intervenções mais longas ou numa mudança na etapa do diálogo, era apresentado um botão “Continuar” [Balsa et al., 2020]. Quando o utilizador possui controlo total sobre o que responde ao agente, não podemos esperar que responda casualmente com um “Continuar”. O agente

tem de ser capaz de capturar a atenção e o interesse do utilizador, mantendo-o envolvido na conversa.

No Dialogflow CX, todas as respostas do agente são tratadas na página, e uma página pode ter um *fulfillment* de entrada, e uma resposta estática para cada rota. O *fulfillment* de entrada é opcional e é o que o agente responderá ao utilizador quando uma página se torna inicialmente ativa. Para cada rota adicionada a uma página, existe um campo de *fulfillment*, onde é possível adicionar vários tipos de mensagens de resposta, embora, neste trabalho, tenham sido utilizadas apenas mensagens de resposta de texto.

O recurso de *fulfillment* de entrada é uma grande vantagem, pois permite que o agente envie algo ao utilizador sem a necessidade de um *input* anterior. Isto proporciona ao agente maior iniciativa e controlo sobre a conversa, sendo extremamente útil na transição entre etapas do diálogo.

4.4 Processo de treino

Quando um agente é treinado, o Dialogflow usa os dados de treino para criar modelos de *machine learning* específicos para esse agente. Cada fluxo do agente tem o seu próprio modelo. Os dados de treino consistem principalmente em *intents*, frases de treino fornecidas em cada *intent*, e entidades referidas num fluxo, que são efetivamente usadas como rótulos de dados de *machine learning*. No entanto, os modelos do agente são construídos usando as respostas de *prompt* de parâmetros, *state handlers*, configurações do agente, e muitos outros dados associados ao agente.

Nas configurações do agente, é possível escolher treinar o agente automaticamente ou manualmente. Por omissão, o treino é executado automaticamente, mostrando uma notificação *popup* na consola sempre que é feita uma atualização ao fluxo.

4.5 Conclusão

Neste capítulo foi descrito o processo de escolha da ferramenta de criação e gestão de diálogo que foi usada neste trabalho, sendo também descrito em profundidade quais os seus componentes e como foram definidos em função do contexto do trabalho. Foram também aprofundados outros detalhes que complementam o tópico principal do capítulo. Neste caso, detalhou-se a forma como o agente lida com as situações em que deve assumir o controlo do diálogo, e por fim, o processo de treino do agente.

No capítulo que se segue encontra-se descrita a implementação da solução desenvolvida, que faz a integração entre a ontologia descrita no capítulo anterior, e o agente conversacional descrito no presente capítulo. É descrita a arquitetura do sistema implementado e as suas componentes, evidenciando a comunicação entre elas.

Capítulo 5

Desenvolvimento da solução

No presente capítulo é descrita a implementação da solução desenvolvida em função deste trabalho. Primeiramente é feito um levantamento das ferramentas utilizadas na concretização da solução (Secção 5.1). De seguida, é descrita a arquitetura do sistema implementado e as suas componentes (Secção 5.2), sendo depois destacada a forma como é feita a comunicação entre as várias componentes do sistema, ilustrando as partes mais importantes através de amostras do código desenvolvido (Secções 5.3 e 5.4).

5.1 Ferramentas utilizadas

A solução que se decidiu implementar foi uma aplicação *web* simples, que evidencie a capacidade do agente conversacional em manter uma conversa com o utilizador ao mesmo tempo que tira partido do módulo de conhecimento na operacionalização de BCTs oferecido pela ontologia.

Para a concretização desta solução foi necessário adquirir e consolidar competências em diversas ferramentas existentes, estando estas descritas nos seguintes parágrafos.

Python Para aceder facilmente à API do Dialogflow e integrar o agente criado neste sistema, o Dialogflow fornece bibliotecas de cliente e guias para as seguintes linguagens de programação: Java, Node.js e Python, tendo esta última sido a utilizada neste trabalho.

Flask O Flask¹ é uma *microframework* que utiliza a linguagem Python para criar aplicações *web*. O seu *design* foi pensado de forma a tornar os primeiros passos rápidos e fáceis, mas com a capacidade de escalar para aplicações mais complexas.

OWLready Muitos métodos e ferramentas têm sido propostos para o *design*, edição, manutenção, alinhamento ou avaliação de ontologias, incluindo o editor Protégé e o *reasoner* HermiT, porém, poucas opções estão disponíveis para a interface de programação

¹<https://palletsprojects.com/p/flask/>

de ontologias (ferramentas de acesso e modificação de uma ontologia dentro de uma linguagem de programação) [Lamy, 2017]. Neste trabalho tirou-se partido do OWLready, um módulo de programação orientado a ontologias, que possibilita o acesso e edição da ontologia através da linguagem Python.

HTML HTML é a abreviatura da expressão inglesa *HyperText Markup Language*, que é uma linguagem de marcação utilizada para estruturar e dar significado ao conteúdo das páginas *web* (por exemplo, definindo parágrafos, cabeçalhos, tabelas, ou inserindo imagens e vídeos na página).

CSS Abreviatura de *Cascading Style Sheets*, o CSS é uma linguagem de regras de estilo que é usada para aplicar estilo ao conteúdo HTML (para definir cores de fundo e fontes, adicionar margens ao conteúdo, entre outros).

JavaScript O JavaScript (JS) é uma linguagem de programação leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como sendo a linguagem de *script* para as páginas *web*, ou seja, é a linguagem que permite que uma página *web* faça algo mais do que simplesmente mostrar informação estática (criar conteúdo que se atualiza dinamicamente, controlar imagens animadas, entre outros). O JS é a terceira camada do conjunto de tecnologias *standard* da *web*, dando seguimento ao HTML e ao CSS.

5.2 Arquitetura do agente

Previamente, foi mostrado um diagrama simplificado do sistema que se pretendia implementar (Figura 1.1), no qual se evidenciava a integração entre o módulo de conhecimento e o gestor de diálogo. De forma a privilegiar a interação em tempo real, e simplificar os testes realizados durante a implementação do agente, optou-se por implementar uma aplicação *web* simples, focada apenas na troca de mensagens entre o agente e o utilizador.

A arquitetura do agente compreende três componentes principais: o *Integration Core*, a *Dialogflow Engine* e o *BCT Knowledge* (Figura 5.1). O *Integration Core* controla a *user interface* (UI) e, juntamente com a *Dialogflow Engine*, controla o fluxo de execução e da fala do agente. O módulo *BCT Knowledge* contém dados externos com conhecimento acerca da operacionalização das BCTs (ontologia) e de diálogos adicionais necessários a algumas operacionalizações, que podem ser consultados pelo *Core* sempre que for necessário.

De modo genérico, quando um utilizador envia uma mensagem através da *user interface*, ela é enviada ao *Dialogflow* para tentar detetar a *intent*. O *Dialogflow* processa o texto e envia de volta uma resposta (estática ou dinâmica, por meio do serviço de

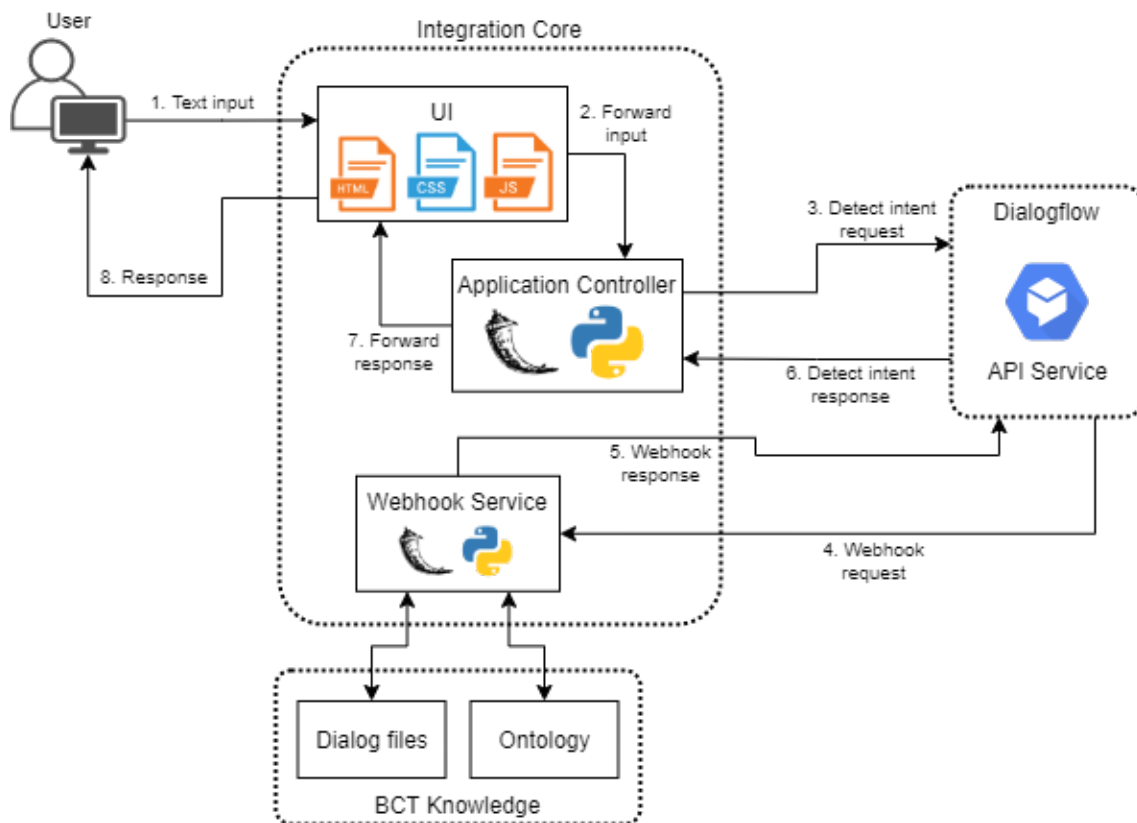


Figura 5.1: Arquitetura do sistema.

webhook). Os números indicados na Figura 5.1 correspondem às etapas que ocorrem a cada troca de mensagem numa sessão:

1. O utilizador digita uma mensagem (em formato de texto), que é conhecida como *user input*.
2. A UI, responsável pela visualização fornecida ao utilizador, recebe o *user input* e encaminha-o para a API do Dialogflow através de um pedido de detecção de *intent* (método gerido pelo *Application Controller* que faz a comunicação entre o *script Python* e o Dialogflow).
3. A API do Dialogflow recebe o pedido de detecção de *intent*. O Dialogflow quando recebe o *user input* tenta fazer a correspondência entre a mensagem e uma *intent* ou parâmetro, define parâmetros conforme necessário e atualiza o estado da sessão. Caso o *webhook* esteja ativado e seja necessário completar informação dinamicamente, ele envia uma solicitação para o serviço de *webhook*; caso contrário, vai diretamente para a etapa 6.
4. O serviço de *webhook* recebe a solicitação e executa todas as ações necessárias para obter uma resposta apropriada, como consultar a ontologia e/ou extrair o diálogo de fontes externas (ficheiros JSON).

5. O serviço de *webhook* cria uma resposta e envia-a de volta ao Dialogflow.
6. O Dialogflow cria uma resposta de detecção de *intent*. Se um *webhook* foi chamado, ele usa a resposta fornecida na resposta do *webhook*. Se nenhum *webhook* foi chamado, ele usa a resposta estática definida no agente do Dialogflow. A resposta de detecção de *intent* é enviada para a UI.
7. A UI recebe a resposta de detecção de *intent* e encaminha-a ao utilizador.
8. O utilizador vê a resposta do agente.

Nas secções que se seguem é descrito com maior detalhe as várias componentes do agente, a forma como estão implementadas e como são feitas as ligações entre elas. No que diz respeito à componente do Dialogflow, a sua descrição foi coberta na Secção 4.2, onde é descrito o processo de definição do agente, aqui apenas será mencionado como é feita a comunicação com o mesmo através da API.

5.3 *Integration Core*

Como mencionado anteriormente, o *Integration Core* controla a interface da aplicação e, juntamente com a *Dialogflow Engine*, controla o fluxo de execução e da fala do agente. Fazem parte desta componente o *Application Controller*, o *Webhook Service* e a *User Interface*.

5.3.1 *Application Controller*

Para estabelecer a ligação entre o *Application Controller* e o Dialogflow, é necessário apresentar as credenciais relacionadas com o acesso à API e ao agente do Dialogflow. Na Figura 5.2 encontram-se as variáveis necessárias para ter acesso ao agente, nomeadamente qual o ID do projeto a que está associado (criado na Google Cloud Platform²), qual a sua localização, e o seu ID. Não se encontra visível na figura, mas é também necessário definir a variável de ambiente “GOOGLE_APPLICATION_CREDENTIALS”, que se obtém através da Google Cloud Platform. Para o ID da sessão, é gerado um ID aleatório único, que é usado durante uma interação completa (desde a página *Start* até à *End Session*).

Tendo a autenticação feita, para fazer a comunicação com o Dialogflow, de forma a enviar a mensagem do utilizador e receber a do agente, basta incorporar no código o método “`detect_intent_texts()`”, fornecido na documentação³, e adaptá-lo conforme os requisitos da aplicação.

O Dialogflow, sendo tipicamente um serviço de criação de *chatbots* que lidam com perguntas do utilizador, necessita de receber um *input* antes de dar início à conversa.

²<https://console.cloud.google.com/>

³<https://cloud.google.com/dialogflow/cx/docs/quick/api#detect-intent>

```
project_id = os.getenv('project_id')
location_id = os.getenv('location_id')
agent_id = os.getenv('agent_id')
session_id = uuid.uuid4()
agent = f"projects/{project_id}/locations/{location_id}/agents/{agent_id}"
```

Figura 5.2: Variáveis necessárias à autenticação do agente do Dialogflow.

Como neste trabalho se pretende que o agente tenha o controlo do fluxo de conversação, deve ser ele a iniciar a conversa com o utilizador e enviar a primeira mensagem. Para contornar esta necessidade do serviço, na página *Start* foi criada uma rota com uma *intent* específica para detetar uma mensagem com o valor “DEFAULT_WELCOME_INTENT”, que é enviada quando o utilizador faz *login* e é redirecionado para a página de conversação, com o endereço “/main” (Figura 5.3). Essa mensagem é enviada ao Dialogflow, através da chamada do método “detect_intent_texts()”, que retorna a resposta do agente.

```
@app.route('/main')
def main():
    global firstInt
    firstInt = True
    message = "DEFAULT_WELCOME_INTENT"
    fulfillment_text = detect_intent_texts(agent, session_id, message, 'pt')
    return render_template('main.html', initial_content = fulfillment_text)
```

Figura 5.3: Função “main()”, executada quando o utilizador é redirecionado para a página de conversação.

O Flask usa o indicador “route()” para vincular uma dada função a um determinado endereço. Assim, a *route* que se vê na Figura 5.3 serve para informar o Flask que, quando o utilizador é redirecionado para o endereço “/main”, a função “main()” deve ser acionada. A função retorna a mensagem que queremos exibir no navegador do utilizador, neste caso, renderizar a página de conversação, apresentando a primeira mensagem do agente.

