

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



# **Implementação de um assistente virtual para rede doméstica controlado por voz**

**Diogo Leite Martins**

DISSERTAÇÃO

MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Orientador: Pedro Santos

Co-orientador: Bruno Parreira

31 de Julho de 2020

# Resumo

O presente documento descreve o projeto que se realizou no âmbito da unidade curricular da dissertação bem como a revisão literária e o estudo do estado-da-arte acerca dos temas envolventes. O projeto consiste no desenvolvimento e implementação de um sistema que informa o utilizador acerca do estado da sua rede doméstica e procura soluções quando ocorrem problemas na mesma. O sistema é constituído por um assistente virtual que recebe os pedidos do utilizador através de comandos de voz e um router capaz de gerir e diagnosticar a rede em causa.

As tecnologias de assistentes virtuais e redes domésticas, que são no fundo o principal foco do projeto, estão inseridas na categoria de tecnologias IoT. Esta vertente tecnológica tem sido bastante desenvolvida nos últimos anos e promete ter um envolvimento muito forte na vida pessoal de uma grande parte da população num futuro próximo.

Um dos objetivos deste trabalho é demonstrar que é possível integrar assistentes virtuais com a gestão de redes domésticas e facilitar a interação do utilizador com o ambiente tecnológico presente no seu lar. A presença dos assistentes virtuais não só na gestão de redes domésticas, mas também na gestão da vida pessoal do utilizador vem trazer algum conforto e facilitismo ao próprio utilizador. Naturalmente, nem todos os utilizadores de redes domésticas têm conhecimento suficiente na área para saber lidar com todo o tipo de problemas que podem, potencialmente, surgir. É nestes casos que o sistema traz mais vantagens e vem substituir o suporte atual para resolução de problemas associados a redes domésticas.

O sistema foi avaliado na perspetiva da experiência do utilizador e verificou-se que existe uma melhoria de performance em relação ao sistema de apoio ao cliente atual. Na área linguística o sistema apresentou algumas fragilidades, nomeadamente a falta de expressões na estrutura de diálogos com o utilizador.

# Abstract

The virtual assistants and the domestic networks are the main focus of this project. This technological area has been developing in the past years and it can have a strong presence in the personal lives of the majority of the population in the near future.

One of the objectives of this project is to demonstrate that it is possible to integrate virtual assistants in the management of domestic networks and to facilitate the user interaction with his technological environment. The presence of virtual assistants not only on network management, but also on the user's personal life management brings some confort and easiness to the user himself. Naturally, not all users have enough knowledge on this technological area to solve any network problem that may occur. On these cases the implemented system brings some advantages and comes as a substitute to the current customer support.

The project consists on the development and implementation of a system that informs the user about the state of his domestic network and tries to solve any occurring problems. The system is composed by a virtual assistant that captures the user requests by listening to voice commands and a router capable of network managment and diagnosis.

The system was evaluated on a user experience perspective. On the user interaction area, the system showed some weaknessses, mainly the lack of user inputs stated on the dialogue structure.

**Palavras-chave:** Assistentes Virtual, router, redes domésticas, IoT, OpenWRT, Alexa.

# Conteúdo

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>  | <b>2</b>  |
| 1.1      | Motivação e contexto . . . . .                                 | 2         |
| 1.2      | Desafio e Solução . . . . .                                    | 3         |
| 1.3      | Estrutura do documento . . . . .                               | 4         |
| <b>2</b> | <b>Tecnologias, Literatura e Trabalhos Relacionados</b>        | <b>5</b>  |
| 2.1      | Tecnologias relevantes . . . . .                               | 5         |
| 2.1.1    | Assistentes Virtuais de referência . . . . .                   | 5         |
| 2.1.2    | Assistentes virtuais <i>open source</i> . . . . .              | 6         |
| 2.1.3    | Tecnologias associadas a assistentes virtuais . . . . .        | 7         |
| 2.1.4    | Sistemas operativos para ambientes de router . . . . .         | 9         |
| 2.1.5    | Protocolos de comunicação . . . . .                            | 10        |
| 2.2      | Análise de literatura e trabalhos relacionados . . . . .       | 11        |
| 2.2.1    | Home Automation Systems (HAS) . . . . .                        | 11        |
| 2.2.2    | Sistemas que usam/implementam Voice Assistants . . . . .       | 12        |
| 2.3      | Considerações finais . . . . .                                 | 14        |
| <b>3</b> | <b>Assistente virtual para redes domésticas</b>                | <b>16</b> |
| 3.1      | Arquitetura e funcionamento do sistema . . . . .               | 16        |
| 3.1.1    | Interação com o utilizador . . . . .                           | 16        |
| 3.1.2    | Operação interna do sistema . . . . .                          | 18        |
| 3.2      | Casos de uso . . . . .   | 19        |
| 3.2.1    | Criação de rede Guest . . . . .                                | 20        |
| 3.2.2    | Identificação da razão de problemas de conectividade . . . . . | 20        |
| 3.3      | Implementação . . . . .  | 22        |
| 3.3.1    | Seleção das tecnologias . . . . .                              | 22        |
| 3.3.2    | Implementação do gestor do assistente . . . . .                | 23        |
| 3.3.3    | Mecanismos implementados no OpenWRT . . . . .                  | 26        |
| 3.3.4    | Implementação dos casos de uso . . . . .                       | 29        |
| 3.4      | Considerações finais . . . . .                                 | 36        |
| <b>4</b> | <b>Avaliação da experiência do utilizador</b>                  | <b>37</b> |
| 4.1      | Métricas . . . . .   | 37        |
| 4.2      | Metodologia/Protocolo dos testes . . . . .                     | 39        |
| 4.2.1    | Teste T1: Criação da rede Guest . . . . .                      | 39        |
| 4.2.2    | Teste T2: Sinal do dispositivo fraco . . . . .                 | 39        |
| 4.2.3    | Teste T3: Limitação de largura de banda . . . . .              | 40        |
| 4.3      | Análise de resultados . . . . .                                | 41        |

|          |                                       |           |
|----------|---------------------------------------|-----------|
| 4.3.1    | Métricas objetivas . . . . .          | 41        |
| 4.3.2    | Resultados do questionário . . . . .  | 42        |
| 4.3.3    | Resultados não considerados . . . . . | 46        |
| 4.4      | Considerações finais . . . . .        | 46        |
| <b>5</b> | <b>Conclusão</b>                      | <b>47</b> |
| 5.1      | Trabalho futuro . . . . .             | 48        |
|          | <b>Referências</b>                    | <b>49</b> |

# Lista de Figuras

|      |   |    |
|------|---|----|
| 2.1  | Distribuição dos assistentes virtuais nos EUA (2017) [6] . . . . .          | 6  |
| 2.2  | Arquitetura do modelo STT para reconhecimento de fala . . . . .             | 7  |
| 2.3  | Arquitetura do modelo TTS para síntese de fala . . . . .                    | 9  |
| 2.4  | Estrutura do modelo multi-agent [27] . . . . .                              | 14 |
| 3.1  | Arquitetura do sistema . . . . .  | 17 |
| 3.2  | Fluxograma de operação interna do sistema numa interação genérica . . . . . | 19 |
| 3.3  | Arquitetura do gestor do assistente virtual . . . . .                       | 24 |
| 3.4  | Componentes presentes no OpenWRT . . . . .                                  | 26 |
| 3.5  | Arquitetura do agente de configuração de rede . . . . .                     | 28 |
| 3.6  | Processo de ativação de rede Guest . . . . .                                | 30 |
| 3.7  | Problemas de utilização da Internet . . . . .                               | 31 |
| 3.8  | Rotina de diagnóstico de rede . . . . .                                     | 32 |
| 3.9  | Rotina de teste de velocidade e latência . . . . .                          | 33 |
| 3.10 | Rotina para reiniciar o <i>router</i> . . . . .                             | 34 |
| 3.11 | Rotina para contar o número de dispositivos na rede . . . . .               | 35 |
| 4.1  | Resultados da primeira pergunta do questionário . . . . .                   | 43 |
| 4.2  | Resultados da segunda pergunta do questionário . . . . .                    | 43 |
| 4.3  | Resultados da terceira pergunta do questionário . . . . .                   | 44 |
| 4.4  | Resultados da quarta pergunta do questionário . . . . .                     | 44 |
| 4.5  | Resultados da quinta pergunta do questionário . . . . .                     | 45 |
| 4.6  | Resultados da sexta pergunta do questionário . . . . .                      | 45 |

# Lista de Tabelas

|     |  |    |
|-----|--|----|
| 2.1 | Classificação das emoções e a sua intensidade segundo <a href="#">[25]</a> . . . . . | 13 |
| 4.1 | Medição em segundos do tempo de execução dos testes por cada utilizador . . . .      | 42 |

# Abreviaturas e Símbolos

|      |                                     |
|------|-------------------------------------|
| TTS  | Text-To-Speech                      |
| STT  | Speech-To-Text                      |
| NLP  | Natural language processing         |
| IoT  | Internet of things                  |
| HMM  | Hidden Markov Model                 |
| SSH  | Secure Socket Shell                 |
| WLAN | Wireless Local Area Network         |
| WAN  | Wide Area Network                   |
| MQTT | Message Queuing Telemetry Transport |
| QoS  | Quality of Service                  |
| IP   | Internet Protocol                   |
| MAC  | Media Access Control                |
| UCI  | Unified Configuration Interface     |



# Agradecimentos

À minha família, principalmente os meus pais e irmãos que acompanham o meu trabalho todos os dias 24 horas e apoiam-me incondicionalmente, não só agora, mas ao longo de toda a minha vida.

Ao meu orientador e co-orientador, Engenheiro Pedro Miguel Salgueiro dos Santos e Engenheiro Bruno Parreira, respetivamente, pelo acompanhamento incansável durante o período de trabalho e pelos debates de ideias para o desenvolvimento do projeto.

À comunidade FEUP, onde tive oportunidade de ser aluno durante os últimos 5 anos e da qual me orgulho bastante.

À Altran, que me ofereceu a oportunidade de trabalhar nesta tese.

À equipa de R&D Telecom da Altran, que me acompanhou também todos os dias desde o início do projeto e que me recebeu de braços abertos.

À minha namorada, que sempre me apoiou quando eu mais precisava e também por ser compreensiva quando a agenda não era favorável.

A todos os meus amigos que, provavelmente sem saber, contribuíram da melhor forma para a minha boa disposição e motivação para trabalhar.

Diogo Leite Martins

# Capítulo 1

## Introdução

Neste capítulo será, inicialmente, apresentada a motivação e o contexto associados ao desenvolvimento deste projeto, seguido de uma secção que apresenta o desafio e a solução proposta, e finalizando com uma secção que descreve a estrutura do documento.

### 1.1 Motivação e contexto

Os assistentes virtuais têm vindo a mostrar-se cada vez mais relevantes por oferecerem uma forma muito intuitiva de interagir com o ser humano. Um assistente virtual é um software que procura automatizar/simplificar determinadas ações do ser humano bem como auxiliar o seu utilizador através de um vasto leque de funcionalidades. É uma tecnologia inovadora que permite assistir o ser humano em diversos tipos de tarefas. É cada vez mais comum encontrarmos em residências, dispositivos conectados entre si ou com sistemas externos. O próximo passo dessa evolução é a integração destes dispositivos com uma interface que permite interação por comandos de voz. Os assistentes virtuais já são uma realidade em quase todos os smartphones dos grandes fabricantes do mercado; mas o mercado para este tipo de tecnologias associadas a ambientes residenciais ainda está em crescimento. Em [1] é afirmado que este ano (2020) o mercado mundial de IoT (Internet of Things) atingirá US\$1.7 biliões e é previsto que 25% das residências vão estar equipadas com dois ou mais assistentes virtuais inteligentes.

O projeto, para além dos assistentes virtuais, também envolve gestão da rede doméstica através de um router. Segundo a ANACOM [2], o subsetor da Internet fixa está em penúltimo lugar no ranking de satisfação dos clientes portugueses e, com 30,6% dos clientes a apresentarem reclamações, encontra-se no primeiro lugar no ranking de reclamações por setor e subsetor. Estes valores indicam que os utilizadores experienciam bastantes problemas com as suas redes domésticas. Com a introdução destas tecnologias, o número de reclamações e o grau de insatisfação dos utilizadores deve baixar tendo em conta que o principal fator causador da insatisfação dos clientes é o facto de, muitas vezes, não ser trivial a identificação da raiz do problema que provoca os distúrbios na rede doméstica, algo que o assistente virtual, juntamente com a tecnologia implementada no router, poderá facilitar ou tornar mais trivial para uma significativa parte dos problemas.

Segundo a ANACOM [3], 41% dos utilizadores (55% na grande Lisboa) considera o tempo de espera demasiado elevado. Este dado indica alguma insatisfação do lado dos clientes para com o serviço de apoio ao cliente.

A dissertação está enquadrada com o projeto HAL da empresa Altran, onde foi implementado o sistema. O projeto integra-se na equipa de R&D (*Research and Development*).

## 1.2 Desafio e Solução

Nesta dissertação foi desenvolvida uma solução onde os assistentes virtuais poderão adicionar valor na gestão de redes domésticas, através de comandos de voz. A maioria dos utilizadores dos serviços de telecomunicações têm conhecimento limitado acerca do funcionamento da rede. Quando um cliente liga para um call center tem dificuldades em identificar a raiz do problema, complicando a resolução do mesmo. O sistema será capaz de realizar algumas funcionalidades úteis na perspetiva de prestar apoio ao utilizador na gestão da sua própria rede.

Neste projeto foi desenhado e implementado um assistente virtual com comunicação bidirecional acionado por voz que assiste o utilizador na resolução de problemas associados ao mau funcionamento de uma rede de comunicação doméstica, nomeadamente problemas associados a má conexão e perda de funcionalidades.

O sistema acima descrito foi implementado numa plataforma de desenvolvimento para ambientes de router. O utilizador comunica através de uma aplicação associada ao assistente virtual no dispositivo móvel que processa a mensagem de voz do utilizador e envia para o router através de um protocolo de comunicação para ambientes de IoT. Após a chegada da informação ao router é feito o processamento da mensagem, dependente do input do utilizador, que têm como output uma resposta apresentando uma solução ou solicitando mais informação através de mensagens de voz criada pelo assistente.

Resumidamente as contribuições desta tese são:

- Definição de cenários de interação humano-assistente, com identificação de problemas que podem ser resolvidos no contexto de redes domésticas, identificação do campo lexical associado, e desenho dos passos seguidos pelo sistema para depuração do problema.
- Implementação do sistema em routers domésticos e *smartphone* do utilizador, com tecnologia de assistente de voz. O sistema será capaz de identificar o problema e, se possível, resolvê-lo.
- Avaliação do sistema por utilizadores reais. Foi realizada uma avaliação da experiência do utilizador de modo a identificar os pontos fortes e os pontos fracos do sistema.

Os testes ao sistema pretendem não só avaliar a performance do sistema, mas também a experiência do utilizador. Foi selecionado um grupo de pessoas com diferentes perfis para avaliarem o sistema em certos cenários. Os resultados obtidos foram importantes na medida em que evidenciaram aquilo que podia ser melhorado. De um modo geral, os utilizadores ficaram satisfeitos com o desempenho e a abordagem ao assistente virtual.

### 1.3 Estrutura do documento

Nos próximos dois capítulos do presente documento é feito o estudo do estado da arte que se encontra dividido em duas grandes secções, [2](#) e [2.2](#).

Num primeiro capítulo será feita uma revisão tecnológica, mais orientada para o estudo das tecnologias envolventes. No segundo capítulo é realizada uma revisão científica/análise de literatura.

Nos restantes capítulos será descrita a proposta de arquitetura juntamente com o funcionamento do sistema, [3](#). Seguidamente, será explicado como foi implementado o sistema, a explicação dos casos de uso e algum detalhe técnico sobre implementação, [3.3](#).

Na parte final do documento será examinada a avaliação da experiência do utilizador, com os testes realizados e os resultados obtidos, [4](#), acabando com uma conclusão acerca do sistema e do trabalho realizado, juntamente com algumas notas sobre trabalho futuro, [5](#).

## Capítulo 2

# Tecnologias, Literatura e Trabalhos Relacionados

Neste capítulo vão ser abordados tópicos como assistentes virtuais e o seu modo de funcionamento e as tecnologias associadas, bem como sistemas operativos para routers e as suas funcionalidades. Numa outra secção são abordados os trabalhos relacionados que foram revistos no âmbito da realização deste projeto.

### 2.1 Tecnologias relevantes

#### 2.1.1 Assistentes Virtuais de referência

Os assistentes virtuais baseados em comandos de voz estão a crescer em popularidade tanto em dispositivos móveis como domésticos. Assistentes como o Amazon Alexa, Google assistant, Apple Siri, Microsoft Cortana, entre outros, têm a capacidade de auxiliar o utilizador com uma panóplia de funcionalidades desde informação acerca do tempo até à reserva de quarto de hotel [4]. Estes assistentes são controlados por voz e contêm diversas funcionalidades como tocar música, controlar smart devices, informar o utilizador acerca de diversos assuntos, entre outras funcionalidades.

Verifica-se que entre estas principais referências de assistentes virtuais não há nenhum que se destaque em relação aos outros de uma maneira global. No entanto existem algumas diferenças naquilo que são considerados os indicadores de performance de um assistente virtual (naturalidade da fala, a exatidão na resposta) dependendo da funcionalidade que esteja a ser avaliada [5].

