



Universidade do Minho  
Escola de Engenharia

# F3M: Active Voice

Projeto em Engenharia Informática  
Mestrado em Engenharia Informática

---

## Grupo 8

Gabriela Santos Ferreira da Cunha - pg53829

Gonçalo Semelhe Sousa Braga - pg53845

João António Redondo Martins - pg53905

João Pedro Antunes Gonçalves - pg53932

Miguel de Sousa Braga - pg54095

Millena de Freitas Santos - pg54107

Nuno Guilherme Cruz Varela - pg54117



Repositório GitHub: [github.com/MiguelBraga2/F3M-Active-Voice](https://github.com/MiguelBraga2/F3M-Active-Voice)

novembro, 2024

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Objetivos</b>	<b>3</b>
<b>3</b>	<b>Requisitos</b>	<b>3</b>
3.1	Modelo de Domínio . . . . .	3
3.2	Breve Descrição dos <i>Use Cases</i> . . . . .	4
3.3	Requisitos Funcionais . . . . .	6
3.4	Restrições à Solução e Requisitos Não Funcionais . . . . .	6
<b>4</b>	<b>Metodologia</b>	<b>7</b>
<b>5</b>	<b>Análise de Modelos</b>	<b>8</b>
5.1	Reconhecimento de Voz . . . . .	8
5.2	Processamento de Linguagem Natural . . . . .	9
5.3	Restrições de Financiamento . . . . .	11
<b>6</b>	<b>Arquitetura</b>	<b>13</b>
<b>7</b>	<b>Implementação</b>	<b>14</b>
7.1	Reconhecimento de Voz . . . . .	14
7.2	Processamento de Linguagem Natural . . . . .	14
7.3	Comandos de Voz . . . . .	15
7.4	Funcionalidades Adicionais . . . . .	16
<b>8</b>	<b>Resultados</b>	<b>17</b>
<b>9</b>	<b>Modelo de Negócio</b>	<b>18</b>
9.1	Sistema Desenvolvido . . . . .	18
9.1.1	Mercado/Cliente . . . . .	18
9.1.2	Produto Proposto . . . . .	18
9.1.3	Tecnologia . . . . .	19
9.2	Produtização . . . . .	19
9.2.1	Tecnologia . . . . .	19
9.2.2	Mercado . . . . .	19
9.2.3	Produto <i>Mass Market</i> . . . . .	20
9.3	Modelo e Estratégia de Negócio . . . . .	20
9.3.1	Modelo de Negócio . . . . .	20
9.3.2	Viabilidade Financeira . . . . .	21
9.3.3	Viabilidade Económica . . . . .	22
9.3.4	Estratégia de Entrada no Mercado . . . . .	22
9.3.5	Estratégia de Crescimento . . . . .	22
<b>10</b>	<b>Conclusão</b>	<b>23</b>

# 1 Introdução

No âmbito da unidade curricular Projeto de Informática, foi-nos proposto o desafio de desenvolver um projeto em colaboração com a empresa F3M. Este projeto visa investigar e desenvolver uma interface de voz inovadora para integrar em soluções *mobile* já existentes na empresa.

A tecnologia desempenha um papel essencial na simplificação de tarefas diárias e na melhoria da eficiência em diversas áreas. Na área da saúde, os profissionais precisam várias vezes de introduzir dados enquanto estão em contacto com o paciente e não podem manusear dispositivos devido a questões de higiene. Deste modo, com este projeto, pretende-se efetuar uma melhoria na acessibilidade e eficiência na inserção de dados, através da integração de comandos de voz no preenchimento de formulários em aplicações móveis dessa área.

## 2 Objetivos

Neste projeto, espera-se que a investigação se foque em compreender as melhores práticas de *design* de interface, de reconhecimento de voz e de processamento de linguagem natural para criar uma experiência de utilizador fluída e eficiente, resultando numa interface de voz robusta e amigável que possa ser integrada nas soluções *mobile* F3M. Com o propósito específico de orientar e estruturar, de forma precisa, as etapas necessárias para a realização bem-sucedida deste projeto, foram definidos os seguintes objetivos por parte da empresa:

- Investigar as tecnologias existentes de reconhecimento de voz e processamento de linguagem natural aplicáveis ao contexto de formulários móveis;
- Desenvolver protótipos de interface de voz que integrem com soluções *mobile* existentes;
- Avaliar a usabilidade e precisão dos protótipos em diferentes cenários de uso;
- Propor diretrizes de *design* para a implementação efetiva de interfaces de voz em aplicações móveis.

## 3 Requisitos

Após o estabelecimento dos objetivos, procedeu-se, em conjunto com a equipa da F3M, ao levantamento de requisitos, através de métodos como o *brainstorming*, o contacto semanal com os *stakeholders* do produto, a introspeção e a análise dos formulários existentes.

### 3.1 Modelo de Domínio

De forma a obter uma melhor compreensão sobre o problema que temos em mãos, assim como as entidades que estarão envolvidas no sistema, procedeu-se à implementação do seguinte modelo de domínio.

