UNIVERSIDADE D
COIMBRA

Pedro Manuel Vicente de Almeida

Interação de um Robô Social com Utilizadores Humanos através de Linguagem Natural

Dissertação no âmbito do Mestrado de Engenharia Eletrotécnica e de Computadores na Área da Robótica, Controlo e Inteligência Artificial orientada pelo Professor Doutor Rui Paulo Pinto da Rocha e coorientada pelo Professor Doutor Fernando Manuel dos Santos Perdigão e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Universidade de Coimbra

Julho de 2024

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Interação de um Robô Social com Utilizadores Humanos através de Linguagem Natural

Pedro Manuel Vicente de Almeida

Coimbra, Julho de 2024

Interação de um Robô Social com Utilizadores Humanos através de Linguagem Natural

Orientador:

Prof. Doutor Rui P. Rocha

Coorientador:

Prof. Doutor Fernando Perdigão

Júri:

Prof. Doutor Lino José Forte Marques

Prof. Doutor Paulo Jorge Carvalho Menezes

Prof. Doutor Rui Paulo Pinto da Rocha

Dissertação apresentada em cumprimento parcial para obtenção do grau de Mestre em Engenharia Eletrotécnica e de Computadores.

Coimbra, Julho de 2024

Este trabalho de dissertação de mestrado teve o apoio financeiro do Programa Interreg Espanha-Portugal (POCTEP 2021-2027), através de uma bolsa de investigação do projeto EuroAGE+ financiado por este programa com a referência 0124_EUROAGE_MAS_4_E.

Agradecimentos

Gostaria de expressar a minha mais profunda gratidão a todos os que, de diversas maneiras, fizeram parte deste importante percurso académico e contribuíram para a realização desta dissertação.

Primeiramente, gostaria de agradecer aos meus orientadores, Prof. Rui P. Rocha e Prof. Fernando Perdigão, pelo apoio incansável, pela motivação constante, pela orientação precisa, disponibilidade e pela confiança depositada em mim ao longo deste projeto. Foi uma honra poder realizar este trabalho sob a orientação do vosso vasto conhecimento, que decerto tanto colaborou para o sucesso do mesmo.

Um agradecimento especial ao Projeto EuroAGE+, financiado pelo programa Interreg Espanha-Portugal, pela oportunidade de apoio financeiro através de uma bolsa de investigação com referência 0124_EUROAGE_MAS_4_E.

Gostaria também de agradecer à Cáritas Diocesana de Coimbra, especialmente às terapeutas ocupacionais Carina e à Ana, pela incansável disponibilidade e profissionalismo. Um sincero agradecimento à Dr^{a}. Flávia e à Dr^{a}. Andrea do Departamento de Inovação desta instituição, que permitiram a validação experimental do trabalho desenvolvido e sem a qual não poderia ter sido finalizado com sucesso.

Não poderia deixar de expressar o meu eterno agradecimento à minha família. Em especial aos meus pais, à minha irmã e ao meu avô, que tanto influenciaram na pessoa que sou hoje e na escolha do tema deste trabalho. Sempre fizeram questão de estar presentes, mesmo quando achava desnecessário, oferecendo sempre o devido apoio incondicional, incentivo constante e paciência. Sem o vosso apoio, este trabalho não teria sido possível.

Aos meus amigos e colegas de curso, em especial aos que marcaram esta minha passagem, à Ana, ao Mário e ao Lucas que mesmo não estando presente nestes dois últimos anos, a sua influência prevaleceu. As experiências partilhadas convosco ao longo destes anos tornaram esta jornada ainda mais especial e memorável.

Pela vida académica e por vezes boémia demais, gostaria de agradecer à Estudantina

Universitária de Coimbra, que me proporcionou das melhores vivências nesta cidade e que, certamente, continuará a proporcionar. Os momentos passados com este grupo serão sempre recordados com carinho e gratidão.

Finalmente, à minha namorada, um agradecimento especial pelo apoio emocional nesta fase mais atribulada. Obrigado pela tua presença e compreensão que foram fundamentais para que eu pudesse concluir este trabalho com sucesso.

A todos, que de forma direta ou indireta colaboraram para que este momento fosse possível, o meu mais sincero obrigado.

Resumo

Este trabalho de dissertação tem como objetivo investigar a conceção e desenvolvimento de um protótipo de robô social assistencial capacitado para interação humano-robô através de linguagem natural, incluindo reconhecimento de voz, síntese de voz e gestão de diálogo.

Inspirado pela crescente importância dos robôs sociais assistenciais, especialmente no contexto de *Ambient Assisted Living* (AAL), optou-se pelo uso do robô social GrowMu e pela integração de *Application Programming Interface* (API) de reconhecimento e síntese de voz complementado com algoritmos de sincronização labial.

O principal objetivo passou por criar um protótipo que permita uma interação natural entre o robô e os utilizadores para dois modos de funcionamento: narrativa, onde o robô visa contar uma história interativa ao utilizador idoso consoante o seu nível de défice cognitivo tendo como base as suas informações (e.g., passatempos, profissão exercida, etc.); e um modo livre, destinado a diversos utilizadores que apresenta a capacidade de reconhecimento facial dos mesmos, e consoante essa informação é gerido o diálogo. Ambos os modos foram testados, com uma ênfase particular no modo de narrativa, avaliado por utilizadores finais idosos.

Tendo como base a revisão da literatura da interação humano-robô, foram adotadas as melhores soluções de engenharia para este trabalho, onde se utilizou modelos de linguagem pré-treinados (e.g., da OpenAI), e também a primeira linguagem de programação para agentes virtuais, a *Artificial Intelligence Markup Language* (AIML).

Este trabalho, desenvolvido no âmbito do projeto EuroAGE+ financiado pelo programa Interreg Espanha-Portugal, onde se pretende demonstrar a aplicação de robôs sociais na promoção do envelhecimento ativo, visa contribuir para o avanço da interação humano-robô em ambientes assistenciais, proporcionando uma base sólida para futuras investigações e desenvolvimentos nesta área em constante evolução.

Palavras-chave: robô social assistencial; interação humano-robô; *speech to text*; *text to speech*; gestão de diálogo;

Abstract

This dissertation aims to investigate the design and development of a prototype of an assistive social robot capable of human-robot interaction through natural language, including voice recognition, speech synthesis, and dialogue management.

Inspired by the growing importance of assistive social robots, especially in the context of AAL, the social robot GrowMu was chosen, and an API for voice recognition and synthesis was integrated, complemented with lip synchronization algorithms.

The main objective was to create a prototype that allows natural interaction between the robot and users for two modes of operation: narrative, where the robot aims to tell an interactive story to the elderly user based on their level of cognitive impairment and personal information (e.g., hobbies, past profession, etc.); and a free mode, intended for various users, which features facial recognition and manages the dialogue accordingly. Both modes were tested, with particular emphasis on the narrative mode, which was evaluated by elderly end users.

Based on a review of the literature on human-robot interaction, the best engineering solutions were adopted for this work, utilizing pre-trained language models (e.g., from OpenAI), as well as the first programming language for virtual agents, AIML.

This work, developed within the EuroAGE+ project funded by the Interreg Spain-Portugal program, which aims to demonstrate the application of social robots in promoting active aging, seeks to contribute to the advancement of human-robot interaction in assistive environments, providing a solid foundation for future research and developments in this constantly evolving field.

Keywords: assistive social robot; human-robot interaction; speech to text; text to speech; dialogue management;

"Perseverança na fé."

— Pe. Alberto Galassi

Índice

Lista de Acrónimos

AAL	*Ambient Assisted Living*
AIML	*Artificial Intelligence Markup Language*
API	*Application Programming Interface*
AWS	*Amazon Web Services*
CDC	Cáritas Diocesana de Coimbra
CNN	*Convolutional Neural Network*
IA	Inteligência Artificial
IPA	Alfabeto Fonético Internacional
ISR	Instituto de Sistemas e Robótica
LED	*Light Emitting Diode*
ML	*Machine Learning*
MMSE	*Mini Mental State Exam*
NER	*Named Entity Recognition*
NLTK	*Natural Language Toolkit*
NLP	*Natural Language Processing*
POS Tagging	*Part-of-Speech Tagging*
RNN	*Recurrent Neural Network*
ROS	*Robot Operating System*
SQL	*Structured Query Language*

SSML	*Speech Synthesis Markup Language*
STT	*Speech to Text*
SUS	*System Usability Scale*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
TIC	Tecnologias da Informação e Comunicação
TTS	*Text to Speech*

Índice de Figuras

Índice de Tabelas

1 Introdução

A interação entre robôs e seres humanos tem fascinado mentes ao longo da história. Um exemplo notável remonta a 1495, quando Leonardo Da Vinci criou o Cavaleiro Mecânico, uma armadura medieval capaz de replicar movimentos semelhantes aos humanos no campo de batalha. Avançando no tempo, em 1977, surge o C-3PO na saga de ficção Star Wars, cuja ação passa-se numa galáxia muito distante, cativando audiências para os ecrãs gigantes. Contudo, o que outrora parecia um sonho distante tem-se tornado cada vez mais palpável devido aos notáveis avanços computacionais alcançados e, consequentemente, à viabilização da utilização da inteligência artificial, que cada vez mais tem contribuído para a humanização dos robôs e das máquinas em geral [1].

Ao longo destes anos, o conceito de robô social assistencial tem sido progressivamente idealizado e aprofundado. A sua definição é sugerida da seguinte forma: "Robôs sociais assistenciais são robôs que interagem com humanos e entre si de maneira socialmente aceitável, transmitindo intenções de forma percetível aos humanos, e conseguem alcançar objetivos em colaboração com outros agentes, sejam eles humanos ou robôs" [21]. Consequentemente, outros conceitos e ideias têm ganho mais perspetiva, como AAL, que, ao utilizar este tipo de robôs, visa auxiliar pessoas da terceira idade a manterem-se socialmente conectadas e ativas por mais tempo [30, 28, 33]. Um exemplo prático destes ambientes é a promoção da atividade física regular entre pessoas idosas [28]. De facto, é necessário reconhecer o desafio demográfico global, especialmente em países como Portugal, caracterizado pelo envelhecimento da população e pela sua litoralização. Este cenário suscita a necessidade de explorar soluções inovadoras baseadas neste tipo de tecnologias [3, 2].

1.1 Contexto e motivação

Este trabalho de dissertação foi desenvolvido no âmbito do projeto EuroAGE+[1], financiado pelo programa Interreg Espanha-Portugal. O EuroAGE+ é uma iniciativa internacional de investigação, inovação e transferência de tecnologia voltada para a promoção do envelhecimento ativo. O projeto visa promover a colaboração entre instituições espanholas e portuguesas, incluindo a participação ativa da Universidade de Coimbra através do Instituto de Sistemas e Robótica (ISR), para contribuir para o envelhecimento ativo dos idosos através da promoção de atividades físicas, cognitivas e socioemocionais usando de Tecnologias da Informação e Comunicação (TIC).

Como parte integrante deste projeto, este trabalho de dissertação aspira contribuir ativamente para os objetivos delineados anteriormente, focando mais especificamente na interação entre o robô e o utilizador humano através de diálogo, promovendo assim atividades socioemocionais.

1.2 Definição do problema

Apesar do significativo desenvolvimento dos sistemas de interação humano-robô nas últimas décadas [9], continua a existir uma necessidade persistente de adaptação destes sistemas às necessidades específicas e preferências dos utilizadores idosos, particularmente no que diz respeito à promoção de interações socioemocionais significativas.

O problema central abordado por esta dissertação envolve a implementação de módulos numa plataforma robótica social, permitindo a comunicação do robô com os utilizadores através de diálogo. O foco está na busca por soluções eficazes para a implementação desses módulos, visando capacitar o robô com habilidades avançadas de processamento e contextualização da realidade. Com isso, será permitida a integração desses módulos em robôs sociais assistenciais, possibilitando o apoio ao envelhecimento ativo através da promoção de interações humanizadas e personalizadas.

1.3 Objetivos e contribuições

Para este trabalho de dissertação, os objetivos incluem a conceção de um sistema de gestão de diálogo para um robô social assistencial, com dois modos de funcionamento.

[1] `https://euroageplus.unex.es/pt-pt/`

Um dos modos visa funcionar como ferramenta terapêutica, fornecendo histórias interativas adaptadas a diferentes níveis de défice cognitivo dos utilizadores idosos, simplificando tanto a narrativa quanto o vocabulário conforme necessário. Para além desta característica, é desejado que o sistema narre histórias personalizadas com base em informação específicas de cada utilizador.

O outro modo, tem por objetivo capacitar o robô social assistencial de reconhecer os utilizadores e manter diálogos com base nesse reconhecimento e no conhecimento acerca dos utilizadores adquirido em interações anteriores, respondendo adequadamente às interações dos utilizadores.

Contudo, antes de ser possível a utilização destes dois modos, também tem-se como objetivo dotar o robô das capacidades de se expressar através da fala e de compreender as falas dos seus utilizadores, possibilitando interações com pessoas através de diálogo, melhorando assim a interação e a personalização das interações em contextos assistenciais.

1.4 Estrutura da dissertação

O presente documento de dissertação está estruturado conforme se descreve a seguir.

No capítulo 2, é realizada uma breve revisão de literatura e fundamentos sobre o potencial da utilização de robôs sociais assistenciais para AAL, incluindo uma comparação de serviços que visam dotar estes robôs da capacidade de expressão e compreensão através da fala. No capítulo 3 (página 13) é realizado o planeamento dos dois modos de gestão de diálogo com base na revisão da literatura existente sobre interação homem-máquina, onde se inclui o *design* dos diferentes módulos e a apresentação da arquitetura final para ambos os modos. Com base nesta conceção, é apresentado no capítulo 4 (página 33) todo o processo de implementação, explicitando os desafios técnicos enfrentados ao longo do desenvolvimento desta dissertação. No capítulo 5 (página 49), é apresentada a validação experimental do sistema implementado para os dois modos de funcionamento propostos. Finalmente, no capítulo 6 (59) é apresentada a conclusão desta dissertação, bem como algumas linhas de trabalho futuro suscitadas pelo desenvolvimento deste trabalho.

No final deste documento, são disponibilizados alguns anexos que complementam e apresentam mais detalhes sobre determinadas partes do corpo principal da dissertação.

2 Revisão da literatura e fundamentos

A evolução da interação humano-robô tem sido um campo de investigação ativo, refletindo-se em inúmeros estudos e projetos dedicados a aprimorar esta forma de interação [9]. Neste capítulo, é apresentada uma análise de alguns destes projetos de investigação e desenvolvimento, abordando especificamente a utilização de TIC para promoção do envelhecimento ativo, bem como o papel dos robôs sociais assistenciais neste contexto.

2.1 Literatura

Diversos estudos e projetos de investigação têm demonstrado que a utilização de TIC permite a criação de AAL tanto em residências como em locais de trabalho, possibilitando o envelhecimento ativo aos seus utilizadores [30]. As TIC desempenham um papel fundamental na promoção destes ambientes, que visam melhorar a qualidade de vida dos idosos através da automação e monitorização contínua.

Como foi discutido anteriormente, o envelhecimento da população global, particularmente nos países desenvolvidos, apresenta desafios que podem ser mitigados pela promoção da atividade física e do bem-estar social e emocional, permitindo uma maior autonomia na terceira idade. A promoção de um estilo de vida ativo e socialmente envolvente é essencial para reduzir os riscos associados ao envelhecimento, como a solidão, depressão e declínio cognitivo [31].

Um estudo relevante foi conduzido no âmbito do projeto EuroAGE[1], demonstrando como agentes virtuais em colaboração com um robô companheiro podem incentivar os idosos a manterem-se ativos fisicamente [28]. Para além desta motivação, com base numa avaliação sensorial recolhida pelo sistema de exercício, o mesmo consegue avaliar a prestação do idoso, melhorando a participação dos utilizadores nas atividades físicas e contribuindo assim para o seu bem-estar e autonomia.

[1] `https://euroage.eu/pt/home/`

Concentrando agora na utilização de robôs sociais assistenciais nestes ambientes, em [33] é apresentado um estudo que contribuiu ativamente para a identificação e discussão dos mais variados desafios técnicos e não técnicos na conceção destes agentes. Este trabalho foi realizado no âmbito do projeto GrowMeUp[2], que teve como objetivo investigar como robôs que interagem através da voz podem ser aceites e adotados pela sociedade idosa. Os resultados destacaram desafios na qualidade da interação, como a precisão do reconhecimento de voz, a fala à distância e a expressividade do robô ao falar. No entanto, os idosos deram parecer positivo sobre a capacidade do robô se lembrar de tarefas e atividades, facilitando a vida daqueles que vivem sozinhos. As características deste estudo são importantes para poder compreender os requisitos específicos que a classe idosa coloca no desenvolvimento de sistemas interativos baseados na fala.

Adicionalmente, a aceitação e eficácia dos robôs sociais dependem fortemente da capacidade destes de interagir de maneira natural e empática. Estudos mostram que robôs sociais podem fornecer benefícios significativos em termos de companhia e assistência, ajudando a reduzir a solidão e promovendo uma maior independência [10]. No entanto, para serem efetivamente aceites, estes robôs devem ser projetados para entender e responder adequadamente às necessidades emocionais e físicas dos utilizadores, exigindo avanços em áreas como inteligência artificial e processamento de linguagem natural [20].

A revisão de literatura destes últimos conceitos será apresentada com mais detalhe na secção 3.2 (página 16), onde será apresentada uma revisão de toda a história da interação homem-máquina desde o início dos tempos. Contudo, pelos diversos projetos de investigação aqui apresentados, é possível desde já verificar que a integração de TIC e robôs sociais assistenciais em ambientes de AAL pode desempenhar um papel crucial na promoção do envelhecimento ativo e saudável.

2.2 Comparação de serviços na cloud para processamento de voz

Nesta secção, são analisados e comparados diversos serviços de conversão de texto em fala (TTS) e de reconhecimento de fala para texto (STT) oferecidos no mercado por diferentes empresas tecnológicas. Sendo que estes serviços são processados em nuvem, apresentam custos às empresas que os detêm, pois, estas disponibilizam os seus recursos computacionais

[2] `https://cordis.europa.eu/project/id/643647/it`

para o efeito. Portanto, é necessário fazer uma comparação dos principais prestadores de serviços, escolhendo assim o mais rentável com base na relação qualidade-preço. Posto isto, o objetivo passa por avaliar as características, custos e eficácia de cada serviço, de modo a selecionar a melhor opção para este projeto de dissertação.

Os serviços abordados nesta secção são oferecidos pelas empresas OpenAI, Microsoft, Amazon, Google e ElevenLabs, sendo destacadas as principais informações de cada uma. Para cada, são apresentados detalhes sobre a oferta de serviços gratuitos e os custos associados às opções pagas, bem como observações sobre a qualidade do áudio gerado, a facilidade de uso das suas API e as eventuais limitações linguísticas. No final desta análise, será apresentado um resumo que indica os prestadores de serviços selecionados.

2.2.1 OpenAI

A OpenAI[3] é uma empresa líder em inteligência artificial, fundada em 2015, conhecida por desenvolver modelos de linguagem avançados, como é o caso do ChatGPT. A mesma visa realizar pesquisas e promover o acesso aberto à tecnologia, destacando-se como uma das principais forças inovadoras na vanguarda da Inteligência Artificial (IA).

Nos serviços disponibilizados por esta empresa não existe nenhum plano gratuito limitado; apenas existe o modelo de custos *pay as you go*. Contudo é importante notar que, dentre todos os prestadores de serviços, a OpenAI é a que tem o menor custo no modelo referido. O *Speech to Text* (STT), também conhecido como Whisper, é taxado a $0,006 USD por minuto de áudio processado, custando por isso $0,36 USD por hora, o que é equivalente a 0,33€[4]. A conversão da fala para texto, tanto em português como em inglês, apresenta uma grande taxa de sucesso, estando entre os 98.5% e 95%. Além disto, uma característica interessante deste serviço é que o mesmo consegue detetar idiomas automaticamente. O TTS tem um custo de $0,015 USD por mil caracteres, ou seja 0,014€. Não existe o modo em português europeu, contudo as vozes geradas por este serviço apresentam uma grande naturalidade além do seu excelente processamento.

Como foi mencionado anteriormente, a principal característica que distingue esta empresa são os seus modelos de linguagem de grande escala. Com base nestes, a mesma oferece a possibilidade de criar um assistente, ou seja um "gerador de respostas", baseado num modelo de linguagem disponível, além de permitir o ajuste dos seus parâmetros, tornando-o assim adequado para diversas situações. Tendo por base o modelo `gpt-3.5-turbo-1106`, um dos

[3] `https://openai.com/`

[4] Conversão calculada à data da escrita da dissertação.

modelos indicados para o nosso caso específico, os custos associados são de $0,0010 por mil "tokens" de entrada e $0,0020 USD por mil "tokens" de saída. Os "tokens" referidos, tanto de entrada como de saída, são associados a caracteres. No entanto, importa notar que nem sempre um caractere corresponde a um "token", pois a sua definição depende do idioma e de alguns fatores associados ao modelo a usar.