Estes assistentes virtuais de referência podem ser integrados num equipamento que permite a conexão a vários dispositivos IoT de modo a criar uma rede de conexões na residência gerida pelo utilizador através do assistente (Smart Home). Dados estatísticos indicam que o Amazon Echo, que integra a Alexa, e o Google Home, que integra a Google assistant, são os assistentes virtuais mais usados nos EUA, que por sua vez é o país onde esta tecnologia tem mais influência. Os dois produtos acima referidos conseguem ser bastante mais completos do que o produto desenvolvido pela Apple (Apple HomePod) ou pela Microsoft (Microsoft/Harman Kardon's Cortana). Este

dado deve-se principalmente à compatibilidade que estes dois assistentes apresentam perante os periféricos disponíveis e também devido ao número de funcionalidades que apresentam [6] [7].

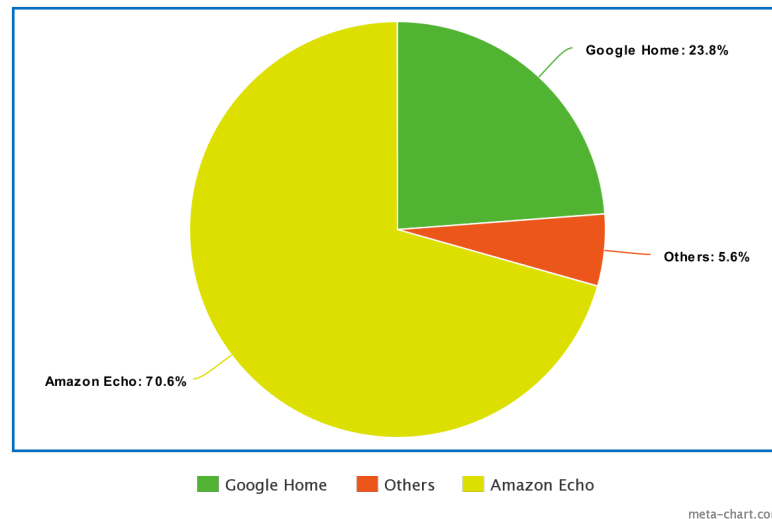


Figura 2.1: Distribuição dos assistentes virtuais nos EUA (2017) [6]

### 2.1.2 Assistentes virtuais *open source*

Uma outra alternativa relativamente a este tópico será o uso de assistentes virtuais *open source* que oferecem diversos tipos de funcionalidades e permitem ao utilizador a implementação de novas. Assistentes como a Snips [8] e o Mycroft [9] permitem que seja feita uma adaptação por parte do utilizador de modo a que cumpram determinada função estabelecida pelo mesmo.

O assistente virtual Mycroft é desenvolvido por uma comunidade ativa que contribui para a implementação de novas funcionalidades do assistente virtual. O utilizador poderá adquirir as funcionalidades desejadas através de um marketplace disponível online. O Mark I e o Mark II são dispositivos análogos ao Google Home e ao Amazon Echo que integram o Mycroft. No entanto este pode ser integrado em diversas plataformas tais como Linux, Windows, Android e Raspberry Pi 3 ou 3B+. Neste momento o Mycroft também pode ser usado como assistente para Smart Homes, sendo possível o emparelhamento com dispositivos inteligentes [9].

A assistente virtual Snips segue a mesma lógica que o Mycroft, ou seja, é desenvolvido por uma comunidade, está disponível para diversas plataformas e contém uma vasta biblioteca de funcionalidades disponíveis online. Uma grande diferença que este assistente apresenta relativamente ao Mycroft é o facto de poder realizar o processamento offline (sem acesso a uma cloud). Esta característica reforça a segurança do sistema em que o assistente está envolvido na medida em que não há disseminação dos dados pessoais para a cloud, os *inputs* de voz são processados offline e nunca deixam o dispositivo em questão. Este reforço de segurança vem com uma consequência, o espaço ocupado pelo módulo de processamento do assistente virtual vai ser bastante maior devido a ter que integrar as tecnologias de reconhecimento/síntese de fala [8].

### 2.1.3 Tecnologias associadas a assistentes virtuais

A comunicação humana é maioritariamente realizada através da fala. Os diferentes pensamentos formados na mente de cada um são comunicadas na forma de palavras, frases ou expressões aplicando um conjunto de regras gramaticais. A fala humana pode ser decomposta num aglomerado de sons produzidos por cada letra do alfabeto que constitui uma palavra, chamados fonemas. Os métodos de conversão de STT (Speech-to-text) e TTS (Text-to-speech) passam essencialmente pela interpretação e categorização de fonemas de modo a conseguir obter uma correta interpretação daquilo que foi capturado no input, tanto texto como fala [10].

Todos os assistentes virtuais usam tecnologias para processar os comandos de voz que recebem, transformando-os em texto, e de maneira análoga transformar texto em sinais sonoros que pretendem imitar a linguagem natural. Desta forma são usadas tecnologias de reconhecimento de voz (STT e TTS) em paralelo, de modo a criar uma interação entre o utilizador e o assistente.

Apesar de não terem sido implementadas, consideramos relevante o estudo destas tecnologias de modo a ter uma visão global do funcionamento das diferentes tecnologias presentes no sistema.

#### 2.1.3.1 Speech-to-text

A técnica de STT passa por receber um input de um microfone (um sinal acústico), e a partir desse input gerar um conjunto de palavras. Este processo pode ser dividido em 2 grandes etapas. Primeiramente, uma fase de treino, onde o sistema aprende os padrões de referência que representam os diferentes sons produzidos na fala. Em segundo, uma fase de reconhecimento de fala, onde é capturado um sinal acústico desconhecido que posteriormente será identificado considerando um conjunto de referências. Esta segunda etapa pode ser subdividida em 4 fases segundo [10].

Na fase de análise é retratado o perfil do orador. O sinal acústico terá diferentes características dependendo de quem o produz. De modo a ser feita uma correta conversão para texto é necessário previamente traçar este perfil vocal do orador. Concluída esta fase, dá-se início à extração de características do sinal acústico, sendo esta a fase mais importante deste processo por fazer corresponder o reconhecimento do sinal acústico com o modelo treinado de fala. A maioria dos modelos de fala são gerados a partir da implementação de um HMM (Hidden Markov Model), sendo esta a maneira mais bem sucedida e flexível na atualidade [11]. Um HMM é um modelo probabilístico que faz a correspondência entre a fala e o texto [12].

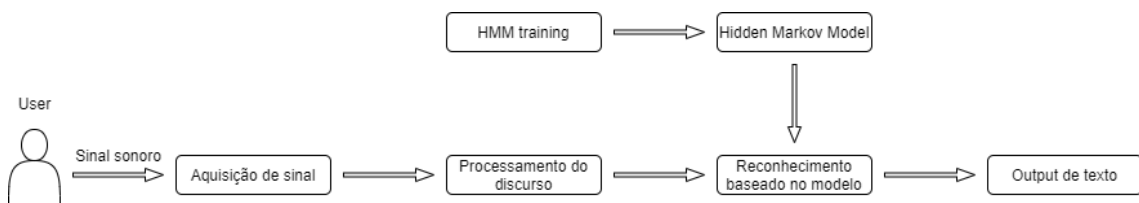


Figura 2.2: Arquitetura do modelo STT para reconhecimento de fala

### 2.1.3.2 Text-to-speech

A síntese de fala resume-se à transformação de um input textual numa onda sonora artificial que tenta retratar aquilo que é a fala humana. De maneira análoga aos sistemas STT, o modelo standard para sistemas TTS divide-se em duas partes. Numa primeira fase é feita a análise e processamento de texto (também chamada de front-end). O input de um sistema TTS é muitas vezes texto presente em documentos eletrónicos, jornais, emails, blogs etc. Este tipo de texto contém várias irregularidades, sendo estas palavras não standard que não estão presentes no dicionário da língua envolvente tais como, números, abreviações e símbolos construídos com sinais de pontuação tais como ‘!’ ou ‘:-)’ [13]. O objetivo deste módulo é processar o texto de input de modo a normalizar as secções de texto não standard e gerar a correta sequência de sons. Existem também palavras que são pronunciadas de maneira diferente dependendo do contexto em que se encontram. De modo a compreender e aplicar a correta entoação às palavras não standard são usados métodos de machine learning como redes neuronais ou HMMs. Nesta fase também é realizada uma análise prosódica, ou seja, são analisadas as propriedades fónicas de cada sílaba/palavra de modo a estabelecer a duração e a entoação de cada som para cada sílaba/palavra, são usadas técnicas como o mapeamento de um conjunto de regras ou através de machine learning [13].

A segunda fase deste processo (também chamada de back-end) é responsável pela síntese da fala através da concatenação dos sons gerados na primeira fase. Os métodos usados na conversão dos sons para a forma de onda da fala a produzir podem ser categorizados em síntese paramétrica, concatenativa e estatístico-paramétrica [13].

- Síntese paramétrica: Parâmetros como formantes (indicador da forma espectral do som através de valores de pico de frequência) e coeficientes de previsão linear são extraídos de cada fonema. Desta forma é possível gerar sinais sonoros através articulação destes parâmetros com o output do módulo de processamento de texto. Este método na atualidade não é usado devido ao facto do resultado final produzido ser um som demasiado “robótico”.
- Síntese concatenativa: Esta técnica baseia-se na concatenação de sons previamente gravados com a duração estabelecida pela fase de processamento de texto com o objetivo de gerar um sinal sonoro. A maneira mais eficaz de usar este método é usando a concatenação de unidades, sendo estas fonemas, bifonemas, sílabas, palavras e até frases. O resultado final deste método é reconhecido como o que produz a onda sonora mais próxima do natural, no entanto apresenta alguma inconsistência na qualidade do som.
- Síntese estatístico-paramétrica: A abordagem estatístico-paramétrica é a mais recente dos métodos estudados e baseia-se na produção de som através de parâmetros que são resultado de um processo de machine learning. O processo é equiparável à síntese paramétrica, a diferença mais notória é a aplicação de um processo de machine learning (HMM ou deep neural networks) para que a concatenação dos sons seja feita da maneira mais natural possível. Este método produz uma fala consistente e próximo do natural comparando com os outros métodos acima referidos.



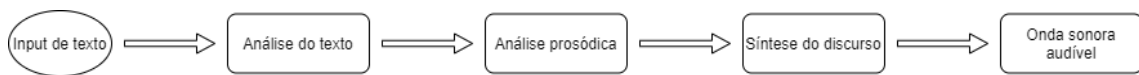


Figura 2.3: Arquitetura do modelo TTS para síntese de fala

### 2.1.3.3 Wake Word listener

A tecnologia *Wake Word listener* tem como objetivo detetar quando determinada palavra ou expressão é pronunciada pelo utilizador e, a partir desse momento, processar aquilo que é dito. O Mycroft Precise é uma tecnologia que implementa essa mesma funcionalidade, é *open source* e baseada em *machine learning*. Qualquer utilizador pode definir uma *wake word* treinando o modelo para reconhecer a expressão desejada [14].

Esta tecnologia trás alguns problemas a nível de segurança devido à constante captura de dados que o assistente virtual faz. Dados pessoais, rotinas, localização, etc. são informações que o assistente virtual consegue facilmente obter e, no caso dos assistentes cloud-based, esses dados vão ser colocados numa rede global onde podem ser facilmente acedidos por hackers/piratas [15].

### 2.1.3.4 Meaning/intent parser

Esta tecnologia pretende atribuir um significado/intenção àquilo que é dito pelo utilizador, transformando a fala numa estrutura de dados perceptível pelo computador associada a uma percentagem de confiança que indica a probabilidade da interpretação feita pelo intent parser ser correta. O assistente virtual Mycroft possui um serviço que implementa esta tecnologia, o Mycroft Adapt [16].

## 2.1.4 Sistemas operativos para ambientes de router

Existem diversas soluções para o sistema operativo a usar no router. Nesta secção pretende-se compará-las e caracterizá-las, dando mais ênfase à solução escolhida, o OpenWRT.

A escolha do sistema operativo vai-nos permitir desenvolver a aplicação que será capaz de interagir com a rede doméstica, obtendo informação acerca do seu estado e reparando eventuais não-conformidades.

O DD-WRT é uma alternativa *open source* baseada em Linux preparada para funcionar em diversos routers WLAN (Wireless Local Area Network) e sistemas embebidos. O sistema procura ser de simples interface mas ao mesmo tempo com uma grande diversidade de funcionalidades. Pode ser configurado através de um browser web, o que facilita a interação com um utilizador não experiente e contém algumas características de destaque, assim como: a capacidade de poder trabalhar com frequências melhoradas (melhoram o alcance e velocidade de transmissão do sinal) e a possibilidade de gerir as larguras de banda alocadas a cada dispositivo conectado [17].

O OpenWRT/LEDE é também uma alternativa *open source* baseada em Linux/GNU desenvolvida para operar em sistemas embebidos, tipicamente implementada em routers sem fios. O

OpenWRT é desenvolvido de forma a tornar-se num sistema operativo com todo o tipo de funcionalidades e de fácil manuseamento. É possível desenvolver aplicações sem a criação de firmware e a interface é feita por linha de comando ou página web. De maneira análoga ao DD-WRT, o OpenWRT também permite a gestão de larguras de banda [18].

Existem também sistemas operativos para routers de carácter comercial como o Cisco IOS que não opera apenas no router; é um pacote de roteamento, switching, *internetworking* e funções de telecomunicações totalmente integrado com um sistema operacional de computador multi-tasking [19].

### 2.1.5 Protocolos de comunicação

O leque de soluções para a implementação de um protocolo de comunicação numa rede IoT é vasto. Protocolos como MQTT (Message Queuing Telemetry Transfer), AMQP (Advanced Message Queuing Protocol) e DDS (Data Distribution Service) encaixam bem na solução proposta devido ao facto de implementarem o modelo publisher/subscriber que se enquadra bem com o tipo de interação pretendida, isto é, vários processos vão estar a ser executados em paralelo e apenas recebem a mensagem correspondente ao serviço que subscreveram.

O protocolo MQTT funciona com um broker que serve de componente central na via de comunicação e a sua função passa por receber as mensagens publicadas e distribuí-las pelos clientes que subscreveram os respetivos tópicos. O protocolo destaca-se pelo facto de usar poucos recursos. É ideal para ambientes em que o consumo de energia ou a memória do sistema são limitados. Outra característica relevante é o facto de ser possível armazenar mensagens para novos subscritores através de uma flag "retain". Com esta flag a *true*, o broker é informado que deve guardar a mensagem para que seja entregue aos novos subscritores. O MQTT funciona sob uma rede TCP (Transmission Control Protocol) o que garante a fiabilidade acrescida [20].

O DDS funciona de maneira análoga (Publisher/Subscriber) mas contém a particularidade de ser descentralizado, ou seja, não possui um elemento central (broker) que gere a entrega das mensagens. A comunicação é feita ponto a ponto, o que faz com que a fiabilidade do canal de comunicação seja maior porque deixa de existir um elemento onde possa haver erros na propagação da mensagem. Este protocolo pode ser implementado tanto sob TCP como UDP [20].

O AMQP segue também o modelo publisher/subscriber e pode funcionar com ou sem broker (ponto a ponto). A sua característica de destaque é o facto de ter compatibilidade com diversas aplicações e sistemas. No entanto isto torna o protocolo pesado e apenas indicado para redes sem restrições de latência e largura de banda [20].

## 2.2 Análise de literatura e trabalhos relacionados

Esta secção dá início aos à apresentação dos casos de estudo em que foram implementadas tecnologias que foram usadas neste projeto, é feita a interpretação dos casos e é também evidenciado aquilo em que se destacam.

### 2.2.1 Home Automation Systems (HAS)

Nesta secção vão ser analisados alguns trabalhos realizados no âmbito da domótica. O estudo destes trabalhos é relevante para a realização deste projeto na medida em que são projetadas arquiteturas de comunicação/interação entre assistentes virtuais e equipamentos físicos que realizam determinadas funções dentro da residência.

Em [21] foi implementada uma plataforma de middleware baseado em mensagens que se conecta com diferentes tipos de dispositivos IoT de forma a conseguir monitorizar a energia gasta pelos aparelhos domésticos. A interface é feita através da aplicação Telegram (aplicação para troca de mensagens de texto e conteúdos multimédia), onde o utilizador comunicava com um assistente virtual (chatbot) que administrava a rede doméstica e recolhia informações sobre os consumos dos dispositivos conectados à rede.

A arquitetura para o chatbot baseia-se em compreender comandos de voz, mensagens de texto ou comandos pré-definidos e dar como resposta uma mensagem de texto ou voz. Após a implementação do chatbot era necessário desenvolver o middleware responsável por recolher os dados em tempo real e também conectar-se com uma base de dados que contém informação acerca das interações com o utilizador e um historial de consumos dos aparelhos.

Para a monitorização e demonstração de resultados foi usada uma API (Application programming interface) para tratamento e demonstração de dados estatísticos, ThingSpeak. Esta API mostra o consumo em KWh e o custo associado em diferentes escalas de tempo (por dia, mês, etc).

Como resultado deste projeto, obteve-se a informação relativa a dois meses de consumo e foi possível analisar que aparelhos é que consumiam mais energia de forma a rentabilizar os gastos.

Vários projetos foram realizados com o objetivo de desenhar e implementar uma arquitetura de controlo de equipamentos em ambiente doméstico. Por exemplo em [22] foi desenvolvida uma arquitetura deste tipo que, para além do controlo via assistente virtual, suporta controlo remoto dos equipamentos através de uma interface web. Foram colocados microcontroladores em diferentes divisões com sensorização capaz de medir temperatura e humidade, juntamente com um microcontrolador central que controla a luz de todas as divisões. Os estados das luzes e os valores dos sensores são capturados e guardados numa base de dados Firebase, que por sua vez, também contém os dados dos utilizadores que têm acesso à aplicação através de um Login. O microcontrolador responsável pelo controlo das luzes recebe as mensagens de controlo por MQTT (Message Queuing Telemetry Transport) e controla assim o estado das luzes.