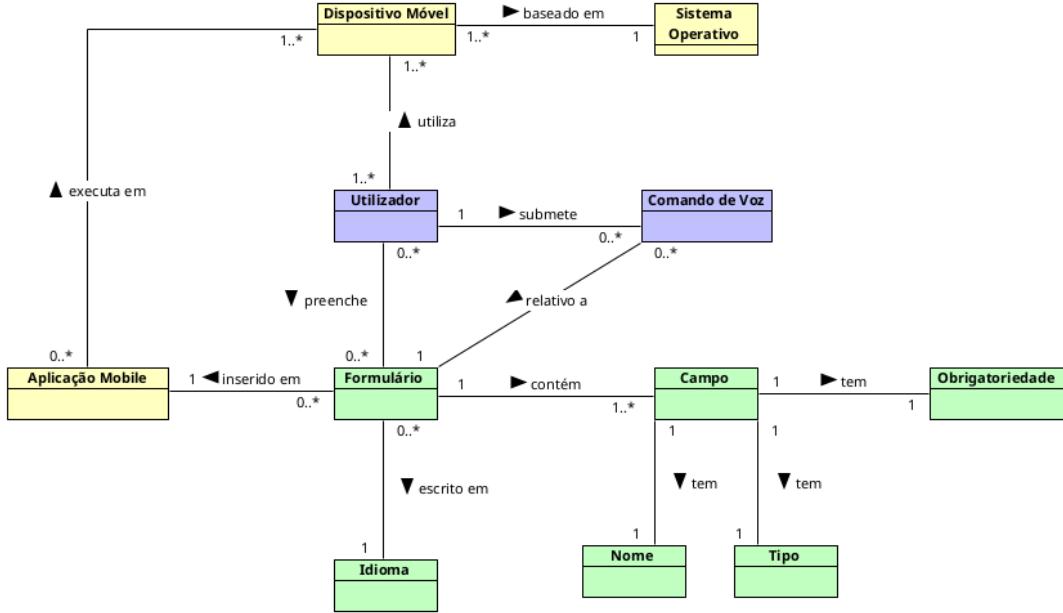


Figura 1: Modelo de domínio.

Este modelo reflete o foco da aplicação no conceito de Formulário que, na nossa aplicação, irá ser estendido para permitir o seu preenchimento de forma automática pelo uso de comandos de voz.

### 3.2 Breve Descrição dos *Use Cases*

No que diz respeito aos casos de uso, especifica-se na tabela 1 o *use case* mais comum na nossa solução, uma vez que esta foi especificamente concebida para permitir o preenchimento de formulários através da interação por voz. Para além deste *use case*, o sistema abrange ainda outros casos relevantes, que foram igualmente considerados no seu desenvolvimento e contribuem para a versatilidade e abrangência da solução apresentada:

- **“Interagir com Assistente de Voz”**: Permite ao utilizador ter informação extra sobre o estado atual do formulário;
- **“Avançar/Recuar Formulário”**: Permite realizar a navegação entre páginas com o uso da voz;
- **“Consultar Manual de Instruções”**: Permite ouvir/saber mais sobre o modo de uso da aplicação.

Esta especificação tabelar visa facilitar o processo de implementação de cada funcionalidade do nosso sistema. Deste modo, consideramos que é bastante percutível o fluxo sequencial da interação do ator com o sistema. Este *use case* permite que os profissionais de saúde preencham os formulários

na aplicação utilizando comandos de voz. O formulário inclui campos como dados pessoais do paciente, sintomas, diagnósticos e tratamento recomendado. Existem dois cenários alternativos: O primeiro quando o utilizador opta pela inserção manual e o segundo quando o sistema não consegue interpretar corretamente o comando de voz, sendo solicitada a correção dos campos ou a inserção manual de dados.

<b>Use case</b>	1- Preencher Formulário com Voz	
<b>Ator principal</b>	Profissional de Saúde	
<b>Ator secundário</b>	Paciente	
<b>Pré-Condições</b>	O Profissional de Saúde está autenticado e o formulário está disponível para preenchimento.	
<b>Pós-Condições</b>	O formulário foi preenchido corretamente e submetido com sucesso.	
	<b>Input do Ator</b>	<b>Resposta do Sistema</b>
<b>Cenário Normal</b>	1 - O Utilizador indica que quer iniciar a gravação por voz dos dados do formulário.	
	3 - O Utilizador começa a preencher o formulário por voz (ex.: "O nome do paciente é João Silva com 34 anos e possui uma ferida com 3 cm de diâmetro").	2 - O Sistema apresenta um sinal sonoro a indicar o início da gravação.
	6 - O utilizador confirma a submissão do formulário com um comando de voz ("Submeter").	4 - O Sistema transcreve os dados ditados e preenche os campos correspondentes. 5 - O Sistema comunica por voz/por escrito ao utilizador quais os campos que não foram preenchidos.
		7 - O Sistema valida o formulário e guarda os dados.
<b>Alternativa 1</b> [Inserção manual] (Passo 1)	1.1 - O Utilizador preenche o formulário manualmente  1.3 - O Utilizador submete o formulário.	1.2 - O Sistema atualiza os campos preenchidos.  1.4 - O Sistema valida o formulário e guarda os dados.
<b>Alternativa 2</b> [Campo não reconhecido corretamente] (Passo 6)	6.1 - O Utilizador corrige o campo com um novo comando de voz.	6.2 - O Sistema atualiza o campo, corrigindo-o com o novo dado fornecido. 6.3 - O Sistema informa o utilizador que o campo foi alterado com a respetiva informação. 6.4 - O Sistema volta a 6.
<b>Exceção 1</b> [Formulário incompleto] (Passo 7)	7.1 - O Sistema informa que campos obrigatórios estão em falta.	7.2 - O Sistema termina o processo.