Por último, é importante notar que, apesar desta empresa apresentar os custos mais baixos do mercado, para o contexto deste trabalho de dissertação, considerando que não serão utilizadas muitas horas destes serviços, será mais indicado um prestador que ofereça um limite gratuito, reduzindo assim os custos em causa.

2.2.2 Microsoft

A Microsoft[5] é uma empresa de tecnologia mundialmente conhecida, contando com um histórico de inovação em software e hardware. Fundada por Bill Gates e Paul Allen em 1975, é reconhecida pelos sistemas operativos Windows, pelas ferramentas Office e por plataformas como Azure e Xbox. Além deste notável histórico, é líder em inteligência artificial e computação em nuvem, oferecendo soluções empresariais e de consumo em todo o mundo.

Entre os prestadores de serviços apresentados, a Microsoft destaca-se pela facilidade de uso e pela excelente relação qualidade-preço. No STT, são oferecidas mensalmente 5 horas de processamento de áudio gratuito; quando excedido, é taxado a 0,911€ por hora. Este serviço é rapidamente distinguido pelo seu reduzido tempo de latência em comparação com outros. A partir do mesmo, há a capacidade de transcrever texto com a respetiva pontuação, assemelhando-se ao Whisper da OpenAI. Através da sua biblioteca, necessária para a utilização do serviço, é possível captar áudio e transcrever o mesmo em tempo real devido ao seu gravador já incorporado.

No TTS, a empresa oferece 500 mil caracteres gratuitos por mês. Após essa utilização, o custo é de 0,0146€ por mil caracteres. Apesar de contar com vozes em português europeu, há uma limitação significativa relacionada com a entoação de interpretar perguntas. Esta é a única falha notável, pois a qualidade do áudio processado em língua portuguesa e inglesa é indiscutível. Outras características positivas passam pela capacidade de transformar o texto sintetizado numa sequência de visemas[6], que permite a sincronização do áudio com as expressões labiais. Além disso, a biblioteca incorpora um sintetizador que reproduz imediatamente

[5] `https://www.microsoft.com/pt-pt`

[6] Um visema é a descrição visual de um fonema na linguagem falada, representando, portanto, a posição da face e da boca ao dizer uma palavra.

o áudio gerado, reduzindo todos os tempos críticos de latência.

2.2.3 Amazon

A Amazon[7] é uma empresa de comércio eletrónico e serviços em nuvem fundada por Jeff Bezos em 1994. Mundialmente conhecida pela sua plataforma líder de comércio online, a Amazon também é reconhecida pelos seus serviços de tecnologia, como a *Amazon Web Services* (AWS), que oferece uma ampla gama de serviços para empresas e desenvolvedores.

Com o STT da Amazon, a empresa oferece uma hora de áudio por mês gratuita. No entanto, este plano é apenas válido para os primeiros 12 meses aquando da criação da conta AWS. Esgotando os tempos gratuitos mensais e eventualmente o período anual, o serviço é taxado a $0,0240 USD por minuto, equivalente a $1,44 USD por hora, ou seja, 1,32€. Relativamente aos resultados da transcrição usando este serviço, os mesmos enfrentam alguns problemas relativamente à compreensão na entoação de perguntas por parte dos interlocutores.

Quanto ao serviço TTS, também conhecido como Amazon Polly, são disponibilizados 5 milhões de caracteres gratuitos mensais nos primeiros 12 meses. Após esse limite, é cobrado $0,004 USD, ou seja 0,00367€ por mil caracteres. As vozes portuguesas europeias disponíveis são a Inês e o Cristiano, porém ambas apresentam pouca naturalidade. Outro aspeto negativo é a incapacidade de permitir "multilanguage" num determinado idioma, impossibilitando o uso de certas palavras emprestadas. Apesar disso, o serviço oferece a capacidade de entoação a partir da pontuação do texto. É importante também notar que as vozes em inglês são mais naturais e, portanto, mais viáveis de serem utilizadas. Outro aspeto deste serviço é a deteção de visemas, semelhante ao da Microsoft, porém com uma capacidade inferior.

Em suma, os serviços oferecidos por esta empresa, no caso específico deste trabalho de dissertação, acabam por não ser uma das melhores soluções devido à limitação do período gratuito nos primeiros 12 meses, bem como à menor qualidade dos serviços de STT e TTS. Contudo, em última instância, o TTS poderá eventualmente ser uma solução viável em inglês devido aos resultados alcançados.

[7] `https://www.amazon.com/-/pt`

2.2.4 Google

Sendo uma das maiores empresas de tecnologia do mundo, a Google[8] fundada em 1998 por Larry Page e Sergey Brin, é conhecida pelo seu motor de pesquisa universal. Esta empresa oferece uma ampla gama de produtos e serviços, incluindo Android, Gmail, Google Cloud Platform, Google Workspace, e, além disso, é pioneira em inteligência artificial e aprendizagem automática[9], impulsionando inovações em áreas como assistentes virtuais, Google Assistant, carros autónomos e reconhecimento de imagem.

Ao utilizar o STT da Google, a oferta gratuita mensal é de 60 minutos de conversão de áudio que, quando excedida, passa a ser taxada a $0,024 USD por minuto, equivalendo aproximadamente a $1,44 USD por hora, ou 1,32€. O modo português deste serviço (STT) não consegue interpretar a pontuação da fala do interlocutor, por exemplo, na entoação de perguntas. Contudo, em inglês, o mesmo já não acontece. Apesar desta limitação, a taxa de sucesso da transcrição do áudio aparenta ser bastante elevada.

Quanto ao TTS, a empresa oferece 1 milhão de caracteres gratuitos mensais. Quando este limite é excedido, são cobrados $0,000016 USD por caracter, ou seja, $0,016 USD ou 0,0147€ por mil caracteres. Ambas as línguas apresentam boa qualidade, conseguindo interpretar a pontuação presente no texto.

No geral, a utilização dos serviços da Google assemelha-se à simplicidade dos serviços da Microsoft, apenas tendo a desvantagem de a sua biblioteca não possuir um sintetizador integrado, o que resulta num atraso ligeiramente maior, mesmo que impercetível. Outra desvantagem advém da incapacidade de deteção de visemas conforme o texto e áudio sintetizados.

2.2.5 ElevenLabs

A empresa ElevenLabs[10] foi criada em 2022 por Piotr Dąbkowski e Mateusz Staniszewski. É uma empresa de software especializada em síntese de fala natural, sendo conhecida pela conversão de texto em fala com elevadíssima qualidade, recorrendo ao uso de IA.

Ao contrário dos outros prestadores de serviços apresentados, esta empresa contém apenas TTS, e, além disso, possui uma estrutura de planos com cotas distintas das até agora apresentadas.

[8] `https://www.google.com/intl/pt-PT_pt/business/`

[9] Tradução da expressão inglesa "machine learning".

[10] `https://elevenlabs.io/`

Os planos mensais oferecidos pela ElevenLabs são os seguintes:

- **Gratuito:** 10.000 caracteres;
- **Iniciante** – $5 USD – 30.000 caracteres com a capacidade de clonar vozes;
- **Criador** – $22 USD – 100.000 caracteres com a capacidade de clonar vozes profissionalmente;
- **Editora independente** – $99 USD – 500.000 caracteres com saída de áudio a 44.1kHz PCM;
- **Negócios em crescimento** – $330 USD – 2.000.000 caracteres;
- **Empreendimento – Personalizado** – plano ajustado às necessidades específicas com cotas personalizadas.

De todos os serviços TTS referidos, o desta empresa é o melhor. Este serviço é caracterizado pela fácil utilização e pela grande qualidade e naturalidade do áudio gerado, estando o mesmo atento às pontuações. As vozes disponíveis são de grande diversidade, cada uma com características distintas, desde o sotaque até ao nível de expressividade, sendo também possível controlar estes parâmetros. Em contrapartida, este serviço é apenas viável para inglês; em português, apenas está disponível a versão brasileira. Além disso, o serviço apresenta grandes tempos de latência comparativamente com os outros prestadores de serviços e não possui a funcionalidade de deteção de visemas.

2.2.6 Análise comparativa

Após analisar todas as informações recolhidas sobre os prestadores de serviços revistos nas subsecções anteriores, chegamos a algumas conclusões importantes que nos permitem selecionar os prestadores de serviços mais adequados aos objetivos desta dissertação. Para sintetizar os custos associados a cada empresa e proporcionar uma vista um pouco mais geral sobre todos os prestadores de serviços aqui apresentados, nas tabelas 2.1 e 2.2 encontram-se, respetivamente, as condições oferecidas por cada empresa em modo gratuito e *pay as you go*.

É evidente que, entre todas as empresas apresentadas, a Microsoft destaca-se por oferecer serviços de alta qualidade a custos bastante acessíveis. Além disso, como já foi mencionado anteriormente, não será necessário efetuar pagamentos usando o seu modo gratuito, uma

	Microsoft	**Amazon (12 meses)**	**Google**	**ElevenLabs**
STT	5h	1h	1h	-
TTS (em caracteres)	500 mil	5 milhões	1 milhão	10 mil

Tabela 2.1: Comparação de serviços de STT e TTS em modo gratuito.

	Microsoft	**Amazon (12 meses)**	**Google**	**OpenAI**
STT	0,911€/hora	1,32€/hora	1,32€/hora	0,33€/hora
TTS (em caracteres)	0,014571€/mil	0,00367€/mil	0,0147€/mil	0,014€/mil
Assistente (por mil 'tokens') Entrada + Saída	0,00092€ + 0,0018€	-	-	-

Tabela 2.2: Comparação de serviços de STT e TTS em modo *pay as you go*.

vez que dificilmente excederemos os limites impostos, representando assim uma vantagem significativa. No caso da Amazon, a solução não pode ser considerada ótima, pois o acesso aos limites gratuitos é concedido apenas nos primeiros 12 meses, o que pode comprometer a continuidade do projeto para anos futuros. Contudo, para casos inevitáveis, poderá eventualmente ser uma solução. A ElevenLabs, como foi mencionado anteriormente, destaca-se pelo seu serviço TTS, proporcionando uma excelente qualidade de áudio. No entanto, o seu muito menor limite de caracteres gratuitos tende a desincentivar a sua utilização. Por fim, apesar de ser uma das escolhas mais económicas em modo *pay as you go* e uma das que mais se destaca pela qualidade dos serviços que disponibiliza, a OpenAI não poderá ser considerada para STT e TTS visto que é mais rentável utilizar em modo gratuito qualquer um dos outros serviços apresentados. Portanto, para esta empresa, o único serviço a ser considerado é o assistente que disponibiliza a capacidade de uso de um dos modelos de linguagem para gestor de diálogo.

Podemos assim concluir que a solução STT da Microsoft é preferível. No entanto, se necessário for, o serviço da Google pode ser considerado uma alternativa equiparável. Para TTS, sendo necessário no nosso caso assegurar duas línguas – português e inglês –, as soluções são as seguintes, por ordem de preferência:

Inglês:

1. ElevenLabs
2. Microsoft
3. Amazon

Português:

1. Microsoft
2. Google
3. Amazon

2.3 Sumário

No presente capítulo, foi apresentada uma breve revisão da literatura e fundamentos, para comprovar a importância da utilização de TIC e robótica assistencial social em ambientes de AAL no contexto do envelhecimento ativo. Foram discutidos diversos estudos que demonstram como estas tecnologias podem melhorar a qualidade de vida dos idosos, promovendo a atividade física e a interação social. A continuação da revisão da literatura e fundamentos é complementada na secção 3.2 (página 16), onde é apresentada a história da interação entre homem e robô desde o início dos tempos.

Para além disto, foi apresentada uma comparação dos serviços STT e TTS disponibilizados pelos prestadores de serviços de computação em nuvem OpenAI, Microsoft, Amazon, Google e Eleven Labs, que são relevantes para o objetivo final deste trabalho de dissertação.

No capítulo seguinte é apresentada a conceção do sistema de gestão de diálogo.

3 Conceção do sistema de gestão de diálogo

Para além de se dotar o robô social da capacidade de comunicação com o utilizador humano baseada em linguagem natural utilizando os serviços de computação em nuvem STT e TTS, conforme apresentado no capítulo anterior, é também necessário dotar o mesmo da capacidade de estabelecer diálogos. Por outras palavras, é preciso que o robô consiga gerar uma resposta adequada a partir de um determinado texto de entrada obtido da transcrição da fala do utilizador humano gerada pelo serviço de STT. Neste contexto, o principal objetivo passa pela conceção de um sistema de gestão de diálogo responsável pela compreensão e adaptação do robô a situações e conversações diversas.

Tendo em conta o contexto dos últimos avanços tecnológicos que transformaram a humanidade num passado recente, quando pesquisamos literatura sobre máquinas capazes de usarem linguagem natural, sobressaem conceitos e capacidades importantes, tais como IA [38], *Machine Learning* (ML) [8], NLP [22], etc. Contudo, o processo de evolução do conhecimento até estas tecnologias avançadas surgirem e moldarem o que é hoje a forma de compreensão e expressão das máquinas, foi bastante longo e demorado, havendo a necessidade de remontar até à década de 50 e 60 do século XX, onde os primeiros investigadores começaram a explorar métodos computacionais para processar linguagem natural.

3.1 Análise de Requisitos

Como já foi referido, este trabalho de dissertação foi desenvolvido no âmbito do projeto EuroAGE+, uma rede internacional de investigação, inovação e transferência de tecnologia destinada à promoção do envelhecimento ativo. O principal objetivo a partir da colaboração entre instituições espanholas e portuguesas, nas quais a Universidade de Coimbra participa ativamente através do ISR, é contribuir para o envelhecimento ativo dos idosos através

da promoção de atividade física, cognitiva e socioemocional. Pretende-se ainda incentivar a vida autónoma saudável dos idosos e pessoas dependentes através da promoção de iniciativas inovadoras com recurso às TIC, incluindo robôs sociais assistenciais.

Uma das contribuições do ISR neste projeto é a criação de um sistema de gestão de diálogo para narração de histórias interativas com fins terapêuticos. O principal objetivo é estimular cognitivamente idosos com diferentes níveis de défice cognitivo, proporcionando-lhes a capacidade de tomar decisões que influenciem a continuidade da narrativa. Pretende-se ainda que este jogo se torne uma atividade recorrente no acompanhamento terapêutico em diversas instituições de cuidados a idosos, possibilitando a avaliação do desempenho dos jogadores por parte de terapeutas ocupacionais, ajudando a identificar potenciais necessidades de trabalho cognitivo para cada paciente.

Além do modo de funcionamento de narração interativa, o plano de trabalho da presente dissertação inclui a conceção preliminar de um modo "livre", no qual o robô consegue identificar uma pessoa e guardar algumas informações importantes da mesma a partir do diálogo estabelecido. Ao contrário do outro modo de funcionamento, este modo não se destina exclusivamente a utilizadores finais idosos, tendo como objetivo principal assegurar interações quotidianas do robô com diferentes utilizadores, possivelmente de diferentes idades.

Com dois modos de sistema de gestão de diálogo a serem concebidos, é essencial identificar e compreender todas as necessidades dos utilizadores finais. Como primeiro passo, é necessário definir os requisitos para desenvolver uma solução eficaz e eficiente. A seguir, são apresentados os requisitos funcionais e não funcionais para os dois modos de funcionamento.

3.1.1 Modo Narrativa

Requisitos Funcionais

Como requisitos funcionais, o sistema de gestão de diálogo deve:

1. Possibilitar que o robô fale e conte a história de forma que possa ser escutada e compreendida pelo utilizador;

2. Permitir que os idosos interajam com a narrativa tomando decisões que influenciem o desenrolar da história;

3. Ser possível adaptar o nível de complexidade das histórias ao nível de défice cognitivo quantificado através da métrica *Mini Mental State Exam* (MMSE) [18];

4. Reproduzir narrativas originais consoante um conjunto de parâmetros pré-estabelecidos a serem introduzidos pelo terapeuta, tais como:

   (a) Tema

   (b) Assuntos Sensíveis

   (c) Nome do idoso

   (d) Idade

   (e) Nível de défice cognitivo

   (f) Profissão passada

   (g) Passatempos

   (h) Nomes de relações importantes, i. e. Irmãos, Cão, etc.

5. Permitir ao terapeuta pausar e retomar a narrativa ou, se necessário, parar definitivamente a narração da história e finalizar permanentemente a sessão (i. e. paragem de emergência).

6. Identificar quando a história termina.

Requisitos Não Funcionais

Como requisitos não funcionais, o sistema de gestão de diálogo deve:

1. Possibilitar que o robô interprete corretamente as falas do utilizador;

2. Ser responsivo e eficiente, garantindo tempos de resposta o mais próximo possível de 200 milissegundos, sendo este valor a latência de resposta média em conversas face a face entre utilizadores humanos [41];

3. Assegurar a privacidade e a segurança dos dados dos utilizadores, conforme as regulamentações de proteção de dados;

4. Apresentar uma interface amigável e intuitiva para o terapeuta ocupacional;

3.1.2 Modo Livre

Requisitos Funcionais

No modo livre, o sistema de gestão de diálogo deve:

1. Possibilitar que o robô fale de forma que possa ser escutado e compreendido pelo utilizador;
2. Conseguir identificar e reconhecer o idoso utilizando tecnologias de reconhecimento facial;
3. Armazenar informações relevantes sobre o idoso, como nome e idade;
4. Permitir que o robô mantenha um diálogo natural, dando respostas contextualizadas;

Requisitos Não Funcionais

No modo livre, o sistema de gestão de diálogo deve:

1. Possibilitar que o robô interprete corretamente as falas do utilizador;
2. Ser responsivo e eficiente, garantindo tempos de resposta o mais próximo possível de 200 milissegundos, sendo este valor a latência de resposta média em conversas face a face entre utilizadores humanos [41];
3. Assegurar a privacidade e a segurança dos dados dos utilizadores, conforme as regulamentações de proteção de dados;
4. Ser facilmente escalável, permitindo a adição de novos utilizadores.

Tendo identificado os diferentes requisitos para os dois modos de funcionamento, a seguir, é apresentada uma revisão da teoria e da literatura, começando pelo NLP, antes de se apresentar a solução proposta nesta dissertação.

3.2 Processamento de Linguagem Natural

O NLP foca-se na interação entre computadores e a linguagem humana, visando permitir que os computadores compreendam e interpretem esta forma de comunicação, assemelhando-se aos seres humanos [22].

Sendo este um campo de investigação com muitos anos de pesquisa, foi em 1966 que surgiu um dos primeiros sistemas notáveis desta área, desenvolvido por Joseph Weizenbaum e apresentado ao mundo como ELIZA [48]. Este sistema foi um dos primeiros a demonstrar fielmente as capacidades e o potencial do NLP, simulando uma conversa terapêutica.

Nas décadas seguintes, houve avanços significativos em várias áreas dentro do NLP, incluindo análise sintática [12], análise semântica [16], tradução automática [23] e muito mais. Além disso, com o aumento exponencial do poder computacional nos anos subsequentes, impulsionou-se a criação de sistemas mais capazes e robustos. Esta característica foi importante, uma vez que possibilitou a aplicação de técnicas de ML e o uso de redes neuronais [19], permitindo então a aprendizagem de modelos a partir de grandes volumes de dados. Com esta nova abordagem, a realização de tarefas ainda mais complexas tornou-se viável, permitindo assim construir sistemas de sumarização de texto [32], análise de sentimento [40] e muito mais.

Atualmente, esta área de pesquisa continua a contar com uma grande comunidade que colabora ativamente para o seu avanço. Um exemplo deste tipo de contribuições é a biblioteca *Natural Language Toolkit* (NLTK)[1], uma biblioteca de referência em Python para processamento de linguagem natural. A mesma foi desenvolvida originalmente por Steven Bird e Edward Loper [7], oferecendo uma variedade de ferramentas e recursos para lidar com tarefas comuns de NLP.

3.2.1 Principais técnicas de NLP

Para compreender um pouco mais em que se baseia o NLP, é importante conhecer algumas das suas principais técnicas. A seguir, são apresentadas algumas das funcionalidades oferecidas pela biblioteca NLTK.