A arquitetura desenvolvida contém uma camada integra vários tipos de serviços de fonte aberta, o Firebase para a base de dados, WebSpeech da Google para o reconhecimento de fala e o DialogFlow para a integração do sistema de Natural language processing (NLP) com o sistema IoT. Sendo esta camada o middleware, é responsável por comunicar com o utilizador através da interface web, e atuar no hardware baseada nos comandos de input. Para avaliar a performance do produto final foram realizados testes baseados na medição de tempos de resposta do sistema em diferentes tipos de redes, ao serem introduzidos comandos pelo utilizador. Verificou-se que no pior caso a resposta numa rede lenta de 3G (0.36 Mbps) ronda os 2,5 segundos e que no melhor dos casos, rede ADSL (Assymetrical Digital Subscriber Line, a 3.93 Mbps) o tempo de resposta ronda os 0.5 segundos. É de salientar que cerca do 70% do tempo de resposta é gasto pelo módulo de NLP e os restantes 30% são gastos pelo processamento nos microcontroladores e pela comunicação entre módulos.

Outro exemplo do mesmo género está descrito em [23] no qual foi desenvolvido um Home Automation System (HAS) controlado a partir de uma aplicação Android conectada a um microcontrolador que por sua vez comunica com os equipamentos da casa e com um web server. O funcionamento deste sistema passa por receber comandos de voz do utilizador através do dispositivo móvel e com uma ferramenta de reconhecimento de voz é feita a interpretação desse comando e enviado, em forma de texto, para o microcontrolador e, posteriormente, reencaminhado para o web server. Nesta fase é feita uma seleção de keywords presentes numa base de dados que associa o comando do utilizador com uma determinada ação. O resultado desta operação é enviado para o microcontrolador e aí é tomada a ação desejada no dispositivo definido.

Em [24] também foi desenvolvido um projeto semelhante mas com a particularidade de poder comunicar com o utilizador via SMS (Short Message Service) ou e-mail para o envio de alertas.

### **2.2.2 Sistemas que usam/implementam Voice Assistants**

Nesta secção serão estudados alguns projetos relativos a assistentes virtuais. A cobertura deste tópico é importante para perceber as diversas aplicações que os assistentes virtuais têm nos dias de hoje.

Diversos trabalhos foram realizados com assistentes virtuais. Por exemplo em [25] foi desenvolvido um protótipo de assistente virtual que reage em função das emoções que o utilizador apresenta quando em contacto com o assistente. Neste projeto as emoções do ser humano foram categorizadas em oito diferentes tipos e cinco diferentes intensidades, representadas em 2.1. Para a captura de emoções são localizados os pontos chave na face do utilizador (olhos, nariz e boca) e é recolhida informação acerca da posição e movimento desses pontos. Após a recolha de dados é criado um modelo 3D que representa a face em questão e através de um HMM são associadas emoções à imagem capturada. Foram conseguidos resultados satisfatórios no que toca à identificação das emoções mas foram encontrados alguns aspetos a melhorar, a intensidade da emoção tornou-se bastante difícil de identificar e ainda há margem de melhoria no que toca à identificação das emoções.

| Intensity<br>type | None | Little | middle | strong | Very strong |
|-------------------|------|--------|--------|--------|-------------|
| Happiness         | 1    | 2      | 3      | 4      | 5           |
| Acceptance        | 1    | 2      | 3      | 4      | 5           |
| Surprise          | 1    | 2      | 3      | 4      | 5           |
| Fear              | 1    | 2      | 3      | 4      | 5           |
| Sadness           | 1    | 2      | 3      | 4      | 5           |
| Disgust           | 1    | 2      | 3      | 4      | 5           |
| Interesting       | 1    | 2      | 3      | 4      | 5           |
| Anger             | 1    | 2      | 3      | 4      | 5           |

Tabela 2.1: Classificação das emoções e a sua intensidade segundo [25]

Um outro exemplo de implementação de assistente virtual está em [26]. Foi criado um assistente através de uma página web que interage com alunos universitários e esclarece dúvidas em relação às funcionalidades/serviços disponibilizados pela universidade. Inicialmente foi feita uma recolha de dados de um grupo de estudantes para aferir que tipo de dúvidas e que tipo de linguagem os estudantes usavam para interagir com os serviços académicos. Após a recolha destes dados foi então implementado o assistente virtual e feito um questionário de satisfação no fim. A aprovação dos estudantes foi bastante positiva, o que mostra que há uma grande aceitação deste tipo de tecnologia por parte dos jovens.

Também em [27] foi desenvolvido um assistente virtual capaz de auxiliar o utilizador em compras online. O assistente é dividido em agentes responsáveis por 4 diferentes etapas na interação com o utilizador. Existe um agente responsável pela interface do utilizador que recebe os inputs textuais e os traduz num ficheiro XML, um segundo agente que analisa o ficheiro XML (Extensible Markup Language) e determina a próxima ação, outro agente que faz a pesquisa dos artigos na base de dados, que por sua vez, é o último agente. Na figura 2.4 está descrita a interação entre os diversos agentes. O próximo passo deste projeto seria a implementação de uma busca gerida por voz.

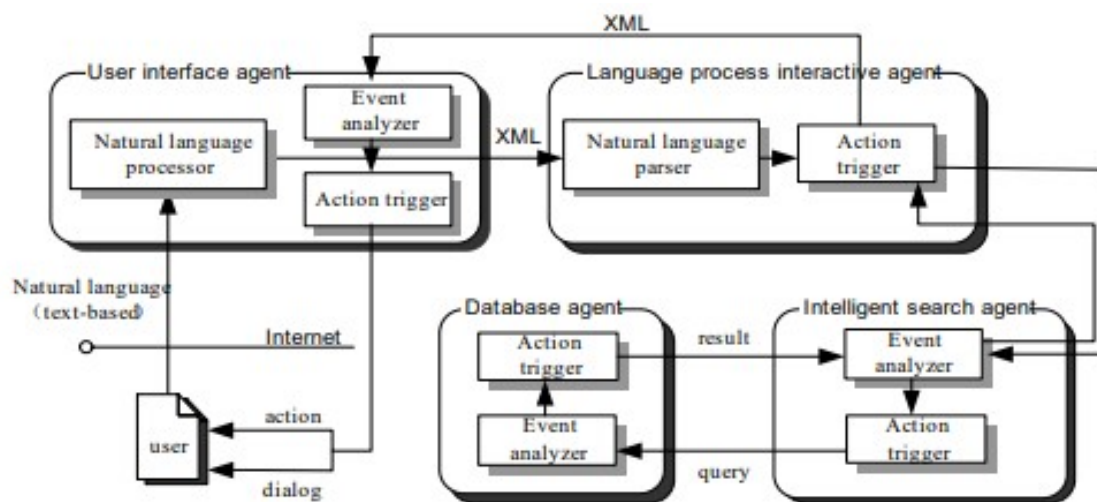


Figura 2.4: Estrutura do modelo multi-agente [27]

Por último em [28] foi desenhado um assistente virtual capaz de assistir interação humano-humano em ambientes virtuais. A função do assistente é dar sugestões de tópicos de conversa aos participantes da conversa baseado em informação que ele recolhe durante a interação. O objetivo deste assistente está focado nas interações entre pessoas de culturas diferentes de modo a colmatar as diferenças culturais através de sugestões. Foram realizados testes bem sucedidos com interações entre Americanos e Japoneses.

## 2.3 Considerações finais

Neste capítulo começamos por rever a oferta de assistentes virtuais com suporte de voz, tanto proprietários como de fonte-aberta.

Apesar de não implementarmos as tecnologias associadas ao reconhecimento/síntese de fala (TTS e STT), achamos relevante apresentá-las por serem um aspeto crítico dos assistentes virtuais baseados em voz.

O nosso sistema assentará em tecnologias IoT, nomeadamente tecnologias de programação para routers e mecanismos de comunicação do tipo *Publisher-Subscriber*; neste capítulo procuramos dar uma visão global do estado da tecnologia nestes dois campos, salientando as vantagens das tecnologias OpenWRT e MQTT, respetivamente.

Os vários estudos revistos mostram a relevâncias dos assistentes virtuais e as inúmeras possibilidades de integração com outras tecnologias. O uso dos assistentes vem trazer uma melhoria significativa na experiência do utilizador em diversas áreas tecnológicas e promove a interação do utilizador com os sistemas que o rodeiam no quotidiano.

Não foi encontrado nenhum estudo que relaciona diretamente o uso de assistentes virtuais na gestão de redes domésticas, no entanto foram estudadas diversas implementações de assistentes virtuais em outras áreas.

Retirou-se como aspetos mais relevantes a medição das métricas de desempenho do sistema, ou seja, a percentagem de sucesso numa interação com o assistente, e de experiência do utilizador, na qual foi avaliado o sistema na perspetiva do utilizador.

## Capítulo 3

# Assistente virtual para redes domésticas

Neste capítulo é apresentada a arquitetura conceptual do sistema bem como o seu modo de funcionamento seguido de uma descrição dos casos de uso considerados. Após a apresentação inicial do sistema, serão explicados os casos de uso considerados seguido da secção onde é explicada a implementação do sistema.

### 3.1 Arquitetura e funcionamento do sistema

A arquitetura do sistema foi desenhada para operar num contexto doméstico em que o utilizador interage com o assistente através de um equipamento próprio (p.ex. smartphone). É necessária uma conexão à Internet para o sistema funcionar.

Na figura 3.1 estão listados os diferentes componentes lógicos do sistema e a maneira como interagem:

- O sistema de reconhecimento e síntese de voz, constituído pelo dispositivo móvel do utilizador e pelas tecnologias de STT e TTS, é responsável pela captura do comando do utilizador e reencaminhamento da mensagem para a *cloud*, onde é feito o processamento. No caso de ser o assistente a falar com o utilizador, é realizada a operação contrária.
- O gestor do assistente é então o elemento responsável pela interpretação da mensagem do utilizador e toma as decisões necessárias para realizar a ação pretendida pelo utilizador.
- O Agente de configuração de rede é o elemento que faz a gestão da rede diretamente, ou seja, altera configurações, realiza diagnósticos e ativa funcionalidades.

#### 3.1.1 Interação com o utilizador

O sistema entra em ação quando há uma interação com o utilizador: a partir do momento que é pronunciada a *wake word* o assistente virtual irá capturar aquilo que o utilizador disser. O *input* do utilizador é capturado e enviado para os serviços de *cloud* do assistente virtual, onde entra em ação a tecnologia de STT e então é convertido o discurso do utilizador em texto que vai servir de



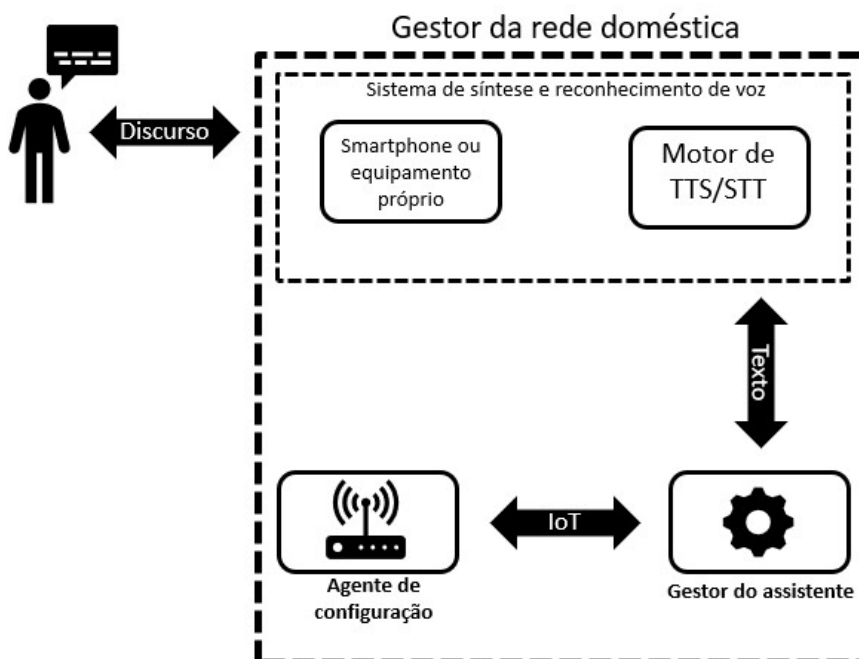


Figura 3.1: Arquitetura do sistema

input para o gestor do assistente. Após receber o input, o gestor analisa-o e extrai a intenção do utilizador associando-a a uma possível ação e/ou resposta. Este mecanismo será explicado com mais detalhe na secção de implementação, 3.3.2, do presente documento. Concluído este passo existem duas hipóteses: o gestor gera uma resposta imediata para o utilizador sem ter que recorrer ao *router* ou então é enviada uma mensagem para o *router* cujo conteúdo indica a tarefa que ele terá que completar.

Na primeira hipótese a resposta imediata pode ser de diversos tipos:

- Um pedido ao utilizador para dar permissão para realizar determinada tarefa;
- Uma pergunta para recolher mais informação sobre o estado da rede;
- Uma afirmação que indica ao utilizador que terminou a interação com o assistente virtual.

Na segunda hipótese o *router* receberá então a mensagem vinda do gestor do assistente e irá efetuar o processamento necessário de modo a recolher os resultados dos testes feitos ou a alterar qualquer configuração da rede. Após este processamento, o *router* irá enviar uma mensagem de resposta para o gestor com uma confirmação daquilo que foi alterado ou com os resultados dos testes feitos. O gestor, recebendo esta mensagem, começa a construir a resposta para o utilizador com os dados do *router* e envia-a para *cloud* de novo para ser feita a operação contrária à que foi feita inicialmente, ou seja, transformar o texto em discurso através do software de TTS da *cloud*.

É assim construída a resposta ao utilizador que é enviada para o dispositivo que contém o assistente virtual e, por fim, é comunicada a resposta. Este processo ocorre sempre que o utilizador fala para o assistente virtual.

### 3.1.2 Operação interna do sistema

Na figura 3.2 é apresentado sob a forma de diagrama a interação que o utilizador tem com o sistema de uma maneira genérica, adaptada a qualquer caso de uso.

O utilizador vai ativar o sistema quando ocorrer alguma espécie de problema ou quando simplesmente quiser usar uma funcionalidade do assistente. Esse *input* do utilizador vai ser processado e o assistente vai decidir se é necessária a interação com o *router* ou não face ao que foi dito pelo utilizador. Em caso afirmativo significa que há informação necessária para realizar uma ação no *router* e então o assistente vai comunicar com o aparelho para dar *trigger* à rotina que realiza a ação esperada. Em caso negativo tem-se duas hipóteses: é o fim da interação (não há mais nada a fazer por parte do assistente ou o utilizador não quis continuar a conversa) ou o assistente virtual precisa de mais informação para tomar uma ação e vai esperar pela resposta do utilizador.

Em qualquer situação em que o assistente esteja à espera de uma resposta do utilizador é ativado um *timeout* de cerca de quatro segundos. No caso do utilizador não responder a tempo é efetuada uma segunda tentativa por parte do assistente de forma a tentar obter a resposta do utilizador, ou seja, o assistente coloca novamente a pergunta, podendo ou não ser de maneira diferente. Após a segunda tentativa é ativado, também uma segunda vez, o *timeout* de quatro segundos e se o utilizador não responder a interação é dada por terminada por parte do assistente.

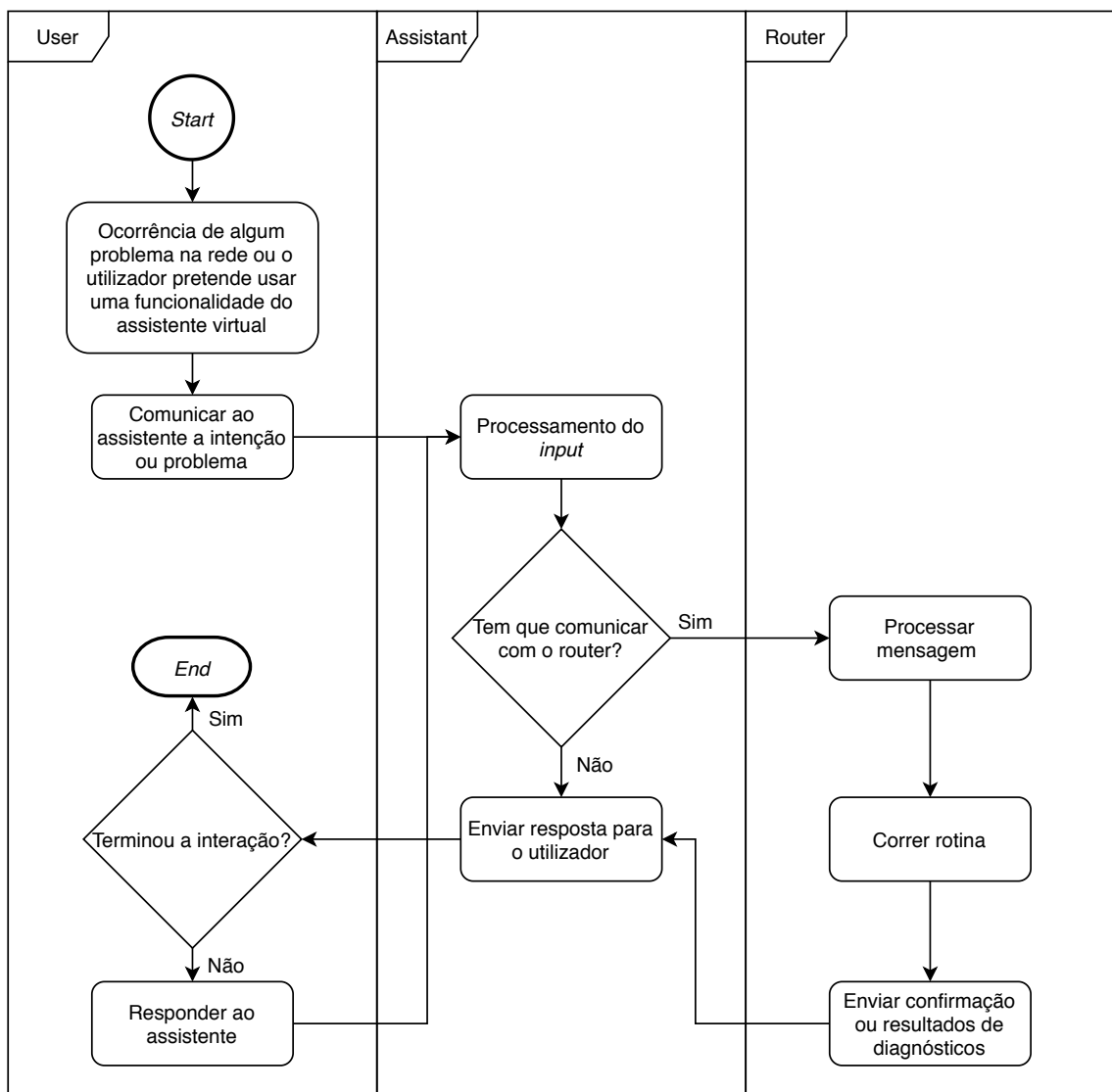


Figura 3.2: Fluxograma de operação interna do sistema numa interação genérica

## 3.2 Casos de uso

Nesta secção vão ser introduzidos os casos de uso implementados no sistema. Os casos de uso que se seguem foram pensados e desenhados na ótica de explorar as funcionalidades e capacidades dos *routers* domésticos. A integração do assistente virtual com o sistema operativo do *router* possibilita a utilização de funcionalidades de rede sem necessidade de conhecimentos específicos por parte do utilizador.