Ao longo da conversa, sempre que o agente tem uma resposta estática para enviar ao utilizador, é executada a função “send_message()”. A UI encaminha a mensagem do utilizador para esta função, que, novamente, recorre ao método “detect_intent_texts()” para a enviar ao Dialogflow de forma a receber a resposta, que é retornada à UI e posteriormente mostrada ao utilizador. Quando a resposta do agente é dinâmica, a função chamada é a “webhook()”, detalhada na secção seguinte.

5.3.2 Webhook Service

Um *webhook* é um HTTP *request* enviado automaticamente sempre que determinados critérios são atendidos, neste caso, sempre que uma determinada rota no Dialogflow tem

```

@app.route('/send_message', methods=['POST'])
def send_message():
    message = request.form['message']
    fulfillment_text = detect_intent_texts(agent, session_id, message, 'pt')
    response_text = { "message": fulfillment_text }
    return jsonify(response_text)

```

Figura 5.4: Função “send_message()”, que recebe a mensagem do utilizador e retorna a resposta do agente.

a opção de *webhook* ativada. Um *webhook* pode ser criado em qualquer linguagem de programação do lado do servidor, neste caso, Python, com auxílio do Flask.

O Dialogflow não oferece suporte a *localhost*, o local onde a aplicação está, para já, a ser executada. Para integrá-lo como *webhook* foi necessário contornar esta limitação e arranjar forma de o tornar ativo. Para isso, tirou-se partido do ngrok⁴, que é uma ferramenta que permite criar túneis seguros de um *endpoint* público, como a Internet, para um servidor local.

Considerando que o servidor *web* está na porta 5000, para gerar um endereço basta executar o seguinte comando: `ngrok http 5000`. Na Figura 5.5 encontra-se o resultado da execução do comando, a partir do qual basta copiar o endereço que é gerado para o Dialogflow, na secção “Webhook URL”, seguido de “/webhook”, que será a *route* indicada para o Flask saber qual a função a executar.

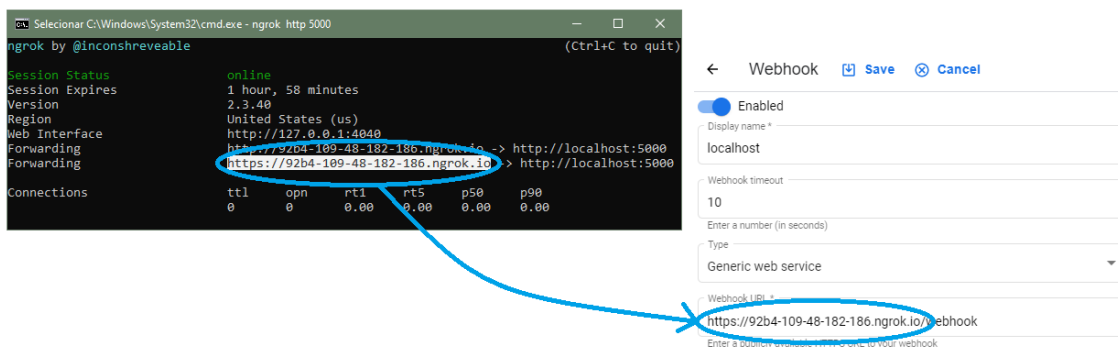


Figura 5.5: Processo de geração e definição do URL para o *webhook*.

A solicitação de *webhook* é feita através de um HTTP POST *request*, o que significa que, na *route* definida para o serviço de *webhook*, deve também estar especificado o método POST.

Na Figura 5.6 encontra-se uma amostra de código que demonstra o que se sucede quando o Dialogflow faz uma chamada de *webhook*. Cada página ou rota que tenha ativada a opção de *webhook* tem anexada uma *tag* específica que a distingue das restantes. Quando o sistema recebe um *webhook request*, acede-se ao conteúdo da *tag* contida na

⁴<https://ngrok.com/>

solicitação e, dependendo do seu valor, é executado o método correspondente a esse momento da interação, que consulta o que for necessário (da componente *BCT Knowledge*) e devolve a resposta adequada.

```
# evoked whenever there's a webhook call in order to give a dynamic response
@app.route('/webhook', methods=['POST'])
def webhook():
    data = request.get_json(silent=True)
    tag = data.get('fulfillmentInfo').get('tag')
    if tag == "review_tasks":
        fulfillment_text = oc.reviewTasks()
    elif tag == "determinant":
        det = data.get('sessionInfo').get('parameters').get('determinant')
        fulfillment_text = oc.problemSolvingDet(det)
    elif tag == "counselling":
        # different response because it can also contain parameters
        response = oc.counselling()
        return jsonify(response)
    elif tag == "assign_tasks":
        fulfillment_text = oc.assignTasks()
    response = {
        "fulfillment_response": {
            "messages": [ { "text": { "text": [ fulfillment_text ] } } ]
        }
    }
    return jsonify(response)
```

Figura 5.6: Função “webhook()”, executada sempre que o agente faz um pedido ao serviço de *webhook*.

5.3.3 User Interface

A *user interface* é definida pelo conteúdo dos ficheiros HTML (*index.html* e *main.html*), CSS (*style.css*) e JS (*custom.js*). O Flask requer que os ficheiros HTML estejam todos na diretoria “*templates*”, e os ficheiros CSS e JS na diretoria “*static*”.

O ficheiro *index.html* contém a estrutura de uma página inicial simples, onde o utilizador pode fazer *login* através do seu *username*. Caso o *username* digitado se encontre na ontologia, o utilizador é redirecionado para a página principal, onde decorrerá a conversa com o agente; caso o *username* não esteja na ontologia, a página é recarregada (para já ainda não está disponível a criação de um novo utilizador através da aplicação, apenas diretamente na ontologia).

O ficheiro *main.html* contém a estrutura da página onde decorre a conversa. A estrutura que a página apresenta assemelha-se à das aplicações de mensagens existentes nos

telemóveis, com um campo para o utilizador digitar a mensagem, com um balão colorido mais à esquerda que contém a mensagem do agente, e com um balão colorido mais à direita, que contém a mensagem que o utilizador envia (ver Figura 6.4).

A função mais importante desta componente encontra-se no ficheiro `custom.js`, que é a função que é acionada quando o utilizador envia uma mensagem ao agente: “`submit_message(message)`” (Figura 5.7). O parâmetro “`message`” contém a resposta que o utilizador digitou e clicou para enviar. Após adicionar a mensagem do utilizador à página, é chamada a função “`submit_message`”, que faz POST na *route* “`/send_message`” (sublinhado a azul na Figura 5.7). A função aí presente trata de enviar a mensagem do utilizador ao Dialogflow, e após receber a resposta do agente, retorna-a para esta função. Ao receber a mensagem do agente, a função “`handle_response`” é acionada, que trata de adicionar à página *web* uma secção com a resposta do agente, para o utilizador ver.

```
function submit_message(message) {  
  $.post( "/send_message", {message: message}, handle_response);  
  
  function handle_response(data) {  
    document.getElementById('response').id = "past-response"  
    // append the agent's response to the div  
    $('chat-container').append(`  
      <div class="chat-message col-md-6 agent-message" id="response">  
        ${data.message}  
      </div>  
    `)  
    // remove the loading indicator  
    $("#loading").remove();  
    document.getElementById('response').scrollIntoView({block: 'start', behavior: 'smooth'});  
  }  
}
```

Figura 5.7: Função “`submit_message`”.

5.4 *BCT Knowledge*

Esta componente inclui o conhecimento com que o agente é dotado e que pode consultar sempre que for necessário de forma a obter a melhor resposta para dar ao utilizador. Este módulo divide-se na integração da ontologia definida e no acesso a dois ficheiros JSON (`problem_solving_det.json` e `counselling.json`) que contêm diálogo extra que diz respeito à etapa *Counselling*.

A classe “`OntologyConnection`” é responsável por todas as comunicações realizadas com a ontologia, feitas com recurso ao OWLready. Na Figura 5.8 encontra-se parte do código necessário para integrar a ontologia definida na aplicação. É criada uma instância da classe “`OntologyConnection`”, onde é carregada a ontologia (ficheiro `BCCO.owl`) para

a variável “self._onto”, a partir da qual se pode aceder a qualquer entidade definida na ontologia (classe, propriedade, ou instância, por exemplo).

```

# ontology connection index.py
script_location = os.path.dirname(sys.argv[0])
oc = OntologyConnection.OntologyConnection(script_location)
-----
class OntologyConnection: OntologyConnection.py
    """
    Class that deals with everything that is Ontology related.
    """
    def __init__(self, script_location):
        onto_path.append(script_location)
        self._onto = get_ontology("file://BCCO.owl").load()
        self._int = 0

```

Figura 5.8: Integração da ontologia através da classe “OntologyConnection”.

Uma das alturas em que o agente tem necessidade de procurar mais informação é no caso descrito na Secção 3.3.1, quando o utilizador não cumpriu o objetivo acordado e foi questionado sobre o porquê de não o cumprir. Após receber a resposta do utilizador (com o determinante de não adesão), o agente consulta a ontologia para ver qual o objetivo que o utilizador tem ativo no momento da interação. Na Figura 5.9 encontra-se a parte de código em que, a partir do determinante (variável “det”) e do objetivo (variável “activeGoal”) identificados, o agente pesquisa na ontologia qual operacionalização (propriedade “is_a”) relacionada com o objetivo ativo (propriedade “relatedTo”) é acionada por esse determinante (propriedade “triggeredBy”).

```

op = self._onto.search_one(
    is_a = self._onto.Operationalization,
    relatedTo = self._onto.search_one(is_a = self._onto.Behavior_Goal, iri = "*" + activeGoal[5:]),
    triggeredBy = self._onto.search_one(is_a = self._onto.Behavior_Determinant, iri = "*" + det)
)

```

Figura 5.9: Amostra de código onde é feita uma consulta à ontologia.

Ao identificar a operacionalização correta, o agente verifica quais as BCTs que devem ser concretizadas (accedendo à propriedade “triggers”). No caso da operacionalização ser sequencial, o agente verifica também qual a ordem com que as BCTs devem ser concretizadas no diálogo, acedendo à propriedade “bctOrder”.

Após recolher toda a informação que necessita da ontologia, o agente acede ao ficheiro `problem_solving_det.json`, guardando o seu conteúdo na variável “dialog”, e vai buscar a resposta adequada para devolver ao utilizador.

O ficheiro `problem_solving_det.json` está organizado em formato de dicionário, acedendo à resposta correta da seguinte forma:

- No caso da operacionalização ser da classe *Basic*:
 - “response = dialog[activeGoal][str(op.triggers[0])]”, no caso da concretização da BCT ser independente do determinante.
 - “response = dialog[activeGoal][str(op.triggers[0])[det]”, no caso da concretização da BCT depender diretamente do determinante.
- No caso da operacionalização ser da classe *Sequential* (percorrendo cada uma das BCTs que concretiza):
 - “response.append(dialog[activeGoal][bct])”, no caso da concretização da BCT ser independente do determinante.
 - “response.append(dialog[activeGoal][bct][det])”, no caso da concretização da BCT depender diretamente do determinante.

Quando a etapa *Counselling* está ativa, o agente precisa de aceder ao ficheiro `counselling.json`. Esta poderá não ser a solução mais prática, uma vez que ainda está muito rígida e requer algum trabalho manual a definir os diálogos segundo a estrutura do ficheiro, no entanto, foi a melhor solução encontrada até ao momento.

Como esta é a etapa mais longa e que exige mais interação de ambas as partes, foi organizado o ficheiro JSON tendo em conta o tópico ativo e consoante uma variável auxiliar “int” (Figura 5.10). Esta variável é definida logo que é criada a instância da classe “OntologyConnection”, como se pode ver na Figura 5.8 (variável “self.int”), e é incrementada cada vez que o método “counselling()” é chamado. Quando é altura de aceder à resposta, o agente faz a concatenação da *string* “int_” com o valor da variável (“this.int = ”int_”+str(self.int)”), de forma a corresponder com o formato da chave do dicionário, obtendo a resposta através de “response = dialog[activeTopic][this.int]”.

A etapa *Counselling* prolonga-se até o parâmetro “counselling” ser definido como “FINISH”, pelo que, antes de obter a resposta, o agente verifica se o valor da próxima entrada do dicionário é “FINISH”. No caso da Figura 5.10, quando está na quinta interação da etapa (“self.int = 5”), o agente vê que a próxima entrada tem o valor “FINISH”,

```

    },
    "BCTO.Topic_02" : {
      "int_1" : "Hoje vamos falar sobre snacks saudáveis. (...)",
      "int_2" : "Os lanches ajudam a diminuir a sensação (...)",
      "int_3": "A decisão de introduzir uma refeição intermédia (...)",
      "int_4": "Os lanches ou refeições intermédias também (...)",
      "int_5": "O planeamento dos lanches irá ser mais fácil (...)",
      "int_6": "FINISH"
    },
    "BCTO.Topic_03" : {

```

Figura 5.10: Amostra do ficheiro `counselling.json`.

então, além de enviar a resposta que diz respeito à quinta mensagem da etapa, envia também indicação de que o parâmetro de sessão “counselling” passa a ser igual a “FINISH”. Quando isto acontece, o Dialogflow passa à próxima página e etapa (é possível ver esta rota de condição na Figura 4.2).

5.5 Conclusão

Neste capítulo encontra-se descrita a implementação da solução desenvolvida para completar os objetivos deste trabalho, fazendo a integração entre a ontologia e o agente conversacional definidos (descritos nos dois capítulos antecedentes).

Foi descrita a arquitetura do sistema implementado e as suas componentes, sendo depois destacado como é exercida a comunicação entre as várias componentes do sistema, ilustrando as mais importantes através de amostras do código desenvolvido.

No próximo capítulo é apresentado, através de casos de uso práticos, o funcionamento da prova de conceito desenvolvida.

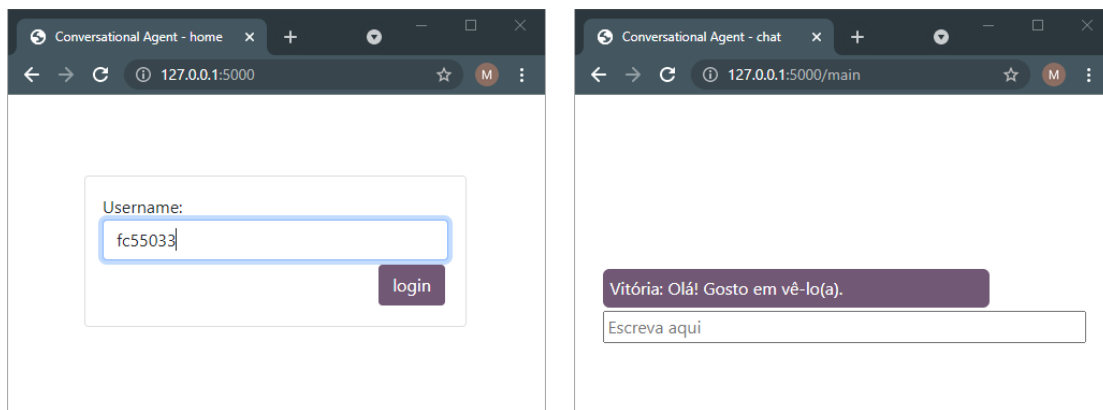
Capítulo 6

Prova de conceito

Este capítulo começa pela apresentação das interfaces criadas para a aplicação (Secção 6.1) e, de seguida, são retratados alguns exemplos que demonstram a funcionalidade da solução implementada, nomeadamente no que diz respeito à correta concretização de diversas BCTs durante as etapas do diálogo, tanto as BCTs mais simples (Secção 6.2) como outras mais complexas (Secção 6.3).