1. **Tokenização**: A tokenização é o processo de dividir o texto em unidades menores, como palavras ou frases. Tomemos o exemplo a seguir:

   ```
   Frase: A tokenização é um passo importante para o processamento de
   linguagem natural.

   ['A', 'tokenização', 'é', 'um', 'passo', 'importante', 'para', 'o',
   'processamento', 'de', 'linguagem', 'natural', '.']
   ```

2. ***Stemming***: O *stemming* é usado para reduzir as palavras à sua forma raiz, ou seja, são removidos os sufixos e prefixos das palavras mantendo apenas o radical. Através

[1]NLTK Python: `https://www.nltk.org/`

desta técnica, é possível auxiliar a normalização das palavras, especialmente em tarefas como recuperação de informações e classificação de texto.

```
Palavra: correndo
Stemming: corr
```

3. **Lematização**: A lematização, ou em inglês *lemmatization*, também é um processo de simplificação das palavras, reduzindo as mesmas à forma de lema.

```
Palavra: correndo
Lematização: correr
```

Apesar de parecerem iguais, importa notar que a lematização considera a categoria gramatical da palavra ao produzir o seu lema, o que a torna mais precisa que o stemming.

4. ***Part-of-Speech Tagging* (POS Tagging)**: O POS Tagging é usado para rotular as partes do discurso num texto. Cada palavra é classificada com a sua respetiva categoria gramatical, como artigo (ART), verbo (V), substantivo (N), adjetivo (ADJ), preposição (PREP), entre outros.

```
Frase: A tokenização é um passo importante para o processamento de
linguagem natural.

[(A, ART), (tokenização, N), (é, V), (um, ART), (passo, N),
(importante, ADJ), (para, PREP), (o, ART), (processamento, N),
(de, PREP), (linguagem, N), (natural, ADJ), (., .)]
```

5. ***Chunking***: Esta funcionalidade é usada para identificar e extrair partes do texto com diferentes funções sintáticas de uma frase como sujeito, predicado, complemento direto, complemento indireto, etc. Tomemos o seguinte exemplo:

```
Frase: A tokenização é um passo importante para o processamento de
linguagem natural.

[ A tokenização ] [ é ] [ um passo ] [ importante ] [ para ]
[ o processamento ][ de linguagem natural ]
```

Cada grupo de palavras nos parênteses retos representa um *chunk* identificado na frase original. Os mesmos representam uma unidade significativa da frase, como um sintagma nominal ou verbal. Este exemplo demonstra como o *chunking* pode ser aplicado para identificar partes importantes de uma frase, facilitando a análise e a extração de informações.

6. ***Named Entity Recognition* (NER)**: O NER é usado para identificar entidades nomeadas num texto, como nomes de pessoas, organizações, locais, etc.

```
Frase: Bill Gates é o fundador da Microsoft.
[('Bill Gates', 'PERSON'), ('Microsoft', 'ORGANIZATION')]
```

7. **WordNet**: O WordNet é uma base de dados lexical amplamente utilizada que fornece informações sobre relações entre palavras, sinónimos, antónimos, entre outros.

```
Palavra: carro
[(automóvel), (veículo), (automóvel), (veículo a motor),
(veículo automotor), (carruagem), (transporte), (auto),
(carreta), (automóvel), (carroça), (veículo rodoviário),
(carrinha), (autocarro), (ônibus), (camião), (táxi),
(carro de passeio), (carro de turismo)]
```

Identificadas que estão estas funcionalidades básicas do NLP, é fácil entender o seu papel fundamental em sistemas de gestão de diálogo usando linguagem natural, visto que é por meio destas técnicas que conseguimos dotar a máquina de organizar e consequentemente "compreender" todo o texto introduzido.

Dito isto, e voltando agora ao início deste milénio, decorria o ano 2001 quando se reafirmou um conceito que havia acompanhado o NLP e que até então passava um pouco

despercebido, denominado como "chatter bot". Nesse mesmo ano, foi apresentada ao público uma nova forma de programação destes agentes virtuais, que viriam mais tarde a ser apelidados de "chatbots", surgindo então o sistema ALICE e a sua base de funcionamento, o AIML [47]. O nome da entidade em questão revelava facilmente do que se tratava: "Artificial Linguistic Internet Computer Entity", que marcou um avanço significativo na interação homem-máquina. Esta "entidade" foi desenvolvida por Richard Wallace em 1995 [46] e foi uma das primeiras tentativas de criar um *chatbot* capaz de simular uma conversa humana de maneira convincente, uma vez que até então tudo se voltava para técnicas de NLP.

Podemos dizer então, que o desenvolvimento do sistema ALICE gerou um interesse acrescido por tecnologias de NLP e IA. A partir daí, o desenvolvimento de *chatbots* tornou-se um campo de investigação e inovação contínuo permitindo que hoje em dia seja comum a utilização em *websites* destes agentes virtuais, vulgarmente designados de "bots", para tornarem mais natural a interação com os utilizadores e permitirem atender a questões, comentários e pedidos formulados pelos utilizadores na forma de texto aberto, i.e. usando linguagem natural escrita. Empresas e investigadores exploraram diversas abordagens para melhorar a inteligência e a capacidade de comunicação destes agentes virtuais. Para tal, e para entender este passo importante, na subsecção seguinte são apresentadas algumas características desta tecnologia que tanto influenciou e continua a influenciar o mundo.

3.3 *Chatbots*

Começando por definir o que são os *chatbots*, podemos afirmar com certeza que são programas baseados em NLP projetados para simular interações humanas por meio de mensagens de texto ou de voz. O funcionamento do programa envolve a interpretação das mensagens de entrada dos utilizadores e, por meio de algoritmos de NLP, responder da maneira mais apropriada consoante a sua lógica. Podemos observar com mais detalhe todo o processo mencionado na figura 3.1.

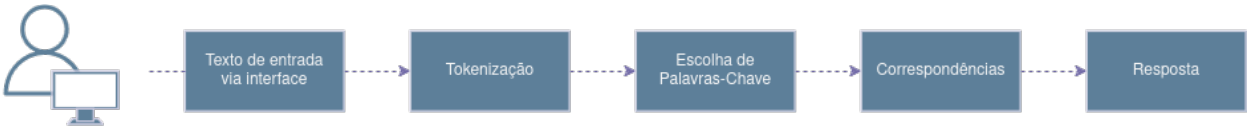


Figura 3.1: Arquitetura típica de um *chatbot* [6].

O funcionamento de um *chatbot* começa pela interação do utilizador por meio de uma

interface (e.g. texto ou voz). Esta é a responsável por permitir a introdução do texto de entrada. Como primeiro passo de processamento, temos a tokenização cujo objetivo é separar o texto em partes menores, como vimos na subsecção anterior e, em seguida, fazer uma análise das palavras principais, removendo assim as "stop words" que não transmitem um significado relevante para as frases. Após esta simplificação do texto, tenta-se encontrar possíveis correspondências que possam fazer sentido para o texto de entrada, sendo este passo também conhecido como "bot logic" e, quando formulada, é retornada a resposta para a interface.

Agora, com base na compreensão do modo de funcionamento dos *chatbots*, é fácil imaginar um programa deste tipo a responder a comandos específicos ou até mesmo a perguntas frequentes e, dependendo da lógica da construção de respostas, a conseguir manter conversas naturais.

De facto, a utilização destes agentes virtuais ao longo dos anos tem-se disseminado em setores como atendimento ao cliente, comércio eletrónico, serviços bancários, saúde e muito mais [37].

3.3.1 Principais técnicas e abordagens para *chatbots*

Um passo importante para a total compreensão da melhor abordagem de implementação para o sistema de diálogo pretendido é explorar as operações que regem um *chatbot*. Assim, nesta subsecção, apresentam-se algumas das principais técnicas e abordagens usadas desde o início até à atualidade na programação destes agentes virtuais.

Parsing

Após uma análise às técnicas de NLP, verifica-se que o *parsing* [14] é uma técnica fundamental desta área. Esta envolve a análise do texto de entrada e a sua manipulação para extrair informações relevantes. No contexto dos *chatbots*, o *parsing* é essencial para compreender as mensagens dos utilizadores e gerar respostas apropriadas.

Pattern Matching

O *Pattern Matching* [26] é uma técnica comum em *chatbots*, sendo amplamente utilizada em sistemas de perguntas e respostas. Esta consiste em identificar padrões nas mensagens dos utilizadores e correspondê-los a respostas predefinidas. Esta abordagem é eficaz para lidar com questões diretas e frequentes.

AIML

O AIML [47] é uma linguagem de marcação de inteligência artificial amplamente utilizada em *chatbots* desde os seus primórdios. Esta fornece uma estrutura para definir padrões de entrada e as suas respostas correspondentes, permitindo que os *chatbots* processem e respondam a uma variedade de entradas.

Chat Script

Quando os padrões definidos em AIML não correspondem à entrada do utilizador, os *chatbots* podem recorrer à abordagem de Chat Script [4]. Esta técnica foca-se na construção de respostas por defeito usando uma sintaxe eficiente, garantindo uma interação coerente mesmo em situações imprevistas.

Structured Query Language (SQL) e Base de Dados

A integração de SQL e bases de dados permite que os *chatbots* memorizem conversas anteriores e recuperem informações relevantes durante interações subsequentes. Esta abordagem contribui para uma experiência mais personalizada e contextualizada para os utilizadores.

Cadeias de Markov

As cadeias de Markov [29] são utilizadas na construção de respostas probabilísticas em *chatbots*. Esta abordagem baseia-se na probabilidade de ocorrência de cada letra ou palavra no conjunto de dados textual, gerando respostas mais aplicáveis e, consequentemente, mais precisas.

Redes Neuronais

As redes neuronais [43] desempenham um papel crucial na compreensão e geração de texto de forma contextualmente relevante em *chatbots*. Através da utilização destas, é possível obter relações semânticas complexas entre palavras e frases, possibilitando conversas naturais e fluídas entre o *chatbot* e o utilizador.

Ao analisar estas diferentes técnicas, é possível concluir que, a partir da sua génese de funcionamento, um *chatbot* pode ser categorizado em dois tipos: "Scripted Chatbots" [29, 48, 46], onde todas as respostas são previamente programadas consoante as entradas, como o sistema ALICE; e os "Artificially Intelligent Chatbots" [45, 39] que recentemente

captaram a atenção do mundo pela grande naturalidade e assertividade de diálogo. Estes últimos, através da aprendizagem com base num conjunto de dados, ou seja, um *dataset*, e considerando o texto de entrada como texto de contexto, conseguem gerar as suas próprias respostas. Importa notar que a qualidade destas depende significativamente do *dataset* bem como da quantidade de texto de contexto que o modelo consegue considerar. Estes *chatbots* são "alimentados" por IA, estando então associados a modelos de linguagem pré-treinados. Como tal, é necessário compreender como estes modelos funcionam.

3.3.2 Modelos de Linguagem Pré-Treinados

O estudo de Modelos de Linguagem tem uma história mais longa do que muitos imaginam, estando enraizado no campo do NLP. No entanto, a sua popularidade disparou com a publicação do artigo "Attention Is All You Need" [44]. Antes disso, os modelos de linguagem enfrentavam desafios significativos devido à necessidade de grande poder computacional. Por isso, os resultados muitas vezes não eram os melhores, pois perdiam o contexto em conversações, levando a imprecisões. A justificação para o sucedido prendia-se com a arquitetura usada até então, que era baseada em técnicas *sequence to sequence*.

O referido artigo propôs uma abordagem inovadora para a geração de texto, introduzindo um mecanismo de atenção. Com este mecanismo, o modelo podia concentrar-se em partes relevantes da sequência de entrada, melhorando assim a capacidade de capturar relações de longo alcance em sequências, sem depender exclusivamente da *Recurrent Neural Network* (RNN) ou da *Convolutional Neural Network* (CNN). Assim, surgiu a arquitetura Transformer, composta principalmente por camadas de codificação e descodificação. No contexto das tarefas de NLP, o codificador processa a entrada, enquanto o descodificador gera a saída.

Os modelos mais conhecidos que utilizam este tipo de arquitetura são o BERT ("Bidirectional Encoder Representations from Transformers") [15], GPT ("Generative Pre-trained Transformer") [36], RoBERTa ("Robustly Optimized BERT Approach") [25], entre outros.

3.3.3 *Chatbots* comerciais

Como já foi mencionado anteriormente, a utilização de *chatbots* continua a despertar cada vez mais interesse pelas empresas, uma vez que, com esta tecnologia, conseguem reduzir os humanos necessários no atendimento ao cliente. A prova de que esta utilização já está consolidada na atualidade é a frequência com que interagimos com este tipo de tecnologias

num *site* de vendas *online* qualquer. Por norma, o *chatbot* consegue responder às nossas questões e muitas vezes ajudar-nos eficazmente e, quando não consegue, somos encaminhados para um atendimento personalizado assegurado por uma pessoa (e não um *chatbot*).

Para além deste benefício, a implementação de sistemas de *chatbots* tem sido facilitada por grandes empresas tecnológicas como Microsoft, Google, Amazon, entre outras. Estas empresas, ao perceberem o potencial destas tecnologias, passaram a oferecer serviços para a construção de "Scripted Chatbots" para diversos cenários, sem a necessidade de grandes conhecimentos de programação. Para além dos serviços *cloud* mencionados anteriormente, esta solução é amplamente popular em diferentes canais de comunicação, como WhatsApp, Messenger, Instagram, entre outros, pela facilidade na sua implementação, bem como os seus reduzidos custos.

Ainda no contexto da comercialização de *chatbots*, e com o aumento da popularidade dos modelos de linguagem pré-treinados mencionados na secção 3.3.2, certas empresas tecnológicas também oferecem a oportunidade de usar estes modelos para a criação de "Artificially Intelligent Chatbots", muitas vezes categorizados como "assistentes". Para utilizar estes modelos, é necessário inserir um conjunto de instruções que o sistema tem de cumprir, dando assim o contexto necessário ao determinado modelo de linguagem. Ao contrário dos serviços de "Scripted Chatbots", para a implementação destes sistemas, já é preciso ter algum conhecimento de programação, pois a comercialização é feita através do uso de API. Uma vez que a utilização destes modelos requer um elevado poder computacional, para garantir a qualidade no tempo de resposta, esta é a maneira mais simples e indicada. Um exemplo dessa comercialização é a OpenAI, conforme mencionado na secção 2.2.1.

3.4 Conceção do sistema proposto

Tendo em conta toda a revisão da literatura aqui apresentada, cujo objetivo foi orientar o melhor caminho para a conceção dos dois modos de funcionamento definidos na secção 3.1, conclui-se que, a solução passa pela implementação de dois *chatbots* distintos, uma vez que em ambas as situações é necessária uma resposta interativa, i.e., uma resposta consoante um texto de entrada.

É apresentada detalhadamente a seguir a solução adotada para cada sistema de gestão de diálogo.

3.4.1 Modo Narrativa

Considerando os requisitos específicos deste modo e priorizando a originalidade constante das histórias contadas pelo sistema de diálogo, bem como o cumprimento de todos os parâmetros introduzidos pelo terapeuta, que variam de idoso para idoso, a melhor solução é a implementação de um "Artificially Intelligent Chatbot", "alimentado" por um modelo de linguagem pré-treinado (ver secção 3.3.2). Como demonstrado na secção 3.3.3, a utilização comercial desta tecnologia é ideal para o caso de estudo, ao assegurar qualidade e eficiência nos tempos de resposta, o que no caso da criação do zero deste modelo, não seria garantido. O modelo escolhido é o da OpenAI, especificamente o `gpt-3.5-turbo-1106` como apresentado na secção 2.2.1 (página 6).

Optou-se por este *chatbot* comercial, porque possibilita obter uma variedade de narrativas fornecendo instruções específicas sobre como criar a história conforme os parâmetros introduzidos pelo terapeuta. Desta forma, o sistema consegue gerar histórias personalizadas e adaptadas às necessidades individuais de cada idoso, cumprindo um dos principais requisitos apresentados na secção 3.1. Até ao momento, poucas empresas no mercado apresentam a capacidade de criação de agentes capazes de seguir instruções, uma vez que a disponibilização destes serviços é recente e ainda apresenta algum trabalho pela frente. Contudo, na procura por alternativas deste *chatbot*, surgiu a hipótese de utilização da API da groqcloud[2], conhecida por fornecer respostas de grandes modelos de linguagem em tempos recordes. Esta alternativa seria proveitosa para o nosso caso de estudo, dado que o tempo de geração das interações é um fator crítico. Infelizmente, e como já referido, esta empresa ainda não disponibiliza a criação de agentes que cumpram instruções personalizadas e como tal, por enquanto, não é uma solução viável.

Definido o *chatbot* a ser usado, e para garantir que o terapeuta introduza os parâmetros de cada história e realize paragens de emergência e pausas, haverá uma interface específica para o mesmo, permitindo o controlo total destas funcionalidades.

A Figura 3.2 apresenta a arquitetura do sistema de gestão de diálogo para o modo narrativa, onde consta a organização dos diferentes módulos e a sua interligação.

Conforme apresentado na figura, a solução proposta inclui quatro entidades distintas: Terapeuta, Robô GrowMu, Cloud e Utilizador Final Idoso. Estas entidades interagem com o robô para cumprir os requisitos funcionais e não funcionais exigidos por este modo.

[2] `https://groq.com/`

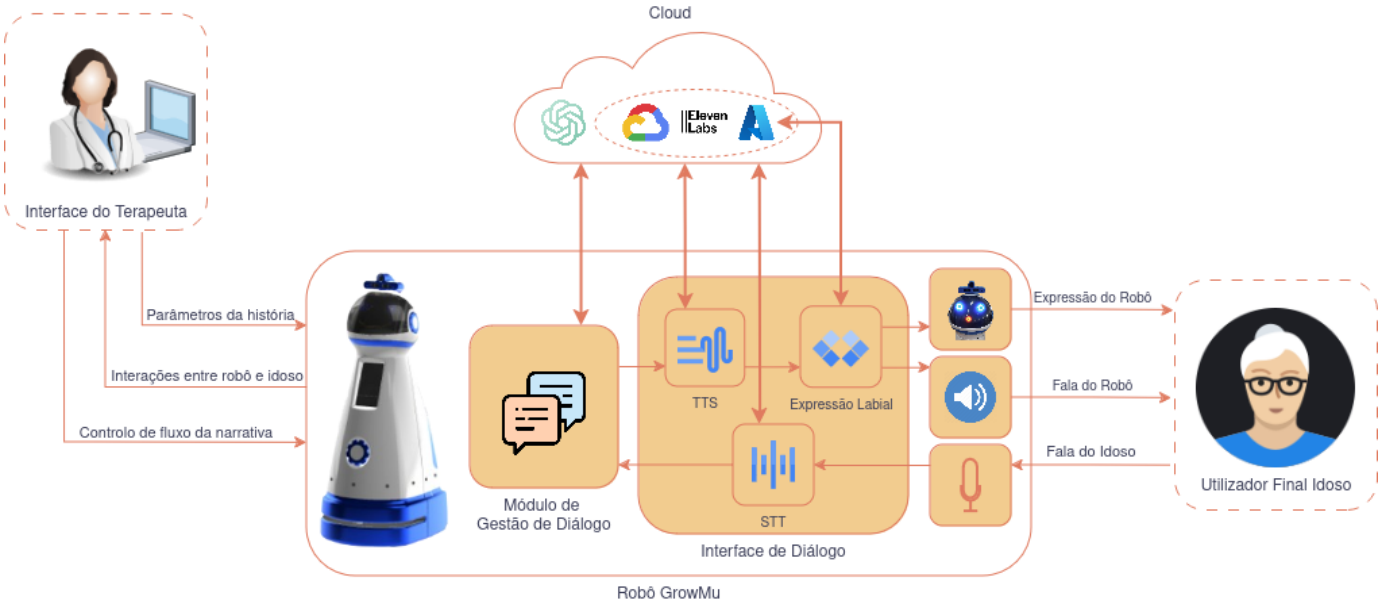


Figura 3.2: Arquitetura do sistema de gestão de diálogo para o Modo Narrativa.

- **Interação entre Terapeuta e Robô:** O terapeuta utiliza uma interface específica que permite definir todos os parâmetros da história e controlar o fluxo da narrativa, como pausas e paragens de emergência. Além disso, o terapeuta recebe em primeira instância, através desta interface transcrições de todas as interações ocorridas entre o idoso e o robô.

- **Interação entre Robô e Cloud:** A conexão entre o robô e a Cloud é essencial, ao ser a mesma responsável por "alimentar" os seus diferentes módulos de *software*, como a gestão de diálogo e a interface de diálogo. O módulo de gestão de diálogo é especificamente suportado pela API do assistente ChatGPT, conforme mencionado anteriormente, e a interface de diálogo permite utilizar diferentes serviços, como Azure, Google e Eleven Labs, permitindo ao operador do sistema optar por diferentes prestadores para as suas funcionalidades, i.e., síntese de fala (TTS) disponível em todos os prestadores, conversão de fala para texto (STT) disponível apenas na Azure e na Google, e a sincronização labial, sendo esta uma característica exclusiva do TTS da Azure, como apresentado na secção 2.2. No caso da utilização do TTS de outros prestadores, esta sincronização da expressão labial e do áudio é feita através do cálculo da energia do sinal, como será apresentado na secção 3.5.