Tabela 1: Especificação do *use case* “Preencher Formulário com Voz“.

### **3.3 Requisitos Funcionais**

Com base nos casos de uso, foram levantados os seguintes requisitos. Por motivos de limitações de espaço no relatório, estes são apresentados de uma forma bastante resumida. Uma apresentação mais detalhada é feita no documento de requisitos.

- O sistema deve permitir responder por voz a questões de seleção;
- O sistema deve permitir responder por voz a questões de texto;
- O sistema deve permitir responder por voz a questões de escolha múltipla;
- O sistema deve permitir reconhecer, por voz, os comandos de avançar e retroceder formulário;
- O sistema deve permitir que os utilizadores façam correções ou editem os dados inseridos usando comandos de voz;
- O sistema deve fornecer *feedback* aos utilizadores após o reconhecimento dos comandos de voz;
- O sistema deverá reconhecer automaticamente o fim da fala do utilizador;
- O sistema deve pedir ao utilizador permissão para a gravação da sua voz.

### **3.4 Restrições à Solução e Requisitos Não Funcionais**

Foram também levantados os seguintes requisitos não funcionais (mais detalhes no documento de requisitos):

- A solução tem de ser desenvolvida em React Native;
- O sistema deve ser capaz de compreender diversas línguas, tais como o Português e o Inglês;
- O preenchimento de formulários por voz deve ser eficiente;
- A interface de voz deve ser robusta e amigável;
- A GUI deve ser amigável para o cliente;
- O produto deve poder ser usável por pessoas com deficiências visuais, nomeadamente daltonismo;
- Deve ser possível substituir facilmente os modelos de inteligência artificial integrados na solução;
- O sistema deve ser compatível com diferentes dispositivos móveis.

## 4 Metodologia

A metodologia adotada pela equipa revelou-se crucial para o sucesso do projeto. Desde cedo, foram definidos os responsáveis pelos contactos com a empresa e com os docentes da UC, bem como um gestor de projeto, responsável por designar as funções e responsabilidades de cada membro da equipa, assegurando que todas as tarefas fossem distribuídas de forma equilibrada e eficiente. Não obstante, o gestor de projeto acabou por contribuir para a documentação e, sempre que possível, para o código da solução.

Para assegurar que todos estivessem constantemente a par do progresso e do estado das tarefas, utilizamos a ferramenta Jira. Esta ferramenta facilitou-nos a visualização do fluxo de trabalho e permitiu mantermo-nos alinhados com os prazos e objetivos definidos. Para além disso, adotamos uma abordagem próxima do Scrum, com ciclos de tarefas curtos e reuniões semanais com vista a monitorizar o progresso das tarefas e comunicar eventuais dificuldades e problemas que pudessem ter surgido. Para assegurar uma comunicação eficaz com a empresa, foi estabelecido um formato de reunião semanal, por forma a manter a empresa informada sobre o estado das atividades e receber *feedback* contínuo, ajustando as estratégias seguidas conforme necessário.

Numa fase inicial, estabelecemos os seguintes 3 principais objetivos para o projeto:

- **Desenvolvimento de um sistema de reconhecimento de voz:** Criar um sistema capaz de identificar e interpretar comandos de voz do utilizador para preenchimento de formulários;
- **Integração com tecnologias de processamento de linguagem natural (PLN):** Utilizar PLN para entender o contexto e o conteúdo dos campos de formulário, permitindo uma interação mais natural e intuitiva;
- **Garantia de compatibilidade com diferentes dispositivos móveis:** Assegurar que a interface de voz funciona de forma eficiente em diversos dispositivos e sistemas operacionais.