6.1 Interfaces da aplicação

Como mencionado na Secção 5.3.3, foram estruturadas duas páginas com interfaces bastante simples (Figura 6.1), que permitem testar facilmente a solução implementada.



(a) Página inicial de *login*.

(b) Página de conversação.

Figura 6.1: Interfaces da aplicação.

A primeira página que o utilizador visualiza é uma página inicial, onde o utilizador pode fazer *login* através do seu *username* (Figura 6.1a). Caso o *username* digitado se encontre na ontologia, o utilizador é redirecionado para a página principal, onde decorrerá a conversa com o agente; caso o *username* não esteja na ontologia, a página é recarregada.

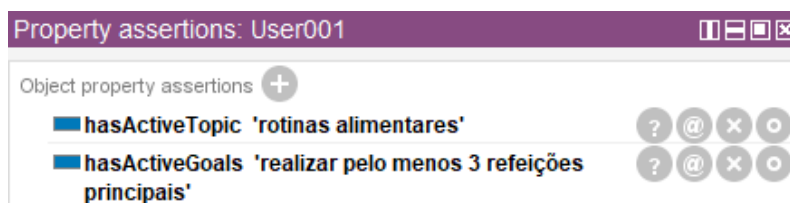
A página principal da aplicação, onde decorre a conversa, foi estruturada conforme as aplicações de mensagens existentes nos telemóveis, com um campo para o utilizador

digitar a mensagem, com um balão colorido mais à esquerda que contém a mensagem do agente, e com um balão colorido mais à direita, que contém a mensagem que o utilizador envia. A Figura 6.1b mostra a página apresentada ao utilizador assim que ele é redirecionado para lá, com a mensagem de saudação do agente.

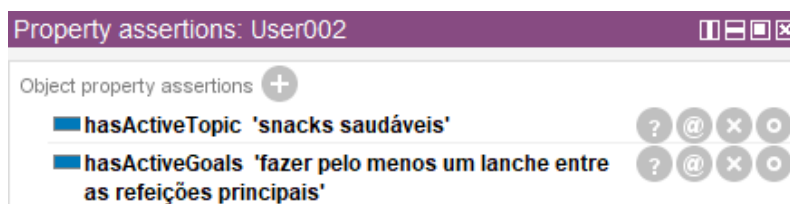
6.2 Exemplo 1: Revisão dos objetivos

Como foi observado durante a análise feita ao agente precedente, em determinadas etapas do diálogo, existem BCTs específicas que são sempre operacionalizados da mesma maneira, sendo mais simples de executar. Uma dessas etapas é a *Review Tasks*, na qual a BCT *Self-monitoring of behavior (2.3)* é sempre executada e concretizada da mesma forma: recolhendo os dados sobre o comportamento do utilizador. Para isso, o sistema tira proveito das seguintes classes de ontologia: *User* e *Behavior Goal*. Estas classes interligam-se a partir da propriedade *hasActiveGoals*, que relaciona um ou mais objetivos comportamentais a um determinado utilizador.

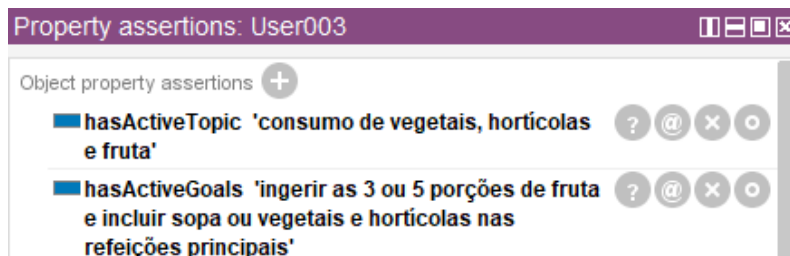
Na Figura 6.2 encontram-se três utilizadores diferentes (instâncias User001, User002 e User003), e as *object properties* que têm associadas. Cada um dos utilizadores tem um tópico e objetivo ativos diferente dos restantes.



(a) Propriedades da instância User001.



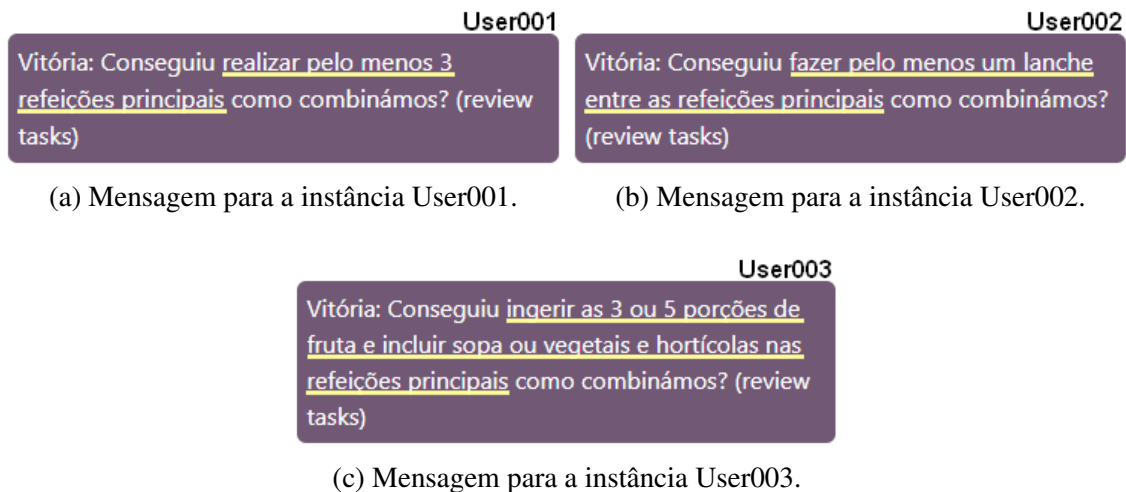
(b) Propriedades da instância User002.



(c) Propriedades da instância User003.

Figura 6.2: Propriedades associadas a três utilizadores diferentes.

Quando o agente passa para a etapa *Review Tasks*, o sistema faz uma pesquisa na ontologia para saber qual o objetivo que o utilizador tem ativo, e acede ao seu *label*, que é definido na ontologia de forma a ser inserido corretamente na *string* “Conseguiu <label do objetivo> como combinámos?”. Na Figura 6.3 encontram-se as mensagens do agente para cada um dos utilizadores da Figura 6.2, onde é possível verificar a inserção dos diferentes objetivos na frase.



(a) Mensagem para a instância User001.

(b) Mensagem para a instância User002.

(c) Mensagem para a instância User003.

Figura 6.3: Concretização da BCT *Self-monitoring of behavior* (2.3) em três utilizadores diferentes.

Na etapa *Assign Tasks*, o método de concretização das BCTs é semelhante, integrando o *label* do objetivo na *string* “Com base nesta conversa, gostava que tentasse <label do objetivo>. Pode ser?”. Antes de o propor ao utilizador, o sistema atualiza a ontologia, alterando o valor da propriedade *hasActiveGoals* para o objetivo seguinte.

Como só há uma maneira de operacionalizar as duas BCTs mencionadas nesta secção, e tendo em conta que dependem apenas do valor das propriedades às quais estão relacionadas, não houve necessidade de associar a sua operacionalização à classe *Operationalization* (onde seriam instâncias da classe *Basic*).

6.3 Exemplo 2: Não adesão ao objetivo

Durante as etapas *Assess* e *Counselling*, o procedimento é um pouco mais complexo que o das BCTs da secção anterior, uma vez que as BCTs operacionalizadas nestas etapas são executadas de várias maneiras diferentes, dependendo de mais do que uma condição.

Seguindo o exemplo mencionado na secção anterior, depois da etapa *Review Tasks*, segue-se a etapa *Assess*, e durante essa etapa, o sistema tira proveito das seguintes classes: *Operationalization*, *User*, *Behavior Goal*, *Behavior Determinant* e *Behavior Change Technique*. Como foi mencionado na Secção 3.3.1, quando o sistema recebe o determinante de não adesão, ele consulta a ontologia para descobrir qual operacionalização,

relacionada com o objetivo ativo, é acionada pelo determinante especificado. De seguida, o sistema acede à propriedade `triggers`, para ver que BCTs são acionadas por essa operacionalização. Caso a operacionalização seja do tipo *Sequential*, o sistema acede à propriedade `bctOrder`, de forma a concretizar as BCTs na ordem adequada.

A Figura 6.4 contém uma demonstração que cobre as etapas mencionadas nos parágrafos anteriores. Ainda na etapa *Review Tasks*, o agente pergunta ao utilizador se ele cumpriu a meta acordada (“fazer pelo menos 3 refeições principais”, sublinhado a rosa na primeira mensagem no canto superior esquerdo), ao qual o utilizador responde negativamente. Depois disso, é operacionalizada outra BCT mais simples, *Feedback on behavior* (2.2) (“Teria sido importante, mas vamos falar sobre isso.”), concretizada estaticamente pelo Dialogflow, uma vez que depende apenas da *intent* (“Sim” ou “Não”). O utilizador responde com “Não tenho motivação”, e a partir dessa frase o sistema capta o determinante “motivação”.

Agent: Conseguiu realizar pelo menos 3 refeições principais como combinámos? (review tasks)

Não

Agent: Teria sido importante mas vamos falar sobre isso. (assess)
Vamos esperar que a nossa conversa possa ajudar. Porque é que não conseguiu alcançar esta meta? (assess) [Respostas exemplificativas: Não tenho motivação; Está a ser difícil mudar os velhos hábitos; Etc...]

Não tenho motivação

Agent: Ocasionalmente, pode optar por comer fora ou adquirir refeições prontas e levá-las para casa. Já pensou em inscrever-se num centro de dia, que pode levar as refeições a sua casa? Para manter valores de glicemia estáveis, é importante distribuir a ingestão de hidratos de carbono ao longo do dia, sem saltar refeições. Saltar refeições leva muitas vezes a que se coma demasiado na refeição a seguir. E pode, com alguns medicamentos para tratar a diabetes, levar a que o açúcar do sangue baixe demasiado. Pergunte aos seus profissionais de saúde se é o caso dos seus medicamentos para a diabetes. Parece-lhe bem? (problem solving)

Escreva aqui

Figura 6.4: Demonstração da concretização de diversas BCTs.

A operacionalização relacionada ao objetivo ativo, e desencadeada pelo determinante “motivação”, é a instância *OP_Alim_Det1_2*, que aciona as BCTs *Problem solving (1.2)* e *Information of health consequences (5.1)*, tendo “1.2, 5.1” como valor da propriedade `bctOrder`. Tendo em conta estas condições, o sistema acede ao ficheiro `problem_solving_det.json`, e extrai as partes do diálogo que correspondem às BCTs acionadas, ordena-as e envia-as ao utilizador. Na última mensagem do agente, no canto inferior esquerdo da Figura 6.4, está sublinhado a azul o diálogo que concretiza a BCT *Problem solving (1.2)*, e a amarelo o diálogo que concretiza a BCT *Information of health consequences (5.1)*.

Como mencionado na Secção 4.2, no caso do utilizador indicar um motivo ainda não definido, impossibilitando o agente de identificar alguma das entidades definidas no Dialogflow, foi adicionado um *event handler* na página *Get Determinant* para lidar com a situação e não deixar o utilizador sem resposta, enviando uma resposta *default* e fazendo a transição para a página seguinte. Este caso encontra-se exemplificado na Figura 6.5, na qual o utilizador diz não saber bem o porquê de não ter alcançado o objetivo proposto.

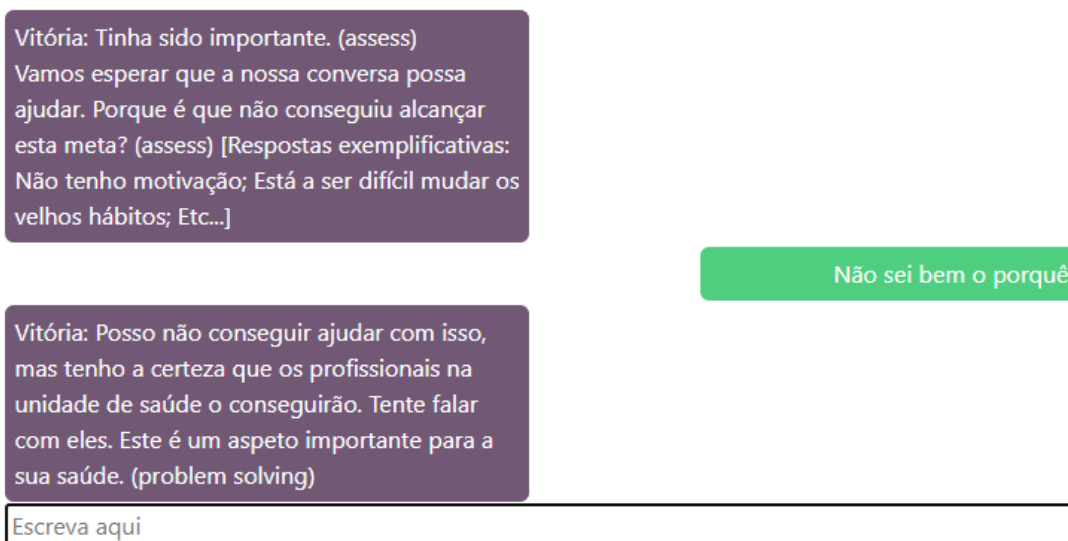


Figura 6.5: Resposta *default*, retornada pelo *event handler* da página *Get Determinant*.

No caso do utilizador responder afirmativamente, este processo é evitado, passando logo à etapa *Counselling*, na qual também se acede a um ficheiro JSON, tirando partido da propriedade `hasActiveTopic`. Nesta etapa, as BCTs concretizadas não estão associadas a operacionalizações na ontologia, estando embutidas no diálogo. Como evidenciado na Secção 5.4, a solução aplicada nesta etapa ainda está bastante rígida, pelo que, deve continuar a ser trabalhada de forma a otimizar e generalizar o procedimento de concretização de BCTs nesta etapa.

6.4 Conclusão

Neste capítulo foram apresentadas as interfaces criadas para a aplicação e expostos alguns exemplos que demonstram a funcionalidade da solução implementada, nomeadamente no que diz respeito à correta concretização de diversas BCTs (umas mais simples e outras mais complexas) durante as etapas do diálogo.