- **Interação entre Robô e Utilizador Final:** A interação entre o robô e o utilizador final idoso é realizada pelo *hardware* que constitui o robô. A fala gerada pela interface de diálogo é reproduzida através das colunas de som e acompanhada pela matriz *Light*

Emitting Diode (LED) da boca (ver Figura 3.3b), permitindo assim a representação das expressões labiais em conjunto com o áudio. A captação da fala pelo idoso é feita através do microfone incorporado no robô.

(a) Cabeça do robô GrowMu

(b) Matriz LED do robô GrowMu para representação de expressões labiais

Figura 3.3: Imagens da cabeça do robô GrowMu.

Desta forma, o funcionamento do sistema começa com a interação do Terapeuta, através da sua interface, em que define e envia os parâmetros da história para o robô. De seguida, já no robô, a partir do módulo de gestão de diálogo, é formulado o início da narrativa vinda da Cloud, que depois é convertida em fala pelo módulo de TTS na interface de diálogo, e realizada a sincronização labial com o áudio. A interação é expressa através da reprodução do mesmo pelas colunas de som do robô, em conjunto com as respetivas expressões labiais na sua matriz de LED. Quando terminada a sua intervenção, o utilizador final pronuncia a sua resposta, que consequentemente é captada pelo microfone e transcrita para texto pela interface de diálogo usando o STT. A interação do utilizador em texto é recebida pelo módulo de gestão de diálogo, sendo devolvida a continuação da história, ou seja, resposta do robô, consoante a interação do idoso. Durante todo o processo, a interface do terapeuta recebe as transcrições de todas as interações do robô e do idoso. O modo de funcionamento descrito decorre até a história terminar, e finaliza com a interação do robô com o idoso.

3.4.2 Modo Livre

Para o modo livre, a melhor solução passa pela implementação de um sistema de diálogo através da linguagem AIML, construindo assim um "Scripted Chatbot". A possível utilização comercial deste *chatbot*, independentemente de ser uma solução bastante atrativa, não pode ser aplicada uma vez que após alguma pesquisa percebe-se facilmente que este tipo de solução não é vantajoso ao ser pensado apenas para fins comerciais como referido na secção 3.3.3.

Já a criação de modelos de linguagem, ou seja, "Artificially Intelligent Chatbots", também não é uma solução viável na janela temporal deste trabalho de dissertação. A conceção de um modelo deste género implicaria a criação de um *dataset* específico, visto haver a necessidade de total controlo da entidade do robô. Contudo, a utilização da versão comercial deste *chatbot* poderá ser útil para o caso de quando o texto de entrada não correspondesse ao script pré-programado. Como tal, em vez de o sistema se limitar a uma resposta padrão, como sugerido pela técnica "Chat Script" (ver secção 3.3.1), poder-se-á utilizar um modelo de linguagem em substituição. A resposta dada poderá ainda ser adicionada ao conjunto de respostas originalmente programadas em AIML, permitindo que o "Scripted Chatbot", aprenda com um modelo de linguagem, reduzindo assim os custos de utilização deste sistema ao longo do tempo. No entanto, tendo em conta a limitação temporal para o término deste trabalho de dissertação, esta abordagem fica para trabalho futuro.

A Figura 3.4 apresenta a arquitetura do sistema de gestão de diálogo para o modo livre, onde consta a organização dos diferentes módulos e a sua interligação.

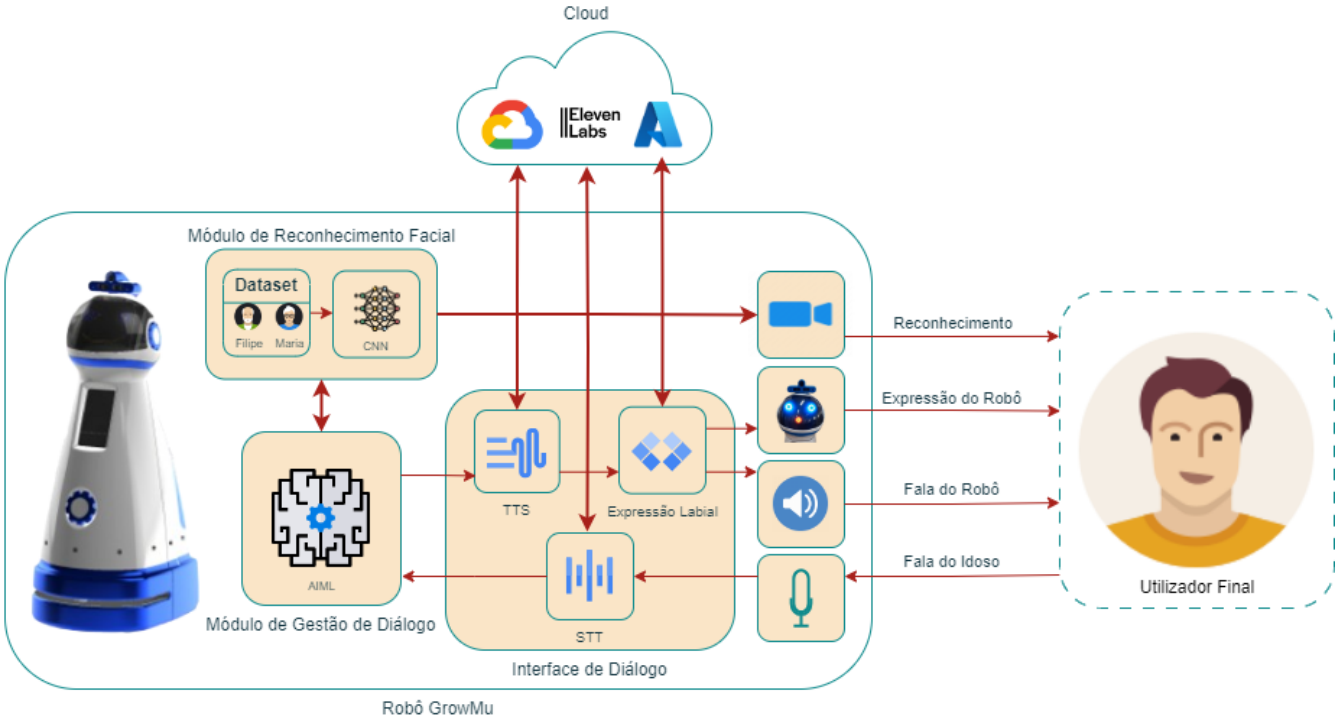


Figura 3.4: Arquitetura do sistema de gestão de diálogo para o Modo Livre.

Para este modo, como a figura ilustra, contamos com três entidades distintas: Robô GrowMu, Cloud e Utilizador Final. Estas entidades interagem com o robô para cumprir os

requisitos funcionais e não funcionais exigidos por este modo.

- **Interação entre Robô e Cloud:** Para este modo, a conexão entre o robô e a Cloud é apenas responsável por alimentar a interface de diálogo. À semelhança do outro modo, a Cloud, que fornece diferentes serviços como Azure, Google e Eleven Labs, permite a escolha dos diferentes prestadores para as suas funcionalidades, i.e., síntese de fala (TTS) disponível em todos os prestadores, conversão de fala para texto (STT) disponível apenas na Azure e na Google, e a sincronização labial, sendo esta uma característica exclusiva do serviço TTS da Azure, como foi apresentado na secção 2.2. No caso da utilização do serviço de TTS de outros prestadores, esta sincronização é feita através do cálculo da energia do sinal, como será apresentado na secção 3.5.

- **Interação entre Robô e Utilizador Final:** A interação entre o robô e o utilizador final é realizada pelo *hardware* que constitui o robô. A fala gerada pela interface de diálogo é reproduzida através das colunas de som e acompanhada pela matriz LED da boca, permitindo assim a representação das expressões labiais em conjunto com o áudio. A captação da fala do utilizador final é feita através do microfone incorporado no robô e a captação de imagem para reconhecimento facial do utilizador é feita através da câmara do robô.

O funcionamento deste sistema encontra-se em permanente escuta através do microfone do robô até à interação do utilizador final ser captada. Enquanto é realizada a captação do áudio, é utilizada a câmara para enviar *frames* do interlocutor para o módulo de reconhecimento facial baseado em redes neuronais, que, tendo sido treinado previamente com base num *dataset*, é feita a correspondência. A fala é transcrita para texto usando o STT da interface de diálogo, e o texto resultante é recebido pelo módulo de gestão de diálogo. Conforme a programação realizada em AIML, é devolvida uma resposta consoante o texto de entrada, bem como a correspondência recebida do módulo de reconhecimento facial. Caso a pessoa seja conhecida, o diálogo é baseado nas informações previamente guardadas referente a essa pessoa; caso seja uma pessoa desconhecida pelo sistema (ou não reconhecida), o diálogo será baseado em perguntas para conhecer a pessoa. Com as respostas vindas do módulo de gestão de diálogo, as mesmas são sintetizadas pelo módulo TTS da interface de diálogo, e realizada a sincronização labial com o áudio resultante. A interação é realizada através da reprodução do áudio pelas colunas de som do robô, em conjunto com as respetivas expressões labiais na sua matriz de LED. O modo de funcionamento decorre desta forma até o utilizador terminar a conversa com uma das seguintes despedidas, “Adeus” ou “até amanhã”.

3.5 Expressão labial do robô sincronizada com a fala

Uma vez concluída a conceção do sistema de gestão de diálogo, a capacidade do robô de se comunicar com o utilizador final está assegurada, uma vez que, através dos serviços de STT e TTS disponíveis na *cloud*, é possível a reprodução da interação gerada por este sistema em resposta à transcrição de fala do utilizador humano. Contudo, para completar este importante módulo de expressividade do robô, é necessário dotá-lo com a capacidade de sincronização labial [34, 17] com o áudio.

A sincronização labial é um elemento essencial para tornar a interação por meio do diálogo mais natural, tanto em avatares quanto em robôs físicos. A precisão na sincronização entre os movimentos dos lábios e o áudio falado é fundamental para criar uma experiência de comunicação mais natural e envolvente com os utilizadores, uma vez que a sincronização da boca ajuda a manter a atenção destes na interação com o robô. Assim, pode-se afirmar que a interação humano-robô pode ser aprimorada utilizando-se informações visuais em conjunto com a voz sintetizada, promovendo uma maior aceitação por parte dos utilizadores. Essa informação visual é denominada visema [42], que consiste na representação visual da menor unidade sonora de uma língua, um fonema [24], que, quando articulado e combinado com outros fonemas, constituem sílabas, palavras e frases.

Os visemas desempenham um papel essencial na melhoria da inteligibilidade da fala em sistemas de comunicação visual. A literatura existente demonstra que a combinação de faixas auditivas e visuais melhora significativamente a compreensão da fala, e que a ausência dessa habilidade pode conduzir a uma desconexão percetiva do utilizador [27]. Além disso, estudos indicam que a expressividade facial e a comunicação não verbal são componentes críticos para a eficácia dos robôs sociais [**?**, 13]. Essas capacidades não melhoram apenas a clareza da comunicação, mas também contribuem para o resultado desejado de interação, o mais natural possível.

Tendo em conta a plataforma robótica GrowMu utilizada para este trabalho de dissertação, a implementação desta funcionalidade é realizada por meio de uma matriz LED, como a figura 3.3b ilustra. Esta matriz LED permite mostrar diferentes visemas.

Uma vez que esta característica está totalmente correlacionada com o áudio sintetizado pelo TTS, é necessária uma boa sincronização entre o áudio da fala e os visemas apresentados. Portanto, a abordagem a ser adotada dependerá do prestador de serviços *cloud* utilizado no módulo de TTS.

Como foi referido no capítulo 2.2, o serviço TTS da Microsoft Azure já inclui a fun-

cionalidade de mapeamento de visemas, retornando *IDs* e tempos específicos dos mesmos, conforme os fonemas correspondentes pelo IPA [5]. Com esses dados, é possível realizar a sincronização labial com o sintetizador específico da Azure, implementando uma função de *callback* que modifique a expressão labial.

Já nos outros prestadores de serviços, esta funcionalidade não está disponível. Em último caso, seria apenas necessário a disponibilização dos tempos de cada fonema pelos serviços que sintetizam as vozes. A partir disto, bastaria converter o texto sintetizado em fonemas, e corresponder o visema específico ao tempo obtido pela *cloud*. Contudo, todos os outros prestadores de serviços exceto a Azure, não disponibilizam esta informação, que poderia ser útil para todos os seus clientes e que não envolveria grande trabalho por parte destas empresas, uma vez que a síntese da fala é baseada em fonemas e, portanto, esta é uma informação que as empresas já possuem certamente.

Posto isto, é necessária a procura por outra técnica para alcançar esta sincronização, restando apenas a consulta à literatura existente sobre algoritmos de sincronização labial. Tendo em conta o trabalho de Cintas *et al.* [13], a solução proposta envolve a análise do sinal de áudio resultante do TTS. Esta análise, contudo, não pode ser demasiado detalhada, i.e., ao nível de identificação de fonemas, pois esse processo exigiria um algoritmo complexo alimentado por redes neuronais. Por isso, a solução concebida neste trabalho parte por uma análise simplificada da energia do sinal do áudio. Importa então notar que esta alternativa contará com uma redução drástica nos visemas a serem representados comparativamente à solução usada no TTS da Microsoft, uma vez que não se conseguirá obter os visemas específicos dos fonemas a serem apresentados.

A expressão matemática usada para calcular a energia do sinal de áudio é:

$$E = \sum_{n=0}^{N-1} |x[n]|^2,$$

onde:

- $x[n]$ representa o valor do sinal numa trama de sinal de N amostras;
- N é o número total de amostras na trama do sinal;

A partir dos diferentes valores de energia, são associados diferentes visemas que representam adequadamente a abertura e fechamento da boca durante a reprodução do áudio, possibilitando uma sincronização labial precisa em tempo real.

3.6 Sumário

Neste capítulo, foi definido detalhadamente o problema brevemente apresentado na secção 1.2, definindo cuidadosamente os requisitos funcionais e não funcionais dos diferentes modos de funcionamento. Foi realizada uma revisão da teoria e da literatura sobre sistemas de gestão de diálogo, abordando conceitos como NLP, *chatbots* e respetivamente as suas principais técnicas de funcionamento, procurando assim a melhor solução para a implementação deste sistema.

Com base nos resultados desta revisão, foram concebidas duas arquiteturas para os dois modos de funcionamento, detalhando cada módulo presente na sua constituição, justificando com base nos requisitos mencionados anteriormente.

No capítulo 4, abordaremos com maior detalhe todo o processo de implementação do sistema de diálogo aqui apresentado, focando nos conhecimentos técnicos e nas tarefas de engenharia desenvolvidas para implementar a solução projetada.

4 Implementação do sistema de gestão de diálogo

Concluída a conceção do sistema de gestão de diálogo (ver capítulo 3), é detalhado a seguir todo o processo da sua implementação, i.e. a programação dos diferentes módulos que compõem a arquitetura do sistema nos seus dois modos de funcionamento.

Em ambos os casos de estudo, é necessária a implementação da interface que garante a compreensão e expressividade com os utilizadores finais através da plataforma robótica. Por isso, é apresentada em primeiro lugar a implementação da interface de diálogo.

4.1 Interface de diálogo

Conforme mostrado no capítulo anterior, para ambos os modos de funcionamento é necessário uma interface de diálogo que assegure a capacidade de transcrição da fala do utilizador final para texto e uma sincronização labial com o áudio da conversão do texto gerado em fala.

Para melhor organização desta interface, foi criada uma biblioteca Python intitulada `speech_library.py` que conta com todos os serviços necessários para esta interface, i.e., STT e TTS, este último incluindo a sincronização labial. Desta forma, ao importar esta biblioteca, é possível chamar as funções nelas implementadas pelo sistema de gestão de diálogo quando necessário.

4.1.1 Módulo STT

Este módulo é responsável pela transcrição da fala do utilizador em texto. Com base na análise apresentada na secção 2.2.6 (página 10), decidiu-se utilizar os serviços da Microsoft para esta funcionalidade.

Para utilizar a API da Microsoft, foi necessário criar uma conta Azure[1] e, de seguida, um recurso da categoria "Speech Services" para obtenção das chaves secretas. Para a implementação deste módulo, foi criada uma classe `SpeechToText` contendo a função `microsoft(self, language: str) -> str`, onde o parâmetro de entrada `language` contém a informação da língua a ser reconhecida (e.g., "pt-PT", "en-US").

A implementação do STT utilizando a biblioteca Python `Speech SDK`[2] é realizada da seguinte maneira:

Primeiro, configura-se a ligação com o serviço de *cloud* utilizando as informações obtidas aquando da criação do recurso na conta Azure. Depois define-se a língua de transcrição e define-se a configuração do áudio, selecionando o microfone a ser usado e inicializando o reconhecedor a partir dessa configuração:

```
1 speech_config = speechsdk.SpeechConfig(subscription="Speech_Key", region="
    westeurope")
2 speech_config.speech_recognition_language = language # Parametro de
    entrada (pt-PT ou en-US)
3 audio_config = speechsdk.audio.AudioConfig(device_name="default")
4 speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config
    , audio_config=audio_config)
```

A transcrição é obtida através da função assíncrona *recognize_once_async()*, confirmando de seguida a validade da conversão, usando o código apresentado no anexo C, página 72. Caso a conversão seja bem-sucedida, o texto da transcrição é devolvido possibilitando a sua utilização noutros módulos do sistema de gestão de diálogo.

4.1.2 Módulo TTS

Este módulo é responsável pela conversão de texto em fala, servindo como meio de comunicação do sistema de gestão de diálogo com o utilizador final. Considerando a informação apresentada na secção 2.2.6 (página 10), decidiu-se possibilitar ao sistema implementado a utilização como prestadores de serviços para esta funcionalidade a Microsoft, a Google e a ElevenLabs.

A implementação deste módulo contou com a criação da classe `TextToSpeech`, contendo diversas funções específicas para cada prestador de serviços.

[1] `https://portal.azure.com/`

[2] `https://pypi.org/project/azure-cognitiveservices-speech/`

Microsoft

À semelhança do STT apresentado anteriormente (ver secção 4.1.1), para a implementação deste serviço é necessária uma conta Azure. Uma vez que estes passos já foram dados anteriormente, basta apenas a utilização das mesmas configurações, i.e., a mesma chave secreta e região selecionada. Para o TTS deste prestador de serviços foi criada a função `microsoft(self, text: str, language: str, rate: str) -> None:`, onde os parâmetros de entrada `text`, `language` e `rate` contêm o texto a ser sintetizado, a língua a ser usada e a variável de controlo de velocidade da síntese de voz.

Utilizando a biblioteca `Speech SDK`, a implementação deste serviço é feita da seguinte maneira:

Primeiro, configura-se a ligação com a *cloud* utilizando as informações obtidas aquando a criação do recurso na conta Azure e define-se a linguagem de síntese. Em seguida, define-se a voz a ser utilizada para a síntese de fala consoante a linguagem escolhida:

```
speech_config = speechsdk.SpeechConfig(subscription="Speech_Key", region="
    westeurope")
speech_config.speech_synthesis_language = language# Parametro de entrada (
    pt-PT ou en-US)
voice_name = 'en-US-JennyNeural' if language == "en-US" else 'pt-PT-
    FernandaNeural'
speech_config.speech_synthesis_voice_name = voice_name

speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=
    speech_config)
```

Cria-se um documento em formato *Speech Synthesis Markup Language* (SSML) para ser possível a alteração da velocidade de síntese da fala. A possibilidade de controlo desta característica é importante para o modo narrativa, uma vez que é necessária a adaptação da velocidade da fala da plataforma robótica consoante o nível de défice cognitivo do utilizador final idoso. Atribuindo valores percentuais positivos ou negativos à variável `rate`, a fala é acelerada ou desacelerada, respetivamente, em comparação com o áudio original (e.g. com -20.00%, o áudio é sintetizado vinte por cento mais lento).

```
ssml_doc = f"""
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis" xml:lang
    ="{language}">
     <voice name="{voice_name}">
```

```
        <prosody rate="{rate}%">
            {text}
        </prosody>
    </voice>
</speak>
"""
```

Por fim, realiza-se a síntese e a reprodução de fala usando a função assíncrona *speak_ssml_async()*, usando o código apresentado no anexo C, página 72.

Google

Como na Microsoft, para a utilização do TTS da Google, houve necessidade de se criar uma conta na Google Cloud[3] e, consequentemente, um projeto na mesma para a obtenção das chaves secretas, permitindo assim a utilização da sua API. Criou-se a função `google(self, text: str, language: str) -> None`, onde os parâmetros de entrada `text` e `language` contêm o texto a ser sintetizado e a língua de pronunciação.