Desta forma, a equipa foi dividida para que o trabalho sobre os modelos de reconhecimento de voz e processamento de linguagem natural pudesse ser realizado de forma paralela. Também houve a necessidade de designar um responsável por avançar com o desenvolvimento do protótipo em *React Native*, para que depois pudesse ser possível iniciar a integração dos modelos por parte de cada sub-equipa. Neste sentido, foram investigados diversos modelos, definidos os que seriam efetivamente utilizados e, por último, integrados no protótipo por parte de cada sub-equipa. A partir desta integração, o foco deslocou-se para o desenvolvimento da interface, onde foram adicionados vários tipos de perguntas idênticas aos formulários da empresa, com vista a demonstrar a eficácia dos modelos para qualquer tipo de questão. Após a adição das questões, a atenção voltou-se para a *user experience*, onde foram incorporados comandos de voz de acessibilidade, para funções tais como avançar, retroceder ou submeter os formulários. Por fim, como último aspeto indispensável para a solução a apresentar, lidamos com as situações em que campos obrigatórios não são preenchidos, impedindo a submissão do formulário. Desta forma, seguindo o propósito da aplicação em evitar manuseamento por parte dos profissionais de saúde, é emitido um *feedback* auditivo que informa os campos que não foram preenchidos, para que os profissionais possam, então, preenchê-los.

Após o entendimento por parte da equipa de que a solução já atendia aos objetivos do projeto, procedeu-se à realização de testes e refinamento dos componentes abordados anteriormente. Em

diálogo com a empresa, foi também decidida a implementação de novas funcionalidades, como um assistente virtual no protótipo. Esta ferramenta foi considerada útil, pois os profissionais de saúde precisam de um suporte claro para qualquer dúvida quando do preenchimento dos formulários. Além disso, foram integradas mais perguntas e o protótipo foi progressivamente aproximando-se dos formulários utilizados pela empresa, de forma a que a integração no *software* existente fosse mais direta.

## 5 Análise de Modelos

A primeira fase do projeto, após recolhidos os requisitos, consistiu em implementar os modelos para este projeto, tanto para a componente do reconhecimento de voz como para a componente do processamento de linguagem natural. A ideia passou por utilizar um modelo já treinado para ambas as componentes, uma vez que existem diversas soluções prontas e amplamente testadas, que prometem boa precisão e desempenho. Estes modelos pré-treinados oferecem a vantagem de serem desenvolvidos e otimizados por grandes comunidades de investigadores e empresas, trazendo um desempenho sólido para uma ampla variedade de aplicações. Neste sentido, para a escolha final dos modelos, consideramos os seguintes fatores:

- **Preço:** Valor necessário a pagar para a utilização do modelo;
- **API disponível:** É necessário a avaliação da quantidade de chamadas possíveis a serem realizadas à API;
- **Desempenho:** É necessário avaliar o seu desempenho antes de ser utilizado de uma forma mais intensiva, garantindo assim que cumpre certos padrões, como por exemplo, um tempo aceitável de resposta.

Esta secção apresenta todo o processo de avaliação dos modelos e os respetivos resultados após investigar e testar vários modelos já existentes, com o objetivo de encontrar a melhor opção que pudesse ser integrada de maneira eficiente no nosso sistema para atender às várias particularidades.

### 5.1 Reconhecimento de Voz

Relativamente à tarefa do reconhecimento/transcrição de áudio, para realizarmos a comparação entre os modelos, foi utilizado um pequeno excerto composto por 10 áudios, cada um com uma duração de 20 segundos, do *dataset MLS (MultiLingual Librispeech)*, todos em português e abrangendo diferentes vozes. Essa seleção foi feita para garantir uma diversidade de oradores, incluindo variações de timbre e entoação, permitindo avaliar a capacidade dos modelos de lidar com a heterogeneidade natural da fala.

No que toca às métricas utilizadas para a comparação, a escolha recaiu sobre o *Word Error Rate* (WER) e a latência. Esta primeira métrica é essencial para quantificar a precisão dos modelos, medindo o número de substituições, inserções e deleções efetuadas pela transcrição obtida, relativamente à transcrição correta. A latência consiste, também, num fator crucial, uma vez que a capacidade de gerar transcrições em tempo real ou com o menor atraso possível é determinante para garantir uma experiência satisfatória ao utilizador, especialmente em sistemas interativos.

Nas tabelas 2 e 3, encontram-se os resultados obtidos na nossa comparação de modelos. Foram testados os modelos Whisper (OpenAI), Speech Studio (Microsoft Azure), Speech-to-Text (Google) e Nova (Deepgram). De notar que ainda foram considerados os modelos Wav2vec (MetaAI) e DeepSpeech (Mozilla). Contudo, o primeiro modelo não suportava a língua portuguesa e o segundo encontrava-se *deprecated*, apresentando diversos problemas de dependências e incompatibilidades. Dos modelos testados, todos foram testados na respetiva plataforma, à exceção do Whisper que foi testado localmente sem GPU, justificando os seus elevados valores de latência.