A utilização da rede *WI-FI* é dos problemas mais comuns nos clientes, desta maneira é necessário demonstrar a resolução de problemas associados às redes *WI-FI*.

### 3.2.1 Criação de rede Guest

O primeiro caso de uso a analisar é uma funcionalidade que permite ao utilizador ativar uma nova rede de acesso à Internet sem password.

Numa situação em que o utilizador pretenda ter convidados em sua casa é uma questão de tempo até que alguém peça acesso à rede *WI-FI*. Com esta funcionalidade, o utilizador pode através de um comando de voz criar a tal rede que permite aos convidados ter acesso à Internet sem a necessidade da password. Isto substitui o facto de o utilizador ter que ditar a password aos utilizadores por uma maneira mais eficiente de resolver o problema. Também é possível agendar a ativação da rede para uma determinada hora. Da mesma forma também será possível desativar a rede sempre que for necessário usando os comandos de voz.

A criação de uma rede *Guest* isolada contribui para a segurança e privacidade da rede doméstica. Especialmente quando consideramos a presença de sistemas de domótica, com a rede isolada os convidados não terão acesso a estes.

Na seguinte listagem estão apresentadas algumas das frases que podem ser pronunciadas para o assistente de modo a despoletar este caso de uso:

- «Share my Internet»
- «Turn on the Guest wi-fi»
- «Activate the wi-fi for my guests»
- (...)

### 3.2.2 Identificação da razão de problemas de conectividade

O segundo caso de uso explora o contexto em que o utilizador está a experienciar problemas no uso da rede. Existem várias origens possíveis para este tipo de problema que podem ser categorizadas como internas ou externas. As falhas internas podem ser resumidas a:

- Falta de sinal *wi-fi*;
- Uso excessivo da rede por parte de um outro aparelho/utilizador;
- Vários dispositivos conectados;
- Falha de hardware.

Estes problemas, excluindo a falha de hardware, podem ser facilmente resolvidos com a intervenção do utilizador. A falta de sinal pode ser colmatada com uma maior aproximação física do *router*, o uso excessivo por parte de outro aparelho/utilizador é resolvido com uma interrupção do processo que está a acontecer no momento, e o número de dispositivos conectados também pode ser sempre reduzido. Olhando agora para as falhas externas, existem duas principais:

- Falha no servidor de um determinado serviço que esteja a ser utilizado;

- Falha na rede externa da operadora.

Neste tipo de falhas não há nada que o utilizador possa fazer visto que a origem do problema não está presente na sua rede doméstica. O utilizador terá que esperar que o serviço esteja *online* novamente. Nestas ocasiões o assistente virtual é particularmente útil para os utilizadores que não possuem conhecimento vasto nesta área da tecnologia e pode ajudar na identificação dos problemas bem como uma sugestão para os resolver.

Com o teste de diagnóstico feito pelo sistema é possível distinguir se o problema tem origem internamente ou externamente. É muito provável que o utilizador esteja numa situação de falha externa quando é feito o teste de diagnóstico e se verifica que existe bom sinal no dispositivo do utilizador, não há saturação da rede a nível de número de dispositivos e a largura de banda é suficiente para realizar qualquer tipo de atividade. Embora mais raro, também poderá ser falha no *hardware* e nestes casos a única solução é esperar pela retoma do serviço ou contactar a operadora para resolver um eventual problema de hardware.

A rotina de diagnóstico pode ser ativada pelo utilizador utilizando diversas expressões, das quais alguns exemplos são:

- «I am having Internet problems»;
- «I need help with my computer»;
- «I can't access the internet»;
- «The internet pages won't load»;
- «I can't watch videos on youtube»;
- «I'm having trouble gaming»;
- (...)

O sistema também é capaz de aceitar ordens diretas que realizam funcionalidades simples como:

- Medir a velocidade da Internet;
- Medir a latência;
- Contar o número de dispositivos ligados à rede;
- Reiniciar o *router*.

O utilizador consegue usufruir destas funcionalidades através de comandos de voz que tenham a intenção direta de as realizar.

### 3.3 Implementação

#### 3.3.1 Seleção das tecnologias

De modo a criar condições de comodidade e conforto, um dos objetivos iniciais da arquitetura era possuir o mínimo de equipamentos físicos possível. Do ponto de vista do utilizador é mais apelativo e natural interagir com apenas uma aplicação no telemóvel ou um dispositivo que contenha um assistente virtual. Desta maneira o ideal seria reduzir o número de equipamentos para dois, o dispositivo do assistente virtual e o *router*.

A assistente virtual Snips, referida no capítulo 2, adequava-se perfeitamente a esta arquitetura devido à sua capacidade de funcionar *offline* e conter todas as ferramentas localmente. Inesperadamente o suporte e desenvolvimento deste assistente foi descontinuado em Janeiro de 2020 impedindo então a possibilidade de o integrar na arquitetura. Os restantes assistentes virtuais *open source* necessitavam sempre de uma máquina com sistema operativo linux para que o assistente possa operar. Esta condição iria acrescentar um novo elemento físico à arquitetura, o que seria indesejável. Inicialmente foi testada a instalação do Mycroft no sistema operativo do *router* (OpenWRT) com o auxílio de memória externa. Esta tentativa não foi um sucesso devido à incapacidade em termos de memória RAM (*random access memory*) que o *router* possuía. Esta incapacidade impossibilita o assistente virtual de operar normalmente.

##### 3.3.1.1 Sistema de reconhecimento/síntese de voz

A solução adotada para o assistente virtual teria que realizar todo o processamento remotamente, ou seja, através de uma *cloud*. Nestas condições restavam duas opções para o papel de assistente, a Amazon Alexa ou a Google assistant. Foi escolhida uma framework que permitia o desenvolvimento de aplicações para estas duas plataformas e, posteriormente, o *deployment* das mesmas para uma *cloud*. A Jovo framework oferece as ferramentas necessárias para o desenvolvimento e *deployment* das aplicações em ambas as plataformas. Foi escolhida a Alexa como assistente virtual devido ao fácil acesso a ferramentas da *cloud*.

A plataforma de desenvolvimento da Alexa (*Alexa Skills Kit*) foi substituída pela *framework* Jovo, no entanto será sempre necessário fazer o *deploy* da estrutura linguística para a plataforma, pois é lá que é feito o reconhecimento e a síntese de discurso. Para além disso, a plataforma também é responsável por associar as aplicações desenvolvidas a uma conta Amazon, e assim disponibilizar as aplicações de voz para qualquer dispositivo com sessão iniciada na respetiva conta.

##### 3.3.1.2 Jovo Framework

A Jovo framework permite-nos desenvolver aplicações tanto para a Alexa como para a Google assistant e também, agora com a última versão, para alguns chatbots.

A interação com a framework é por CLI (*command line interface*) onde são geridos os diferentes projetos e é feita a interligação com as plataformas exteriores (neste caso a Amazon). No que

toca ao desenvolvimento das aplicações, é apenas suportada uma linguagem de programação, o nodejs. As aplicações podem ser desenvolvidas em qualquer ambiente de desenvolvimento, neste projeto foi usado o VS code.

A framework possui também uma ferramenta de *debug* para testar os diversos «caminhos» percorríveis numa interação com o assistente virtual, isto é, verificar se os diálogos estão bem encadeados de forma a ter um diálogo com sentido.

A associação da aplicação com a consola online da Alexa é feita através de um *webhook* criado pela própria *framework*.

### 3.3.1.3 OpenWRT

Do ponto de vista do *router* foi escolhido o sistema operativo OpenWRT devido à sua flexibilidade e suporte disponível. Nesta ferramenta foi criada uma aplicação que era responsável por receber uma mensagem do Gestor do assistente (aplicação Jovo) e correr uma rotina capaz de realizar as tarefas pretendidas.

O sistema operativo OpenWRT permite-nos fazer a configuração da rede de uma maneira bastante flexível. Através do desenvolvimento de uma aplicação em Python (agente de configuração de rede) e do aproveitamento das funcionalidades do sistema é possível gerir toda a rede sem grandes complicações.

O OpenWRT é baseado em Linux e as suas funcionalidades vêm sob a forma de «*packages*» instaláveis. Os *packages* são geridos por uma ferramenta, o *package manager* [29]. O broker MQTT implementado é um exemplo de package instalado no sistema.

O sistema tem duas interfaces disponíveis: uma interface web (LuCI) para uma interação mais visualmente apelativa que pode ser acedida através do endereço IP do *router* em qualquer browser, ou então uma interface de linha de comandos para uma interação mais prática num terminal Linux.

A troca de mensagens entre o *router* e o assistente foi realizada através de MQTT, um protocolo leve e eficiente, ideal para ambientes IoT. Neste ambiente em específico o *broker* foi implementado como um processo no *router* e os clientes eram as duas aplicações, o gestor do assistente e o agente de configuração de rede.

Para este projeto foi instalada a versão 18.06.4 do OpenWRT.

### 3.3.2 Implementação do gestor do assistente

A *framework* Jovo oferece aos seus utilizadores uma arquitetura para a aplicação pré configurada que torna a integração com as plataformas dos assistentes virtuais mais simplificada. Neste projeto foi escolhida a plataforma da Amazon e é possível estabelecer uma sincronização quase automática entre a *framework* e a plataforma. A sincronização é estabelecida através da linha de comandos com a introdução sequencial de dois comandos: «Jovo build» e «Jovo deploy». Após a introdução dos comandos referidos a aplicação estará quase pronta para utilizar em qualquer dispositivo com a Alexa integrada. O último passo é introduzir o comando «Jovo run» para que a

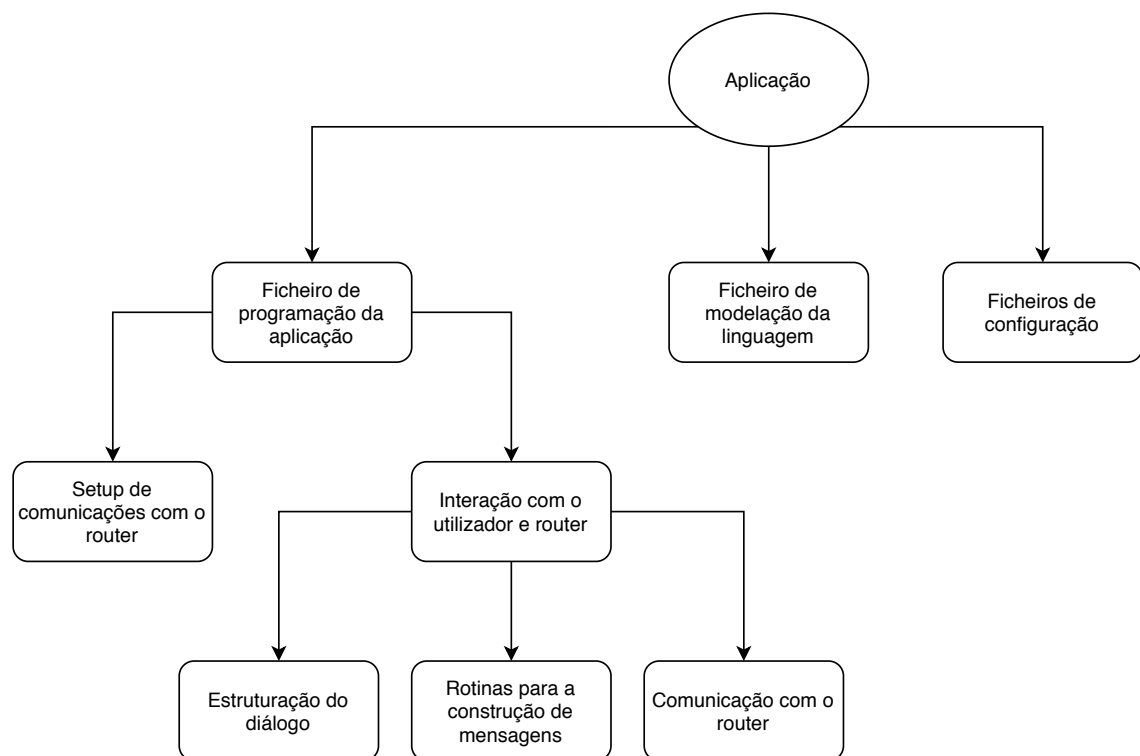


Figura 3.3: Arquitetura do gestor do assistente virtual

aplicação desenvolvida entre em funcionamento e a partir desse momento é possível interagir com o assistente.

Na figura 3.3 é demonstrada gráficamente a arquitetura dos ficheiros presentes no gestor do assistente que são guardados localmente:

O gestor divide-se em três grandes categorias de ficheiros, o próprio ficheiro onde é estruturado o código para a aplicação, um ficheiro onde são mapeados os *inputs* do utilizador e associados a uma intenção pré definida e, por fim, vários ficheiros de configuração responsáveis por:

- Sincronizar a aplicação com a plataforma;
- Definir a linguagem do sistema;
- Guardar os dados do utilizador;
- Gerir a versão do software;
- Guardar as dependências necessárias para a compilação da aplicação.

### 3.3.2.1 Configuração dos *inputs* do utilizador

Os ficheiros de configuração são criados pela *framework* automaticamente e não necessitam de ser alterados. Os que realmente são alterados e que manipulam diretamente a interação do utilizador com o sistema são os restantes dois anteriormente referidos: o ficheiro responsável pela



programação da aplicação e o ficheiro que mapeia os *inputs* do utilizador. Os *inputs* do utilizador estão então organizados numa estrutura de dados dentro de um ficheiro JSON (linguagem usada para estruturação de dados). Dentro do ficheiro estão presentes todos os possíveis comandos de voz que o utilizador pode pronunciar durante a interação com o sistema. Estas frases são organizadas em diversas categorias, chamadas de intenções. As intenções são a ideia ou vontade que o utilizador quer transmitir quando interage com o assistente. No fundo, aquilo que existe neste componente do gestor é uma associação dos possíveis diálogos que o utilizador poderá ter com o assistente e as suas intenções, por exemplo: as frases «*Run a speed test*» e «*How fast is my Internet?*» pertencem à mesma intenção, que seria «*Speed test*».

Outro aspeto importante deste componente é a possibilidade de criar «variáveis» naquilo que são os diálogos do utilizador. Mais concretamente, estas «variáveis» têm o nome de *slots* e o seu objetivo é agregar certas palavras ou expressões numa categoria e facilitar a organização do diálogo. Um exemplo de utilização deste mecanismo pode ser visto quando o sistema está à espera que o utilizador transmita uma determinada hora do dia. Neste caso é criada uma lista com todas as horas do dia e são associadas ao *slot* «*hours*». Deste ponto em diante sempre que for preciso obter um *input* que indique as horas do dia podemos utilizar o *slot* criado ao invés de criar 24 instâncias, em que cada uma corresponde a uma hora diferente. Um exemplo de frase usada neste caso seria: «*Turn the lights on at hours*».

### 3.3.2.2 Lógica do programa

O último, e maior componente do gestor, é o ficheiro com toda a lógica e que «guia» o diálogo entre o assistente e o utilizador e também comunica com o *router*. Este componente tem duas secções principais. Na primeira onde existe um *setup* das comunicações: é definido o endereço do *broker* MQTT, são subscritos os tópicos para os quais o *router* vai publicar mensagens e também são definidas algumas mensagens de erro caso a conexão inicial com o *broker* falhe.