A implementação do TTS utilizando a biblioteca Python `google-cloud`[4] é realizada da seguinte maneira:

Primeiro, inicializa-se o cliente de *Text-to-Speech* utilizando a chave secreta gerada aquando a criação da conta Google Cloud:

```
client = texttospeech.TextToSpeechClient.from_service_account_json('/path/
    to/your/service_account.json')
```

Configura-se o texto de entrada, a seleção de voz consoante a língua a utilizar e as configurações do áudio e seguidamente, é gerada a síntese de áudio usando o código apresentado no anexo C, página 72. Importa notar que, ao contrário da Microsoft, esta biblioteca não possui um sintetizador próprio, como foi referido na secção 2.2.6 (página 10), sendo que o áudio terá de ser reproduzido de outra forma. Esta parte está diretamente relacionada com a sincronização labial que será apresentada mais adiante (ver secção 4.1.3).

ElevenLabs

Para este prestador de serviços, também foi criada uma conta[5]. No entanto, por esta ser destinada apenas a serviços de voz, o processo de criação da chave secreta da API foi bas-

[3] https://cloud.google.com/

[4] https://pypi.org/project/google-cloud/

[5] https://elevenlabs.io/

tante simplificado, sendo logo atribuída no momento da sua criação. Para a implementação do TTS, apesar de existir uma biblioteca Python[6], no momento da escrita desta dissertação a biblioteca *elevenlabs* ainda apresenta algumas limitações, como a impossibilidade de reprodução de vozes em língua portuguesa de Portugal, tornando inviável a sua utilização neste projeto de dissertação.

Posto isto, utilizou-se a biblioteca `Requests`[7] realizando a ligação ao URL associado à voz de síntese da ElevenLabs escolhida.

```
url = "https://api.elevenlabs.io/v1/text-to-speech/pMsXgVXv3BLzUgSXRplE"
headers = {
    "Accept": "audio/mpeg",
    "Content-Type": "application/json",
    "xi-api-key": "API_KEY"
}
```

Ao realizar a chamada POST com os devidos parâmetros, i.e. o URL e dados do áudio escolhidos, é possível a geração da síntese de fala como apresentado no anexo C, página 72. À semelhança do TTS da Google, o processo de reprodução do áudio gerado terá de ser reproduzido por meio de outra forma. Esta parte está diretamente relacionada com a sincronização labial, que será apresentada na seguinte secção 4.1.3.

4.1.3 Implementação de Sincronização Labial

Como já foi mencionado várias vezes ao longo desta dissertação, a sincronização labial está diretamente relacionada com os serviços TTS utilizados. Foram desenvolvidas duas abordagens distintas para realizar a referida sincronização: uma para usar o TTS da Microsoft e outra para usar o TTS da Google e da ElevenLabs.

Ambas as abordagens utilizam a matriz LED do robô GrowMu para simular as expressões labiais (ver Figura 3.3b, página 27). Portanto, em todos os serviços, é necessário fazer uso dessa matriz. Sendo o ambiente de desenvolvimento da plataforma robótica o *Robot Operating System* (ROS)[35], a forma de executar o pretendido é através do uso de `Publishers`, ou seja, um publicador que envia mensagens a um tópico específico, neste caso, à matriz LED, subscrito pelo *driver monarch* desenvolvido em C++ pelo fabricante do robô e compilado em Ubuntu 20.04 e ROS Noetic para este projeto de dissertação.

[6] `https://pypi.org/project/elevenlabs/0.1/`

[7] `https://pypi.org/project/requests/#description`

Os tópicos do robô são definidos por este *driver*. No desenvolvimento desta biblioteca específica, é preciso identificar o tópico da boca, chamado `/mouth_shape`, e atribuir o `Publisher` a esse tópico específico. Assim, no início da criação da classe `TextToSpeech`, é necessário usar a função `__init__` para criar o `Publisher`. O código resultante é apresentado a seguir:

```
def __init__(self):
    rospy.init_node('speech_node', anonymous=True)
    self.mouth_pub = rospy.Publisher('mouth_shape', UInt8, queue_size=10)
```

A partir da implementação desta função, é possível agora a qualquer momento utilizar a linha de código abaixo para alterar a expressão labial do robô através do *ID* de cada visema (num número inteiro).

```
self.mouth_pub.publish("ID_VISEMA")
```

Garantida a representação da expressão labial do robô, é apresentado a seguir com mais detalhe as duas abordagens adotadas para o processamento desta representação.

Microsoft

Sendo a Microsoft o único prestador de serviços capaz de fornecer diretamente informações de visemas com base no áudio gerado, a implementação na função `microsoft` da classe `TextToSpeech` anteriormente apresentada (ver secção 4.1.2), requer a adição de uma função *callback* na função principal `microsoft` da seguinte maneira:

```
def viseme_cb(evt) -> None:
    viseme_id_converted = {0: 0, 1: 20, 2: 17, 3: 11, 4: 18, 5: 18, 6: 21,
        7: 16, 8: 22, 9: 17, 10: 11, 11: 17, 12: 20, 13: 20, 14: 14, 15:
       13, 16: 20, 17: 21, 18: 15, 19: 14, 20: 21, 21: 12}
    if evt.viseme_id in viseme_id_converted:
        self.mouth_pub.publish(viseme_id_converted[evt.viseme_id])

speech_synthesizer.viseme_received.connect(viseme_cb)
```

A função `viseme_cb` é chamada sempre que um evento de `viseme` é recebido pelo objeto `speech_synthesizer`. O dicionário `viseme_id_converted` mapeia o ID dos visemas disponíveis na API da Microsoft para o conjunto de ID específicos que controlam a expressão labial do robô. Quando um visema é recebido, o *ID* correspondente é publicado no tópico `mouth_shape` através do *publisher* `mouth_pub`.

Os visemas disponíveis inicialmente na plataforma robótica eram em menor número do que os disponibilizados pela API da Microsoft, que utiliza os visemas baseados nos fonemas correspondentes ao IPA [5] como foi discutido na secção 3.5 (página 30). Para contar com resultados o mais fidedignos possíveis, foram implementados novos visemas ao *driver* do robô. A tabela de correspondência entre os visemas da Microsoft e os visemas do robô segundo os diferentes fonemas do IPA, pode ser consultada no anexo B (página 69).

O `viseme_id_converted`, baseado na tabela anteriormente referida, associa então os *IDs* de visemas provenientes da Microsoft aos implementados no robô. Importa destacar que esta correspondência não está completa devido às limitações técnicas da matriz, resultando na necessidade de, em algumas situações, usar o mesmo visema do robô para diferentes fonemas do IPA.

Um caso específico desta limitação ocorre quando determinados fonemas necessitam da presença dos dentes para emitir o som, como [s] e [z] (notação segundo o IPA). O número de LED na boca do robô torna a resolução da matriz LED insuficiente para representar virtualmente essa característica na expressão do robô. Este exemplo é ilustrado na figura 4.1.

(a) Representação real do fonema [s], [z].

(b) Representação no robô do fonema [s], [z].

Figura 4.1: Comparação entre a representação real e a representação possível no robô do fonema [s], [z].

Google e ElevenLabs

Ao contrário da Microsoft, os prestadores de serviços Google e ElevenLabs não fornecem diretamente informações de visemas com base no áudio gerado. A solução apresentada na secção 3.5 para estes prestadores de serviços passa pela análise da energia do sinal de áudio. Após receber o áudio, na classe `TextToSpeech` é necessária a criação de uma nova função que receba como parâmetros de entrada os dados do áudio e o seu formato e calcule a energia do sinal em tramas.

Para isso, implementou-se a função `__analyze_audio_energy(self, audio_data: np.ndarray, sample_rate: int, language: str) -> None`, onde os parâmetros de entrada são os dados do áudio, a taxa de amostragem e o idioma da gravação. A partir desta função, é possível calcular a energia e identificar visemas com base nesta análise, da seguinte forma.

Inicializam-se os parâmetros como `frame_length`, `energy_limits` e `hop_length` com base na taxa de amostragem do áudio e do idioma e normaliza-se os dados de áudio para facilitar o cálculo de energia. A seguir, é iniciada uma *thread* separada para reproduzir o áudio, garantindo processamento simultâneo sem bloquear o fluxo principal de execução. O excerto de código correspondente a estas tarefas é apresentado no anexo C, página 72.

Enquanto é reproduzida a faixa sonora, itera-se sobre os dados de áudio normalizados em tramas, calcula-se a energia para cada trama, determinando o visema apropriado com base em limiares de energia predefinidos (`energy_limits`) e publica-se o resultado do visema. O uso da última linha do código a seguir apresentado, `time.sleep(hop_length/sample_rate)` faz com que o ciclo `for` pause a sua execução por um tempo correspondente ao avanço de uma janela de análise no tempo real do áudio. Isso é crucial para sincronizar o processamento do áudio com o tempo real, garantindo que o processamento de cada trama ocorra de maneira contínua e sincronizada com a reprodução do áudio. É importante notar que caso o cálculo da energia fosse realizado inteiramente antes da reprodução, resultaria em grandes tempos de espera até o início da reprodução do áudio.

```
1 for i in range(0, len(audio_array_normalized) - frame_length, hop_length):
2     energy_val = sum(abs(audio_array_normalized[i:i+frame_length])**2)
3     viseme = 0
4     if energy_val <= energy_limits[0]:
5         viseme = 0
6     elif energy_limits[0] < energy_val < energy_limits[1]:
7         viseme = 20
```

```
8       elif energy_limits[1] <= energy_val < energy_limits[2]:
9           viseme = 18
10      else:
11          viseme = 17
12      self.mouth_pub.publish(viseme)
13      time.sleep(hop_length/sample_rate)
```

Ainda referente a este último trecho de código, conseguimos entender como tinha sido indicado na secção 3.5 (página 30), uma redução drástica no número de visemas usados para esta solução, limitando a utilização a 4 visemas conforme ilustrado na Figura 4.2.

(a) Visema 0. (b) Visema 20. (c) Visema 18. (d) Visema 17.

Figura 4.2: Representação dos visemas usando o *TTS* da Google e ElevenLabs.

Quanto maior for o valor da energia do sinal, maior é a abertura da boca do robô, emulando assim de forma simplificada a pose dos "lábios" do robô durante a fala. Importa notar que os valores limite de energia no *array* `energy_limits` foram definidos através da realização de testes e analisando os valores máximos de cada voz.

4.2 Sistema de Gestão de Diálogo

Finalizada a implementação da interface de diálogo, foi necessário implementar o sistema responsável por gerar as interações. Conforme foi definido na secção 3.4 (página 24), existem dois modos de funcionamento.

4.2.1 Modo Narrativa

Para a implementação deste sistema de diálogo, foi desenvolvida uma biblioteca Python, `game_library`, que realiza todo o processamento e geração de interações. Ao importar esta biblioteca, o robô consegue utilizar as funções implementadas para a geração da narrativa.

Utilizando a API do assistente da OpenAI, conforme foi apresentado na secção 3.4 (página 24), foi necessária a criação de uma conta na OpenAI *Platform*[8] e, consequentemente, a criação de um assistente. Foi definido um conjunto de instruções específicas para o *chatbot* construir a narrativa segundo os requisitos apresentados anteriormente:

- Criar uma história interativa simples;
- Limitar cada interação a 80 palavras para evitar que o idoso se perca na narrativa;
- Adaptar a história ao nível de défice cognitivo, utilizando vocabulário apropriado;
- Oferecer duas opções para escolha do utilizador em cada continuação da história;
- Limitar a história a um máximo de 6 interações para não se tornar demasiado longa.

Após a criação do assistente, foi necessário obter a chave da API do mesmo para integrá-la no sistema através da biblioteca Python `openai`[9].

Na biblioteca criada para o sistema de gestão de diálogo (`game_library`), foi implementada a classe `StoryTelling`, onde se realiza a conexão com o assistente, da seguinte forma:

```
def __init__(self):
    self.client = openai.OpenAI(api_key="API_KEY")
    self.thread = self.client.beta.threads.create()
```

Havendo a necessidade do envio dos parâmetros da história bem como a fala do utilizador final para o assistente da OpenAI, criaram-se as funções `define_parameters` e `new_message` que podem ser consultadas no anexo C, página 72.

Para se obter a resposta do assistente com tempos de resposta o mais curtos possível, foi implementada a funcionalidade de *streaming*, i.e. a capacidade da resposta começar a ser reproduzida através do módulo TTS mesmo antes de ser totalmente processada pelo servidor.

```
def obtain_response(self) -> None:
    """Obtain response from the OpenAI assistant."""

    event_handler = EventHandler()
    with self.client.beta.threads.runs.create_and_stream(
```

[8] https://platform.openai.com/playground
[9] https://pypi.org/project/openai/

```
        thread_id=self.thread.id,
        assistant_id="ID_ASSISTANT",
        event_handler=event_handler,
    ) as stream:
        stream.until_done()

    while(1):
        if(event_handler.control == False):
            break

    print("\n")
    tts.microsoft(event_handler.text_final, "pt-PT", self.rate)

    event_handler.all_text += event_handler.text_final

    return event_handler.all_text
```

Para tal, criou-se a classe `EventHandler`, para lidar com os eventos do assistente da OpenAI e garantir a referida funcionalidade, através do código apresentado no anexo C, página 72.

A partir do momento em que uma frase é finalizada, dá-se início a um processo em paralelo para reprodução da fala usando o TTS. Assim, enquanto a geração da restante interação é terminada, é feita a síntese de vez da primeira frase.

Interface Gráfica do Terapeuta

Para um melhor planeamento desta interface e se garantir que todos os requisitos fossem cumpridos, foi realizado um esboço da interface gráfica do terapeuta com as diversas funcionalidades, que pode ser consultado no anexo A (página 66).

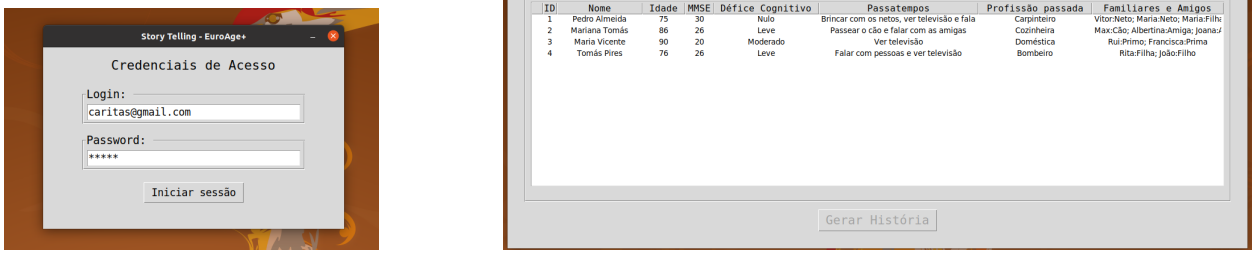
Esta interface foi projetada para ser executada num computador pessoal ou institucional, necessitando de comunicação com o robô GrowMu. Para tal, as mesmas foram implementadas utilizando o protocolo *Transmission Control Protocol/Internet Protocol* (TCP/IP), pois o terapeuta e o robô estarão na mesma sala e, portanto, compartilharão o mesmo ponto de acesso à 'internet'. Além disso, para garantir que esta interface possa ser usada em qualquer computador das variadas instituições, foi criado um servidor que contém as informações dos idosos, assegurando que, em caso de troca de computadores, os dados não sejam perdidos.

Para tal, foram desenvolvidos três `scripts` distintos: `client.py`, que contém toda a programação desta interface gráfica; `server.py`, responsável por armazenar as informações

de todos os utentes das instituições; e, por fim, o `robot.py`, que, utilizando as bibliotecas apresentadas neste capítulo, garante toda a narração da história.

Entrando agora em mais detalhes na implementação da interface gráfica do terapeuta (`client.py`), a mesma foi realizada através da utilização da biblioteca nativa do Python, `tkinter`[10].

A interface é inicializada com um painel de autenticação, onde as credenciais de cada instituição são inseridas, permitindo a escalabilidade do sistema. Após a autenticação bem-sucedida, a primeira página a ser exibida é a de criação de nova história, onde o terapeuta pode introduzir os diversos parâmetros e criar a narrativa. As seguintes janelas gráficas são apresentadas na Figura 4.3.



(a) Janela de autenticação.

(b) Janela da criação de nova história.

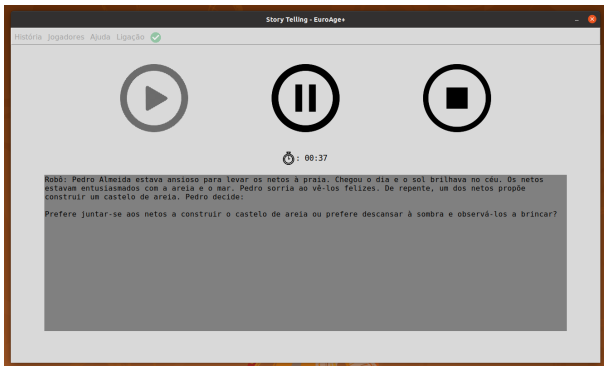
Figura 4.3: Janela de autenticação e criação de nova história.

Ao clicar no botão "Gerar História" na janela de criação de nova história, as informações são enviadas para o robô que, utilizando o sistema de gestão de diálogo anteriormente apresentado, consegue interagir com o *chatbot* da OpenAI. A história começa quando o terapeuta clica no botão de reprodução (ver Figura 4.4a), permitindo pausar ou parar a narração conforme necessário. As interações entre o robô e o utilizador final idoso são exibidas nesta mesma janela, conforme ilustrado na Figura 4.4a. Para adicionar ou editar informações de idosos na base de dados, pode-se utilizar a janela mostrada na Figura 4.4b.

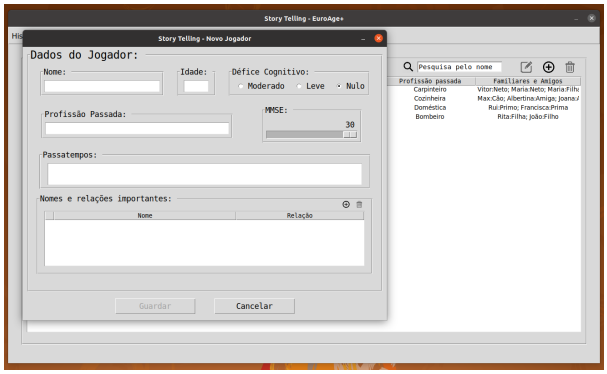
4.2.2 Modo Livre

A implementação do sistema de diálogo para este modo de funcionamento está diretamente relacionado com a identificação de pessoas, pois a sua formulação de interações

[10] `https://pypi.org/project/tk/`



(a) Janela de interação da história gerada.



(b) Janela de adição ou edição de jogadores.

Figura 4.4: Janela de interação da história e de edição da informação de jogadores.

depende desta variável, como foi apresentado na secção 3.4.2 (página 27).

Reconhecimento Facial

Para este módulo foi usada a biblioteca Python `face_recognition`[11], uma biblioteca que conta com grande apreciação por parte da comunidade pela sua grande precisão de análise e simplicidade de utilização.

Para garantir que a comunicação entre este módulo e o sistema de diálogo seja possível, à semelhança do que foi realizado para a exibição de expressões labiais na matriz LED do robô, criou-se um ROS *publisher* para publicar no tópico `/face_names` permitindo assim a subscrição e consequentemente a leitura no sistema de diálogo dos resultados do reconhecimento facial.

Posto isto, foi criado um pacote ROS para a possível utilização do mesmo em outras situações, onde primeiramente, criou-se uma biblioteca Python `simple_facerec` para simplificar ainda mais a utilização da "face_recognition", reduzindo-a a simples funções (o código desta biblioteca pode ser consultado no anexo C, página 72).

Noutro script, realizou-se o reconhecimento da pessoa da seguinte maneira. Primeiramente criou-se o ROS *publisher* e inicializou-se o objeto `sfr` da biblioteca `simple_facerec`, e por fim, fez-se o carregamento das imagens do dataset bem como o `load` da câmara através da biblioteca `OpenCV`[12].

```
1 rospy.init_node('face_recognition_publisher')
2 face_names_pub = rospy.Publisher('face_names', String, queue_size=10)
3
```

[11] https://pypi.org/project/face-recognition/
[12] https://pypi.org/project/opencv-python/

```
sfr = SimpleFacerec()
sfr.load_encoding_images('/path/to/images')

cap = cv2.VideoCapture(0)
```

Evitando um subprocessamento por parte do robô, é necessária a realização do reconhecimento facial apenas quando o serviço de STT esteja a ser usado, ou seja quando o utilizador estiver a falar. Para isso, foi criado um *subscriber* que recebe informações do sistema de diálogo para executar o reconhecimento apenas durante esses momentos. O tratamento dos dados recebidos é feito por uma função *callback*, da seguinte maneira:

```
rospy.Subscriber("flag_get_recognition", String, callback)

def callback(data):
    global flag
    rospy.loginfo("Recebido: %s", data.data)
    flag = True if data.data == "Inicia" else False
```

Entrando agora num ciclo infinito até o *shutdown* do ROS, sempre que a *flag* da *callback* for verdadeira, é realizado o processo de reconhecimento facial usando o código apresentado no anexo C, página 72.