Modelo	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Média
Whisper	6,5	5,5	5,4	4,1	2,2	7,3	6,1	6,9	4,3	4,4	<b>5,7</b>
Nova	13	3,6	5,4	16,3	4,4	7,3	6,1	9,3	6,4	8,9	<b>8,1</b>
Speech-to-Text	28,2	20	8,9	36,7	11,1	14,6	16,3	83,7	36,1	17,8	<b>27,3</b>
Speech Studio	8,7	5,5	1,8	6,1	2,2	9,8	4,1	6,9	14,9	4,4	<b>6,4</b>

Tabela 2: *Word Error Rate*, em %, de cada modelo para cada áudio.

Modelo	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Média
Whisper	32,8	33,1	32,9	32,4	29,2	29,3	34,1	32,2	31,4	30,7	<b>31,8</b>
Nova	3	1	2	4	1	2	3	9	2	1	<b>2,8</b>
Speech-to-Text	6,5	6,2	6,3	13,6	6,2	6,3	6,9	6,6	6,2	6,2	<b>7,1</b>
Speech Studio	11,1	11	10,6	11,1	10,7	10,8	11,4	19,4	10,6	10,5	<b>11,7</b>

Tabela 3: Latência, em segundos, de cada modelo para cada áudio.

Para o modelo da Google, esperava-se um desempenho superior, considerando que se trata de um serviço pago. No entanto, os resultados não corresponderam às expectativas, especialmente em comparação com outros modelos analisados. Observando as métricas, o segundo modelo destacou-se apresentando um bom equilíbrio entre a taxa de erro e o tempo de latência. À medida que a percentagem de erro diminuía, notou-se uma tendência de aumento no tempo de latência, sugerindo uma compensação entre precisão e velocidade. É importante realçar que o Whisper, o único modelo *open source* na nossa análise, demonstrou os melhores resultados em termos de desempenho. Devido à sua natureza de código aberto e assumindo a utilização de uma máquina com GPU para melhores resultados a nível de latência, o Whisper traduziu-se na nossa escolha final para este projeto.

## 5.2 Processamento de Linguagem Natural

Relativamente ao processamento de linguagem natural, a tabela 4 resume as informações investigadas sobre os modelos propostos inicialmente. Os modelos escolhidos para a análise de desempenho foram, então, o GPT (OpenAI), Claude (Anthropic), BERT (Google), Llama 3 (MetaAI), Llama 3.2 (MetaAI), Gemma 2 (Google) e Phi 3.5 (Microsoft).

Modelo	Necessidade de Treino	Gratuito	API Disponível	Tarefas Principais
<b>GPT</b>	Não para tarefas gerais	Não	Sim	Geração e extração de texto, compreensão de linguagem natural, diálogos
<b>Claude</b>	Não para tarefas gerais	Não	Sim	Interpretação de intenções, respostas seguras, PLN geral
<b>BERT</b>	Pode precisar para tarefas específicas	Sim	Sim	Reconhecimento de entidades, classificação de texto, resposta a perguntas
<b>Llama 3</b>	Pode não precisar de treino para extração de informações simples	Sim, localmente	Limitada	Tarefas de PLN com baixo custo computacional, extração de dados, análise de texto
<b>Llama 3.2</b>	Pode não precisar de treino para extração de informações simples	Sim, localmente	Limitada	PLN eficiente, extração de dados, otimização para hardware limitado
<b>Gemma 2</b>	Pode necessitar de <i>fine tuning</i> para precisão em tarefas mais específicas	Sim, localmente	Limitada	Extração de dados em dispositivos de baixo recurso, PLN eficiente
<b>Phi</b>	Pode precisar de <i>fine tuning</i> dependendo da complexidade do texto	Sim, localmente	Limitada	Extração de informações, PLN geral com baixo custo computacional

Tabela 4: Comparação de modelos de processamento de linguagem natural.

O BERT foi descartado por exigir um pré-treino exaustivo para alcançar um desempenho aceitável em comparação com os outros modelos, que não necessitaram desse pré-treino. O Gemma 2 (Google) apresentou um tempo de resposta extremamente elevado, o que impossibilitou a realização de testes completos e, portanto, foi descartado pelo desempenho.

De forma a testar estes modelos, foram elaborados 4 formulários distintos, bem como os respetivos *schemas*. Para cada formulário, foram gerados 3 textos contendo a informação a ser extraída para preencher os seus campos, assim como ficheiros JSON com as respostas corretas esperadas. Todos os modelos receberam o mesmo *prompt*, bem como as mesmas definições de parâmetros e hiperparâmetros.

As métricas utilizadas para a avaliação dos modelos foram a *accuracy* e o tempo de resposta. A *accuracy* é fundamental, uma vez que indica a proporção de previsões corretas em relação ao total de previsões realizadas, permitindo assim medir a eficácia do modelo em identificar corretamente a informação a ser preenchida em cada campo do formulário. Por outro lado, o tempo de resposta é uma métrica crítica, pois reflete a eficiência do modelo na geração de respostas em tempo real.

Desta forma, a combinação destas métricas forneceu-nos uma visão abrangente da *performance* de cada modelo.