Os exemplos apresentados permitem constatar que o propósito deste trabalho foi alcançado, provando que a integração dos dois módulos desenvolvidos é bem sucedida no âmbito da criação de agentes conversacionais avançados para intervenções de mudança comportamental.

No próximo capítulo são expostas algumas conclusões acerca do trabalho desenvolvido, juntamente com algumas expectativas de trabalho futuro.

Capítulo 7

Conclusões e trabalho futuro

7.1 Conclusões

No capítulo introdutório do presente documento, evidenciou-se que este trabalho se propunha a ultrapassar duas limitações do agente precedente (desenvolvido no âmbito do projeto VASelfCare): (1) o *input* do utilizador ser escolhido a partir de um conjunto limitado de opções, e (2) a operacionalização das BCTs ser totalmente delineada aquando da criação prévia dos diálogos.

Apesar do seu início lento, devido ao extenso material existente para analisar, à pandemia COVID-19, que dificultou e atrasou algumas reuniões essenciais para o avanço deste trabalho, e a uma mudança no planeamento inicial, os objetivos definidos foram todos cumpridos.

O objetivo principal deste trabalho foi alcançado através da conceção e implementação de uma nova arquitetura que permite o desenvolvimento de agentes conversacionais para intervenções de mudança comportamental. Esta arquitetura possibilita a integração de um motor de diálogo avançado com capacidade de aprendizagem (Dialogflow) com a representação do conhecimento sobre a operacionalização de técnicas de mudança comportamental, através da definição de uma ontologia de conceitos de mudança comportamental.

As duas características principais que se pretendiam ter no agente conversacional eram: (1) a possibilidade de haver interação da linguagem natural, e (2) algum controlo do fluxo do diálogo, de forma a abordar situações em que o agente é o responsável por conduzir o diálogo (e não o contrário, como acontece em contextos de pergunta-resposta, por exemplo). O uso do Dialogflow viabiliza ambas as características pretendidas, e permitiu ultrapassar a primeira limitação.

A inclusão do módulo que permite a representação explícita do conhecimento sobre a operacionalização das BCTs contorna a segunda limitação, conseguido através da definição da ontologia.

A prova de conceito construída foi bem sucedida no seu propósito, e pode servir de

base para o desenvolvimento de agentes conversacionais avançados, o que levanta várias perspectivas interessantes de trabalho futuro.

A nível pessoal, este trabalho foi bastante gratificante, não só pela concretização dos objetivos que foram propostos, mas também por ter tido a oportunidade de trabalhar com uma equipa multidisciplinar e dinâmica, que sempre se mostrou disponível a ajudar e trocar ideias durante o avanço do trabalho, principalmente na definição da ontologia.

Em termos de competências técnicas, durante a realização deste trabalho foi possível interagir com bastantes ferramentas, incluindo algumas com as quais nunca tinha tido contacto, como é o caso do Dialogflow e de todas as outras ferramentas de criação de diálogo, mencionadas na Secção 2.4. Permitiu-me ainda aprimorar o conhecimento que já tinha de outras ferramentas, como o Protegé, e linguagens de programação, como Python, HTML, CSS e JavaScript.

7.2 Trabalho futuro

A prova de conceito construída mostrou-se bem sucedida, pelo que, a partir daqui surgem várias vertentes de continuação deste trabalho.

Duas das funcionalidades mais urgentes são a incorporação de conhecimentos adicionais, de forma a que a capacidade do agente seja ampliada, e treinar o sistema com um conjunto de diálogos que foram desenvolvidos para o trabalho anterior, de forma a solidificar o reconhecimento do agente do Dialogflow. A ontologia definida, apesar de conter já os conceitos base para estruturar uma intervenção, ainda não é a solução ótima, devendo continuar a ser explorada e ampliada, de forma a generalizar ainda mais a operacionalização das BCTs e a entrega do diálogo do agente, nomeadamente no que diz respeito à etapa de *Counselling*.

Pretende-se também incorporar mecanismos que permitam considerar uma dimensão ética neste tipo de agentes. Zhang et al. apontam que as dimensões éticas relacionadas ao desenvolvimento de agentes conversacionais têm estado completamente ausentes e que há uma falta de entendimento na orientação teórica e nas recomendações práticas sobre o *design* de *chatbots* com inteligência artificial para programas de modificação de estilo de vida [Zhang et al., 2020]. Em contextos cujo objetivo é induzir mudança comportamental em humanos, a incorporação de princípios éticos e a garantia de responsabilidade dos *designers* do sistemas são de grande importância, pelo que, devem procurar incorporar-se.

Na Secção 2.3 é feito um levantamento acerca do uso de *Embodied Conversational Agents* (ECAs) nas aplicações *eHealth*. Neste trabalho, foi desenvolvido o que diz respeito à parte conversacional do agente, ficando por implementar o que diz respeito à sua personificação. No entanto, antes de se dar prioridade à definição de uma ontologia, foram realizadas experiências de integração das ferramentas Watson Assistant e FAtiMA Toolkit

com um ambiente de desenvolvimento que suporte personagens virtuais, o Unity (*software* usado no projeto VASelfCare). O trabalho desenvolvido nesse contexto encontra-se no Apêndice D.

O Dialogflow fornece bibliotecas de cliente e guias para as seguintes linguagens: Java, Node.js e Python, tendo esta última sido a utilizada neste trabalho. O Unity usa o C# como linguagem de programação, pelo que, a integração do agente criado neste trabalho no Unity não seria facilitada. No entanto, há ferramentas de *software* semelhantes disponíveis onde se poderia criar uma aplicação integrando o agente desenvolvido ou, dependendo do tipo de modelo que se vier a incluir (2D ou 3D), fazer a animação num *software* como o OpenToonz¹ (um *software open-source* gratuito para a criação de animações 2D) e exportá-la para ser integrada no sistema desenvolvido.

Como [Bickmore et al., 2005] reconhecem, além das conversas que os ECAs são capazes de manter com os utilizadores, torna-se também necessário incorporar aspetos sociais, emocionais e relacionais da interação face a face num agente computacional, sendo especialmente crucial para agentes projetados para interagir com o utilizador durante um longo período de tempo. Assim sendo, pretende-se alargar a modelação do agente de forma a incluir também estes aspetos.

Ainda na vertente de incorporação de aspetos emocionais, uma das funcionalidades que se pretende desenvolver diz respeito ao *feedback* afetivo do utilizador. O *feedback* afetivo procura associar uma emoção ao utilizador, e pode ser deduzida automaticamente através do comportamento do utilizador para com a aplicação, ou perguntando-lhe diretamente. Este tipo de funcionalidade permite recolher informação sobre as emoções do utilizador em relação a vários aspetos, e pode vir a ser usado para a personalização da intervenção de mudança comportamental, assim como ser uma ferramenta de avaliação e melhoria do produto. Também nesta vertente foi desenvolvida alguma pesquisa, que pode ser consultada no Apêndice E.

A personalização da intervenção de mudança comportamental é um objetivo fulcral a ser implementado no futuro. A solução criada para este trabalho serviu o propósito de comprovar que é possível, e traz vantagens, integrar os diferentes módulos desenvolvidos para obter um agente mais avançado e inteligente que o seu precedente, no entanto, a intervenção não está completa, cobrindo apenas três interações (cada uma com um tópico e objetivo diferentes, relacionados à componente de Alimentação). A ideia é que os objetivos só sejam alterados dependendo de certas condições no estado do utilizador para com a intervenção (por exemplo, nível de risco, adesão aos objetivos). Os ficheiros de código desenvolvidos incluem comentários que indicam as partes que foram feitas de determinada forma em prol do trabalho desenvolvido neste projeto e que deverão ser condicionadas no futuro.

¹<https://opentoonz.github.io/e/>

Bibliografia

- [Balsa et al., 2020] Balsa, J., Félix, I., Cláudio, A. P., Carmo, M. B., Silva, I. C. e., Guerreiro, A., Guedes, M., Henriques, A., and Guerreiro, M. P. (2020). Usability of an Intelligent Virtual Assistant for Promoting Behavior Change and Self-Care in Older People with Type 2 Diabetes. *Journal of Medical Systems*, 44(7):130.
- [Baptista et al., 2020] Baptista, S., Wadley, G., Bird, D., Oldenburg, B., Speight, J., and Group, T. M. D. C. R. (2020). Acceptability of an Embodied Conversational Agent for Type 2 Diabetes Self-Management Education and Support via a Smartphone App: Mixed Methods Study. *JMIR mHealth and uHealth*, 8(7):e17038.
- [Betella and Verschure, 2016] Betella, A. and Verschure, P. (2016). The affective slider: A digital self-assessment scale for the measurement of human emotions. *PLoS ONE*, 11:e0148037.
- [Bhuyan et al., 2016] Bhuyan, S. S., Lu, N., Chandak, A., Kim, H., Wyant, D., Bhatt, J., Kedia, S., and Chang, C. F. (2016). Use of Mobile Health Applications for Health-Seeking Behavior Among US Adults. *Journal of Medical Systems*, 40(6):153.
- [Bickmore et al., 2005] Bickmore, T. W., Caruso, L., Clough-Gorr, K., and Heeren, T. (2005). ‘it’s just like you talk to a friend’ relational agents for older adults. *Interacting with Computers*, 17(6):711 – 735. HCI and the Older Population.
- [Bradley and Lang, 1994] Bradley, M. M. and Lang, P. J. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49 – 59.
- [Broekens and Brinkman, 2009] Broekens, J. and Brinkman, W.-P. (2009). Affectbutton: Towards a standard for dynamic affective user feedback. pages 1 – 8.
- [Cassell et al., 2000] Cassell, J., Sullivan, J., Prevost, S., and Churchill, E. (2000). *Embodied Conversational Agents*.
- [Espanha and Fonseca, 2010] Espanha, R. and Fonseca, R. B. (2010). Plano Nacional de Saúde 2011-2016 - Tecnologias de Informação e Comunicação. *Alto Comissariado da Saúde*.

- [Félix et al., 2019] Félix, I. B., Guerreiro, M. P., Cavaco, A., Cláudio, A. P., Mendes, A., Balsa, J., Carmo, M. B., Pimenta, N., and Henriques, A. (2019). Development of a Complex Intervention to Improve Adherence to Antidiabetic Medication in Older People Using an Anthropomorphic Virtual Assistant Software. *Frontiers in Pharmacology*, 10:680.
- [Global Observatory for eHealth, 2011] Global Observatory for eHealth, W. (2011). mhealth: new horizons for health through mobile technologies: second global survey on ehealth.
- [Gong et al., 2020] Gong, E., Baptista, S., Russell, A., Scuffham, P., Riddell, M., Speight, J., Bird, D., Williams, E., Lotfaliany, M., and Oldenburg, B. (2020). My Diabetes Coach, a Mobile App–Based Interactive Conversational Agent to Support Type 2 Diabetes Self-Management: Randomized Effectiveness-Implementation Trial. *Journal of Medical Internet Research*, 22(11):e20322.
- [Guimarães et al., 2019] Guimarães, M., Mascarenhas, S., Prada, R., Santos, P., and Dias, J. (2019). An accessible toolkit for the creation of socio-emotional agents. *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2357–2359.
- [Hall et al., 2017] Hall, M., Elliott, K., and Meng, J. G. (2017). Using the pad (pleasure, arousal, and dominance) model to explain facebook attitudes and use intentions. *The Journal of Social Media in Society*, 6(1).
- [Hippisley-Cox et al., 2017] Hippisley-Cox, J., Coupland, C., and Brindle, P. (2017). Development and validation of qrisk3 risk prediction algorithms to estimate future risk of cardiovascular disease: prospective cohort study. *BMJ*, 357.
- [IDF, 2014] IDF (2014). How mobile health can help tackle the diabetes epidemic and strengthen health systems: International diabetes federation (IDF) response to the public consultation on the european commission’s green paper on mobile health. *Diabetes research and clinical practice*, 105(2):271–272.
- [Lamy, 2017] Lamy, J.-B. (2017). Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial Intelligence in Medicine*, 80:11–28.
- [Laranjo et al., 2018] Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A. Y. S., and Coiera, E. (2018). Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association : JAMIA*, 25(9):1248—1258.

- [Larsen et al., 2017] Larsen, K., Michie, S., Hekler, E., Gibson, B., Spruijt-Metz, D., Ahern, D., Cole-Lewis, H., Bartlett Ellis, R., Hesse, B., Moser, R., and Yi, J. (2017). Behavior change interventions: the potential of ontologies for advancing science and practice. *Journal of Behavioral Medicine*, 40.
- [Michie et al., 2014] Michie, S., Atkins, L., and West, R. (2014). *The behaviour change wheel : a guide to designing interventions*. Silverback.
- [Michie et al., 2013] Michie, S., Richardson, M., Johnston, M., Abraham, C., Francis, J., Hardeman, W., Eccles, M. P., Cane, J., and Wood, C. E. (2013). The Behavior Change Technique Taxonomy (v1) of 93 Hierarchically Clustered Techniques: Building an International Consensus for the Reporting of Behavior Change Interventions. *Annals of Behavioral Medicine*, 46(1):81–95.
- [Michie et al., 2017a] Michie, S., Thomas, J., Johnston, M., Aonghusa, P. M., Shawe-Taylor, J., Kelly, M. P., Deleris, L. A., Finnerty, A. N., Marques, M. M., Norris, E., O’Mara-Eves, A., and West, R. (2017a). The Human Behaviour-Change Project: harnessing the power of artificial intelligence and machine learning for evidence synthesis and interpretation. *Implementation Science*, 12(1):121.
- [Michie et al., 2021] Michie, S., West, R., Finnerty, A., Norris, E., Wright, A., Marques, M., Johnston, M., Kelly, M., Thomas, J., and Hastings, J. (2021). Representation of behaviour change interventions and their evaluation: Development of the Upper Level of the Behaviour Change Intervention Ontology [version 2; peer review: 2 approved]. *Wellcome Open Research*, 5(123).
- [Michie et al., 2017b] Michie, S., Yardley, L., West, R., Patrick, K., and Greaves, F. (2017b). Developing and evaluating digital interventions to promote behavior change in health and health care: Recommendations resulting from an international workshop. *J Med Internet Res*, 19(6):e232.
- [Raposo, 2020] Raposo, J. F. (2020). Diabetes: Factos e números 2016, 2017 e 2018. *Revista Portuguesa da Diabetes*, 15(1):19–27.
- [Ribeiro and Monteiro, 2015] Ribeiro, A. and Monteiro, L. (2015). A indução afetiva em cenários de realidade virtual: avaliação da sensação de presença. *Psicologia Clínica*, 27:139–160.
- [Short et al., 2018] Short, C. E., DeSmet, A., Woods, C., Williams, S. L., Maher, C., Middelweerd, A., Müller, A. M., Wark, P. A., Vandelanotte, C., Poppe, L., Hingle, M. D., and Crutzen, R. (2018). Measuring engagement in ehealth and mhealth behavior change interventions: Viewpoint of methodologies. *J Med Internet Res*, 20(11):e292.