Na segunda secção do ficheiro é organizada toda a lógica do programa. Inicialmente o diálogo é separado em diferentes estados em que cada um define as intenções esperadas pelo sistema vindas do utilizador, por exemplo: no momento em que o gestor é invocado pelo utilizador, é ativado o estado «*ProblemFinderState*», neste estado o sistema está à espera que o utilizador indique a razão pela qual interagiu com o assistente. Cada estado é composto por várias intenções que se relacionam com as intenções descritas no ficheiro JSON com os diálogos do utilizador estruturados. As intenções na lógica do programa assemelham-se a *if statements* no sentido em que a condição para entrar no *statement* é o utilizador pronunciar uma frase, palavra ou expressão que pertença ao grupo da intenção em causa. Colocando tudo nesta lógica, o programa transforma-se numa cadeia de *if statements* separada por diferentes estados que representam as diferentes etapas de interação entre o utilizador e o assistente. Logicamente, quando o programa interpreta uma dada intenção vinda do utilizador, existe sempre alguma rotina que vai cumprir a ação desejada pelo utilizador. Na aplicação desenvolvida estas ações podem ser:

- Enviar mensagens para o *router*;

- Processar as mensagens enviadas pelo *router*;
- Construir mensagens para o utilizador a partir de respostas do *router*;
- Enviar mensagens para o utilizador;
- Gerir o tempo que o *router* demora a responder quando lhe é enviada uma mensagem.

Esta última ação é importante devido a uma particularidade da plataforma da Alexa. A plataforma não permite que após uma intervenção do utilizador o sistema demore mais que sete segundos a responder. Este detalhe levou a uma pequena alteração em alguns aspetos do gestor, por exemplo no caso em que o sistema precisa de fazer um teste à velocidade da Internet. Esta operação tem uma duração perto de trinta segundos, ou seja, não é possível implementar da maneira mais direta, que seria fazer o utilizador esperar o tempo necessário até ao fim do teste. A solução que foi pensada e implementada baseia-se em continuar o diálogo com o utilizador até que o teste esteja quase pronto. Para este efeito, realizou-se este teste diversas vezes e verificou-se que a duração varia entre 25 a 33 segundos e o diálogo é mantido até cerca de 24 segundos. Findo o diálogo, um mecanismo conta o tempo para que caso o resultado do teste não chegue dentro do limite esperado, levar o assistente a entrar de novo em contacto com o utilizador para se desculpar da demora e perguntar se o utilizador não se importa de aguardar mais uns segundos. Este mecanismo foi testado inúmeras vezes e provou ter uma eficácia de 100% em todas as situações.

### 3.3.3 Mecanismos implementados no OpenWRT

No sistema operativo do *router* foram implementados os mecanismos que permitem ao utilizador fazer a gestão da rede. Através de uma aplicação em Python foram estabelecidas as comunicações com o gestor do assistente e implementadas as rotinas de configuração/diagnóstico da rede.

Na figura 3.4 estão listados os 2 componentes implementados no OpenWRT. O broker MQTT, que é o elemento central das comunicações, e o script Python, que é o agente de configuração de rede e que é responsável por todas as ações realizadas no *router*.

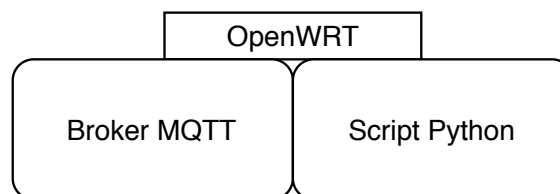


Figura 3.4: Componentes presentes no OpenWRT

#### 3.3.3.1 Criação da rede Guest

A criação da rede Guest é um mecanismo implementado no *router* que ativa e desativa a disponibilidade de acesso de uma rede com acesso exclusivamente à Internet e sem palavra passe.

Esta rede foi criada através de comandos UCI (*unified configuration interface*). Esta ferramenta permite ao utilizador editar configurações do *router*.

A rede Guest é um ponto de acesso com as mesmas características que as redes *WI-FI* comuns em ambientes domésticos mas com a particularidade de ter uma zona na *firewall* que faz *forwarding* do tráfego dos seus utilizadores diretamente para a WAN (Wide Area Network). Isto implica que os dispositivos conectados a esta rede tenham acesso apenas à Internet e não à rede interna. Para ativar e desativar a rede existem dois passos a cumprir:

- Colocar a variável booleana que indica a visibilidade da rede no ficheiro de configuração a «*True*»;
- Reiniciar o serviço WLAN.

### 3.3.3.2 Comunicação

O elemento central da arquitetura da comunicação do sistema, o broker MQTT, está implementado no *router*. Este elemento é responsável pela receção e reencaminhamento das mensagens enviadas pelos restantes 2 elementos que comunicam, o programa do assistente e o programa de configuração do *router*.

As mensagens MQTT são enviadas para o broker com um determinado tópico. Neste sistema o *router* subscreve a todos os tópicos em que o assistente publica mensagens e vice-versa. O processo de subscrever aos tópicos faz parte do *setup* da comunicação em ambos os programas. Existe uma variável que controla a qualidade do serviço MQTT, QoS (Quality of Service). Esta variável está com o valor «1» para garantir que a mensagem chega ao destino pelo menos uma vez.

Com o *setup* das comunicações feito, os 2 elementos podem trocar livremente mensagens dentro dos tópicos que subscreveram. As mensagens enviadas pelo assistente informam o *router* da ação que ele terá que realizar, enquanto que as mensagens do *router* confirmam a realização da ação ou devolvem os resultados obtidos.

Os tópicos associados às mensagens indicam o enquadramento das mesmas, por exemplo: Uma mensagem com o conteúdo «ligar» só faz sentido dentro de um determinado tópico. Neste caso o tópico poderia ser «Rede Guest».

### 3.3.3.3 Agente de configuração de rede

Na fig. 3.5 está descrita aplicação Python implementada no *router* que é responsável por fazer a associação das mensagens recebidas pelo assistente com a ação correspondente. Para além do *setup* das comunicações, o esquema do programa consiste numa cadeia de *if statements*, que é ativada quando o *router* recebe uma mensagem num tópico subscrito, que verificam o conteúdo da mensagem e comparam com as possíveis alternativas. Este processo vai dar origem a uma ação por parte do *router* quando for encontrada a mensagem correta.

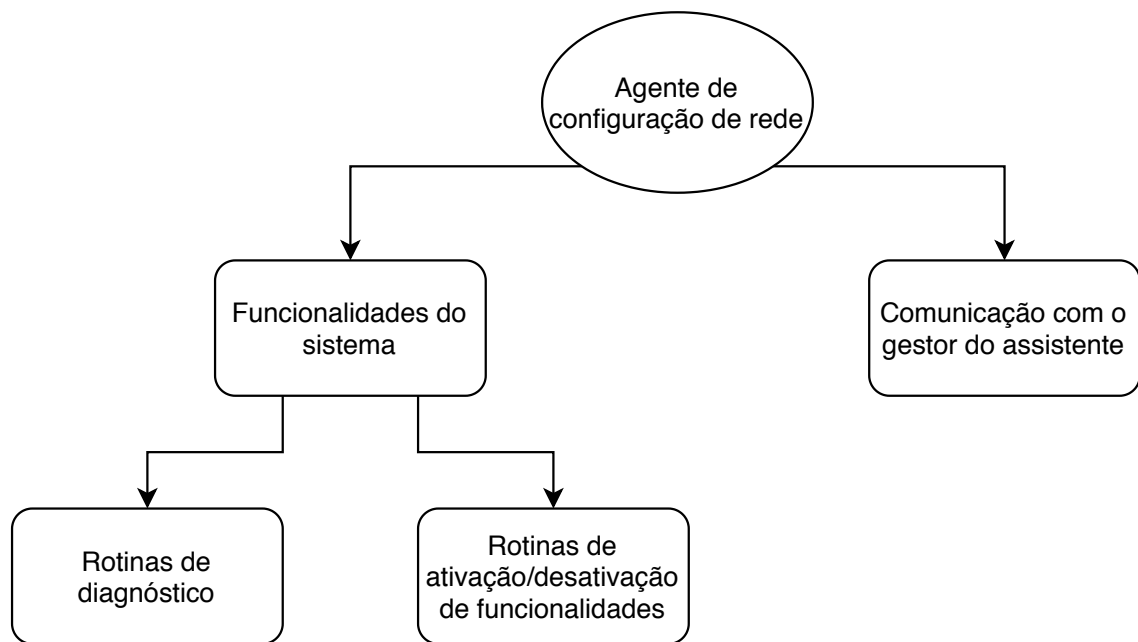


Figura 3.5: Arquitetura do agente de configuração de rede

Para além do mecanismo de verificação de mensagens, o programa é constituído por um conjunto de funções que realiza as operações de diagnóstico e ativa ou desativa funcionalidades no *router*. As operações são as seguintes:

- Ativação ou desativação da rede Guest;
- Medição da velocidade da Internet e da latência;
- Estruturação dos dados para enviar para o assistente;
- Contagem dos dispositivos conectados à rede;
- Medição da força do sinal dos dispositivos conectados à rede.

O teste de velocidade e de latência é realizado através de um módulo de Python que permite a realização do teste através da linha de comandos [30]. Existe também um processo de extração de dados vindos do terminal e organização de valores medidos em variáveis.

Após a estruturação dos dados, é necessário traduzir os valores numéricos do teste de velocidade em valores por extenso para que seja possível fazer a síntese do discurso. De momento, a plataforma da Alexa ainda não permite *inputs* de texto que contenham números na sua forma numérica. Para tal efeito foi usada um programa que transformava o número na sua forma por extenso [31]. Após todo este processamento, os dados são enviados para o gestor do assistente prontos para serem integrados numa mensagem para o utilizador.

A contagem dos dispositivos e a medição da força do sinal usam a mesma ferramenta. O programa usa as ferramentas do OpenWRT para listar todos dispositivos conectados à rede, juntamente com informação relativa aos endereços IP (Internet Protocol) e MAC (Media Access Control), potência do sinal, etc. A contagem é feita através do ciclo que vai listando os dispositivos, ou seja, por cada dispositivo listado existe uma variável que é incrementada. A força do sinal de cada dispositivo é retirada de maneira análoga aos valores do teste de velocidade, ou seja, através de um mecanismo de extração de informação do terminal.

#### 3.3.3.4 Modularidade do sistema

As tecnologias e softwares escolhidos para o sistema fornecem alguma modularidade no que diz respeito à adição de funcionalidades e componentes. As aplicações desenvolvidas para a Alexa podem ser adicionadas de maneira transparente, ou seja, sem causar alterações no serviço atual e adicionando novas funcionalidades ao sistema.

A utilização do MQTT permite o desenvolvimento e integração de novos componentes no sistema, também de maneira transparente. A modularidade do sistema permite, no futuro, a inclusão de:

- Novas funcionalidades a oferecer aos utilizadores;
- Mecanismos de análise e aprendizagem para gestão automatizada da rede e para os operadores adquirirem informação sobre os tipos de problemas enfrentados pelos utilizadores;
- Integração com sistemas de domótica, nos quais é comum a utilização deste protocolo.

### 3.3.4 Implementação dos casos de uso

#### 3.3.4.1 Criação da rede *Guest*

Na fig. 3.6 podemos observar um diagrama que descreve os passos que o sistema executa para a ativação da rede *Guest*. É sempre preciso o *input* inicial do utilizador que transmite a intenção de ligar a rede, seguido de um processamento por parte do gestor do assistente que vai notificar o *router* com uma mensagem MQTT adequada para a ação esperada. No *router* é ativada uma rotina que liga a rede *Guest* e reinicia o serviço WLAN do *router* para que a rede fique disponível. Após este evento o assistente confirma ao utilizador que a rede irá estar disponível dentro de alguns segundos ou dentro de algumas horas, dependendo de haver agendamento ou não.

Para complementar esta funcionalidade foi criado o mecanismo de desativar a rede. Os únicos aspetos alterados no diagrama em 3.6 são: o *input* do utilizador e a mensagem MQTT enviada pelo gestor.

A rede em si foi configurada através do terminal com comandos UCI (*unified configuration interface*) que permitem aceder a todas as configurações de todas as redes, também alterar protocolos e configurar a *firewall*. É apenas permitido aos utilizadores desta rede o acesso à Internet, o que reforça a segurança do sistema dado que é uma rede sem password e qualquer pessoa pode

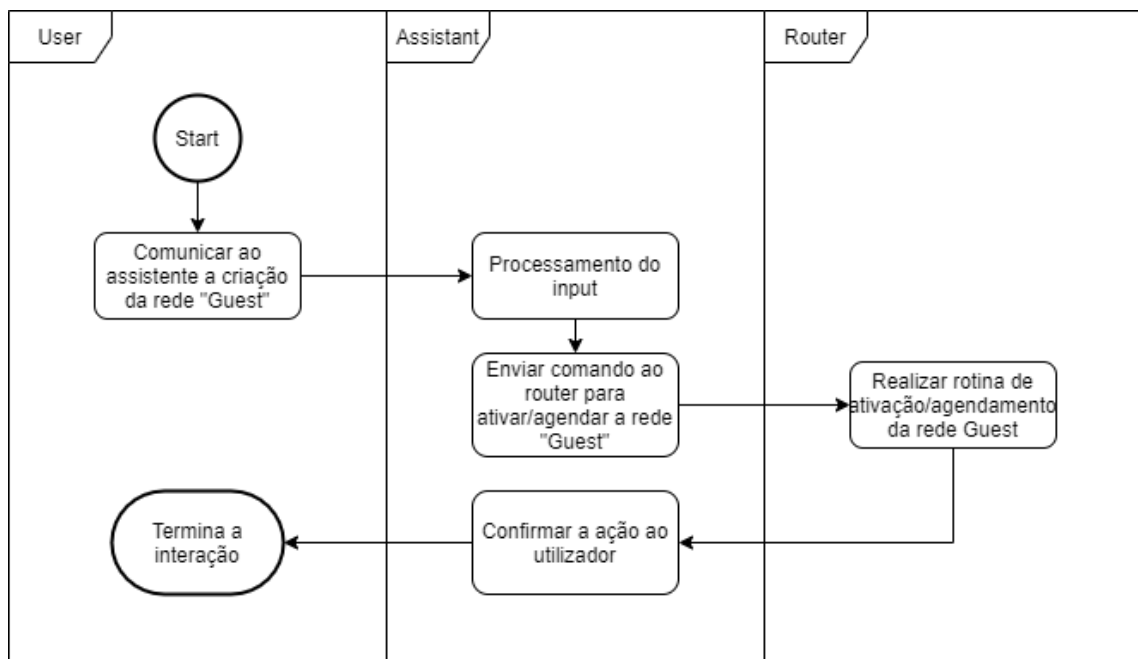


Figura 3.6: Processo de ativação de rede Guest

aceder. Este detalhe foi implementado através de uma regra de *forwarding* na *firewall*; o utilizador que é conectado a esta rede fica automaticamente com acesso apenas à WAN.

### 3.3.4.2 Diagnóstico da rede

O presente caso de uso descreve as situações mais recorrentes no âmbito de problemas de redes domésticas. Os utilizadores tendem a ter problemas associados ao uso da Internet e o assistente entra nestas situações para prestar apoio e capacidade de resolução de problemas. Este caso de uso destina-se aos utilizadores com conhecimento limitado nas tecnologias envolvidas e na maneira com funcionam.

Na fig. 3.7 está representado de uma forma genérica o processo para a abordagem de problemas na rede. O utilizador vai detetar o problema ocorrido e comunicá-lo ao assistente. A partir desse momento, o assistente vai tentar identificar o problema através duma rotina de diagnóstico, representada na Fig. 3.8.

A rotina começa por listar os dispositivos conectados ao *router* usando um *shell script* que faz uso das funcionalidades do OpenWRT para esta finalidade. A listagem dos dispositivos contém informação que é útil para a realização do diagnóstico, como a força do sinal de cada dispositivo. O agente de configuração de rede é responsável por contar o número de dispositivos listados e associá-los à força do sinal *WI-FI* que chega a cada um.

O último procedimento da rotina de diagnóstico é realizar um teste de velocidade e um teste de ping à Internet, o qual só se vai realizar se o teste de potência do sinal for positivo. O teste de ping

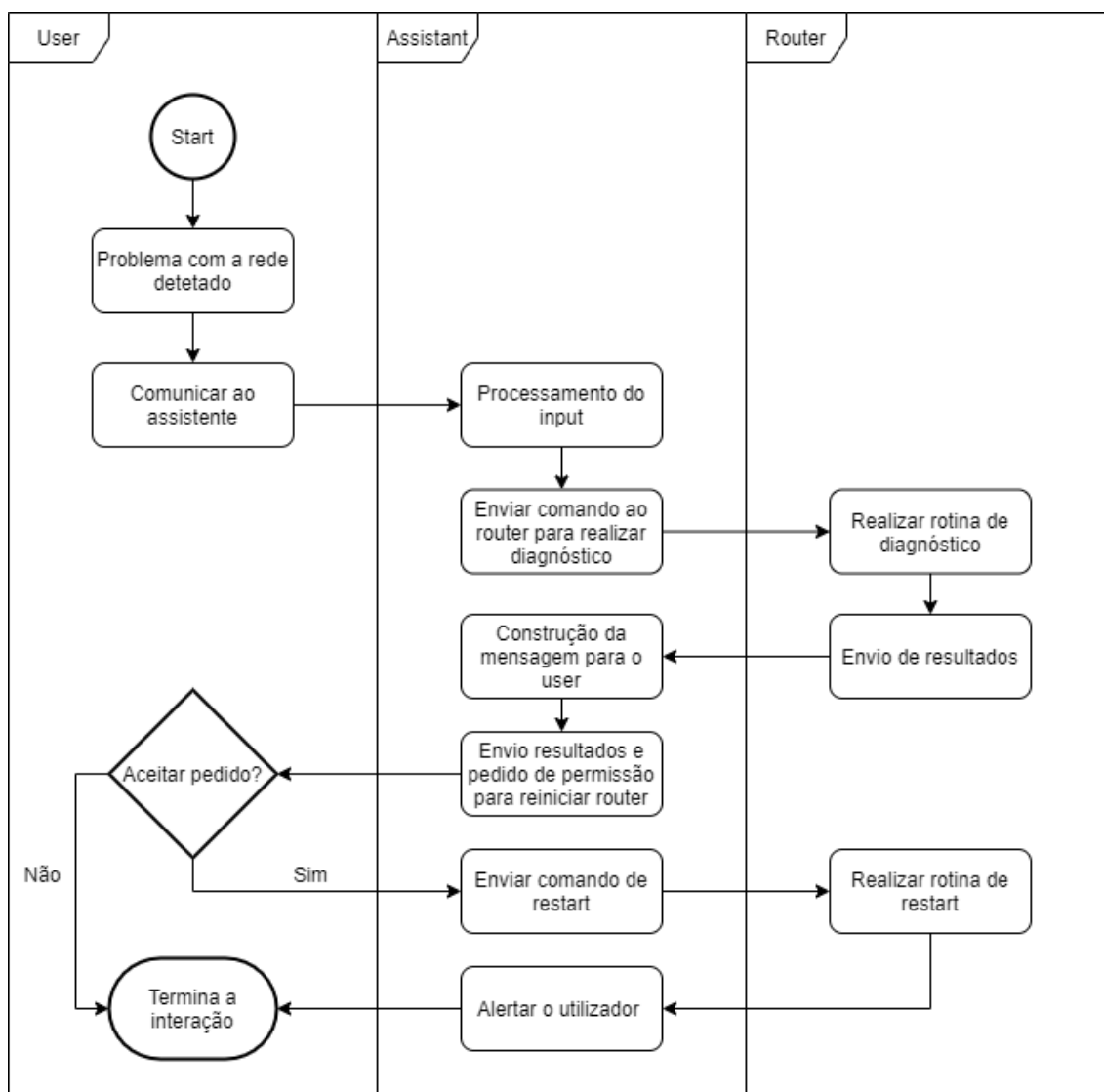


Figura 3.7: Problemas de utilização da Internet

permite verificar o tempo de resposta de um determinado servidor, neste caso o teste é realizado no servidor do ISP (Internet Service Provider) do utilizador.