As tabelas 5 e 6 apresentam todos os resultados obtidos. Foram testados os modelos Llama 3 (MetaAI) com 8 mil milhões de parâmetros, Llama 3.2 (MetaAI) com 3 mil milhões de parâmetros e Phi 3.5 (Microsoft) com 3.8 mil milhões de parâmetros.

Modelo	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	Média
Llama 3	0,0	0,93	1,0	0,92	1,0	0,92	1,0	1,0	1,0	0,93	0,89	0,89	<b>0,87</b>
Llama 3.2	1,0	1,0	1,0	1,0	1,0	1,0	0,92	1,0	1,0	0,96	0,96	0,85	<b>0,97</b>
Phi 3.5	1,0	0,87	1,0	0,92	0,85	1,0	1,0	1,0	1,0	0,89	0,93	0,78	<b>0,94</b>

Tabela 5: *Accuracy* de cada modelo.

Modelo	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	Média
Llama 3	124,86	50,07	47,32	84,70	47,61	42,72	63,07	28,43	26,43	134,01	64,71	70,17	<b>65,34</b>
Llama 3.2	47,46	23,16	24,83	37,77	22,70	20,50	27,71	13,00	12,32	61,29	30,97	37,50	<b>29,93</b>
Phi 3.5	70,18	39,14	37,25	61,57	35,51	32,10	43,67	23,59	21,62	91,90	46,28	57,42	<b>46,69</b>

Tabela 6: Tempo de resposta, em segundos, de cada modelo.

No que diz respeito à *accuracy*, os modelos testados demonstraram desempenhos notáveis, com o Llama 3.2:3b a destacar-se com 0.97. Este resultado supera o Phi 3.5:3.8b com 0.94 e o Llama 3:8b com 0.87, evidenciando que o Llama 3.2:3b, apesar do seu tamanho mais reduzido, consegue uma precisão superior. Estes resultados sugerem que avanços na arquitetura e no treino dos modelos podem ter um impacto mais relevante na precisão do que apenas um aumento do tamanho do modelo.

Quanto ao tempo de resposta, observou-se uma variação considerável entre os modelos. O Llama 3.2:3b destacou-se novamente entre os modelos locais com o menor tempo médio de resposta (29.93 segundos), seguido pelo Phi 3.5:3.8b (46.69 segundos) e, por último, o Llama 3:8b (65.34 segundos). Esta diferença substancial no tempo de processamento é particularmente relevante para que o formulário seja preenchido.

O desempenho do Llama 3.2:3b, tanto em *accuracy* como em tempo de resposta, fez com que este fosse o modelo escolhido pela equipa para o projeto.

### 5.3 Restrições de Financiamento

Após a fase de teste, comparação e seleção dos modelos adequados para a nossa solução, concluímos que, para atender aos requisitos de tempo de resposta, seria indispensável o uso de máquinas equipadas com GPUs dedicadas. Assim, essa necessidade foi comunicada à empresa, juntamente com os requisitos técnicos necessários para implementar esta infraestrutura e foi realizada uma análise de financiamento para avaliar os custos envolvidos. Durante esta análise, considerou-se uma alternativa viável: a utilização da API da OpenAI. Esta escolha foi fundamentada no facto de tanto

o modelo de *Speech-to-Text* como o de NLP pertencerem à mesma empresa. Como esta API opera com um modelo de pagamento por pedido, revelou-se uma solução financeiramente viável e flexível para as necessidades da empresa.

Uma vez que o Whisper já seria a escolha para o reconhecimento de voz, a transição para a API da OpenAI traduziu-se num *downgrade* na versão - o modelo testado localmente foi o `whisper-large-v3` e o modelo utilizado pela API é o `whisper-large-v2`. Assim sendo, foi necessária a realização de testes, por forma a perceber se este *downgrade* impactava significativamente a *accuracy* do modelo.

Métrica	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Média
WER (%)	2,2	9,1	3,6	2	20	26,8	4,1	4,6	4,3	4,4	<b>7,6</b>
Latência (s)	3,9	4,2	2,7	2,7	2,9	2,3	2,5	2	4,6	6,9	<b>3,4</b>

Tabela 7: Resultados do modelo Whisper presente na API da OpenAI.

Em termos de *accuracy*, como esperado, esta versão do Whisper apresentou piores resultados que a versão mais recente. Contudo, a diferença não foi tão significativa ao ponto de comprometer a eficácia geral do sistema. A média de 7,6% demonstra que a versão analisada ainda consegue obter resultados bastante aceitáveis, superando os modelos Nova (Deepgram) e Speech-to-Text (Google). Relativamente à latência, com uma redução média para 3,4 segundos, esta versão apresentou uma resposta muito mais ágil, sendo apenas ligeiramente inferior ao modelo Nova (Deepgram). Este equilíbrio entre um leve declínio no desempenho do modelo e uma redução considerável no tempo de processamento justificou o *downgrade* na versão do Whisper, necessário para a utilização da API da OpenAI.