- [Teixeira et al., 2021] Teixeira, M. S., Maran, V., and Dragoni, M. (2021). The interplay of a conversational ontology and AI planning for health dialogue management. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 611–619, Virtual Event Republic of Korea. ACM.
- [ter Stal et al., 2020] ter Stal, S., Kramer, L. L., Tabak, M., op den Akker, H., and Hermens, H. (2020). Design features of embodied conversational agents in ehealth: a literature review. *International Journal of Human-Computer Studies*, 138:102409.
- [Wang et al., 2020] Wang, I., Buchweitz, L., Smith, J., Bornholdt, L.-S., Grund, J., Ruiz, J., and Korn, O. (2020). Wow, you are terrible at this!-an intercultural study on virtual agents giving mixed feedback.
- [Weizenbaum, 1966] Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.
- [Zapata et al., 2015] Zapata, B. C., Fernández-Alemán, J. L., Idri, A., and Toval, A. (2015). Empirical Studies on Usability of mHealth Apps: A Systematic Literature Review. *Journal of Medical Systems*, 39(2):1.
- [Zhang et al., 2020] Zhang, J., Oh, Y. J., Lange, P., Yu, Z., and Fukuoka, Y. (2020). Artificial Intelligence Chatbot Behavior Change Model for Designing Artificial Intelligence Chatbots to Promote Physical Activity and a Healthy Diet: Viewpoint. *Journal of Medical Internet Research*, 22(9):e22845.

Apêndice A

Caracterização dos elementos de intervenção da componente de Alimentação

Neste apêndice encontra-se uma versão reduzida da tabela resultante da análise dos diálogos de determinadas etapas da componente de Alimentação, incluída na intervenção de mudança comportamental do projeto VASelfCare.

Diet Non-lite Follow-up
Map 3.2

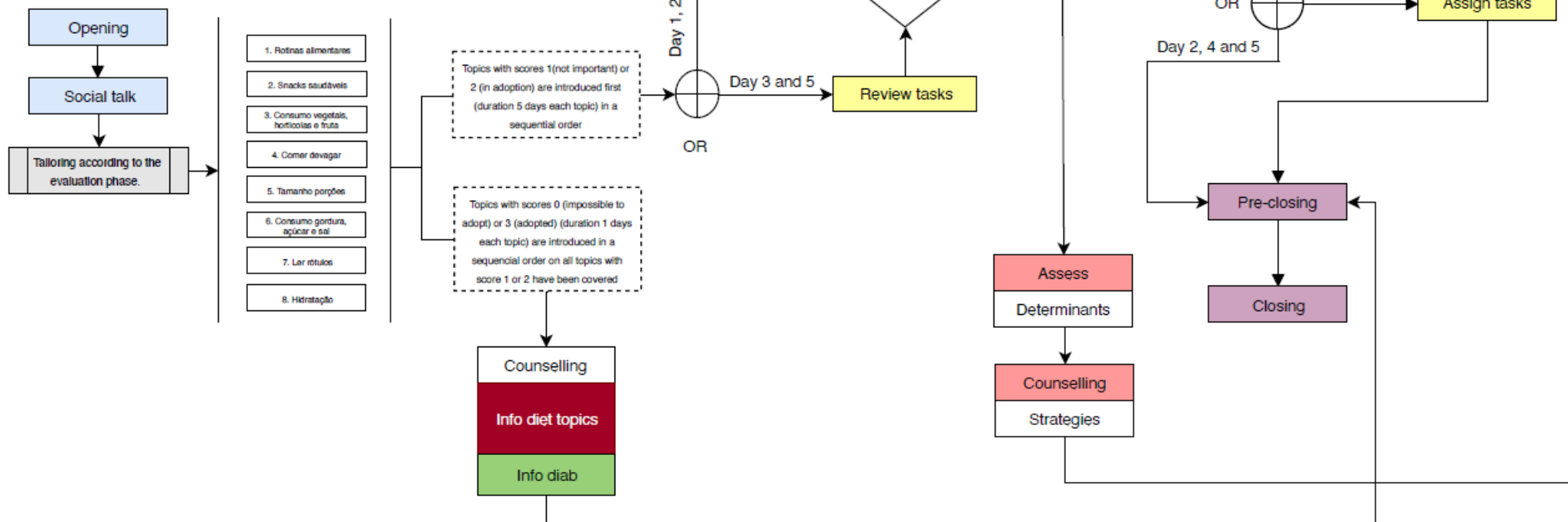


Figura 1: Arquitetura da Intervenção da Alimentação.

Informação importante: Na quinta coluna da Tabela 1, o texto que se encontra dentro de parêntesis após “Diálogo da Vitória” diz respeito ao nome do ficheiro Yarn no qual se encontra o diálogo apresentado abaixo. Nessa mesma coluna, após o diálogo, encontram-se entre parêntesis os nomes que identificam os nós de início e de fim (caso o diálogo esteja dividido por mais do que um nó), nos quais se encontram as frases do diálogo apresentado. As frases sublinhadas com a mesma cor pretendem salientar o facto de serem iguais ou bastante semelhantes, sendo que algumas delas se encontravam escritas mais do que uma vez (i.e., em mais do que um nó) nos ficheiros de diálogo analisados.

Tabela 1: Caracterização dos elementos de intervenção e respectivas BCTs e determinantes (componente da Alimentação).

Tópico	Etapa do diálogo	Determinante	BCT	Elemento	Workflow ¹
1. Rotinas alimentares	Review tasks		Self-monitoring of behavior (2.3)	Diálogo da Vitória (T1.yarn): “Em relação ao que combinámos sobre as refeições: consegui fazer pelo menos 3 refeições principais durante o dia de ontem?” (T1_RewTks_D3_P1)	IF Dia = 3
				Diálogo da Vitória (T1.yarn): “Consegui realizar pelo menos 3 refeições principais ontem?” (T1_RewTks_D5_P1)	IF Dia = 5
	Assess		Feedback on behavior (2.2)	Diálogo da Vitória (T1.yarn): “Muito bem! Assim vai controlar melhor a sua saúde.” (T1_Asse_D3_P2) “Ótimo!” (T1_Asse_D5_P2)	IF Dia = 3, 5 E Foi aderente ao objetivo
				Diálogo da Vitória (T1.yarn + F2determinantes_diet.yarn): “Tinha sido importante. Vamos então conversar um pouco para que possa alcançar esta meta. Porque é que não consegui implementar o número de refeições que combinámos?”(T1_Asse_D3_P3 + 1_F2_P1_2)	IF Dia = 3 E Não foi aderente ao objetivo
	Counselling	“Não gosto de preparar refeições só para mim.”	Problem solving (1.2) + Social support (unspecified) (3.1) + Information about health consequences (5.1) + Instruction on how to perform a behavior (4.1)	Diálogo da Vitória (F2determinantes_diet.yarn): “Como não gosta de cozinhar só para si, sugiro que escolha um dia da semana para confeccionar várias refeições. Já pensou em inscrever-se num centro de dia, que pode levar as refeições a sua casa? Ocasionalmente, pode ainda optar por comer fora ou adquirir refeições prontas e levá-las para casa. Considere convidar amigos ou familiares para comerem consigo ou visite-os ocasionalmente. Para manter valores de glicemia estáveis, é importante distribuir a ingestão de hidratos de carbono ao longo do dia, sem saltar refeições. Saltar refeições leva muitas vezes a que se coma demasiado na refeição a seguir. E pode, com alguns medicamentos para tratar a diabetes, levar a que o açúcar do sangue baixe demasiado. Pergunte aos seus profissionais de saúde se é o caso dos seus medicamentos para a diabetes.” (1_F2_P1_10 – 1_F2_P1_14)	IF Dia = 3, 5 E Não foi aderente ao objetivo (depende do determinante de não adesão)
		“Não tenho motivação para cozinhar.”	Problem solving (1.2) + Information about health consequences (5.1) + Instruction on how to perform a behavior (4.1)	Diálogo da Vitória (F2determinantes_diet.yarn): “Ocasionalmente, pode optar por comer fora ou adquirir refeições prontas e levá-las para casa. Já pensou em inscrever-se num centro de dia, que pode levar as refeições a sua casa? Para manter valores de glicemia estáveis, é importante distribuir a ingestão de hidratos de carbono ao longo do dia, sem saltar refeições. Saltar refeições leva muitas vezes a que se coma demasiado na refeição a seguir. E pode, com alguns medicamentos para tratar a diabetes, levar a que o açúcar do sangue baixe demasiado. Pergunte aos seus profissionais de saúde se é o caso dos seus medicamentos para a diabetes.” (1_F2_P1_9_1 – 1_F2_P1_14)	
		“É difícil mudar de hábitos.”	Problem solving (1.2) + Instruction on how to perform a behavior (4.1) + Information	Diálogo da Vitória (F2determinantes_diet.yarn): “De forma a não introduzir muitas alterações no seu dia-a-dia, sugiro que comece por fazer mais uma refeição do que o habitual. É recomendado	

¹ Com base na Arquitetura de Intervenção (Figura 1).

			about health consequences (5.1)	realizar 3 refeições principais durante o dia: pequeno-almoço, almoço e jantar, com horários regulares. Deve fazer pequenos lanches entre as refeições principais. Vamos falar sobre os lanches noutra altura. Para manter valores de glicemia estáveis, é importante distribuir a ingestão de hidratos de carbono ao longo do dia, sem saltar refeições. Saltar refeições leva muitas vezes a que se coma demasiado na refeição a seguir. E pode, com alguns medicamentos para tratar a diabetes, levar a que o açúcar do sangue baixe demasiado. Pergunte aos seus profissionais de saúde se é o caso dos seus medicamentos para a diabetes.” (1_F2_P1_8 – 1_F2_P1_14)	
	“Não gosto de fazer refeições sozinho(a).”		Social support (unspecified) (3.1) + Problem solving (1.2) + Instruction on how to perform a behavior (4.1) + Information about health consequences (5.1)	Diálogo da Vitória (F2determinantes_diet.yarn): “Considere convidar amigos ou familiares para comerem consigo ou visite-os ocasionalmente. Já pensou em inscrever-se num centro de dia, que pode levar as refeições a sua casa? Para manter valores de glicemia estáveis, é importante distribuir a ingestão de hidratos de carbono ao longo do dia, sem saltar refeições. Saltar refeições leva muitas vezes a que se coma demasiado na refeição a seguir. E pode, com alguns medicamentos para tratar a diabetes, levar a que o açúcar do sangue baixe demasiado. Pergunte aos seus profissionais de saúde se é o caso dos seus medicamentos para a diabetes.” (1_F2_P1_1_1 – 1_F2_P1_14)	
	Assign tasks		Goal setting (behavior) (1.1) + Action planning (1.4)	Diálogo da Vitória (T1.yarn): “Então, compromete-se a realizar a partir de amanhã pelo menos 3 refeições principais por dia?” (T1_AssgTks_D1_P1)	IF Dia = 1
2. Snacks saudáveis	Review tasks		Self-monitoring of behavior (2.3)	Diálogo da Vitória (T2.yarn): “Em relação ao que combinámos sobre os lanches entre as refeições principais: consegui fazer pelo menos um lanche durante o dia de ontem?” (T1_RewTks_D3_P1)	IF Dia = 3
				Diálogo da Vitória (T2.yarn): “Conseguiu realizar pelo menos um lanche entre as refeições principais ontem?”	IF Dia = 5
	Assess		Feedback on behavior (2.2)	Diálogo da Vitória (T2.yarn): “Muito bem! Assim vai controlar melhor a sua saúde.” (T1_Asse_D3_P2) “Ótimo!” (T1_Asse_D5_P2)	IF Dia = 3, 5 E Foi aderente ao objetivo
				Diálogo da Vitória (T2.yarn + F2determinantes_diet.yarn): “Tinha sido importante. Vamos então conversar um pouco para que possa alcançar esta meta. Porque é que considera que tem sido difícil implementar lanches saudáveis?” (T1_Asse_D3_P3 + 1_F2_P2_2)	IF Dia = 3 E Não foi aderente ao objetivo
	Counselling	“Não tenho apetite para fazer tantas refeições.”		Problem solving (1.2) + Instruction on how to perform a behavior (4.1)	Diálogo da Vitória (F2determinantes_diet.yarn): “Uma estratégia que lhe pode permitir fazer mais refeições é comer porções mais pequenas de cada vez. Ao almoço e ao jantar, experimente comer num prato mais pequeno, uma vez que temos tendência a enchê-lo. Nestas refeições, preencha metade do prato com legumes e hortícolas e um quarto com uma fonte de proteína, como carne ou peixe. O resto do prato pode ser preenchido com hidratos de carbono, como massa, arroz ou batata. Ao ingerir porções menores às refeições principais, poderá conseguir mais facilmente fazer lanches entre elas. Um lanche saudável é, por exemplo, uma

				<p>porção de fruta, como uma maçã ou duas clementinas, uma cenoura, ou um iogurte magro ou um copo de leite magro. Para controlar os valores de glicemia, é importante fazer refeições intermédias com alimentos saudáveis. A introdução de lanches entre as refeições ajuda a controlar a fome e a ingerir porções menores nas refeições principais.” (1_F2_P2_10 – 1_F2_P2_13_1)</p>	
		<p>“Não tenho motivação para preparar lanches.”</p>	<p>Problem solving (1.2) + Instruction on how to perform a behavior (4.1)</p>	<p>Diálogo da Vitória (F2determinantes_diet.yarn): “Pode optar por lanches que não requerem preparação. Acredito que o consiga fazer! Pode escolher alimentos que não requerem preparação como um iogurte magro ou uma fruta fresca que não precise de ser preparada. Pode ainda escolher fruta desidratada sem adição de açúcar. Pode começar por introduzir mais um lanche que o habitual. Quando já fizer parte da sua rotina tente introduzir outro. Para controlar os valores de glicemia, é importante fazer refeições intermédias com alimentos saudáveis. A introdução de lanches entre as refeições ajuda a controlar a fome e a ingerir porções menores nas refeições principais.” (1_F2_P2_4_1 – 1_F2_P2_13_1)</p>	
		<p>“É difícil mudar hábitos.”</p>	<p>Information about health consequences (5.1) + Instruction on how to perform a behavior (4.1)</p>	<p>Diálogo da Vitória (F2determinantes_diet.yarn): “Ter sempre consigo alguma comida saudável, como fruta desidratada ou fresca, pode tornar mais fácil comer entre as refeições principais. Apesar de ser difícil, acredito que ao longo do tempo seja capaz de mudar os seus hábitos.” (1_F2_P2_12_2 – 1_F2_P2_12_3)</p>	
		<p>“Tenho dificuldade em deslocar-me para fazer compras.”</p>	<p>Problem solving (1.2) + Social support (unspecified) (3.1)</p>	<p>Diálogo da Vitória (F2determinantes_diet.yarn): “Considere pedir a um familiar ou amigo que adquira esses produtos por si. Pode também pedir que entreguem os produtos desejados em sua casa. Poderão haver pequenas lojas ou grandes superfícies na sua área de residência que disponham desse serviço. Também pode tentar procurar recursos na comunidade. Na sua zona de residência podem existir associações ou grupos de voluntários que podem ajudar. Procure aconselhamento na sua junta de freguesia. Pode também falar com o enfermeiro na sua unidade de saúde, que poderá ajudar a encontrar a melhor solução. Para controlar os valores de glicemia, é importante fazer refeições intermédias com alimentos saudáveis. A introdução de lanches entre as refeições ajuda a controlar a fome e a ingerir porções menores nas refeições principais.” (1_F2_P2_7_2 – 1_F2_P2_13_1)</p>	
Assign tasks			<p>Goal setting (behavior) (1.1) + Action planning (1.4)</p>	<p>Diálogo da Vitória (T2.yarn): “Então, compromete-se a realizar a partir de amanhã pelo menos um lanche saudável entre as refeições principais? (Excelente!/Vamos tentar! Comece por fazer um lanche a meio da manhã ou então entre o almoço e o jantar. Escolha um período que lhe dê mais jeito. Pode ser?) E não se esqueça que deve ser constituído por fruta ou lacticínios.” (T1_AssgTks_D1_P1 – Node593)</p>	IF Dia = 1

Apêndice B

Caracterização dos elementos de intervenção da componente de Atividade Física

Neste apêndice encontra-se uma versão reduzida da tabela resultante da análise dos diálogos de determinadas etapas da componente de Atividade Física, incluída na intervenção de mudança comportamental do projeto VASelfCare.