Programação em AIML

Assegurado o módulo de reconhecimento facial, é preciso criar o guião deste *Scripted Chatbot* utilizando a plataforma de desenvolvimento *Pandorabots*[13].

Esta programação é derivada da linguagem *XML* e portanto é regida por categorias e *templates*. Abaixo está um exemplo de como pode ser implementada a resposta para diversos cumprimentos.

Primeiramente, cria-se um ficheiro *.set*, onde são inseridos diversos cumprimentos.

```
[["Boa Tarde"], ["Bom dia"], ["Boa noite"], ["Ola"], ["Oi"]]
```

Utilizando este ficheiro "alimenta-se" o *pattern* com estas hipóteses, gerando a resposta conforme o conteúdo da variável *nome*.

```
<category>
    <pattern><set>cumprimentos</set></pattern>
```

[13] https://home.pandorabots.com/home.html

```
    <template>
        <condition name="nome">
            <!-- Se nao conhecer -->
            <li value="Unknown">
                <srai>1</srai>
            </li>
            <!-- Se conhecer -->
            <li>
                Ola <get name="nome"/>! Como estas hoje?
            </li>
        </condition>
    </template>
</category>
```

Finalizada a programação dos ficheiros *.aiml*, os mesmos são carregados por um *script* Python através da biblioteca *Pyaiml*[14].

```
k = Kernel()

k.learn_aiml("/PATH/udc.aiml")
k.learn_aiml("/PATH/apresentacao.aiml")
k.init_new_user("1")

k.load_set("cumprimentos", upload_sets("/PATH/cumprimentos.set"))
k.load_set("apresentacao", upload_sets("/PATH/apresentacao.set"))
```

Entrando num ciclo infinito como no código apresentado no anexo C (página 72), o sistema assegura que o chatbot responda adequadamente utilizando reconhecimento facial para personalizar a interação com o utilizador.

4.3 Sumário

Neste capítulo, detalhou-se todo o processo de implementação dos dois modos do sistema de gestão de diálogo, apresentando os detalhes técnicos mais relevantes e as tarefas de engenharia realizadas. Esta implementação materializou a conceção do sistema de diálogo apresentado no capítulo 3 (página 13).

Abordou-se os mais variados módulos de funcionamento do robô tais como a interface de diálogo, o sistema de gestão de diálogo, interface gráfica do terapeuta no modo de narração

[14] https://pypi.org/project/pyaiml21/

interativa e o módulo de reconhecimento facial usado no modo livre.

No próximo capítulo, será apresentada a validação experimental destes sistemas através da avaliação realizada por diversos utilizadores finais.

5 Validação experimental

Para avaliar a eficácia do sistema de gestão de diálogo desenvolvido para cada modo de funcionamento, foram realizadas sessões de teste em diferentes ambientes. No modo narrativa, as sessões foram conduzidas no centro de dia da Cáritas Diocesana de Coimbra (CDC)[1], envolvendo idosos e terapeutas ocupacionais, proporcionando um contexto o mais próximo possível da realidade para a qual este modo foi projetado. Já no modo livre, o mesmo foi testado num ambiente laboratorial, onde se teve como objetivo analisar a funcionalidade do sistema de gestão de diálogo, no estado preliminar de desenvolvimento em que se encontra.

5.1 Modo Narrativa

Para a realização do teste piloto deste modo, foram realizadas sessões de narração de histórias interativas a utilizadores finais idosos, nas quais as terapeutas ocupacionais estiveram presentes para a condução de cada sessão. A duração de cada sessão variou entre 10 e 15 minutos, dos quais cerca de 7 minutos foram dedicados à narração da história e o tempo restante à resposta a um questionário por parte da pessoa idosa.

O principal objetivo foi verificar a boa usabilidade do sistema na perspetiva dos utilizadores finais idosos, bem como dos terapeutas. Pretendeu-se no teste piloto testar as seguintes hipóteses:

Hipótese 1: Os utilizadores finais idosos avaliam o sistema de gestão de diálogo com elevada usabilidade.

Hipótese 2: Os terapeutas ocupacionais avaliam o sistema de gestão de diálogo com elevada usabilidade.

Como ambas as hipóteses se centram na usabilidade do sistema, o formulário usado para os utilizadores idosos e para os terapeutas foi uma adaptação da conhecida *System Usability Scale* (SUS) [11], um questionário com dez itens avaliados por meio de uma escala *Likert*

[1] `https://caritascoimbra.pt/`

com cinco níveis, para uma avaliação subjetiva da usabilidade do sistema.

Houve formulários distintos para os utilizadores idosos e para os terapeutas ocupacionais, uma vez que para os idosos este teria de ser mais simples devido às suas possíveis limitações cognitivas. No caso dos idosos, o questionário foi respondido no final da sessão. Aos terapeutas foi entregue para preenchimento outro questionário no final da realização de todas as sessões.

5.1.1 Participantes

Uma vez que para este sistema de diálogo é necessária uma prévia avaliação do nível de défice cognitivo dos idosos, a equipa de terapeutas da CDC, instituição esta que se disponibilizou para colaborar neste projeto de dissertação, aplicou o MMSE [18] aos idosos aos seus cuidados no centro dia, selecionando aqueles que eram elegíveis para jogar o jogo sério, i.e. idosos que apresentassem no teste um resultado superior a 15 (metade da escala).

Assim, foram selecionados 15 idosos, com escolaridade, como sendo elegíveis para a participação no piloto, dos quais 2 apresentavam nível de défice cognitivo Moderado (entre 15 a 20 pontos no MMSE), 10 com nível de défice cognitivo leve (21 a 26 pontos no MMSE) e 3 com nível de défice cognitivo nulo (27 a 30 pontos no MMSE).

5.1.2 Realização das Sessões

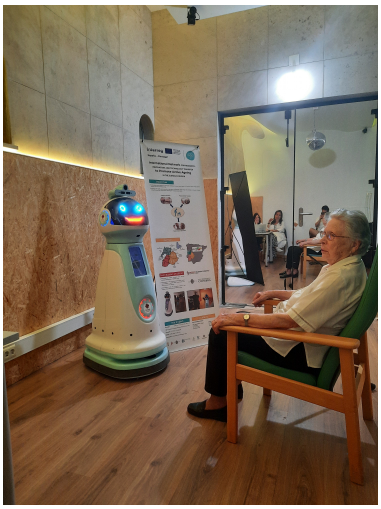
Todas as sessões foram realizadas numa sala cedida pela CDC conforme ilustra a Figura 5.1. Foi utilizado um robô GrowMu que a CDC já possuía do projeto anterior GrowMeUp [33], no qual a instituição também participou como parceira no projeto.

Os idosos sentaram-se à frente do robô como na Figura 5.1b, de modo a favorecer ao máximo a interação entre ambos e melhorar a captação do som. As terapeutas ocupacionais foram dispostas lateralmente conforme mostra a Figura 5.1a, permitindo que interviessem com os idosos a qualquer momento da história usando os mecanismos de controlo disponíveis na sua interface.

Antes do início da sessão, foi explicado ao participante idoso o que iria acontecer, informando-o de que uma história seria contada e ele deveria responder às perguntas feitas pelo robô sobre a continuidade da mesma.

Cada utilizador final idoso teve um tema específico, bem como um conjunto de parâmetros único, contendo as suas informações pessoais, permitindo assim que o sistema de diálogo cria-

(a) Disposição da sala.



(b) Idoso e Robô GrowMu.

Figura 5.1: Imagens das sessões realizadas na Cáritas Diocesana de Coimbra.

se histórias personalizadas. Estes dados foram introduzidos pelo terapeuta através da sua interface que, em caso de necessidade, perguntava algumas informações específicas ao idoso.

É possível consultar com detalhe uma sessão de teste, onde o sistema gera uma narrativa dirigida a um idoso com nível de défice cognitivo leve, no anexo E, página 83.

5.1.3 Resultados

Após as experiências, foi realizada uma análise das respostas aos formulários de avaliação preenchidos pelos idosos e pelos terapeutas. Os formulários, bem como as respostas de ambos os grupos, podem ser consultados em detalhe no anexo D, página 77.

Utilizadores Idosos

Para os utilizadores idosos, no geral, ou seja sem qualquer distinção do seu nível de défice cognitivo, foram obtidos os resultados apresentados na Figura 5.2, em que as colunas representam a média do *score* obtido em cada pergunta do questionário e as setas o intervalo de variação medido através do desvio padrão.

As perguntas 1 (“Gostaria de jogar este jogo com frequência”), 2 (“Acho o jogo interessante e fácil de compreender”), 5 (“A história contada manteve-me interessado e atento o tempo todo”), 8 (“Senti bem-estar ao ouvir a história contada pelo robô”) e 9 (“Senti-me confortável enquanto interagia com o robô”) são as que mais evidenciam a elevada usabilidade do sistema pelos utilizadores finais idosos demonstrando resultados bastante positivos. Ainda em relação

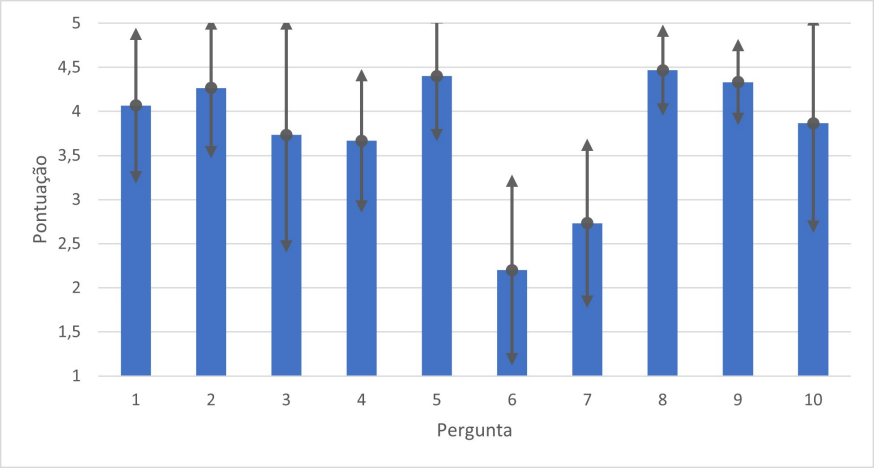


Figura 5.2: Resultados gerais para idosos (SUS — Adaptado).

à usabilidade, a pergunta 6 ("Achei que o jogo tinha muitas limitações ou problemas") obteve uma pontuação média de 2,2 ± 1,08, confirmando os resultados positivos das perguntas anteriormente referidas.

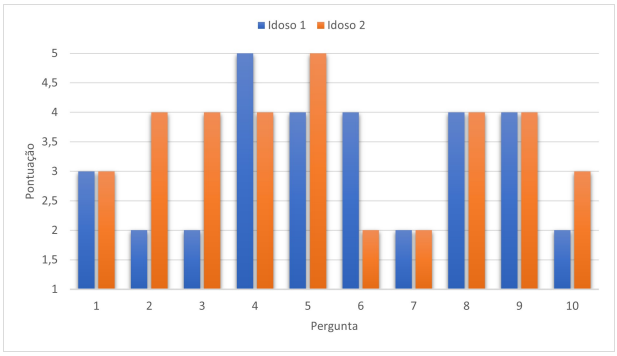
Embora a capacidade de audição não tenha sido diretamente testada durante a seleção da população idosa para realização do teste piloto, a avaliação da compreensão auditiva, medida pela pergunta 3 ("Compreendi com facilidade o que o robô dizia"), revelou uma pontuação média de 3,73 ± 1,33. A avaliação menos positiva desta característica, que apresenta alguma disparidade nas respostas dos idosos, pode estar relacionada com a ausência de uma avaliação prévia da capacidade auditiva.

É notável que as perguntas que obtiveram maior consenso e elevada pontuação foram as 8 e 9. podendo-se concluir que um dos grandes objetivos deste sistema foi cumprido, pois toda a interação entre robô e idoso mostrou-se natural e satisfatória.

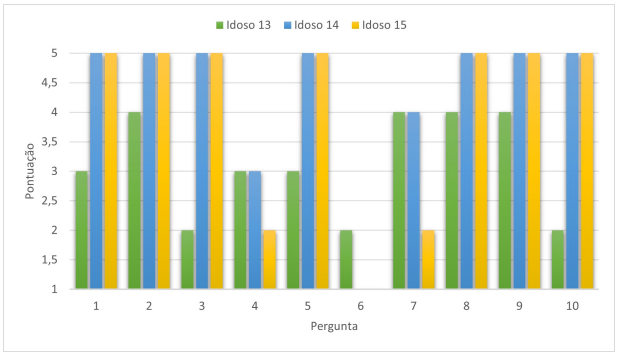
Outra pergunta que merece destaque é a 10 ("Consegui seguir facilmente a história que o robô contou"), com uma avaliação média de 3,87 ± 1,246. Esta pergunta apresentou a maior divergência de respostas, independentemente do seu resultado positivo. Isto pode ser explicado por uma característica relevante do sistema desenvolvido: nenhuma das histórias tinha um tema semelhante e, por vezes, os temas, sendo um pouco mais complexos e não tão bem especificados pelo terapeuta, dificultavam a criação de histórias pelo sistema.

Ao analisar a avaliação realizada pelos idosos com diferentes níveis de défice cognitivo, conforme é apresentado na Figura 5.3, não é possível concluir com clareza se a usabilidade do sistema é avaliada de maneira distinta entre os diferentes níveis de défice cognitivo devido à insuficiência de participantes nos níveis nulo e moderado. Contudo, observando esses mesmos resultados nas Figuras 5.3a e 5.3b, percebe-se uma tendência de que os idosos

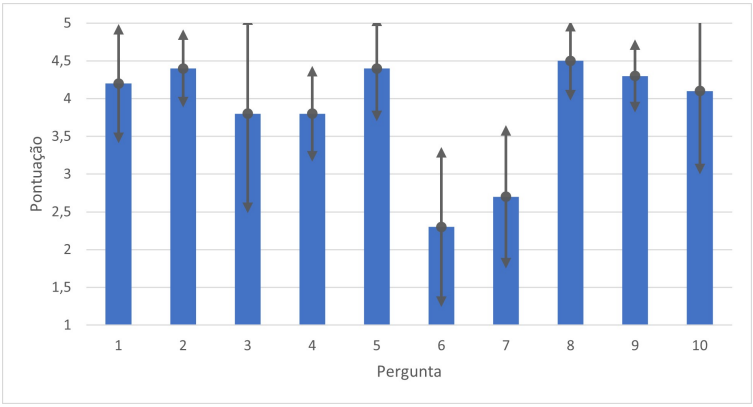
com nível de défice cognitivo moderado apresentam maior dificuldade em usar o sistema, independentemente dos esforços do mesmo para a adaptação da narrativa para este nível mais delicado. É importante notar que alguns destes pacientes já apresentam algum tipo de demência, justificando os resultados obtidos.



(a) Nível moderado (SUS — Adaptado).



(b) Nível nulo (SUS — Adaptado).



(c) Nível leve (SUS — Adaptado).

Figura 5.3: Resultado SUS — Adaptado para os diferentes níveis de défice cognitivo.

Para os idosos com nível de défice cognitivo leve, conforme apresentado na Figura 5.3c, a pontuação é semelhante à pontuação agregada de todos os idosos anteriormente apresentada, uma vez que estes constituem a maioria da população, resultando em avaliações médias similares. Comparando com os idosos de nível moderado, observa-se uma melhoria notável, enquanto que comparando com os de nível nulo há uma semelhança nos resultados, embora estes últimos apresentem uma avaliação um pouco melhor.

Pode-se concluir que, de forma geral, a avaliação pelos utilizadores idosos é bastante positiva, embora não seja possível correlacionar conclusivamente os diferentes níveis de défice cognitivo e determinar a afetividade do sistema. Contudo, como referido anteriormente, pode ser observada uma pequena tendência de melhoria da avaliação da usabilidade do sistema

consoante o menor nível de défice cognitivo.

Terapeutas Ocupacionais

A avaliação da usabilidade do sistema por parte dos terapeutas ocupacionais pode ser visualizada graficamente na Figura 5.4.

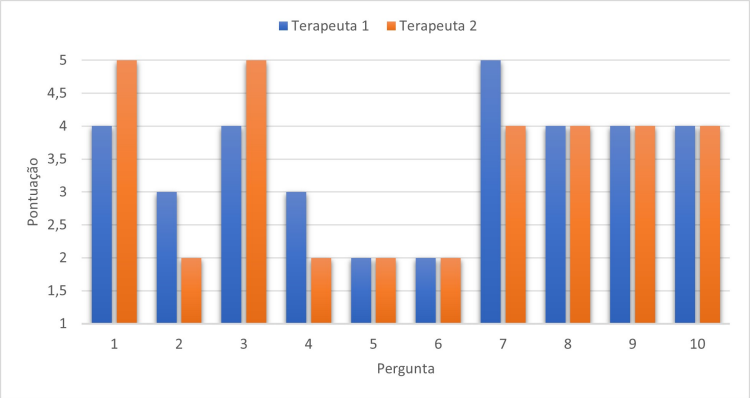


Figura 5.4: Resultados de avaliação dos terapeutas ocupacionais (SUS — Adaptado).

Para este questionário, as perguntas 1 ("Gostaria de utilizar este jogo com frequência com os idosos à minha responsabilidade") e 7 ("Acredito que a maioria dos terapeutas e cuidadores, em geral, conseguiriam usar facilmente o jogo com os seus idosos") são as que mais evidenciam a elevada usabilidade do sistema. A pergunta 2 ("Considero o jogo mais complexo do que necessário para o fim a que se destina") é posta um pouco em causa pelo terapeuta 1, identificando-se como neutro quanto à complexidade do mesmo. Contudo, na pergunta 9 ("Acho que o jogo pode ser usado na implementação de estratégias terapêuticas para ajudar os idosos à minha responsabilidade") e 6 ("Acho que o jogo apresenta muitas inconsistências e limitações durante o seu uso com os idosos"), confirma-se positivamente a usabilidade deste sistema como ferramenta terapêutica para todas as diversas instituições de cuidados a idosos.

Outros dados importantes são a avaliação positiva da capacidade do sistema moldar a história conforme o nível de défice cognitivo do idoso (pergunta 8), bem como o interesse que os idosos manifestaram pela história contada pelo robô (pergunta 10). A interface destinada aos terapeutas também foi avaliada positivamente (pergunta 3), encorajando-os a avaliar positivamente a usabilidade deste sistema sem a ajuda de um técnico (pergunta 4). Esta característica é bastante importante, uma vez que é de grande interesse garantir que este sistema possa ser utilizado com grande facilidade pelos utilizadores finais.

Por último, importa notar que o aspeto que apresentou uma avaliação negativa foi a forma de falar do robô (pergunta 5), sendo avaliada negativamente pelos terapeutas, não concordando que a forma de falar do robô seja natural e fácil de compreender pelos idosos. Possivelmente, a pergunta não terá sido formulada da melhor forma, o que poderá ter polarizado negativamente as respostas dadas à pergunta. De qualquer forma, a avaliação negativa merece reflexão e poderá ser também explicada pelo facto de que a voz do robô não apresenta uma grande naturalidade ao contar as histórias a nível da entoação. Contudo, a fala implementada é clara e de boa qualidade, tanto que, quando questionados sobre a compreensão ao que o robô disse, os idosos avaliaram o sistema com uma pontuação média de 3,73 $\pm$ 1,33, ou seja, positiva. Assim, é possível concluir que, para este modo, pode haver a necessidade de utilizar outros serviços de STT que apresentem uma sensibilidade específica de entoação para contar histórias. No futuro, as vozes a serem usadas deverão procurar ser menos sintetizadas e mais naturais, através do uso de IA.

Em suma, os resultados obtidos a partir dos formulários SUS adaptados, tanto para os terapeutas ocupacionais quanto para os utilizadores finais idosos vão no sentido da aceitação da Hipótese 1 e Hipótese 2 apresentadas na secção 5.1. É importante destacar, no entanto, que há algumas características do sistema que necessitam de melhorias, para aprimorar ainda mais a usabilidade do sistema, como foi referido anteriormente.

5.2 Modo Livre

Para avaliar a funcionalidade do modo livre, foram conduzidas duas sessões em dias distintos com um único utilizador final.

Conforme já referido anteriormente, o desenvolvimento deste modo à data da escrita desta dissertação não se encontra finalizado. No entanto, para validar o estado atual do sistema, bem como o potencial que este apresenta para trabalhos futuros, foram realizadas duas sessões com um utilizador. Cada sessão teve uma duração aproximada de um minuto, durante a qual se realizou um breve diálogo entre o utilizador e o robô. Na primeira sessão, o utilizador era desconhecido pelo robô; na segunda sessão, o robô reconheceu a face da pessoa e desenvolveu um diálogo baseado no conhecimento prévio da pessoa em questão.