Na rotina de diagnóstico, o teste de velocidade da Internet é feito no sistema operativo do *router* e caso haja algum problema associado a largura de banda nos dispositivos conectados à rede é possível distinguir entre problemas externos e internos. O teste de velocidade vai medir os valores no *router* e, perante o resultado, identificar a origem do problema reportado pelo utilizador, ou seja, caso os valores estejam normais, quer dizer que a largura de banda providenciada pela rede externa está a funcionar corretamente e que a origem do problema é interna. A questão é descobrir a razão pela qual há falta de largura de banda, que será ou falta de sinal ou limitação por parte de outros dispositivos. O algoritmo implementado verifica primeiramente a potência do sinal e apenas no caso de serem medidos valores aceitáveis é que o sistema tenta despistar o problema

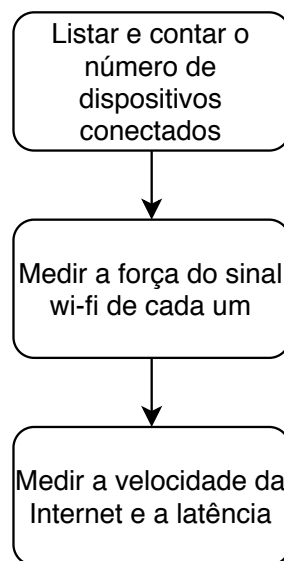


Figura 3.8: Rotina de diagnóstico de rede

fazendo o teste de velocidade à Internet, que permite perceber se o problema é externo ou interno. Caso seja interno, o sistema assume que existem dispositivos a consumir largura de banda em excesso.

O sistema consegue também detetar anomalias externas à rede doméstica através do teste de velocidade. Devido ao teste ser feito no sistema operativo do *router*, os valores baixos no teste de velocidade significam que a largura de banda que chega à rede doméstica do utilizador vêm do exterior, ou seja, não há soluções que possam envolver o utilizador para uma resolução imediata do problema. Nestes casos o utilizador pode esperar para que o serviço esteja em alta outra vez ou contactar a operadora para reportar o problema, tendo a certeza de que o problema é externo.

A ferramenta usada para a realização do teste de velocidade e *ping* é um módulo Python («*speedtest-cli*») que permite fazer tanto o teste de velocidade como o teste de *ping* através do terminal. Mais uma vez, o agente de configuração de rede é responsável por recolher os dados e, por fim, organizá-los de forma estruturada para enviar uma resposta ao gestor do assistente virtual.

### 3.3.4.3 Mecanismos de resposta direta

Os mecanismos de resposta direta são rotinas usadas quando o utilizador dá uma ordem direta ao assistente virtual. No final do capítulo anterior foram listadas as rotinas implementadas que respondem a ordens diretas. A interação do utilizador com o assistente, ao nível conceptual, não varia muito de rotina para rotina; mudando apenas aquilo que é configurado ou testado ao nível da rede e também a própria comunicação com o utilizador.

Os diagramas que se seguem demonstram graficamente como foram implementadas estas rotinas.



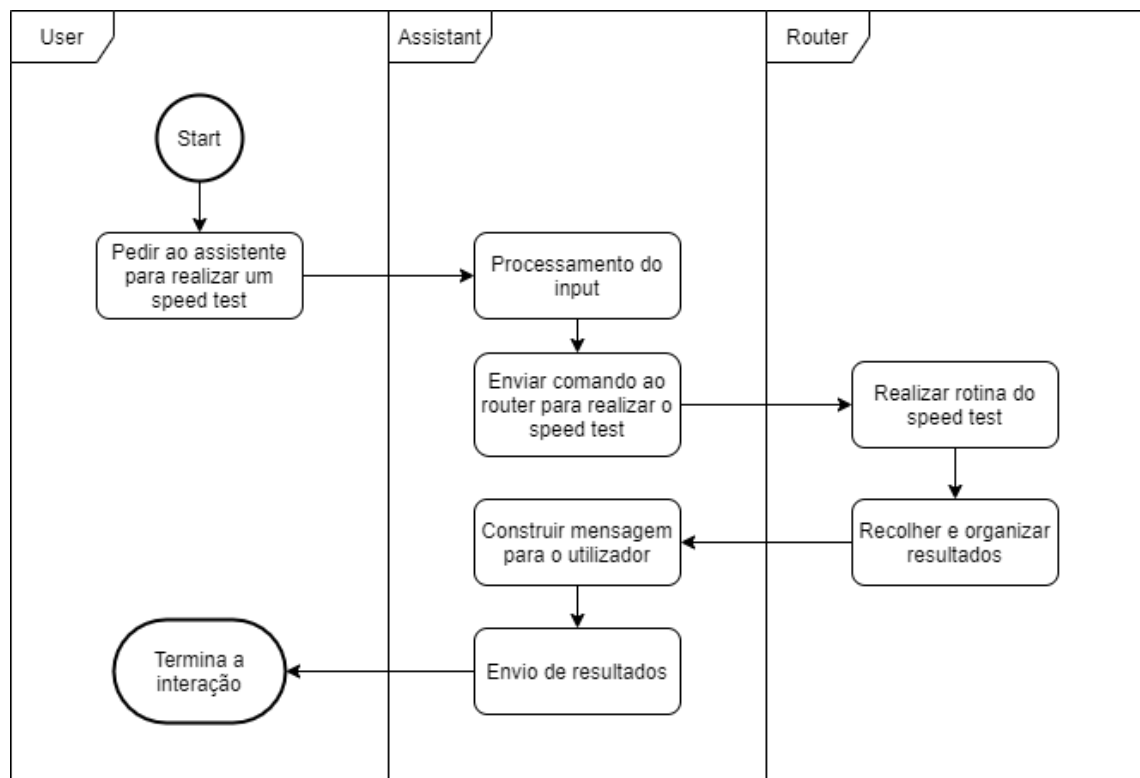


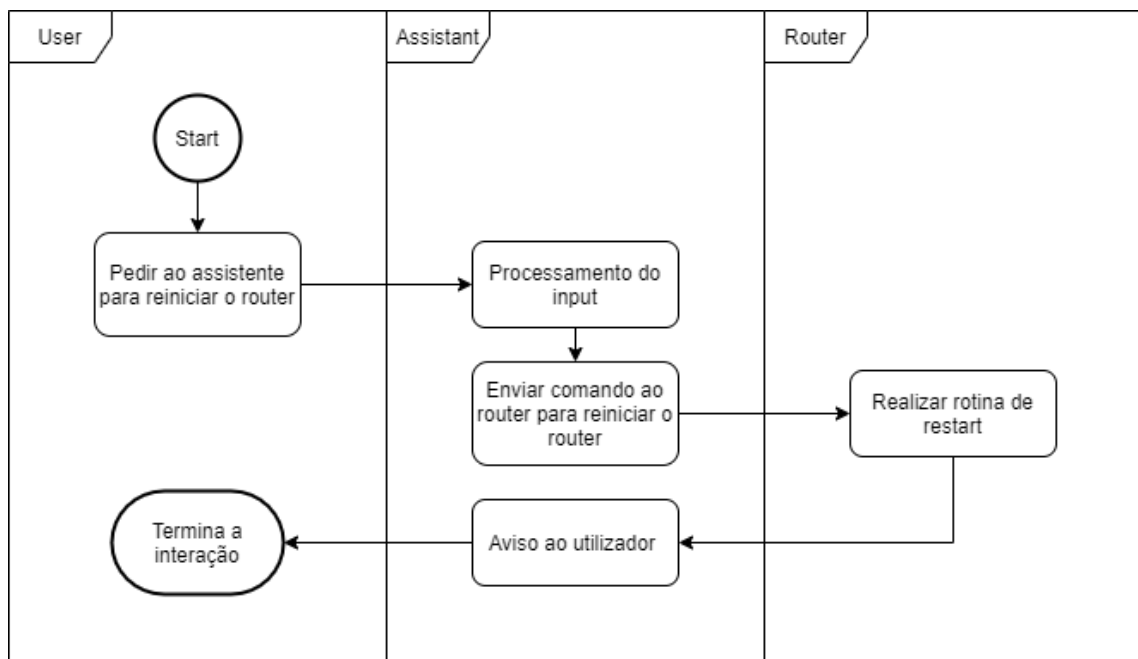
Figura 3.9: Rotina de teste de velocidade e latência

#### 3.3.4.4 Teste de velocidade e latência

Na figura 3.9 é demonstrada a rotina que é ativada pela ordem do utilizador que quer realizar um teste de velocidade e latência à sua rede. O utilizador começa naturalmente por informar ao assistente que pretende realizar o teste de velocidade, seguidamente o gestor do assistente vai perceber a intenção do utilizador e enviar ao *router* o comando para ativar a rotina do teste de velocidade e devolver os resultados, finalmente, os resultados são devolvidos e traduzidos numa frase com discurso de linguagem natural e transmitidos ao utilizador. Os resultados transmitidos ao utilizador estarão na forma numérica, ou seja, quantitativa e não qualitativa, porque, à partida, é uma operação invocada por utilizadores com maior conhecimento tecnológico. É importante considerar a coexistência de diferentes utilizadores (com maior ou menor conhecimento), e o assistente é capaz de adaptar a resposta a ambos.

Algumas expressões que despoletam esta rotina são:

- «Run a speed test»;
- «Check my internet speed»;
- «How fast is my internet?»;
- «Measure my internet speed»;

Figura 3.10: Rotina para reiniciar o *router*

- «What is my latency?»
- (...)

A resposta final do sistema a este pedido será então:

- «The results of the speed test are the following: your download speed is [x] megabits per second, your upload speed is [y] megabits per second, and your latency is [z] milliseconds.»

### 3.3.4.5 Reiniciar o *router*

Na figura 3.10 está exposta a rotina que faz reiniciar o *router* através do comando do utilizador. O sistema começa por receber um *input* do utilizador com a intenção de reiniciar o aparelho. O gestor do assistente vai receber o comando e enviar uma mensagem de *restart* ao agente de configuração de rede para que seja accionado o mecanismo. Após o envio da mensagem, o assistente dá um aviso ao utilizador que o informa que o aparelho vai ser reiniciado dentro de poucos segundos.

O mecanismo de *restart* do *router* é a execução de um simples comando («reboot»).

Alguns exemplos de ativação deste mecanismo são:

- «Reset the *router*»;
- «Reboot the *router*»;
- «Turn the *router* off and on again»;
- «Restart the *router*».

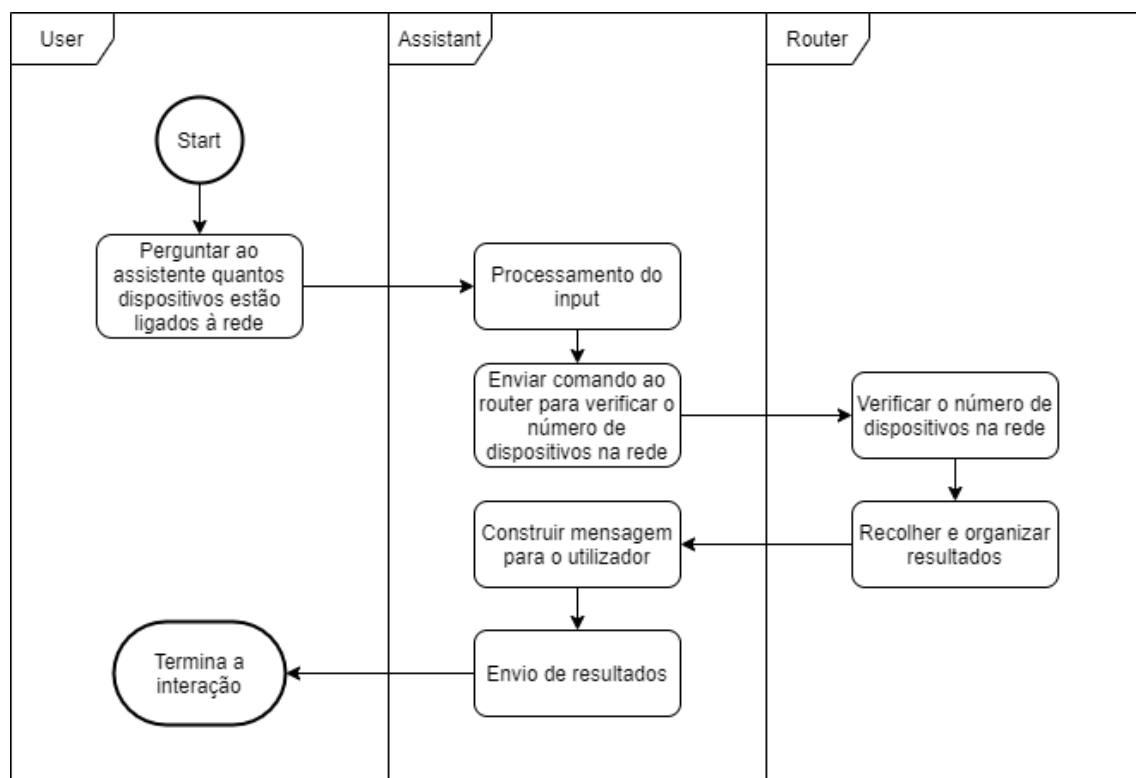


Figura 3.11: Rotina para contar o número de dispositivos na rede

- (...)

A resposta final do sistema ao pedido do utilizador será:

- «Your *router* will reboot in a few seconds.»

#### 3.3.4.6 Contagem do número de dispositivos na rede

Na figura 3.11 é demonstrada a rotina que permite ao utilizador ter conhecimento do número de dispositivos conectados à sua rede. A interação começa naturalmente pelo *input* do utilizador, seguido de uma interpretação por parte do gestor do assistente que vai notificar o agente de configuração da intenção do utilizador. O *router*, por sua vez, vai contar o número de dispositivos conectados à rede usando o *shell script* que foi mencionado no caso de uso da rotina de diagnóstico. Após a contagem o aparelho vai enviar a quantidade de dispositivos ao gestor que, por sua vez, constrói uma mensagem ao utilizador e transmite o resultado.

Alguns exemplos de expressões para ativar este mecanismo são:

- «How many devices are connected?»
- «How many devices are using Internet?»
- «Count the devices connected to the network»

- (...)

A resposta do sistema face ao pedido do utilizador é a seguinte:

- «You have [x] devices connected to your network.»

### 3.4 Considerações finais

Este capítulo procurou descrever o modo de funcionamento das tecnologias implementadas no sistema, bem como a implementação dos diferentes casos de uso. Colocamos especial foco para a criação da rede *guest* e para a identificação de problemas na rede que são os casos de uso mais complexos em termos de implementação e vão ser avaliados no capítulo seguinte, o capítulo de avaliação de experiência do utilizador.

A grande vantagem no uso da voz é que oferece ao utilizador comum, ou seja, um utilizador com conhecimento técnico limitado na área das telecomunicações e redes, uma interface que permite diagnosticar e eventualmente resolver certos tipos de problemas com a linguagem natural e sem o tal conhecimento extra nesta área.

Existe no nosso sistema uma funcionalidade capaz de criar um rede independente que permite o acesso à Internet e não necessita de credenciais de acesso. Existe também uma rotina capaz de diagnosticar e identificar problemas na rede e, por fim, estão implementadas rotinas que respondem a pedidos diretos do utilizador.

## Capítulo 4

# Avaliação da experiência do utilizador

O presente capítulo incide sobre a exposição dos métodos de teste escolhidos para avaliar o sistema em termos de performance e de grau de satisfação do utilizador. Para várias interações entre o sistema e utilizadores de teste, foram medidas métricas objetivas relativas à experiência do utilizador, e quantificada a percepção do utilizador sobre a experiência com o sistema por meio de um questionário. No final do capítulo será feita uma reflexão crítica sobre os resultados obtidos.

### 4.1 Métricas

De modo a testar o desempenho do sistema foram estabelecidas algumas métricas objetivas que pretendem avaliar o adequado comportamento do sistema.