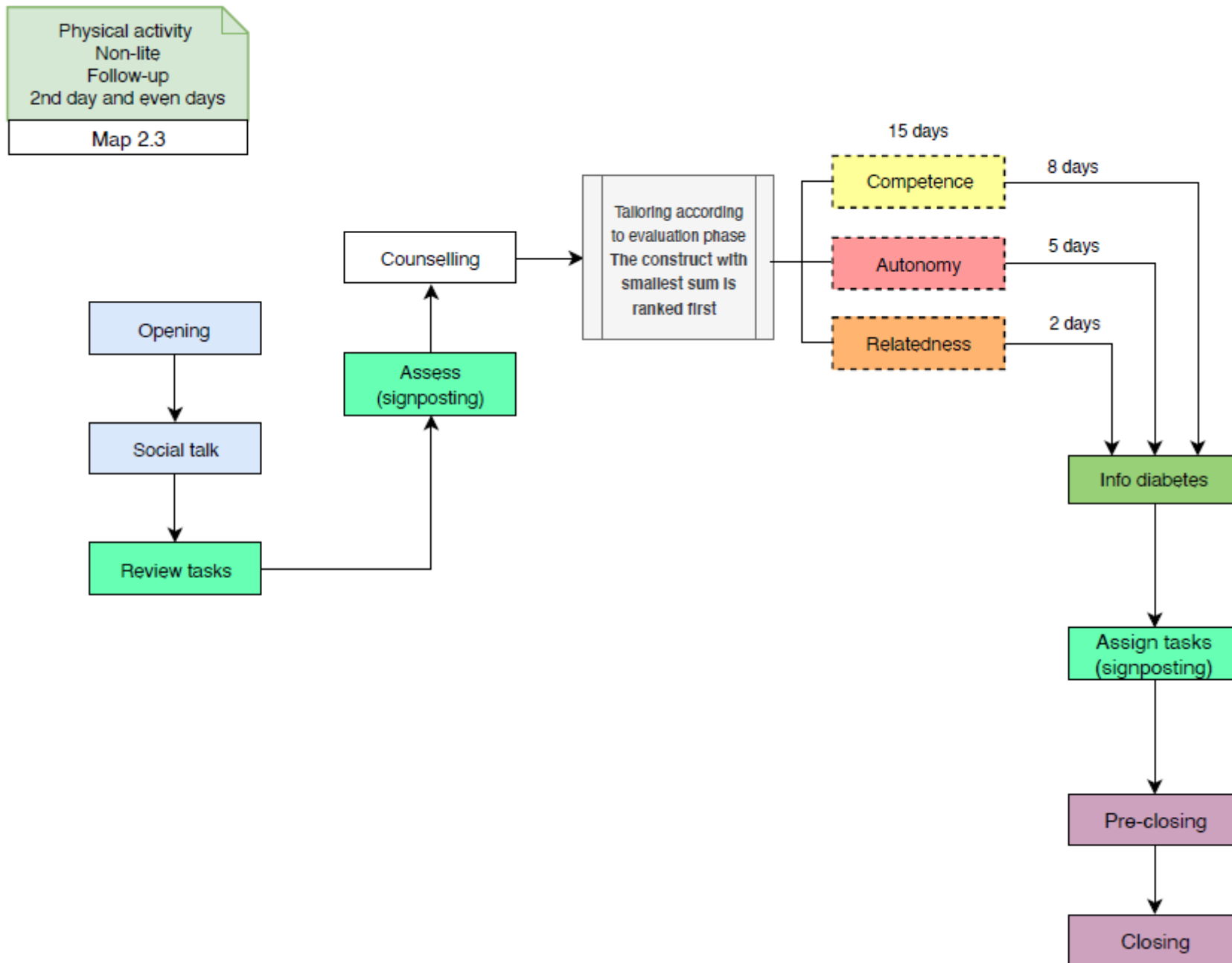


Figura 1: Arquitetura da Intervenção da Atividade Física para dias pares.

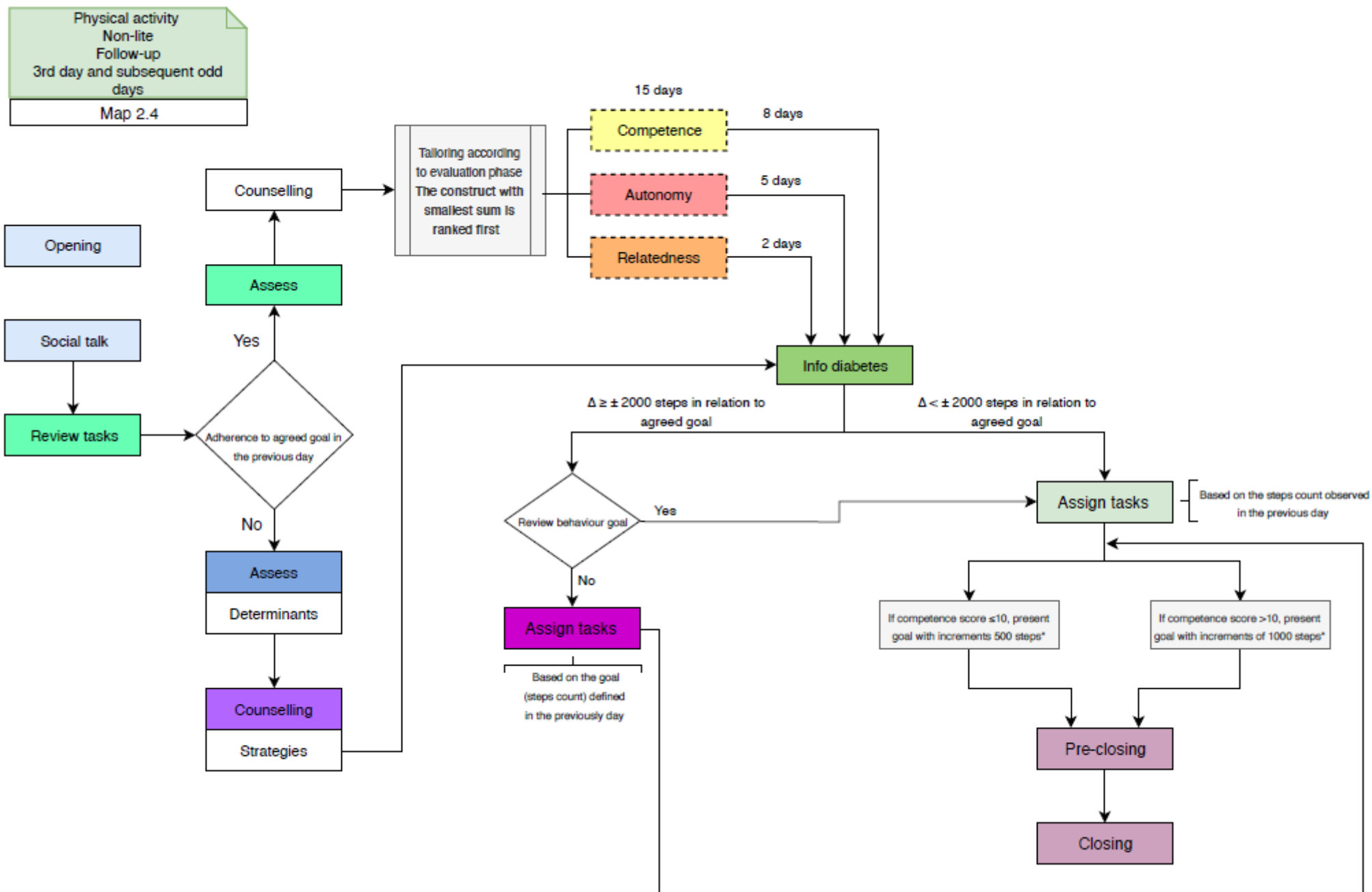


Figura 2: Arquitetura da Intervenção da Atividade Física para dias ímpares.

Informação importante: Na terceira coluna da Tabela 1 e na quarta coluna da Tabela 2, o texto que se encontra dentro de parêntesis após “Diálogo da Vitória” diz respeito ao nome do ficheiro Yarn no qual se encontra o diálogo apresentado abaixo. Nessas mesmas colunas, após o diálogo, encontram-se entre parêntesis os nomes que identificam os nós de início e de fim (caso o diálogo esteja dividido por mais do que um nó), nos quais se encontram as frases do diálogo apresentado. As frases sublinhadas com a mesma cor pretendem salientar o facto de serem iguais ou bastante semelhantes, sendo que algumas delas se encontravam escritas mais do que uma vez (i.e., em mais do que um nó) nos ficheiros de diálogo analisados.

Tabela 1: Caracterização dos elementos de intervenção e respetivas BCTs (componente de Atividade Física).

Etapa do diálogo	BCT	Elemento	Workflow ¹
Review tasks	Self-monitoring of behavior (2.3)	Diálogo da Vitória (Reviewtasks_D1.yarn): “Para começarmos, gostava que consultasse o dispositivo onde estão registados o seu número de passos. Pode ser? Registe a média de passos dos últimos 7 dias.” (F2P1_AF & F2P2_AF)	IF Dia =1
		Diálogo da Vitória (Reviewtasks _ Assess _ Assigntasks_D2 (Dpares) + D3 (Dimpares).yarn): “Pode indicar-me o número de passos que andou ontem? Procure no seu dispositivo e insira a contagem de passos referente ao dia de ontem. Para conseguir ajudar a ser uma pessoa mais ativa, é fundamental monitorizar o seu número de passos. Em qualquer momento depois dos nossos diálogos, pode consultar o número de passos que caminha por dia. Esta informação está disponível no menu ver os meus dados, na parte que respeita ao seu desempenho no menu principal.” (Node33 – Node37)	IF Dia = par
		Diálogo da Vitória (Reviewtasks _ Assess _ Assigntasks_D2 (Dpares) + D3 (Dimpares).yarn): “Ora vamos ver o seu número de passos ontem? Então veja no seu dispositivo e registe-os. Obrigado! É importante irmos mantendo estes registos e ver a sua evolução.” (F2P1_AF – F2P3_AF)	IF Dia = ímpar (exceto 1)
Assess	Feedback on behavior (2.2)	Diálogo da Vitória (Reviewtasks _ Assess _ Assigntasks_D2 (Dpares) + D3 (Dimpares).yarn): “Hoje não vamos discutir o número de passos que caminhou. Vamos fazê-lo amanhã!” (Node38)	IF Dia = par
		Diálogo da Vitória (Reviewtasks _ Assess _ Assigntasks_D2 (Dpares) + D3 (Dimpares).yarn): “Vou apresentar-lhe o gráfico com o número de passos registado. Cada ponto representa o número de passos que caminhou na data indicada. As pessoas que caminham por norma menos que 3000 passos por dia têm maior risco de doenças do coração, por isso representámos a vermelho no gráfico. Quando terminarmos de falar também pode consultar o seu número de passos no menu “Ver os meus dados”. (F2P5_AF – F2P4_AF)	IF Dia = ímpar (exceto 1)
Counselling	Problem solving (1.2)	Diálogo da Vitória (Competence.yarn): “Muitas pessoas acham que fazer atividade física tem que ser num ginásio com um instrutor. Esta ideia é redutora. A atividade física inclui todos os movimentos corporais que impliquem gastar mais energia do que aquela que dependemos em repouso. Por isso, a atividade física está presente no dia-a-dia das pessoas e inclui atividades tão simples como passear o cão, fazer as tarefas domésticas e ir às compras. Não implica necessariamente frequentar um ginásio ou praticar uma modalidade desportiva.” (Node20 – Node24) (D1)	IF Dia = par OU IF Dia = ímpar E aderente ao objetivo de passos; Duração total de 8 dias.
	Social support (unspecified) (3.2)	Diálogo da Vitória (Relatedness.yarn): “Fazer caminhadas sozinho pode ser aborrecido para algumas pessoas. (É ótimo que retire prazer das suas caminhadas.) Caminhar com um familiar ou até um grupo de amigos pode ser muito mais entusiasmante! Que tal desafiar um amigo ou vários para se juntarem a si e realizarem uma caminhada? (Compreendo mas poderá experimentar!) Caso os convença depois poderá repetir a experiência mais vezes e até definir uma periodicidade	IF Dia = par OU IF Dia = ímpar E aderente ao objetivo de passos;

¹ Com base na Arquitetura de Intervenção (Figuras 1 e 2).

		semanal. Repare que iria ter companhia e ainda estaria a contribuir para proporcionar benefícios para a saúde de quem o acompanhar.” (Node1 – Node4) (D1)	Duração total de 2 dias.
Assign tasks	Goal setting (behavior) (1.1) E Action planning (1.4) (codificadas em conjunto)	Diálogo da Vitória (Assigntasks_D1.yarn): “Então agora vamos combinar o número de passos que se propõe caminhar amanhã? Das sugestões apresentadas, qual é o número de passos que quer definir como objetivo para amanhã?” (Node6 & Node2)	IF Dia = 1
		Diálogo da Vitória (Assign tasks _ Δ≥2000 .yarn): (* “Teve um aumento muito significativo do seu número de passos. Fantástico!” (Start) (* “Caminhou mais passos do que o acordado. Bom trabalho!” (Node20)	IF Dia = ímpar E aderente ao objetivo de passos
		Diálogo da Vitória (Assign tasks _ Δ≥2000 .yarn): (* “Vejo que está longe do seu objetivo.” (Node3) (* “Ficou aquém do seu objetivo. Estou aqui para ajudar!” (Node23)	IF Dia = ímpar E não aderente ao objetivo de passos

Tabela 2: Caracterização dos elementos de intervenção e respectivas BCTs e determinantes (componente de Atividade Física).