Por outro lado, a transição para a API da OpenAI exigiu a substituição do modelo de NLP. Na tabela 8 apresentam-se os resultados obtidos pelo modelo utilizado na API da OpenAI, o GPT-4.

Métrica	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	Média
Accuracy	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	<b>1,0</b>
Latência (s)	8,78	6,75	6,35	5,12	6,79	5,42	4,40	2,33	2,09	6,57	7,02	3,08	<b>5,39</b>

Tabela 8: Resultados do modelo GPT-4 presente na API da OpenAI.

O GPT-4 destacou-se pelo seu desempenho impecável, com uma *accuracy* superior ao modelo previamente selecionado, o Llama 3.2. Relativamente à latência, com uma média de 5.39 segundos, este valor é aproximadamente 5.55 vezes mais rápido que o Llama 3.2. De notar que esta diferença entre valores poderia vir a ser atenuada com a utilização de GPU no modelo Llama 3.2.

Embora a latência dos modelos com GPU pudesse ser menor, a diferença relativamente aos valores obtidos nos novos modelos não seria significativa o suficiente para justificar o elevado custo associado à manutenção de máquinas com GPU. Desta forma, dado o baixo custo, o elevado desempenho e o reduzido tempo de resposta de ambos os modelos, a opção mais vantajosa mostrou-se ser a API

da OpenAI, que utiliza os modelos `whisper-large-v2` para o reconhecimento de voz e `gpt-4-0613` para o processamento de linguagem natural.

## 6 Arquitetura

A arquitetura da solução desenvolvida pelo grupo encontra-se resumida pela figura 2.

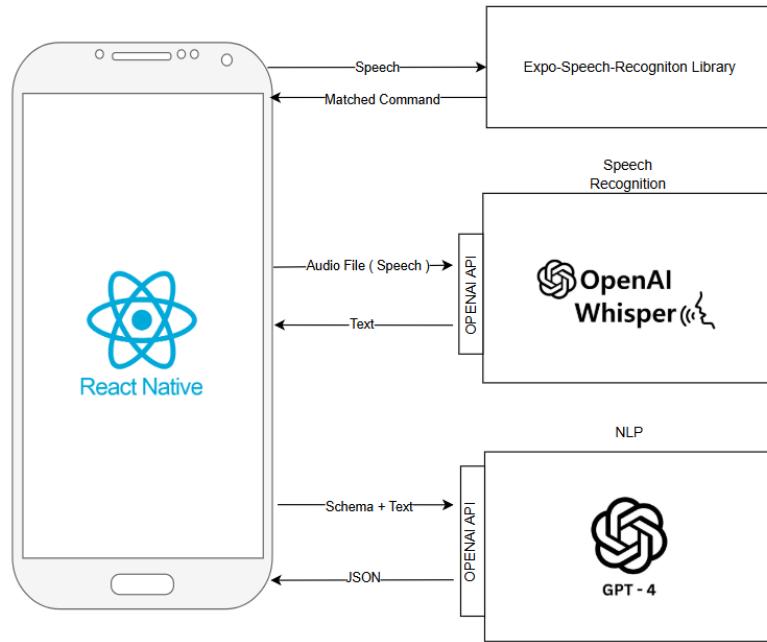


Figura 2: Arquitetura da solução.

Esta arquitetura descreve um sistema que combina reconhecimento de voz e processamento de linguagem natural. A aplicação móvel foi desenvolvida em *React Native*, com vista a facilitar a integração com a solução da F3M, uma vez que é baseada na mesma tecnologia. A aplicação recebe como *input* a voz do utilizador, sendo esta, em seguida, processada de modo a preencher os formulários. Para realizar este processamento da voz, isto é, converter o discurso em texto, utilizamos o modelo Whisper, através da API da OpenAI. Para o preenchimento do formulário é necessário passar como *input* o *schema* JSON pretendido, bem como o *output* de texto proveniente do processamento da voz. Por último, é devolvido o JSON com os campos preenchidos.

Além da funcionalidade principal de preenchimento de formulários com a voz, a aplicação também oferece uma funcionalidade de reconhecimento de comandos de voz para facilitar a interação com o sistema. Para isso, utilizamos a biblioteca Expo Speech Recognition, que disponibiliza um modelo de

reconhecimento de fala diretamente no dispositivo. Este modelo permite que a aplicação reconheça comandos de voz de maneira rápida e eficiente, sem a necessidade de depender de uma API externa.

## 7 Implementação

### 7.1 Reconhecimento de Voz

Para implementar o reconhecimento de voz, foi desenvolvido um *hook* que disponibiliza as funções necessárias para a principal funcionalidade da aplicação. Este *hook* permite realizar a gravação de áudio do utilizador, enviar o áudio ao modelo Whisper para a conversão do discurso em texto e, posteriormente, enviar o texto ao GPT, que interpreta e preenche automaticamente os campos do formulário.