No ambiente criado e no contexto das tecnologias em questão (redes domésticas e assistentes virtuais) foi pensado como seria a melhor maneira de avaliar o sistema. Dado que o sistema tem como objetivo prestar apoio e suporte técnico ao utilizador e, no fundo, substituir aquilo que é a linha de apoio ao cliente por parte das operadoras, as métricas usadas para fazer a avaliação de desempenho basearam-se em:

- Medir a percentagem de sucesso nos testes feitos com utilizadores;
- Medir o tempo médio que demora a interação com o utilizador e comparar com uma possível chamada para uma linha de apoio.

Estas duas métricas permitirão fazer uma comparação em termos de desempenho com o sistema atual da linha de apoio e tirar conclusões acerca da utilidade do sistema.

A linha de apoio é constituída por diferentes linhas em que se encontram:

- A 1ª linha de apoio: que serve para fazer uma triagem de problemas e para tentar resolver os problemas mais comuns;
- Linha de negócio/faturação: que serve para resolver questões de subscrição de serviços, faturação e aspetos contratuais;

- Linha de retenção: Usada quando os clientes querem terminar o contrato ou mudar de operadora, esta linha tem autorização para fornecer descontos e benefícios de forma a reter os clientes;
- Linha técnica: Dotada de conhecimento mais especializado de forma a tratar dos problemas que a 1ª linha não conseguiu resolver.

O assistente tentar emular aquilo que é a 1ª linha de apoio, assistindo o utilizador nos problemas mais comuns e fáceis de resolver.

Os utilizadores que participaram no teste ao sistema também tiveram a oportunidade de o avaliar através do preenchimento de um inquérito com seis questões relativas ao desempenho do sistema em vários aspetos. Os utilizadores responderam às perguntas numa escala de Likert [32] que é habitualmente usada para quantificar uma apreciação subjectiva, sendo habitualmente usada em p.ex. questionários de opinião. A escala consiste na atribuição de diferentes níveis de resposta (p.ex. entre 1 e 5 ou entre 1 e 7) que o utilizador pode escolher, expressando a sua opinião perante a questão colocada. Foram utilizados cinco diferentes níveis para as repostas dos utilizadores.

No caso do questionário em causa, o valor inferior corresponde à experiência menos positiva, e o valor mais elevado à mais positiva. As perguntas utilizadas no questionário foram as seguintes:

1. Como classifica a clareza e objetividade das respostas apresentadas pelo sistema? (1 - Nada claro, 3 - Suficiente, 5 - Bastante claro);
2. Como classifica o grau de terminologia técnica no diálogo? (1 - Alto, 3 - Intermédio, 5 - Baixo);
3. Como classifica a adequação das soluções propostas face ao seu problema/necessidade? (1 - Pouco adequadas, 3 - Ajudou alguma coisa, 5 - Perfeitas para a situação em causa);
4. Como classifica a rapidez com que o sistema identificou o problema/necessidade? (1 - Muito demorado, 3 - Relativamente rápido, 5 - Bastante rápido);
5. De um modo geral, como classifica a sua experiência com o assistente virtual? (1 - Má, 3 - Mediana, 5 Excelente);
6. Considera que no futuro os assistentes virtuais poderão ajudar na gestão de redes domésticas? (1 - Não, 3 - Talvez, 5 - Com certeza).

A primeira e a segunda pergunta visam avaliar aquilo que é a interação do utilizador com o sistema ao nível do diálogo. Através da análise às respostas a estas duas questões é possível tirar conclusões acerca da adequação do diálogo na perspetiva do utilizador.

A terceira e a quarta pergunta incidem sobre o desempenho do sistema. O utilizador dá a sua opinião sobre a resposta do sistema ao problema em causa permitindo avaliar o nível de satisfação do utilizador.

A quinta e a sexta questão pedem a opinião do utilizador acerca do sistema e das tecnologias usadas de um modo geral o que permite tirar conclusões acerca da pertinência na aposta destas tecnologias para o futuro das redes domésticas.

## 4.2 Metodologia/Protocolo dos testes

A elaboração dos testes teve em conta as funcionalidades implementadas do assistente virtual e foram então preparados três diferentes testes para cada utilizador. Em cada instância de teste os utilizadores eram colocados nas condições ideais para a realização dos mesmos e foi apenas transmitida a informação necessária no que diz respeito à interação com o assistente de modo a criar o ambiente desejável para a avaliação do sistema.

### 4.2.1 Teste T1: Criação da rede Guest

No primeiro teste foi avaliada a funcionalidade da criação da rede para convidados. Neste teste não há necessidade de criar condições especiais na rede.

O teste consiste em colocar o utilizador na situação hipotética de que precisa de partilhar a sua Internet com os seus convidados. Terá então que interagir com o assistente virtual para surtir esse efeito. A sequência de passos para a realização do teste é a seguinte:

1. O utilizador é colocado numa situação hipotética em que terá que partilhar o acesso à Internet da sua rede doméstica com os convidados;
2. O utilizador interage com assistente virtual, passando a intenção de que quer partilhar o acesso à rede;
3. O *router* ativa o acesso à rede *Guest*;
4. É feita uma verificação de sucesso através do acesso à rede pelo dispositivo móvel do utilizador;
5. O utilizador, novamente, interage com o assistente, desta vez para desativar a rede;
6. O *router* desativa a rede;
7. É feita uma nova verificação no dispositivo móvel do utilizador com o intuito de observar a rede *Guest* a desaparecer das redes disponíveis.

### 4.2.2 Teste T2: Sinal do dispositivo fraco

O objetivo do segundo teste é avaliar a resposta do sistema numa situação onde o utilizador está a experienciar falhas no acesso à Internet devido à falta de potência do sinal no dispositivo móvel. O ambiente de teste requer que o utilizador seja colocado a tal distância do *router* que a potência que chega ao seu dispositivo móvel seja inferior a -80 dBm. Este valor foi retirado de [33] e também comprovado experimentalmente antes da realização dos testes.

O teste tem a seguinte sequência de etapas:

1. O utilizador é colocado a uma distância tal que a potência do sinal seja próxima de -80 dBm;
2. O utilizador interage com o assistente virtual, queixando-se da qualidade do serviço;

3. O sistema realiza a rotina de diagnóstico (Secção 3.3.4.2) e identifica a causa do problema;
4. O assistente comunica o problema encontrado e sugere uma solução;
5. O utilizador aplica a solução sugerida;
6. É verificado se a qualidade do serviço melhorou.

Apesar da conexão ter um sinal fraco, o utilizador consegue na mesma interagir com o assistente virtual por *WI-FI* devido ao fraco uso de largura de banda por parte do assistente.

A solução apresentada pelo sistema neste teste é simplesmente um pedido ao utilizador para aproximar o seu dispositivo do *router* de maneira a receber mais potência no sinal e, consequentemente, melhorar a qualidade do serviço.

#### 4.2.3 Teste T3: Limitação de largura de banda

O terceiro e último teste pretende avaliar a resposta do sistema perante um problema que gera uma limitação na largura de banda da rede. Para criar este efeito foi usada uma *package* do OpenWRT (*nft-qos*) que permite limitar manualmente a largura de banda disponível [34] para os dispositivos conectados. O procedimento para a realização do terceiro teste não difere do procedimento para o segundo teste, sendo as principais diferenças as condições a criar que, neste caso, é a limitação de largura de banda em vez da potência de sinal e também a solução proposta pelo assistente. O procedimento é então o seguinte (idêntico ao segundo teste):

1. A largura de banda da rede é limitada por software;
2. O utilizador interage com o assistente virtual, queixando-se da qualidade do serviço;
3. O sistema realiza a rotina de diagnóstico e identifica a causa do problema;
4. O assistente comunica o problema encontrado e sugere uma solução;
5. O utilizador aplica a solução sugerida;
6. É verificado se a qualidade do serviço melhorou.

A solução sugerida pelo assistente vai ter como base a dedução de que existem outros dispositivos na rede e que não está a ser oferecida a qualidade de serviço adequada ao dispositivo do utilizador a realizar o teste. O sistema vai então sugerir que o utilizador verifique os dispositivos que estão ligados à rede e desconectar, se possível, aqueles que estejam a ter algum tipo de atividade.

Na fase de verificação são retiradas as restrições de largura de banda impostas nos dispositivos de modo a simular a ação do utilizador.



### 4.3 Análise de resultados

Após a realização de todas as sessões de teste, foi feita uma análise aos resultados obtidos. Foi possível ter uma amostra de oito utilizadores para testarem o sistema.

O perfil dos utilizadores é o seguinte:

- Adulto de 53 anos, licenciado em contabilidade;
- Adulta de 50 anos, licenciado em engenharia química;
- Adulto de 45 anos, licenciado em engenharia eletrotécnica;
- Jovem adulto de 19 anos, estudante na área de tecnologia de informação;
- Jovem adulto de 23 anos, estudante na área de engenharia eletrotécnica;
- Jovem adulto de 25 anos, com o 12º ano de escolaridade;
- Jovem adulta de 23 anos, estudante na área da música;
- Jovem adulta de 22 anos, estudante na área de enfermagem;

É de salientar que nenhum dos utilizadores tem contacto regular com assistentes virtuais.

#### 4.3.1 Métricas objetivas

A avaliação das métricas objetivas permitem-nos tirar conclusões sobre o desempenho do sistema. No conjunto dos oito utilizadores foram realizados vinte e quatro testes (três por cada utilizador) e os resultados obtidos foram os seguintes:

- Teste 1 : 8/8 testes realizados com sucesso;
- Teste 2 : 8/8 testes realizados com sucesso;
- Teste 3 : 8/8 testes realizados com sucesso.

A segunda métrica avaliada no sistema foi o tempo médio de realização dos testes. Esta métrica foi escolhida para existir um termo de comparação entre a resolução de problemas com o assistente virtual e o uso da alternativa existente na atualidade, a linha de apoio ao cliente. Os resultados obtidos indicam um claro decréscimo no tempo de espera do utilizador em comparação com a utilização da linha de apoio, o que torna o sistema implementado numa solução bastante viável para a situação em causa.

Os resultados obtidos através da medição do tempo de duração de cada teste foram separados pelos três diferentes testes e foi calculada a média aritmética e desvio padrão. Na tabela 4.1 estão representados esses mesmos resultados, com os valores temporais em segundos:

Os resultados do teste um diferem no tempo de ativação e desativação pelo simples facto de haver menos diálogo entre o utilizador e o assistente na desativação da rede.

Tabela 4.1: Medição em segundos do tempo de execução dos testes por cada utilizador

|          | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | Média | Desvio padrão |
|----------|------|------|------|------|------|------|------|------|-------|---------------|
| T1 - On  | 14.0 | 13.9 | 15.1 | 16.1 | 12.4 | 14.3 | 13.6 | 14.8 | 14.3  | 1.030         |
| T1 - Off | 13.0 | 12.5 | 13.6 | 14.0 | 13.2 | 13.7 | 14.2 | 12.4 | 13.1  | 0.661         |
| T2       | 33.3 | 29.8 | 31.1 | 33.0 | 29.6 | 30.3 | 32.1 | 31.9 | 31.4  | 1.323         |
| T3       | 66.8 | 66.5 | 69.7 | 70.6 | 68.1 | 71.0 | 67.6 | 68.3 | 68.6  | 1.580         |

O teste T3 apresenta um tempo médio superior ao teste T2 por serem operações sequenciais, ou seja, o teste T3 só é realizado após a aprovação do teste dois.

O desvio padrão aumenta com o aumento do tempo de realização dos testes. Este aumento seria de esperar porque quanto mais tempo demora a interação, maior será o número de diálogos entre o assistente e o utilizador, o que acrescenta uma maior variação nos valores temporais de realização dos testes.

Com o objetivo de estabelecer uma comparação com os resultados obtidos, foram medidos os tempos de realização dos testes mas sem a interação com o assistente virtual, ou seja, foi medido o tempo que demora a resolver o problema através de uma interação direta com o sistema operativo do router. Obtiveram-se os seguintes resultados:

- Teste T1 - Ativar e desativar: 4.3 segundos;
- Teste T2 - 7.3 segundos;
- Teste T3 - 31.2 segundos.

Como seria de esperar, os tempos são bastante mais curtos do que os apresentados na tabela 4.1, também porque os testes foram realizados por um utilizador com alguma experiência em OpenWRT. No entanto, a interação com o assistente virtual é um peça muito importante no sistema pois é isso que garante a acessibilidade aos utilizadores com conhecimento limitado na área.

O contacto com as linhas de apoio traz sempre algum tempo de espera associado. Esse tempo de espera em linha não deverá ultrapassar os 60 segundos [35]. O sistema para além de não fazer o utilizador esperar, demora, na maioria dos casos, menos de 60 segundos a sugerir uma solução para o problema.

### 4.3.2 Resultados do questionário

Nesta secção serão apresentados e interpretados os resultados do questionário feito aos utilizadores.

1. Como classifica a clareza e objetividade das respostas apresentadas pelo sistema? (1- Nada claro, 3- Suficiente, 5- Bastante claro)

8 respostas

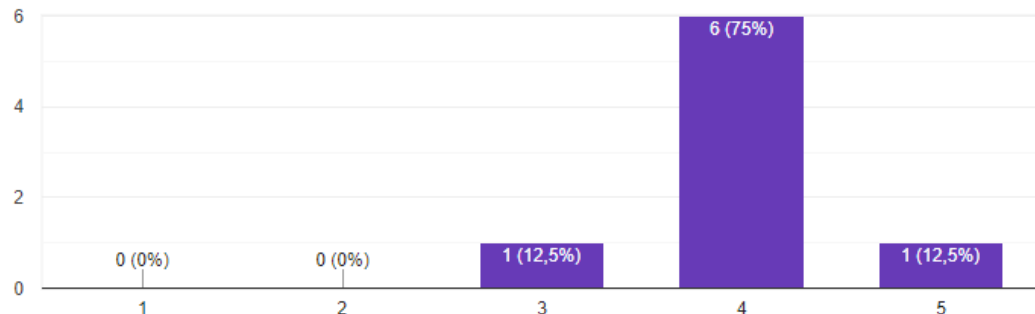


Figura 4.1: Resultados da primeira pergunta do questionário

Na figura 4.1 estão apresentados graficamente os resultados da primeira pergunta do questionário. Os resultados indicam que houve, no geral, um bom entendimento por parte dos utilizadores perante o diálogo feito com o assistente. O sistema, se disponível na língua nativa dos utilizadores, irá proporcionar melhor clareza nas afirmações o que leva a uma melhor percepção por parte do utilizador.

2. Como classifica o grau de terminologia técnica no diálogo? (1 - Alto, 3 - Intermediário, 5 - Baixo);

8 respostas

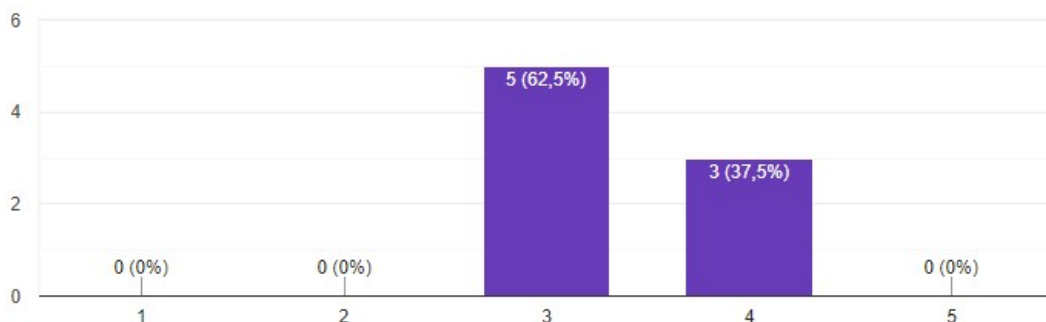


Figura 4.2: Resultados da segunda pergunta do questionário

Na figura 4.2 é possível observar os resultados da questão número dois do questionário. Aqui verifica-se que a maioria dos utilizadores acharam que o grau de terminologia técnica do assistente é intermediário, ou seja, é possível que este grau possa influenciar a compreensão dos utilizadores das intervenções do assistente. Uma solução para combater este ponto menos positivo será, naturalmente, diminuir o número de termos técnicos por parte do assistente. O envolvimento destes termos pode ter um efeito mais grave quanto menor for o conhecimento do utilizador nas áreas

técnicas envolvidas.

3. Como classifica a adequação das soluções propostas face ao seu problema/necessidade? (1- Pouco adequadas, 3- Ajudou alguma coisa, 5- Perfeitas para a situação em causa)

8 respostas

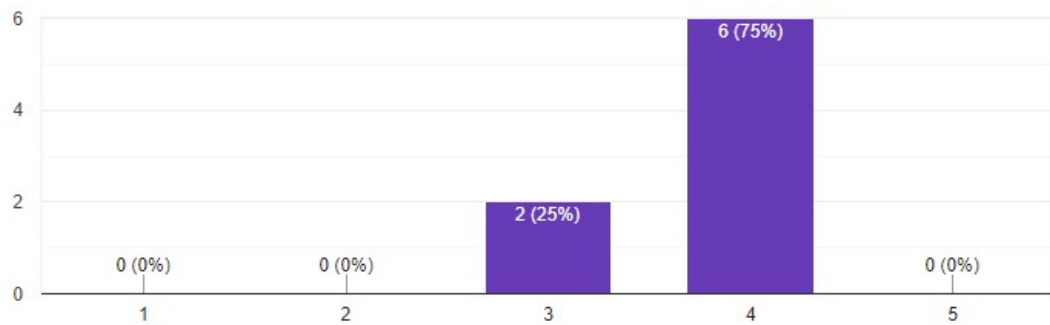


Figura 4.3: Resultados da terceira pergunta do questionário

Na figura 4.3 estão expostos os resultados da terceira pergunta do questionário onde os utilizadores deram a sua opinião acerca das soluções sugeridas pelo assistente. Verificou-se que na maioria dos casos a solução surtiu efeito e houve melhoria na qualidade do serviço.