Uma vez que o robô não estava equipado com qualquer tipo de controlo de pose, ou seja, de manobrabilidade para permitir ficar de frente para o utilizador e numa pose adequada para o diálogo, o utilizador foi posicionado à frente do robô garantindo que o módulo de

reconhecimento facial pudesse funcionar adequadamente, bem como a comunicação entre as duas entidades.

5.2.1 Primeira Sessão

Na primeira sessão, o robô não conhecia ainda a pessoa, adotando um discurso de interação adequado a uma interação pela primeira vez.

O diálogo trocado entre o robô e o utilizador final foi o seguinte:

```
Utilizador: Olá!
Robô: Olá, tudo bem? Ainda não nos conhecemos! Quem és tu?
Utilizador: Eu sou a Ana.
Robô: Muito prazer em te conhecer! Eu sou a Matilde.
Utilizador: O que fazes por aqui?
Robô: Eu sou o robô social assistencial da Universidade de Coimbra. Ainda
estou em desenvolvimento mas já consigo falar e reconhecer pessoas.
Utilizador: Pronto Matilde! Foi um prazer conhecer-te.
Robô: Igualmente, tem um resto de um bom dia.
Utilizador: Até amanhã!
```

Figura 5.5: Primeira sessão.

Figura 5.6: Segunda sessão.

5.2.2 Segunda Sessão

Na segunda sessão, após adicionar a imagem da face do utilizador à base de dados do módulo de reconhecimento facial, o robô pôde reconhecer a pessoa e o diálogo entre as duas partes foi o seguinte:

```
Utilizador: Olá.
```

```
Robô: Olá, Ana! Como estás hoje?
Utilizador: Ainda te lembras de mim?
Robô: Sim, Ana! Eu tenho a capacidade de reconhecer pessoas.
Utilizador: Estás a fazer um bom trabalho.
Robô: Em que te posso ajudar?
Utilizador: Tenho de ir embora!
Robô: Ok! Tem um resto de um bom dia.
Utilizador: Adeus.
```

5.2.3 Resultados

É possível tirar uma conclusão positiva acerca da funcionalidade do sistema de gestão de diálogo para este modo de funcionamento. Através do reconhecimento facial, é obtida a identificação da pessoa e esse dado torna-se relevante para a gestão do diálogo baseado no contexto e nas informações já conhecidas pelo sistema acerca do interlocutor, influenciando as suas respostas. Pode-se afirmar com certeza que, com um aumento significativo de informações nos ficheiros AIML responsáveis por reger o guião deste *chatbot*, é possível dotar o robô de capacidades de interação através de diálogo para as mais diversas situações.

Ainda numa avaliação informal por parte do utilizador, foi dado um parecer positivo acerca do sistema, pelo facto de o mesmo oferecer uma grande naturalidade nos tempos de resposta, bem como na sua assertividade no processo de reconhecimento.

Em suma, o sistema de diálogo neste modo de funcionamento mostrou-se funcional e preciso, possibilitando que, num trabalho futuro, sirva como base para realizar melhorias nas suas interações bem como funcionalidade crítica de controlo de pose e aumento de armazenamento de dados importantes dos utilizadores com quem o robô mantenha interações através do diálogo.

5.3 Sumário

Neste capítulo, apresentou-se o processo de validação experimental, resultados obtidos e as conclusões acerca do desempenho do sistema de gestão de diálogo para os dois modos de funcionamento implementados, que foram descritos no capítulo 4 (página 33).

No modo narrativa, as sessões de teste foram conduzidas na Cáritas Diocesana de Coimbra, envolvendo a sua equipa de terapeutas ocupacionais e idosos do centro dia, tendo como

foco a avaliação da usabilidade do sistema. No modo livre, realizaram-se apenas duas sessões com um único utilizador, tendo como foco a funcionalidade do sistema implementado até então.

Ambas as experiências permitiram identificar vários aspetos que deverão beneficiar de melhorias do sistema em trabalho futuro.

6 Conclusão

A jornada que percorri neste trabalho de dissertação foi muito mais do que uma simples investigação académica; representou um caminho de autodescoberta e crescimento contínuo. Desde a minha infância, fui fascinado pela interação homem-máquina, um interesse alimentado por filmes e livros que retinham a minha imaginação e a mantinham presa num *loop* de "E se...", criando em mim um desejo ardente de poder criar algo semelhante.

Hoje, ao escrever este último capítulo da minha dissertação, posso afirmar com segurança que foi um privilégio investigar um tema tão significativo para mim, especialmente com o apoio financeiro do Programa Interreg Espanha-Portugal mediante uma bolsa de investigação do projeto EuroAGE+. Projeto esse que ganhou não apenas a minha admiração, como também o meu respeito pelas suas nobres metas e objetivos. Esses de encontrar soluções inovadoras para a promoção de um envelhecimento ativo e digno para os idosos, muitas vezes negligenciados e marginalizados pela sociedade pelas limitações que apresentam.

A conceção e implementação dos dois modos de funcionamento do sistema de gestão de diálogo do robô social assistencial que dominaram o meu trabalho de dissertação, foram desafios que me abriram as portas para um mundo novo e fascinante, desde a programação em AIML, até à utilização de modelos de linguagem pré-treinados da revolucionária arquitetura Transformers, trazendo consigo uma nova era de interesse na interação homem-máquina.

Cada passo desta jornada renovou o meu fascínio e alimentou o meu desejo de poder contribuir significativamente para o avanço deste campo de investigação com a esperança de poder um dia melhorar a qualidade de vida de alguém.

Em suma, este trabalho não enriqueceu apenas o meu conhecimento técnico, como também me proporcionou as mais variadas emoções, sendo a de realização a perseverante.

Como principal trabalho futuro desta dissertação, relativamente ao sistema de gestão de diálogo no modo narrativa, é crucial aprimorar os controlos de interação do terapeuta sobre a história. Até à versão atual desta dissertação, foram identificadas limitações, especialmente no que diz respeito à funcionalidade de pausa e de paragem de emergência. Atualmente,

estas funcionalidades são apenas executas aquando a conclusão da fala do robô. Referente às vozes a serem usadas pelo mesmo, é necessário, como trabalho futuro, procurar vozes menos sintetizadas e mais naturais que apresentem uma sensibilidade específica de entoação para contar histórias.

No modo livre, é imperativo melhorar a gestão do diálogo, considerando a implementação de estratégias de aprendizagem contínuo, onde o sistema possa aprender com interações para as quais ainda não está totalmente preparado, utilizando modelos de linguagem pré-treinados. Além disso, é essencial aprimorar a capacidade do sistema em reter informações e preferências dos utilizadores ao longo das conversas.

Bibliografia

[1] Os robôs sociais que já deslizam entre nós. https://www.publico.pt/2017/11/24/tecnologia/noticia/os-robos-que-ja-deslizam-entre-nos-1793831. Acedido em [24/01/2024].

[2] População portuguesa concentrada no litoral, sobretudo nas áreas metropolitanas, diz estudo. https://observador.pt/2023/03/15/populacao-portuguesa-concentrada-no-litoral-sobretudo-nas-areas-metropolitanas-diz-estudo/. Acedido em [24/01/2024].

[3] Portugal está ainda mais envelhecido: há 182 idosos por cada 100 jovens no país, dizem os censos. https://observador.pt/2021/12/16/portugal-esta-ainda-mais-envelhecido-ha-182-idosos-por-cada-100-jovens-no-pais-dizem-os-censos/. Acedido em [24/01/2024].

[4] S. A. and D. John. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7):72–80, 2015.

[5] International Phonetic Association. *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, Cambridge, 1999.

[6] Aza, Muha, Zura, and Nahdatul Akma Ahmad. Review of chatbots design techniques. *International Journal of Computer Applications*, 181:7–10, 08 2018.

[7] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] Cynthia Breazeal. Toward sociable robots. *Robotics and Autonomous Systems*, 42(3-4):167–175, 2003.

[10] Elizabeth Broadbent, Robyn Stafford, and Bruce MacDonald. Acceptance of healthcare robots for the older population: Review and future directions. *International Journal of Social Robotics*, 1(4):319–330, 2009.

[11] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

[12] Noam Chomsky. *Syntactic Structures*. Mouton, 1957.

[13] Felipe Cid, Ramon Cintas, Luis J. Manso, Luis V. Calderita, Agustin Sanchez, and Pedro Nunez. A real-time synchronization algorithm between text-to-speech (TTS) system and robot mouth for social robotic applications. In *Proc. of Workshop of Physical Agents*, pages 81–86, 2011.

[14] Michael Collins. Three generative, lexicalised models for statistical parsing. *ACL '97: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, 1997.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[16] David R Dowty. *Word Meaning and Montague Grammar*. Springer, 1979.

[17] Sefik Emre Eskimez, Kazuhito Koishida, and Hamid Krim. Generating talking face landmarks from speech. *IEEE Signal Processing Letters*, 25(9):1315–1319, 2018.

[18] Marshall F Folstein, Susan E Folstein, and Paul R McHugh. "mini-mental state". a practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research*, 12(3):189–198, 1975.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[20] Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[21] Myounghoon "Philart"Jeon. *Emotions and Affect in Human Factors and Human-Computer Interaction*. Academic Press, 2017.

[22] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, 3 edition, 2023.

[23] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

[24] Peter Ladefoged and Keith Johnson. *A Course in Phonetics*. Thomson Wadsworth, 2006.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[26] Jurgen Masche and Nam Le. A review of technologies for conversational systems. In *Advances in Intelligent Systems and Computing*, volume 629, pages 212–225. Springer, 2018.

[27] Dominic W. Massaro. *Perceiving Talking Faces: From Speech Perception to a Behavioral Principle*. MIT Press, 1993.

[28] Paulo Menezes and Rui P. Rocha. Promotion of active ageing through interactive artificial agents in a smart environment. *SN Applied Sciences*, 3:583, 2021.

[29] Dunja Mladenić and Luka Bradeško. A survey of chatbot systems through a loebner prize competition. In *Proceedings of the 2012 European Conference on Artificial Intelligence*, 2012.

[30] Dorothy Monekosso, Francisco Florez-Revuelta, and Paolo Remagnino. Special issue on ambient assisted living. *IEEE Intelligent Systems*, 30(4), 2015.

[31] United Nations. World population ageing 2019: Highlights, 2019. United Nations Department of Economic and Social Affairs, Population Division.

[32] Ani Nenkova and Kathleen McKeown. Automatic summarization. In *Foundations and Trends in Information Retrieval*, volume 5, pages 103–233. Now Publishers Inc., 2011.

[33] João Oliveira, Gabriel S. Martins, Alexandra Jegundo, Carlos Dantas, Claudio Wings, Luis Santos, João Dias, and Fernando Perdigão. Speaking robots: the challenges of acceptance by the ageing society. In *26th IEEE Int. Symp. on Robot and Human Interactive Communication*, pages 1285–1290, Lisbon, Portugal, 2017.

[34] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W. Senior. Audio-visual speech recognition: An overview. *Issues in visual and audio-visual speech processing*, 22(1):23–37, 2004.

[35] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. 3(3.2):5, 2009.

[36] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[37] Nicole M. Radziwill and Morgan C. Benton. Evaluating quality of chatbots and intelligent conversational agents. *Software Quality Professional*, 19(3):25–36, 2017.

[38] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.

[39] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):3772–3784, 2018.

[40] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[41] Tanya Stivers, N. J. Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan-Peter de Ruiter, Kyung-Eun Yoon, and Stephen C. Levinson. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26):10587–10592, 2009.

[42] A.Q. Summerfield. Lipreading and audio-visual speech perception. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 335(1273):71–78, 1992.

[43] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Generating text with recurrent neural networks. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[45] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[46] Richard S Wallace. The anatomy of a.l.i.c.e. *Pandorabots*, 2003.

[47] Richard S Wallace. *The Elements of AIML Style*. ALICE AI Foundation, 2009.

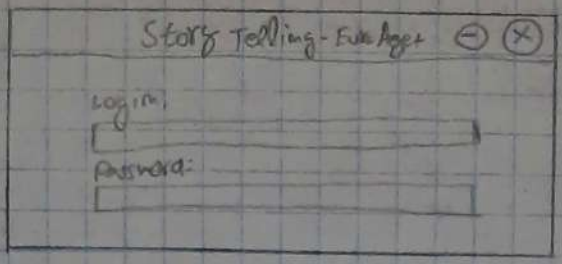
[48] Joseph Weizenbaum. Eliza — a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 1966.

Anexo A

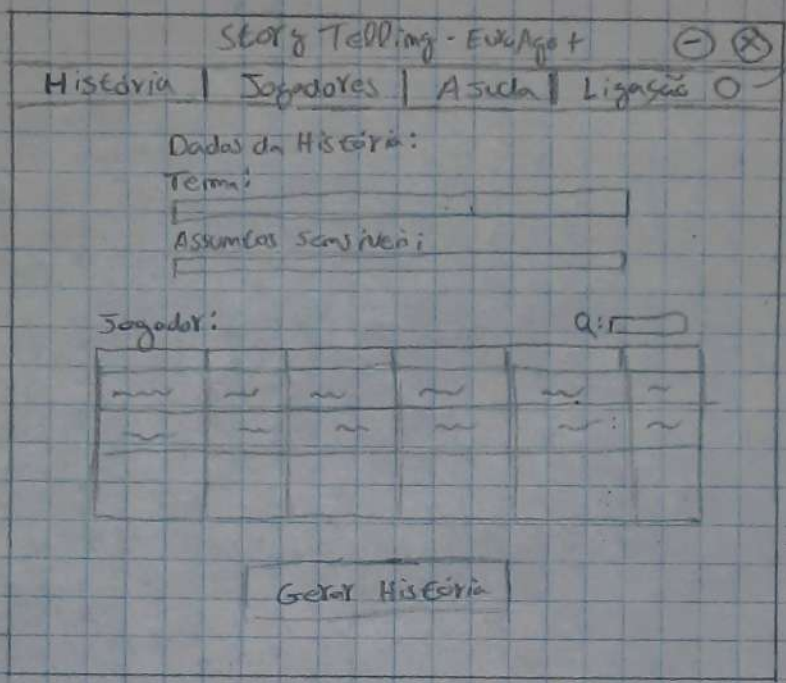
Mockups da Interface

MOCKUPS da Interface

Login (1º)



Credenciais de acesso para cada instituição



Verde ou vermelho segundo o estado de ligação

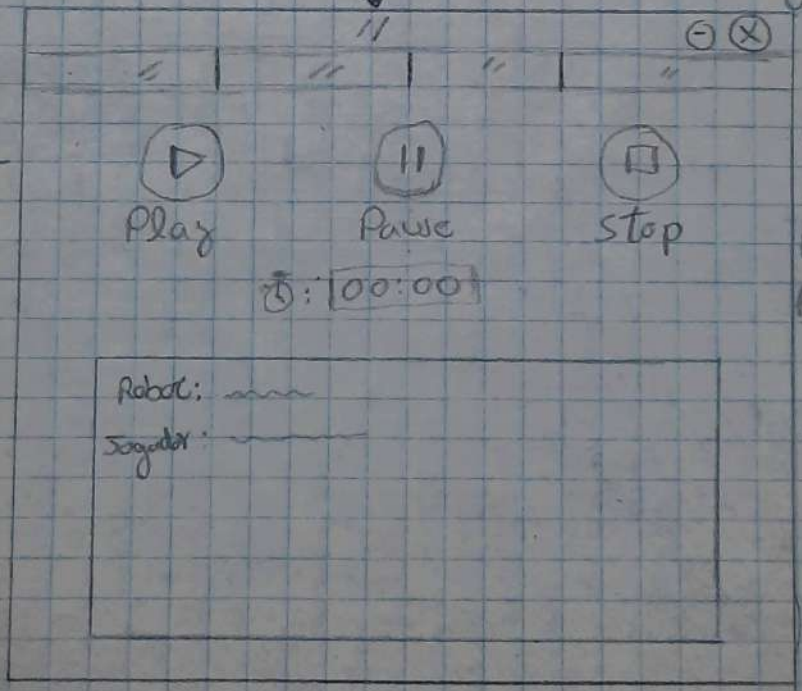
Terapeuta consegue criar história preenchendo os campos.

Na toolbar, o terapeuta consegue mudar a janela consoante a finalidade que cada representa

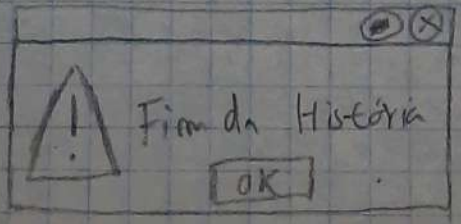
História (Default após o login) → caso haja jogadores na base de dados

Quando clicado em gerar história

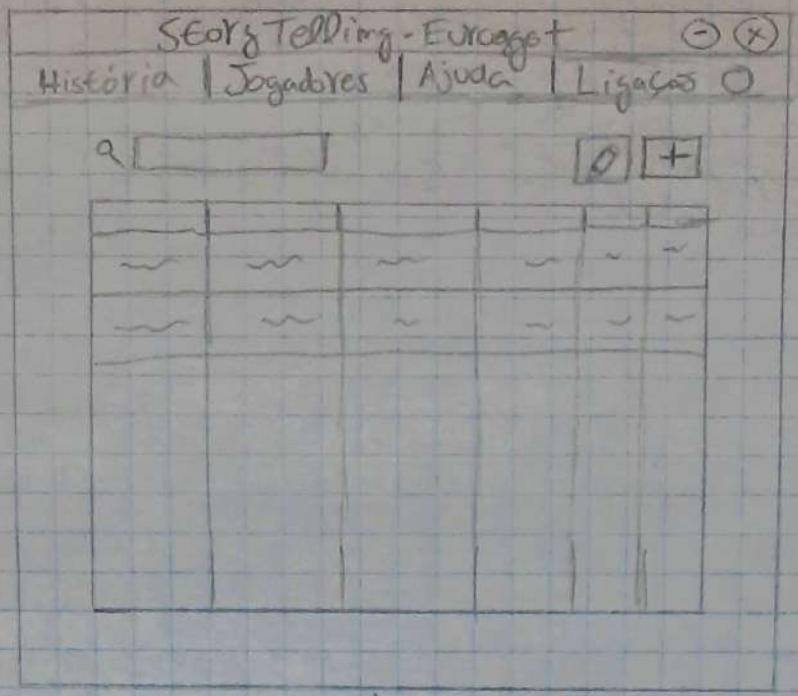
História a decorrer



A toolbar será bloqueada, e o jogo começará logo em pausa, esperando o play do terapeuta. Stop terá dupla confirmação. Timer para contabilizar o tempo da sessão. Quando o jogo acabar aparece message box a avisar e toolbar é desbloqueada.



Jogadores

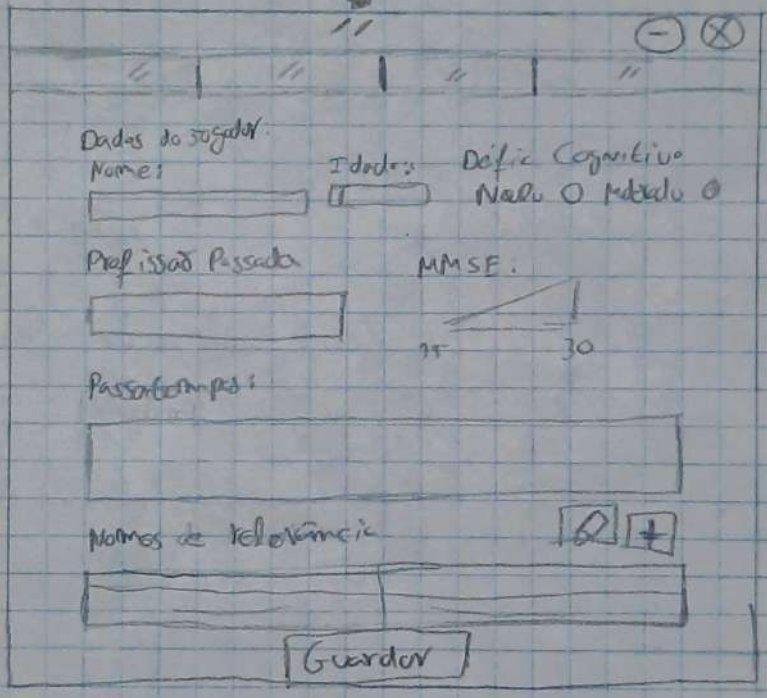


Possibilidade na procura do jogador

Capacidade em adicionar ou editar jogador

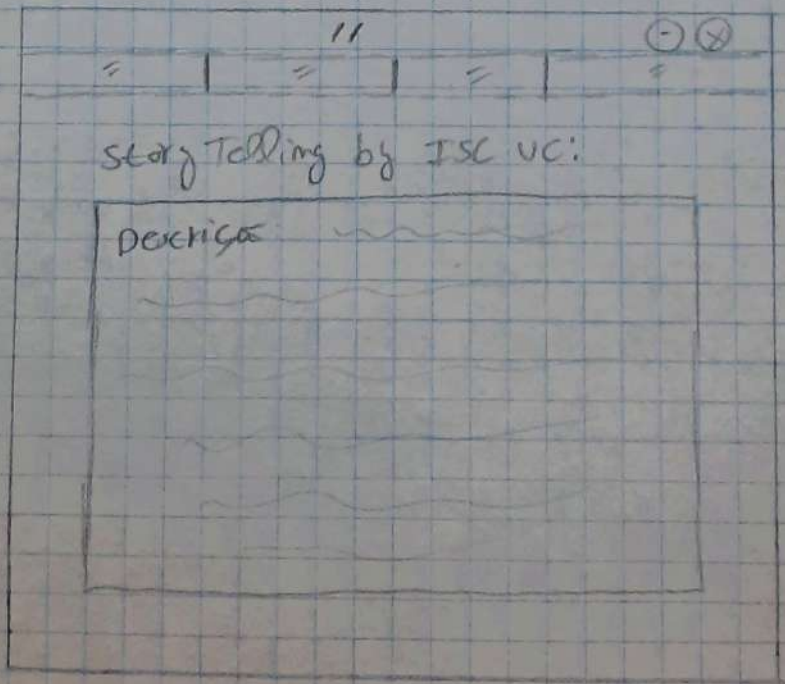
Quando se quer adicionar

Novo Jogador ou Editar Jogador



As informações dos jogadores serão guardadas numa base de dados que estará num servidor. Posto isto, a comunicação com o robô será feita sempre pelo server.