Etapa do diálogo	Determinante	BCT	Elemento	Workflow ²
Counselling	“Não tenho motivação.”	Problem solving (1.2)	Diálogo da Vitória (DeterminantesAF.yarn): “A desmotivação é um fator apontado por muitas pessoas para não praticarem atividade física. Sabia? Experimente começar por realizar alguma atividade de que goste e lhe dê prazer. Frequentemente, depois de experimentar as primeiras vezes, as pessoas sentem mais vontade de continuar. Força! Verá que irá conseguir.” (F2_strat3_1 – F2_strat3_4)	IF Dia = ímpar E não aderente ao objetivo de passos (depende do determinante de não adesão)
	“Sinto-me desconfortável a caminhar na rua.”	Problem solving (1.2)	Diálogo da Vitória (DeterminantesAF.yarn): “Atualmente, existem campanhas para inspirar as pessoas a praticar atividade física, mostrando que fazer exercício pode ser fácil, acessível e entusiasmante. Uma grande parte da população adulta portuguesa tem dificuldades em integrar atividade física no seu dia-a-dia por diversas razões. Repare nas campanhas que surgem na televisão! Acho que podem ajudar a perceber que é transversal a todas as pessoas e que pode ser também importante para si. Em atividades distintas do dia-a-dia podemos aumentar a contagem do número de passos e tornarmo-nos fisicamente ativos.” (F2_strat7_18 – F2_strat7_21)	
	“Não me sinto capaz.”	Problem solving (1.2) + Instruction on how to perform the behavior (4.1)	Diálogo da Vitória (DeterminantesAF.yarn): “Atualmente, existem campanhas para inspirar as pessoas a praticar atividade física, mostrando que fazer exercício pode ser fácil, acessível e entusiasmante. Uma grande parte da população adulta portuguesa tem dificuldades em integrar a prática de atividade física no seu dia-a-dia por diversas razões. Repare nas campanhas que surgem na televisão! A prática de atividade física é importante em todas as idades. E não se restringe à prática de uma modalidade desportiva. Pode ser integrando pequenas modificações no seu dia-a-dia! Por exemplo, pode utilizar as escadas sempre que possível. Estabelecer trajetos mais longos que promovam a marcha. Por exemplo, escolher o caminho mais longo para ir ao café. Estou confiante que irá conseguir!” (F2_strat8_1 – F2_strat8_8)	

² Com base na Arquitetura de Intervenção (Figuras 1 e 2).

		<p>Problem solving (1.2) + Social support (unspecified) (3.1)</p>	<p>Diálogo da Vitória (DeterminantesAF.yarn): “Muitas pessoas sentem que também não são capazes de serem fisicamente ativas. Contudo, é importante perceber a razão que gera esse sentimento. Acha que não é capaz de aumentar a contagem do número de passos porquê? É compreensível. Há alguém na sua família ou grupo de amigos que faça caminhadas?” (F2_strat8_17 – F2_strat8_19) (Sim.) -> “Boa! Que tal fazer caminhadas com o seu familiar ou amigo? Experimente! Pode ser que fique com vontade de o fazer mais vezes. Também pode procurar locais onde habitualmente as pessoas façam caminhadas e sentirá que não está só.” (F2_strat8_20 – F2_strat8_26) (Não.) -> “Existem alguns grupos de caminhada na zona onde mora?” (F2_strat8_21) (Sim.) -> “Que tal juntar-se para realizar caminhadas? Experimente! Pode ser que fique com vontade de o fazer mais vezes. Também pode procurar locais onde habitualmente as pessoas façam caminhadas e sentirá que não está só.” (F2_strat8_22 – F2_strat8_26) (Não.) -> “Também pode procurar locais onde habitualmente as pessoas façam caminhadas e sentirá que não está só. Experimente! Pode ser que fique com vontade de o fazer mais vezes.” (F2_strat8_23 – F2_strat8_24)</p>	
		<p>Problem solving (1.2)</p>	<p>Diálogo da Vitória (DeterminantesAF.yarn): “Muitas pessoas sentem que também não são capazes de serem fisicamente ativas. Contudo, é importante perceber a razão que gera esse sentimento. Acha que não é capaz de aumentar a contagem do número de passos porquê? A desmotivação é um fator apontado por muitas pessoas para não praticarem atividade física. Sabia? Experimente começar a praticar alguma atividade que goste e lhe dê prazer. Frequentemente, depois de experimentar as primeiras vezes, as pessoas sentem vontade de continuar. Para praticar atividade física não é preciso estar em forma. Nem é preciso equipamento especial. Pequenas atividades podem fazer a diferença! Força! Verá que irá conseguir.” (F2_strat8_17 – F2_strat8_40)</p>	
	<p>“Não tenho companhia.”</p>	<p>Problem solving (1.2) + Social support (practical) (3.2)</p>	<p>Diálogo da Vitória (DeterminantesAF.yarn): “É compreensível. Há alguém na sua família ou grupo de amigos que faça caminhadas? Existem alguns grupos de caminhada na zona onde mora? (Que tal juntar-se para realizar caminhadas? Experimente! Pode ser que fique com vontade de o fazer mais vezes.) Também pode procurar locais onde habitualmente as pessoas façam caminhadas e sentirá que não está só.” (F2_strat11_4 – F2_strat11_8)</p>	
		<p>Problem solving (1.2) + Social support (practical) (3.2)</p>	<p>Diálogo da Vitória (DeterminantesAF.yarn): “É compreensível. Há alguém na sua família ou grupo de amigos que faça caminhadas? Boa! Que tal fazer caminhadas com o seu familiar ou amigo? Também pode procurar locais onde habitualmente as pessoas façam caminhadas e sentirá que não está só. Experimente! Pode ser que fique com vontade de o fazer mais vezes.” (F2_strat11_4 – F2_strat11_11)</p>	

Apêndice C

Questionário - Avaliação de risco de Diabetes Tipo 2

Neste anexo encontra-se o questionário disponibilizado pela DGS para avaliação do risco de vir a ter Diabetes Tipo 2 dentro de 10 anos. Este questionário foi extraído da brochura “Será que posso vir a ter Diabetes? Aprenda a cuidar de si!”, do material de campanha¹ do Programa Nacional para a Diabetes.

¹<https://www.dgs.pt/programa-nacional-para-a-diabetes/material-de-campanha.aspx>

Avaliação de risco de Diabetes tipo 2

Assinalar com uma cruz a resposta e somar o total de pontos no final.

1. Idade

- 0 p. Menos de 45 anos
- 2 p. 45-54 anos
- 3 p. 55-64 anos
- 4 p. Mais de 64 anos

2. Índice de Massa Corporal (Peso (Kg)/Alt² (mtl))

- 0 p. Menos de 25 kg/m²
- 1 p. 25-30 kg/m²
- 3 p. Mais de 30 kg/m²

3. Medida da cintura (normalmente ao nível do umbigo)

HOMENS

MULHERES

- | | |
|--|--|
| <input type="checkbox"/> 0 p. Menos de 94 cm | <input type="checkbox"/> 0 p. Menos de 80 cm |
| <input type="checkbox"/> 3 p. 94-102 cm | <input type="checkbox"/> 3 p. 80-88 cm |
| <input type="checkbox"/> 4 p. Mais de 102 cm | <input type="checkbox"/> 4 p. Mais de 88 cm |

4. Prática, diariamente, actividade física pelo menos durante 30 minutos, no trabalho ou durante o tempo livre (incluindo actividades da vida diária)?

- 0 p. Sim
- 2 p. Não

5. Com que regularidade come vegetais e/ou fruta (sopa, salada, legumes cozidos, entre outros)?

- 0 p. Todos os dias
- 1 p. Às vezes

6. Toma regularmente ou já tomou alguns medicamentos para a Hipertensão Arterial?

- 0 p. Não
- 2 p. Sim

7. Alguma vez teve açúcar elevado no sangue (ex.: num exame de saúde, durante um período de doença ou durante a gravidez)?

- 0 p. Não
- 5 p. Sim

8. Tem algum membro de família próximo ou outros familiares a quem foi diagnosticado diabetes (Tipo 1 ou Tipo 2)?

- 0 p. Não
- 3 p. Sim: avós, tias, tios ou primos em 1º grau (excepto pais, irmãos, irmãs ou filhos)?
- 5 p. Sim: pais, irmãos, irmãs ou filhos

Nível de Risco total

O Risco de vir a ter Diabetes Tipo 2 dentro de 10 anos é:

- **< 7 Baixo:** calcula-se que 1 em 100 pessoas desenvolverá a doença
- **7-11 Ligeiro:** calcula-se que 1 em 25 desenvolverá a doença
- **12-14 Moderado:** calcula-se que 1 em 6 desenvolverá a doença
- **15-20 Alto:** calcula-se que 1 em 3 desenvolverá a doença;
- **> 20 Muito alto:** calcula-se que 1 em 2 desenvolverá a doença.

Apêndice D

Integração com o Unity

Das ferramentas de criação de diálogo estudadas, decidiu-se aprofundar os conhecimentos acerca do Watson Assistant e da FATiMA Toolkit, uma vez que são ferramentas pouco ou nada usadas neste contexto e que demonstram bastante potencial. Foi então aprofundado o nível de conhecimento de ambas as ferramentas, e partiu-se para a familiarização com a integração das mesmas com ambientes de desenvolvimento de humanos virtuais, neste caso com o Unity.

D.1 Watson Assistant

Com base num tutorial¹ de como criar um assistente virtual com o Watson Assistant e o Unity, estudou-se a forma como é feita a integração entre as duas ferramentas.

Além do Watson Assistant, tirou-se também partido do serviço Text To Speech², igualmente fornecido pela IBM Cloud, e do sistema Unity Multipurpose Avatar (UMA)³, disponível gratuitamente na Unity Asset Store. O tutorial indicava ainda o uso do serviço Speech To Text, disponível na IBM Cloud, para fazer o reconhecimento da voz do utilizador, no entanto, optou-se por fazer a leitura de um *input* textual que é posteriormente enviado para o Watson Assistant. Por fim, era ainda aconselhado o uso do sistema SALSA LipSync, disponível na Unity Asset Store, no entanto, sendo esta uma ferramenta paga, o seu uso foi descartado.

Para o desenvolvimento deste pequeno projeto, criou-se uma instância do serviço Watson Assistant e outra do serviço Text To Speech, ambos na IBM Cloud. No Watson Assistant exploraram-se alguns dos *Intents* existentes no catálogo, nomeadamente "General_Greetings", "General_Jokes" e "General_Ending". O catálogo disponibiliza um vasto número de Intenções já treinadas em perguntas que os utilizadores costumam fazer. Na Figura D.1 estão visíveis alguns dos exemplos de *user input* da intenção "General_Jokes".

¹<https://developer.ibm.com/recipes/tutorials/create-a-3d-digital-human-with-ibm-watson-assistant-and-unity3d/>

²<https://www.ibm.com/cloud/watson-text-to-speech>

³<https://assetstore.unity.com/packages/3d/characters/uma-2-unity-multipurpose-avatar-35611>

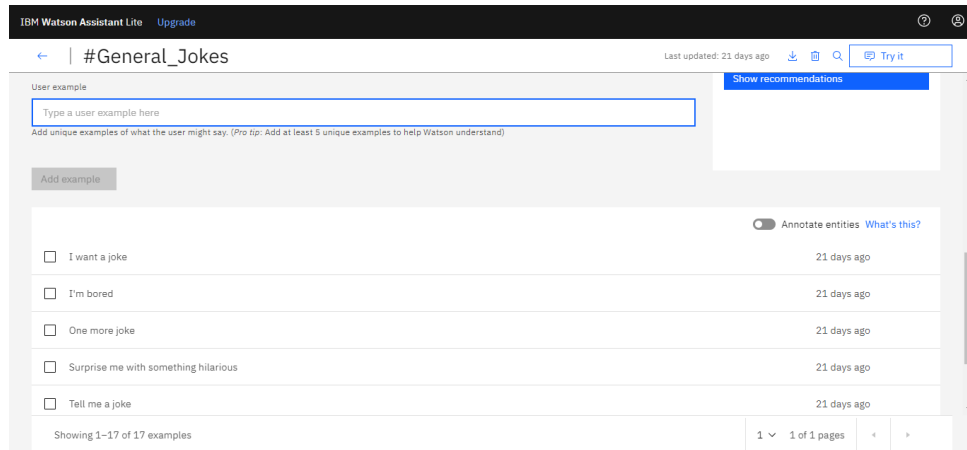


Figura D.1: Exemplo de personalização de uma Intenção.

No menu *Dialog*, acrescentam-se nós à árvore lógica conforme for necessário. Neste caso, adicionou-se um nó para reconhecer a intenção "General_Jokes", estando o seu conteúdo visível na Figura D.2. Quando o assistente reconhece alguma das frases declaradas como exemplo, ele dá uma resposta textual. O mesmo processo foi efetuado para a restante parte da conversa, na qual o assistente procede para o conto de uma anedota, ou não, dependendo da resposta positiva ou negativa do utilizador à pergunta "Seriously, you want me to tell you a joke?".

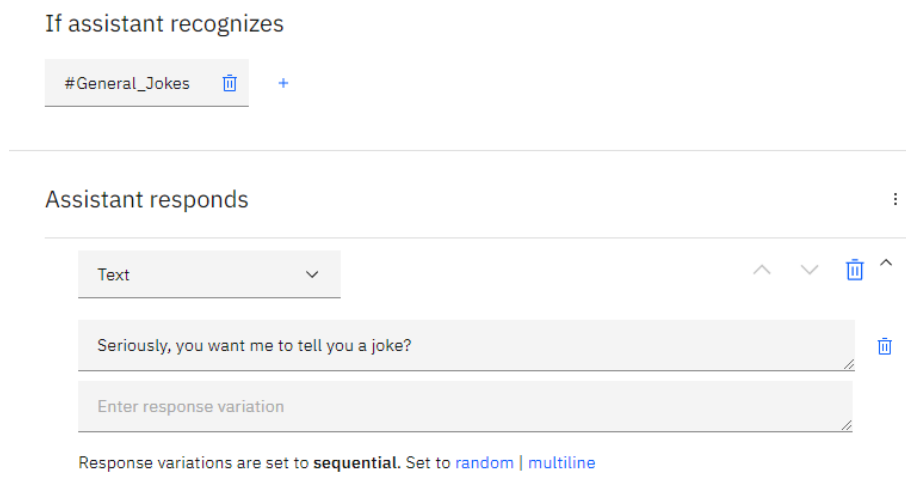


Figura D.2: Exemplo de personalização de um nó de diálogo.

Terminada a modelação do serviço de diálogo, é feita a sua integração no Unity, no qual é necessário um *script* que receba os dados necessários para fazer a autenticação dos sistemas através das respetivas *API keys* e URLs de serviço. Posto isto, preparou-se a cena, introduzindo um humano virtual básico com recurso ao UMA, e uma caixa de texto que recebe a fala do utilizador. O aspeto da interface final pode ser visto na Figura D.3.