4. Como classifica a rapidez com que o sistema resolveu o seu problema/necessidade? (1- Muito demorado, 3- Relativamente rápido, 5- Bastante rápido)

8 respostas

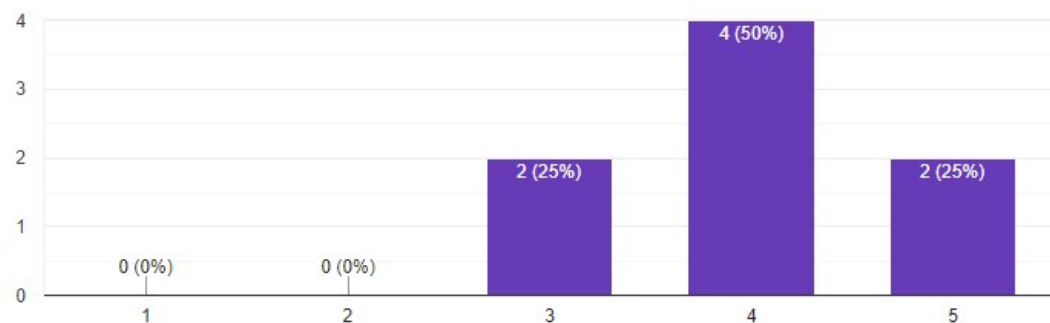


Figura 4.4: Resultados da quarta pergunta do questionário

A figura 4.4 mostra os resultados da pergunta quatro do questionário. Os utilizadores opinaram sobre a velocidade de resposta do sistema. De um modo geral, os resultados são positivos e pode-se afirmar que, de um modo geral, os utilizadores acham que a resposta do sistema é rápida.

5. De um modo geral, como classifica a sua experiência com o assistente virtual? (1- Má, 3- Mediana, 5- Excelente)

8 respostas

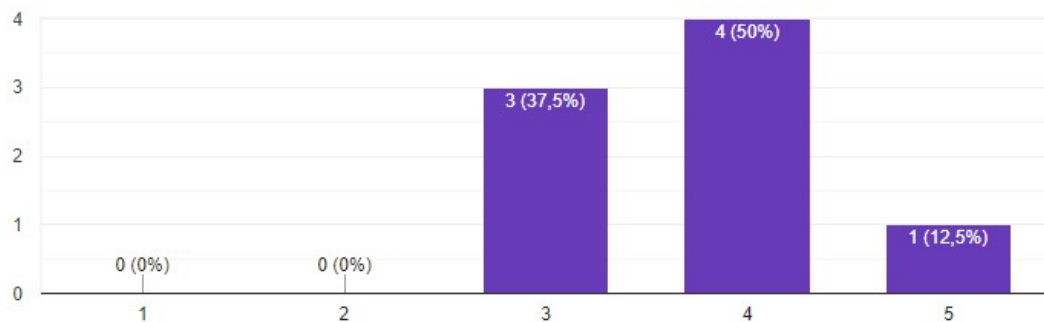


Figura 4.5: Resultados da quinta pergunta do questionário

Na figura 4.5 são demonstrados os resultados da quinta pergunta do questionário. Verificou-se que uma grande parte dos utilizadores não estão completamente à vontade com o uso de assistentes virtuais, apesar disso, a maioria dos utilizadores gostou da experiência com o assistente.

6. Considera que, no futuro, os assistentes virtuais poderão ajudar na gestão de redes domésticas? (1- Não, 3- Talvez, 5- Com certeza)

8 respostas

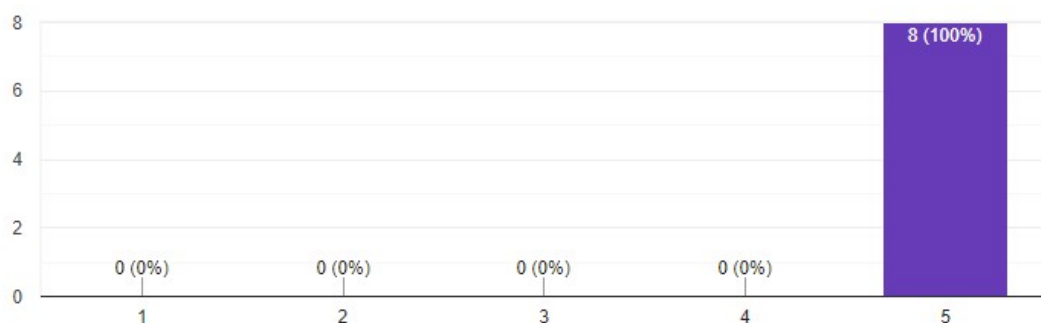


Figura 4.6: Resultados da sexta pergunta do questionário

A figura 4.6 mostra os resultados da sexta e última pergunta do questionário onde os utilizadores deram uma resposta unânime acerca da hipótese dos assistentes virtuais terem uma presença no mercado das redes domésticas no futuro. Todos os utilizadores concordaram que esta tecnologia tem definitivamente espaço para atuar no ambiente doméstico num futuro próximo. Este resultado valida a aposta que é feita no desenvolvimento dos assistentes virtuais e indica que existe algum interesse por parte da população em interagir com esta tecnologia.

### 4.3.3 Resultados não considerados

O sistema ainda está na fase de prova de conceito e por isso não é expectável que tenha um nível de maturidade elevado e que permita responder a qualquer tipo de cenário em ambiente doméstico. Para que o sistema esteja pronto para entrega ao cliente, terá que evoluir em determinados aspetos:

- Aumentar o grau de terminologia, ou seja, aumentar o número de diálogos que podem ser mapeados em intenções de modo a suportar uma maior diversidade de utilizadores;
- Evoluir as linguagens suportadas para suporte de língua portuguesa, ou seja, adaptar o assistente para comunicar na língua nativa dos utilizadores;
- Desenvolver mecanismos de resposta à falha na compreensão das intenções do utilizador, ou seja, no caso do sistema não compreender o pedido do utilizador, deve-se fornecer uma explicação e uma sugestão de alternativa ou uma lista de pedidos possíveis.

Nos 16 testes que foram realizados (testes T2 e T3) houve quatro que necessitaram de repetição. As diferentes razões que espoletaram esta necessidade relacionam-se com a interação com o assistente. As razões são as seguintes:

- Falta de diversidade nos *inputs* do gestor do assistente;
- Língua nativa dos utilizadores diferente da língua do assistente.

Dois testes realizados por dois diferentes utilizadores tiveram que ser repetidos devido à falta de diversidade nos *inputs* do assistente para o *use case*. De maneira semelhante, dois testes também foram repetidos devido à falta de capacidade do utilizador de expressar a sua intenção na língua inglesa.

Embora não haja casos em que o sistema não cumprisse o esperado, as quatro falhas que ocorreram ao nível do diálogo permitem aferir que o sistema necessita de uma maior estrutura de diálogos e também, como seria de esperar, o sistema funcionaria bastante melhor se a sua língua fosse a mesma que a língua nativa dos utilizadores.

## 4.4 Considerações finais

Com a avaliação da experiência do utilizador retirou-se resultados positivos acerca do desempenho do sistema, confirmou-se alguns aspetos que seriam de esperar, como a vantagem em relação ao serviço de apoio ao cliente atual, e constatou-se que existem áreas que podem ser melhoradas no sistema, principalmente a nível linguístico e de interação com o utilizador.

A questão de aumentar a estrutura de diálogos vai proporcionar um maior leque de expressões que o assistente compreende, logo o sistema vai ter menos hipótese de falhar por não compreender o *input* do utilizador. Também é importante ter a língua do assistente igual à língua nativa do utilizador para existir um maior conforto no uso do sistema.

## Capítulo 5

# Conclusão

Este projeto surge na ideia de criar um ambiente doméstico em que o utilizador interage de maneira intuitiva com a sua própria rede. A experiência do utilizador com qualquer ambiente tecnológico melhora significativamente quando é colocada uma interface, neste caso a voz, intuitiva para interagir. Isto verifica-se, principalmente, nos utilizadores com conhecimento limitado na área.

A implementação do projeto focou-se no desenvolvimento de uma estrutura de diálogos para um assistente virtual (Alexa), que permite ao utilizador usar certas funcionalidades de rede. Foram implementados diversas interações que tornam mais rica a experiência do utilizador. Também foram desenvolvidas rotinas de aplicação de funcionalidades do OpenWRT, que fazem a configuração da rede. A associação destas rotinas com a estrutura de diálogos e a arquitetura de comunicação (MQTT) completa o sistema desenvolvido.

Num conjunto de 8 utilizadores, foram realizados 24 testes, dos quais 24 com sucesso num tempo médio de 31,85 segundos, com uma experiência do utilização classificada em média como 3,98 em 5. Alguns testes realizados indicam que é necessário implementar algumas melhorias no sistema de diálogo com o utilizador.

A amostra escolhida para a avaliação do sistema era bastante reduzida, o que traz alguma irrelevância estatística nos resultados obtidos. No entanto os resultados podem ser entendidos como preliminares.

A implementação desta tecnologia na vida dos utilizadores iria trazer benefícios tanto para as operadoras, que iriam apresentar uma menor número de reclamações do serviço de Internet fixa, como para os clientes, que poderiam usufruir mais do serviço com a ajuda de um assistente pronto a prestar apoio na resolução de problemas associados à própria rede doméstica. As operadoras investem muito dinheiro nas linhas de apoio e estas demonstram não conseguir corresponder às expectativas do cliente.

## 5.1 Trabalho futuro

O sistema implementado pode sempre enriquecer com mais funcionalidades e, numa perspectiva de melhorar as capacidades já existentes, também pode ser atualizado com novas tecnologias que vão trazer mais eficiência à resolução dos problemas.

Existe uma funcionalidade que será muito interessante implementar no sistema e que trará muitas vantagens na perspectiva do utilizador. Nomeadamente a implementação de um subsistema de controlo de tráfego na rede. Esta opção permitiria aos utilizadores canalizar a largura de banda disponível na sua totalidade para o próprio dispositivo com apenas o uso da voz. É uma ferramenta muito poderosa e poderá fortificar o caso de uso em que ocorrem problemas na rede devido ao uso excessivo de largura de banda por parte de um ou mais dispositivos. Nesta situação o sistema irá restringir a largura de banda para os vários dispositivos e libertar mais velocidade para o utilizador.

Ao nível dos diálogos seria interessante implementar um mecanismo de *machine learning* na aplicação do assistente virtual que associasse o *input* do utilizador a uma intenção sem a necessidade de ter um estrutura de dados que contenha todas as possibilidades de expressar a intenção em causa.

Também é importante ter a oportunidade de passar a língua do assistente para a língua nativa do utilizador, neste caso o Português. Esta adaptação é fundamental para a exposição do produto no mercado português.

A possibilidade de ter um assistente virtual local, ou seja, as tecnologias de síntese e reconhecimento de fala estariam disponíveis localmente, sem necessitar o acesso à Internet. Neste momento não existe nenhuma solução que permita o armazenamento local das tecnologias.

O sistema também pode ser complementado com mecanismo de análise e aprendizagem do lado das operadoras para que estes tenham uma base e compreensão dos principais problemas sofridos pelos clientes de modo a poderem endereça-los e poderem contribuir para a qualidade de experiência.



# Referências

- [1] Hyunji Chung, Jungheum Park, e Sangjin Lee. Digital forensic approaches for amazon alexa ecosystem. *Digital Investigation*, 22:S15–S25, 2017.
- [2] ANACOM. Divulgação de resultados. [http://www.ecsiportugal.pt/site/assets/files/1019/resultados\\_ecsi\\_2019.pdf](http://www.ecsiportugal.pt/site/assets/files/1019/resultados_ecsi_2019.pdf). accessed in:30/06/2020.
- [3] ANACOM. Decreto-lei n.º 134/2009, de 2 de junho. <https://anacom.pt/render.jsp?categoryId=18200>. accessed in:02/06/2020.
- [4] Alexandra Vtyurina e Adam Fourney. Exploring the role of conversational cues in guided task support with virtual assistants. Em *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, 2018. Association for Computing Machinery. URL: <https://doi.org/10.1145/3173574.3173782>, doi:10.1145/3173574.3173782.
- [5] Gustavo López, Luis Quesada, e Luis A. Guerrero. Alexa vs. siri vs. cortana vs. google assistant: A comparison of speech-based natural user interfaces. Em Isabel L. Nunes, editor, *Advances in Human Factors and Systems Interaction*, páginas 241–250, Cham, 2018. Springer International Publishing.
- [6] Alexa, say what?! voice-enabled speaker usage to grow nearly 130% this year. *eMarketer*, 2017.
- [7] V. Kěpuska e G. Bohouta. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). Em *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, páginas 99–103, Jan 2018. doi:10.1109/CCWC.2018.8301638.
- [8] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018.
- [9] mycroft.ai. Mycroft about. <https://mycroft.ai/about-mycroft/>. accessed in:30/01/2020.
- [10] Prachi Khilari e VP Bhope. A review on speech to text conversion methods. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 4(7):3067–3068, 2015.
- [11] LR Rabiner e BH Juang. Hidden markov models for speech recognition—strengths and limitations. Em *Speech recognition and understanding*, páginas 3–29. Springer, 1992.

- [12] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [13] Kishore Prahallad. Automatic building of synthetic voices from audio books. *Diss. Nagoya Institute of Technology, Japan*, 2010.
- [14] mycroft.ai. Mycroft precise. <https://mycroft-ai.gitbook.io/docs/mycroft-technologies/precise>. accessed in:07/02/2020.
- [15] Hyunji Chung e Sangjin Lee. Intelligent virtual assistant knows your life. *arXiv preprint arXiv:1803.00466*, 2018.
- [16] cisco. cisco ios technologies. <https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-technologies/index.html>. accessed in:06/02/2020.
- [17] dd wrt.com. Ddwrtr about. <https://dd-wrt.com/about/>. accessed in:01/02/2020.
- [18] OpenWRT.org. Openwrt about. <https://openwrt.org/about>. accessed in:01/02/2020.
- [19] cisco. cisco ios technologies. <https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-technologies/index.html>. accessed in:06/02/2020.
- [20] Jasenka Dizdareviundefined, Francisco Carpio, Admela Jukan, e Xavi Masip-Bruin. A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. *ACM Comput. Surv.*, 51(6), Janeiro 2019. URL: <https://doi.org/10.1145/3292674>, doi:10.1145/3292674.
- [21] David Chilcañán, Patricio Navas, e Milton Escobar. Virtual assistant for iot process management, using a middleware. Em *Proceedings of the 2018 2nd International Conference on Algorithms, Computing and Systems*, páginas 209–213, 2018.
- [22] George Alexakis, Spyros Panagiotakis, Alexander Fragkakis, Evangelos Markakis, e Vassilakis Kostas. Control of smart home operations using natural language processing, voice recognition and iot technologies in a multi-tier architecture. *Designs*, 3:32, 07 2019. doi:10.3390/designs3030032.
- [23] M. Tharaniya soundhari e S. Brilly Sangeetha. Intelligent interface based speech recognition for home automation using android application. Em *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, páginas 1–11, March 2015. doi:10.1109/ICIIECS.2015.7192988.
- [24] C. J. Baby, F. A. Khan, e J. N. Swathi. Home automation using iot and a chatbot using natural language processing. Em *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, páginas 1–6, April 2017. doi:10.1109/IPACT.2017.8245185.
- [25] Zhiliang Wang, Ning Cheng, Yumei Fan, Jiwei Liu, e Changsheng Zhu. Construction of virtual assistant based on basic emotions theory. Em Jianhua Tao, Tieniu Tan, e Rosalind W. Picard, editores, *Affective Computing and Intelligent Interaction*, páginas 574–581, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [26] Pirkko H. Harvey, Edward Currie, Padma Daryanani, e Juan C. Augusto. Enhancing student support with a virtual assistant. Em Giovanni Vincenti, Alberto Bucciero, e Carlos Vaz de Carvalho, editores, *E-Learning, E-Education, and Online Training*, páginas 101–109, Cham, 2016. Springer International Publishing.

- [27] Bing Xu, ZG Pan, e HW Yang. Agent-based model for intelligent shopping assistant and its application. Em *The first Conference on Affective Computing and Intelligent Interaction, Beijing*, páginas 306–311. Citeseer, 2003.
- [28] Katherine Isbister, Hideyuki Nakanishi, Toru Ishida, e Cliff Nass. Helper agent: Designing an assistant for human-human interaction in a virtual meeting space. Em *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00*, página 57–64, New York, NY, USA, 2000. Association for Computing Machinery. URL: <https://doi.org/10.1145/332040.332407>, doi:10.1145/332040.332407.
- [29] OpenWRT.org. Opkg package manager. <https://openwrt.org/docs/guide-user/additional-software/opkg>. accessed in:13/07/2020.
- [30] Matt Martz. speed test command line interface. <https://github.com/sivel/speedtest-cli>. accessed in:05/06/2020.
- [31] Collins Abitekaniza. n2w python module. <https://github.com/collin5/python-n2w>. accessed in:05/06/2020.
- [32] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.
- [33] metageek.com. Wi-fi signal strenght basics. <https://www.metageek.com/training/resources/wifi-signal-strength-basics.html>. accessed in:30/06/2020.
- [34] ROSINSON Founder. nft-qos package for openwrt. <https://github.com/rosywrt/nft-qos>. accessed in:30/06/2020.
- [35] ANACOM. Utilização dos serviços de apoio ao cliente. <https://www.anacom.pt/render.jsp?contentId=961775>. accessed in:13/07/2020.