Ajuda



Anexo B

Correspondência dos fonemas do IPA com visemas da Microsoft e do robô GrowMu

Fonema segundo o Alfabeto Fonético Internacional (IPA) [42]	Exemplo numa palavra	Visema ID - Microsoft	Visema ID Driver - Robô	Imagem do Visema
Silêncio	-	0	0	
æ, ə, ʌ	**e**xtremo	1	20 (Novo)	
ɑ	medir**á**	2	17	
ɔ	l**ó**	3	11	
ɛ, ʊ	ám**en**	4	18	
ɝ	b**ir**d	5	18	
j, i, ɪ	escrev**i**	6	21 (Novo)	
w, u	fad**o**	7	16	
o	d**om**	8	22 (Novo)	
aʊ	p**au**	9	17	
ɔɪ	t**oy**	10	11	

aɪ	**i**ce	11	17		
h	a**h**ead	12	20 (Novo)		
ɹ	**re**cordar	13	20 (Novo)		
l	po**l**vo	14	14		
s, z	**s**eco	15	13		
ʃ, tʃ, dʒ, ʒ	**ch**uva	16	20 (Novo)		
ð	mo**th**er	17	21 (Novo)		
f, v	**v**aca	18	15		
d, t, n, θ	input	19	14		
k, g, ŋ	sna**ck**	20	21 (Novo)		
p, b, m	**m**aça	21	12		

Anexo C

Listagens de código fonte

C.1 Interface de diálogo

C.1.1 STT

```
try:
    speech_recognition_result = speech_recognizer.recognize_once_async().get()
except Exception as e:
    rospy.logerr(f"Error recognizing speech with Microsoft Azure Speech-to-Text API: {e}")
    return None

if speech_recognition_result.reason == speechsdk.ResultReason.RecognizedSpeech:
    return speech_recognition_result.text
elif speech_recognition_result.reason == speechsdk.ResultReason.NoMatch:
    rospy.logwarn("No speech recognized")
    return None
elif speech_recognition_result.reason == speechsdk.ResultReason.Canceled:
    cancellation_details = speech_recognition_result.cancellation_details
    rospy.logerr(f"Speech recognition canceled: {cancellation_details.reason}")
    if cancellation_details.reason == speechsdk.CancellationReason.Error:
        rospy.logerr(f"Error details: {cancellation_details.error_details}")
        rospy.logerr("Have you correctly configured the key and region for the speech resource?")
        return None
```

Listing 1: Transcrição de fala usando a API de Speech-to-Text da Microsoft Azure

C.1.2 TTS

```
try:
    result = speech_synthesizer.speak_ssml_async(ssml_doc).get()
    if result.reason != speechsdk.ResultReason.SynthesizingAudioCompleted:
        rospy.logerr(f"Error synthesizing speech with Microsoft Azure Text-to-Speech API: {result.reason}")
except Exception as e:
    rospy.logerr(f"Error synthesizing speech with Microsoft Azure Text-to-Speech API: {e}")
    return
```

Listing 2: Síntese e reprodução da fala usando a API Text-To-Speech da Microsoft Azure

```
synthesis_input = texttospeech.SynthesisInput(text=text)
voice = texttospeech.VoiceSelectionParams(
    language_code=language,
    name='en-US-Neural2-G' if language == 'en-US' else 'pt-PT-Wavenet-D',
    ssml_gender=texttospeech.SsmlVoiceGender.FEMALE
)
audio_config = texttospeech.AudioConfig(
    audio_encoding=texttospeech.AudioEncoding.LINEAR16,
    sample_rate_hertz=16000
)

response = client.synthesize_speech(
    input=synthesis_input,
    voice=voice,
    audio_config=audio_config)
```

Listing 3: Síntese da fala usando a API Text-To-Speech da Google

```
data = {
    "text": text,
    "model_id": "eleven_monolingual_v1",
    "voice_settings": {
        "stability": 0.5,
        "similarity_boost": 0.5
    }
}

try:
```

```
    response = requests.post(url, json=data, headers=headers)
    response.raise_for_status()
except requests.exceptions.RequestException as e:
    rospy.logerr(f"Error sending POST request to Eleven API: {e}")
    return
```

Listing 4: Síntese da fala usando a API Text-To-Speech da ElevenLabs

C.1.3 Sincronização Labial

```
frame_length = 2048 # N (4.6 ms @44100 Hz) (12.8 ms @16000 Hz)
energy_limits = [1, 30, 100] if sample_rate == 44100 else [10, 25, 50] if language == "
    en-US" else [15, 40, 100]
hop_length = 1024 if sample_rate == 44100 else 512

if sample_rate == 44100:
    audio_array_normalized = audio_data.astype(np.float32) / np.max(np.abs(audio_data))
else:
    audio_array_decoded = np.frombuffer(audio_data, dtype=np.int16)
    audio_array_normalized = audio_array_decoded / np.max(np.abs(audio_array_decoded))

audio_thread = threading.Thread(target=self.__play_audio, args=(audio_data, sample_rate)
    )
audio_thread.start()
```

Listing 5: Inicialização dos parâmetros e início da reprodução do áudio

C.2 Sistema de Gestão de Diálogo

C.2.1 Modo Narrativa

```
def define_parameters(self, name, age, brain, hobbies, profession, family, theme,
    forbidden_topics) -> None:
    """Define story parameters based on user input."""
    info = (f"Jogador:\n {name} de {age} anos\n Gosta de {hobbies}\n Nivel de "
            f"deficit cognitivo {brain}\n Profissao passada: {profession}\n "
            f"Familia/Amigos: {family}\n Historia:\n Tema da historia: {theme}\n "
            f"Nao fale sobre {forbidden_topics}")

    self.new_message(info)

def new_message(self, text: str) -> None:
    """Send a new message to the OpenAI assistant."""
    message = self.client.beta.threads.messages.create(
        thread_id=self.thread.id,
        role="user",
        content=text
    )
```

Listing 6: Funções define_parameters e new_message

```
class EventHandler(AssistantEventHandler):
    """Custom event handler for OpenAI assistant events."""

    def __init__(self):
        """Initialize the event handler."""
        super().__init__()
        self.text_final = ""
        self.all_text = ""
        self.control = False
        self.rate = rate

    @override
    def on_text_delta(self, delta, snapshot):
        """Callback function triggered on receiving text delta."""
        self.text_final += delta.value
        print(delta.value, end="", flush=True)
```

```
17
18          if (delta.value in (".", "?", "!")) and not self.control:
19              self.control = True
20              tts_thread = threading.Thread(target=self.call_tts_microsoft, args=(self.
                    text_final,))
21              tts_thread.start()
22              self.all_text += self.text_final
23              self.text_final = ""
24
25      def call_tts_microsoft(self, text):
26          """Invoke Microsoft TTS service."""
27          tts.microsoft(text, "pt-PT", self.rate)
28          self.control = False
```

Listing 7: Classe EventHandler

C.2.2 Modo Livre

C.2.2.1 Reconhecimento Facial

```
1  import face_recognition
2  import cv2
3  import os
4  import glob
5  import numpy as np
6
7  class SimpleFacerec:
8      def __init__(self):
9          self.known_face_encodings = []
10         self.known_face_names = []
11
12         # Resize frame for faster speed
13         self.frame_resizing = 0.25
14
15     def load_encoding_images(self, images_path):
16         """Load encoding images from path."""
17         images_path = glob.glob(os.path.join(images_path, "*.*"))
18
19         print("{} encoding images found.".format(len(images_path)))
20
21         # Store image encoding and names
22         for img_path in images_path:
23             img = cv2.imread(img_path)
24             rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
25
26             # Get the filename only from the initial file path.
27             basename = os.path.basename(img_path)
28             (filename, ext) = os.path.splitext(basename)
29             # Get encoding
30             img_encoding = face_recognition.face_encodings(rgb_img)[0]
31
32             # Store file name and file encoding
33             self.known_face_encodings.append(img_encoding)
34             self.known_face_names.append(filename)
35         print("Encoding images loaded")
36
37     def detect_known_faces(self, frame):
38         small_frame = cv2.resize(frame, (0, 0), fx=self.frame_resizing, fy=self.
                frame_resizing)
39         # Find all the faces and face encodings in the current frame of video
40         rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
41         face_locations = face_recognition.face_locations(rgb_small_frame)
42         face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations
                )
43
44         face_names = []
45         for face_encoding in face_encodings:
46             matches = face_recognition.compare_faces(self.known_face_encodings,
                    face_encoding)
47             name = "Unknown"
48
```

```
            face_distances = face_recognition.face_distance(self.known_face_encodings,
                face_encoding)
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = self.known_face_names[best_match_index]
            face_names.append(name)

        face_locations = np.array(face_locations) / self.frame_resizing
        return face_locations.astype(int), face_names
```

Listing 8: Biblioteca simple_facerec

```
while not rospy.is_shutdown():
    if flag:
        ret, frame = cap.read()

        # Detect Faces
        face_locations, face_names = sfr.detect_known_faces(frame)
        for face_loc, name in zip(face_locations, face_names):

            y1, x2, y2, x1 = face_loc[0], face_loc[1], face_loc[2], face_loc[3]

            cv2.putText(frame, name, (x1, y1 - 10), cv2.FONT_HERSHEY_DUPLEX, 1, (0, 0,
                200), 2)
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 200), 4)
            # Publica os nomes das faces reconhecidas no topico
            face_names_pub.publish(name)

        cv2.imshow("Frame", frame)

        key = cv2.waitKey(1)

        if key == 27:  # Tecla ESC para sair
            break

        rospy.sleep(0.1)

    rospy.sleep(0.1)

cap.release()
cv2.destroyAllWindows()
```

Listing 9: Reconhecimento facial

C.2.2.2 Programação em AIML

```
while True:
    rospy.Subscriber("face_names", String, callback)
    rospy.sleep(0.1)
    get_recognition.publish("Inicia")

    print("Fala\n")
    input = STT.microsoft("pt-PT")

    get_recognition.publish("_")

    if input.upper() == "ADEUS.":
        break

    rospy.sleep(0.1)
    k.setPredicate("nome", nome, "1")

    resposta = k.respond(input, "1")
    print("Bot: ", resposta)
    TTS.microsoft(resposta, "pt-PT", "0")
```

Listing 10: Ciclo infinito do sistema de diálogo em modo livre

Anexo D

Resposta aos formulários dos utilizadores idosos e dos terapeutas ocupacionais

ADAPTED SYSTEM USABILITY SCALE TAILORED TO END USERS

INSTITUTO DE SISTEMAS E ROBÓTICA — UNIV. COIMBRA

NDC: M ☐ L ☐ N ☐

Instruções: Responda às 10 questões seguintes sobre o sistema que acabou de utilizar. Em cada questão, assinale com um X uma única opção.

1. **Gostaria de jogar este jogo com frequência.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

2. **Acho o jogo interessante e fácil de compreender.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

3. **Compreendi com facilidade o que o robô dizia.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

4. **Acho que precisaria de ajuda para jogar este jogo.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

5. **A história contada manteve-me interessado e atento o tempo todo.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

6. **Achei que o jogo tinha muitas limitações ou problemas.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

7. **Acredito que a maioria das pessoas conseguiria jogar este jogo com facilidade.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

8. **Senti bem-estar ao ouvir a história contada pelo robô.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

9. **Senti-me confortável enquanto interagia com o robô.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

10. **Consegui seguir facilmente a história que o robô contou.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

Avaliação dos utilizadores finais idosos — SUS Adaptado

Idosos	Pergunta 1	Pergunta 2	Pergunta 3	Pergunta 4	Pergunta 5	Pergunta 6	Pergunta 7	Pergunta 8	Pergunta 9	Pergunta 10
1	3	2	2	5	4	4	2	4	4	2
2	3	4	4	4	5	2	2	4	4	3
3	4	4	2	4	4	2	3	4	4	2
4	4	4	4	4	5	1	4	5	4	5
5	5	5	5	4	5	4	4	5	5	5
6	5	5	5	2	5	1	1	5	5	5
7	3	4	2	4	4	3	3	4	4	4
8	4	4	5	4	5	2	2	5	5	5
9	5	5	4	4	5	4	3	5	4	5
10	3	4	2	4	3	2	2	4	4	3
11	5	5	5	4	4	2	3	4	4	4
12	4	4	4	4	4	2	2	4	4	3
13	3	4	2	3	3	2	4	4	4	2
14	5	5	5	3	5	1	4	5	5	5
15	5	5	5	2	5	1	2	5	5	5
Média	**4,067**	**4,267**	**3,733**	**3,667**	**4,4**	**2,2**	**2,733**	**4,467**	**4,333**	**3,867**
Desvio padrão	**0,884**	**0,799**	**1,335**	**0,816**	**0,737**	**1,082**	**0,961**	**0,516**	**0,488**	**1,246**

Estatística	Valor
Média de todas as respostas	3,773
Desvio padrão	1,148

Avaliação dos utilizadores finais idosos com Nível de Défice cognitivo Moderado — SUS Adaptado

Idosos	Pergunta 1	Pergunta 2	Pergunta 3	Pergunta 4	Pergunta 5	Pergunta 6	Pergunta 7	Pergunta 8	Pergunta 9	Pergunta 10
1	3	2	2	5	4	4	2	4	4	2
2	3	4	4	4	5	2	2	4	4	3

Avaliação dos utilizadores finais idosos com Nível de Défice cognitivo Leve — SUS Adaptado

Idosos	Pergunta 1	Pergunta 2	Pergunta 3	Pergunta 4	Pergunta 5	Pergunta 6	Pergunta 7	Pergunta 8	Pergunta 9	Pergunta 10
3	4	4	2	4	4	2	3	4	4	2
4	4	4	4	4	5	1	4	5	4	5
5	5	5	5	4	5	4	4	5	5	5
6	5	5	5	2	5	1	1	5	5	5
7	3	4	2	4	4	3	3	4	4	4
8	4	4	5	4	5	2	2	5	5	5
9	5	5	4	4	5	4	3	5	4	5
10	3	4	2	4	3	2	2	4	4	3
11	5	5	5	4	4	2	3	4	4	4
12	4	4	4	4	4	2	2	4	4	3
Média	**4,2**	**4,4**	**3.8**	**3,8**	**4,4**	**2,3**	**2,7**	**4,5**	**4,3**	**4,1**
Desvio padrão	**0,789**	**0,516**	**1,317**	**0,632**	**0,699**	**1,059**	**0,949**	**0,527**	**0,483**	**1,101**

Estatística	Valor
Média de todas as respostas	3,85
Desvio padrão	1,086

Avaliação dos utilizadores finais idosos com Nível de Défice cognitivo Nulo — SUS Adaptado

Idosos	Pergunta 1	Pergunta 2	Pergunta 3	Pergunta 4	Pergunta 5	Pergunta 6	Pergunta 7	Pergunta 8	Pergunta 9	Pergunta 10
13	3	4	2	3	3	2	4	4	4	2
14	5	5	5	3	5	1	4	5	5	5
15	5	5	5	2	5	1	2	5	5	5

ADAPTED SYSTEM USABILITY SCALE TAILORED TO THERAPISTS AND CARERS

INSTITUTO DE SISTEMAS E ROBÓTICA — UNIV. COIMBRA

Instruções: Responda às 10 questões seguintes sobre o sistema que acabou de utilizar. Em cada questão, assinale com um X uma única opção.

1. **Gostaria de utilizar este jogo com frequência com os idosos à minha responsabilidade.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

2. **Considero o jogo mais complexo do que necessário para o fim a que se destina.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

3. **Acho que é adequada e fácil de usar a interface através da qual se faz a configuração da história e a gestão da sessão de "story telling" com o idoso.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

4. **Acho que necessitaria de ajuda de um técnico de engenharia para conseguir utilizar este jogo com os idosos.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

5. **Acho que a forma de falar do robô é natural e fácil de compreender pelos idosos e que a interação proporcionada pelo robô é satisfatória.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

6. **Acho que o jogo apresenta muitas inconsistências e limitações durante o seu uso com os idosos.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

7. **Acredito que a maioria dos terapeutas e cuidadores em geral, conseguiriam usar facilmente o jogo com os seus idosos.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

8. **Acho que o jogo é adequado e se adapta ao nível cognitivo individual de cada idoso.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

9. **Acho que o jogo pode ser usado na implementação de estratégias terapêuticas para ajudar os idosos à minha responsabilidade.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

10. **Acho que o nível de interesse dos idosos pelas histórias contadas pelo robô em geral é alto.**

Discordo completamente	Discordo	Neutro	Concordo	Concordo fortemente

Avaliação dos terapeutas — SUS Adaptado

Terapeuta	Pergunta 1	Pergunta 2	Pergunta 3	Pergunta 4	Pergunta 5	Pergunta 6	Pergunta 7	Pergunta 8	Pergunta 9	Pergunta 10
1	4	3	4	3	2	2	5	4	4	4
2	5	2	5	2	2	2	4	4	4	4

Anexo E

Exemplo de uma sessão narrativa com um idoso de nível de défice cognitivo leve

Tema da história: Comer bolo do caco e beber poncha
R: Robô **I:** Idoso
Nível de Défice cognitivo: Leve
Passatempos: Passear pela praia
Relações: Sara — Neta e Rita — Bisneta

R: Clara recebeu um convite da sua neta, Sara, para irem comer bolo do caco e beber poncha. O que achas que a Clara deve fazer? Ir com a neta ou preferes que ela convite a bisneta Rita?
I: Gostava muito que a Rita fosse.
R: Clara decidiu convidar a bisneta Rita para irem comer bolo do caco e beber poncha. O que achas que elas devem fazer a seguir? Ir ao café local ou preferes que voltem à praia desfrutar da paisagem?
I: Ir à praia.
R: Clara e Rita foram até à praia para desfrutar da paisagem. Enquanto caminham pela areia avistam um vendedor de gelados ao longe. O que achas que elas devem fazer? Comprar um gelado ou continuar a caminhar pela praia?
I: Comer um gelado.
R: Clara e Rita decidiram comprar um gelado de chocolate para saborearem enquanto caminham pela praia. Enquanto apreciam o gelado, avistam um grupo de golfinhos a saltar nas ondas. O que achas que elas devem fazer? Continuar a observar os golfinhos ou preferes que vão dar um mergulho no mar?
I: Não sei nadar, uma tristeza.
R: Clara e Rita continuam a observar os golfinhos a saltarem das ondas, maravilhadas com a beleza dos animais marinhos. O sol começa a pôr-se no horizonte pintando céu de tons alaranjados. O que achas que elas devem fazer agora? Regressar a casa ou preferes que fiquem mais um pouco na praia a desfrutar do final de tarde.
I: Apreciar o pôr do sol e estar mais bocadinho ao fresquinho.
R: Clara e Rita ficaram mais um pouco na praia a desfrutar do final de tarde contemplando o por do sol, sentadas na areia partilham histórias e risos aproveitando o momento de cumplicidade. Quando o sol desaparece do horizonte, Clara olha para Rita e sorri. É altura de regressar a casa com o coração cheio de felicidade. Fim.