Na Figura D.1 é possível verificar que a frase "I'm bored" faz parte dos exemplos de *user input* da intenção "General_Jokes". A Figura D.3 mostra essa mesma frase escrita na



Figura D.3: Exemplo de *input* textual, seguido da cena em que o assistente responde, acompanhado de fala.

caixa de texto, introduzida pelo utilizado. Quando o utilizador clica na tecla Enter para enviar a frase para o sistema e o assistente a reconhece, a resposta a dar é "Seriously, you want me to tell you a joke?", como se pode entender pela Figura D.2. A resposta do assistente virtual pode ser ouvida, graças ao serviço Text To Speech, e aparece também escrita abaixo da caixa de texto, como se pode ver na Figura D.3.

D.2 FAtiMA Toolkit

Para a familiarização com a FAtiMA Toolkit e a sua integração com o Unity, tirou-se partido do Starter-Kit⁴ disponibilizado pelos criadores da ferramenta de forma a facilitar a sua compreensão e aprendizagem.

O cenário incluído no Starter-Kit tinha um diálogo bastante curto, apenas com duas interações. Assim, recorrendo à *Authoring Tool*, adicionaram-se mais algumas interações, seguindo o exemplo apresentado na Figura D.4. A cada interação foram atribuídos diferentes *Meanings* - Neutro, Feliz ou Triste - consoante a conotação emocional da frase.

Tendo o fluxo de diálogo terminado, geraram-se os respetivos áudios através da funcionalidade Text-To-Speech presente na FAtiMA Toolkit. Na Figura D.5 está representada a interface da funcionalidade, na qual se pode escolher gerar apenas um diálogo ou todos os diálogos disponíveis no cenário, assim como escolher qual a voz pretendida e a velocidade com que dirá a frase.

Como ao gerar os diálogos se escolheu uma voz feminina, no Unity procedeu-se à alteração do humano virtual, que inicialmente tinha uma aparência masculina, para um avatar com aparência feminina, também disponibilizado nos *Assets* do Starter-Kit. O resultado final pode ser visto na Figura D.6, onde está visível o humano virtual, de nome Mary, e a sua fala, assim como as três opções de resposta entre as quais o utilizador pode escolher.

⁴<https://fatima-toolkit.eu/fatima-starter-kit-and-3-way-conversations/>

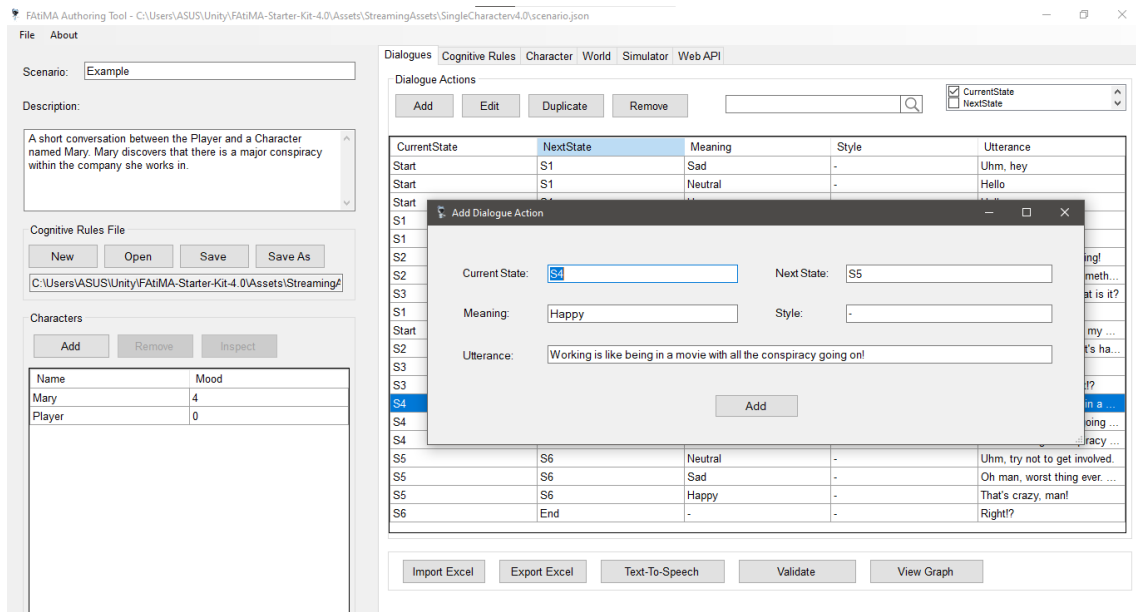


Figura D.4: Customização do diálogo através da *FAtiMA Authoring Tool*.

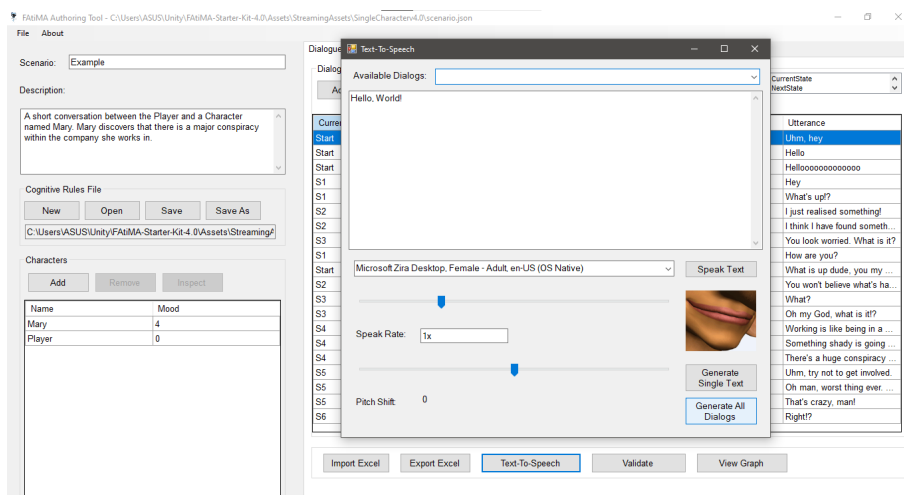


Figura D.5: Funcionalidade Text-To-Speech, presente na *FAtiMA Authoring Tool*.



Figura D.6: Interface do utilizador da experiência de integração da FATiMA Toolkit com o Unity.

Apêndice E

Escalas de autorrelato de resposta emocional

As emoções são um aspeto bastante importante da vida, sendo igualmente importantes no uso de produtos e de tecnologia. Assim sendo, é cada vez mais importante sermos capazes de avaliar essas emoções de forma a desenvolver produtos melhores, a entender melhor como o utilizador interage com os produtos, e também porque o estado afetivo do utilizador é importante para o próprio produto [Broekens and Brinkman, 2009].

O *feedback* afetivo do utilizador pode ser dividido em duas categorias: implícito e explícito. Os métodos implícitos detetam o comportamento do utilizador e deduzem uma emoção. Os métodos explícitos pedem ao utilizador para inserir o *feedback* afetivo diretamente [Broekens and Brinkman, 2009]. A forma mais simples de avaliar a experiência subjetiva é recorrer ao *feedback* explícito e perguntar diretamente ao indivíduo em que medida um determinado estímulo ou evento é agradável ou desagradável e qual a intensidade do estado emocional sentido [Ribeiro and Monteiro, 2015].

O *Self Assessment Manikin* – SAM, desenvolvido por Lang (1980) com cuidados psicométricos, é especialmente adequado para estudar a vertente experiencial da resposta emocional, uma vez que é livre de influências culturais e não dispensa a conduta verbal. Permite fazer a avaliação das três dimensões emocionais introduzidas por Mehrabian e Russel em 1974 – valência (*valence* ou *pleasure*), intensidade (*arousal*) e dominância (*dominance*) – através de três escalas pictográficas com figuras humanóides, como se pode visualizar na Figura E.1 [Bradley and Lang, 1994]. Esta é uma escala visual, normalmente utilizada em formato de papel e caneta, no entanto, existe também uma versão em computador, cujo acesso requer que seja feito um pedido aos seus autores.

A dimensão de valência avalia se o estímulo provocou um sentimento positivo ou negativo ao utilizador, deixando-o feliz ou triste; a dimensão de intensidade avalia a intensidade da emoção experienciada, de uma sensação calmante a entusiasmante; e por fim, a dimensão de dominância captura se o indivíduo se sente no controlo do ambiente ou não [Bradley and Lang, 1994].

[Broekens and Brinkman, 2009] apresentaram o AffectButton, um instrumento de me-

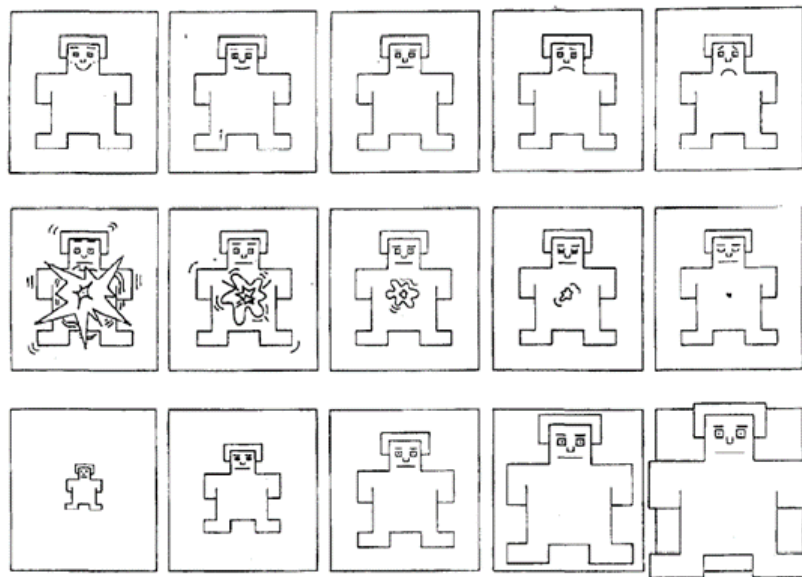


Figura E.1: O *Self-Assessment Manikin* (SAM), usado para classificar as dimensões afetivas de valência (linha do topo), intensidade (linha central), e dominância (linha de fundo) [Bradley and Lang, 1994].

didada que permite ao utilizador dar o seu *feedback* emocional de forma detalhada e subtil em relação às pessoas ou produtos. Movendo o rato/dedo (dependendo da plataforma utilizada, computador ou dispositivo móvel), o utilizador selecciona a expressão facial que melhor combina com o seu sentimento. Na Figura E.2 estão representadas algumas das expressões faciais que o utilizador pode escolher, consoante a localização do cursor. Após clicar no *AffectButton*, são gerados três valores, um para cada uma das dimensões afetivas, de forma semelhante ao que acontece com o SAM. Comparado a outros métodos de autorrelato, o *AffectButton* permite que o *input* afetivo tridimensional seja feito com apenas um clique e fornece um *output* válido e confiável, tornando-se um instrumento útil para reunir informações afetivas.



Figura E.2: Exemplos de várias expressões faciais a escolher pelo utilizador ao usar o *AffectButton* [Broekens and Brinkman, 2009].

As dimensões de valência e de intensidade são consideradas primárias, sendo estas duas dimensões normalmente responsáveis pela maior parte da variação nos julgamentos emocionais, incluindo quando o diferencial semântico é o instrumento de medição [Bradley and Lang, 1994]. Posto isto, a dominância é várias vezes deixada de lado nos estudos académicos [Hall et al., 2017].

Grande parte dos estudos realizados com recurso a este tipo de escalas compreendia uma amostra de imagens (estímulos) às quais os indivíduos eram expostos, fornecendo depois o seu *feedback* afetivo em relação a cada uma delas. Mas e durante o uso de uma aplicação, o que é que pode ser percebido como dominância? [Hall et al., 2017] conduziu um estudo sobre o uso do modelo PAD (*Pleasure-Arousal-Dominance*) para explicar as atitudes e intenções de uso do Facebook. Inicialmente é debatida a inclusão ou não da dimensão de dominância no estudo e, após decidir mantê-la, é explorado aquilo que pode ser percebido como dominância para os utilizadores durante o uso do Facebook, enumerando várias hipóteses.

Mais tarde, [Betella and Verschure, 2016] propuseram uma nova escala de autorrelato para medição da resposta emocional, denominada Affective Slider (AS). O AS é uma ferramenta digital de autorrelato composta por dois *sliders* que medem a valência e a intensidade numa escala contínua, retirando da escala a dimensão da dominância. Abaixo dos *sliders* encontram-se dois triângulos isósceles, simetricamente espelhados, que servem como uma dica visual para a intensidade da resposta emocional. De cada um dos lados dos *sliders* encontram-se *emojicons*, de forma a representar visualmente os estados afetivos bipolares das escalas de emoção de Mehrabian e Russell: sonolento ou bem acordado para a intensidade, e infeliz e feliz para a valência. O AS não requer instruções escritas e é intencionalmente exibido usando uma paleta cromática neutra, de forma a evitar distorções nas avaliações devido à conotação emocional das cores (Figura E.3).

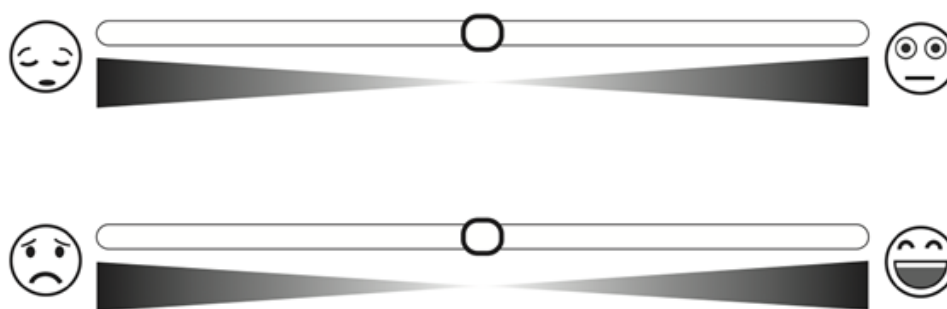


Figura E.3: O Affective Slider, uma ferramenta digital de autorrelato constituída por dois *sliders* para a medição da intensidade (acima) e da valência (abaixo) numa escala contínua [Betella and Verschure, 2016].

Os resultados apresentados pelos autores mostram uma correlação muito forte entre as classificações do SAM e do AS, demonstrando empiricamente que o AS pode substituir o SAM no autorrelato de valência e intensidade, com as vantagens adicionais de

ser independente e facilmente reproduzível em dispositivos digitais de última geração, estando o seu *source code* disponível em: <https://github.com/albertobeta/AffectiveSlider> [Betella and Verschure, 2016].

Para avaliar não só o estado emocional do utilizador, mas também o nível de *user engagement*, os questionários são considerados o método mais acessível em termos monetários. No entanto, existe uma falta de questionários de autorrelato validados para uso no contexto de *eHealth* e *mHealth*, sendo a sua maioria criada para fins específicos [Short et al., 2018].