Para efetuar a gravação do áudio, foi utilizado um algoritmo para o sistema detetar o final da fala do utilizador. Esta funcionalidade melhora a fluidez da interação, evitando gravações desnecessariamente longas e proporcionando uma experiência mais natural e eficiente. Neste sentido, o algoritmo monitoriza continuamente os níveis de volume capturados durante a gravação. Em intervalos regulares, os valores de volume são armazenados numa janela de amostras e analisados para calcular uma média móvel, esta que, por sua vez, é usada para definir um limiar dinâmico ajustado ao ambiente. Quando o volume permanece consistentemente abaixo deste limiar por um período pré-definido, o sistema interpreta que o utilizador concluiu a sua fala, interrompe a gravação automaticamente e retorna o URI do áudio.

### 7.2 Processamento de Linguagem Natural

Para realizar este processamento, o GPT-4 utiliza como *input* o texto gerado pelo Whisper, como referido na arquitetura. Em seguida, o modelo gera um *output* estruturado no formato JSON, de acordo com um esquema definido pelo utilizador. Este esquema segue os vários campos do formulário, facilitando o preenchimento por voz. Para este processo, foi utilizada a *feature* “Structured Outputs” fornecida pela própria OpenAI.

Em seguida, é mostrado um exemplo de um esquema que é passado ao GPT-4, num dos formulários da nossa aplicação:

```
1 const formRequestSchema = {
2   type: "object",
3   properties: {
4     painLevel: { type: "number", description: "The pain level on a
5       scale from 0 (no pain) to 10 (worst pain imaginable). Acceptable
6       values range from 0 to 10." },
7     painLocation: { type: "string", enum: painLocationOptions },
8     observations: { type: "string", description: "Observations relative
9       to the injury" },
10    boneOrMucosa: { type: "string", enum: ['yes', 'no'], description: "
11      If the injury is on a bony or mucosal prominence" },
12    conditions: { type: "array", items: { type: "string", enum:
13      fieldConfig.find(field => field.name === 'conditions')?.props.
14      options } },
```

```

9      time: { type: "number", description: "The estimated time in months
10         since the injury started. Only an integer value representing the
11         number of months is accepted." },
12     },
13   },
14 }
```

O *schema* apresentado tem como objetivo estruturar as respostas que serão fornecidas ao modelo GPT, funcionando como um *prompt* detalhado e específico. Este processo garante que as respostas geradas pelo GPT sigam um formato predeterminado e atendam às expectativas do utilizador. A seguir, discutimos os principais aspectos que devem ser considerados na criação de um *schema* eficaz:

- **Nomes explícitos nos campos:** Os nomes das propriedades devem ser claros e descriptivos para evitar ambiguidades. Por exemplo, o campo `painLevel` deixa claro que se refere ao nível de dor, facilitando a compreensão tanto para o modelo quanto para o utilizador;
- **Tipos corretos para cada campo:** A definição precisa dos tipos de dados (por exemplo, `number`, `string`, `array`) é essencial para garantir que as respostas sejam formatadas corretamente. No caso do campo `painLevel`, é utilizado o tipo `number`, com uma descrição que limita os valores aceitáveis entre 0 e 10;
- **Descrição apropriada:** Cada campo do *schema* deve ser acompanhado de uma descrição detalhada. Isto ajuda o modelo GPT a entender o propósito de cada campo, além de guiar o utilizador no preenchimento adequado. Por exemplo, a descrição do campo `painLocation` pode incluir exemplos ou explicações sobre as opções válidas;
- **Limitação de opções com `enum` e `array`:** Para campos que exigem uma escolha única, como o campo `boneOrMucosa`, utiliza-se o tipo `enum` para apresentar as opções `['yes', 'no']`. Já para campos que permitem múltiplas escolhas, como `conditions`, o tipo `array` combinado com itens enumerados restringe as respostas a um conjunto de opções predefinidas.

### 7.3 Comandos de Voz

De modo a tornar a aplicação completamente interativa e eliminar a necessidade de toque no ecrã, foram implementados comandos de voz, atendendo aos requisitos específicos relacionados aos profissionais de saúde. Esta funcionalidade visa evitar que os utilizadores necessitem de remover as luvas, reduzindo assim a necessidade de refazer processos de preparação e esterilização. Com esta abordagem, procura-se garantir uma experiência fluida no preenchimento dos formulários, dispensando o acesso ou a visualização da tela. Os comandos de voz foram definidos utilizando expressões regulares (*regex*), de forma a oferecer maior flexibilidade e opções aos utilizadores. Os seguintes comandos foram implementados:

- **Avançar:** Permite avançar para a próxima página do formulário;

- **Recuar:** Permite retornar à página anterior do formulário;
- **Iniciar ou Começar:** Inicia a gravação da voz do utilizador para o preenchimento do formulário;
- **Exibir transcrição:** Apresenta a transcrição do texto falado pelo utilizador ao preencher o formulário;
- **Ajuda:** Ativa o assistente para responder a perguntas relacionadas aos campos do formulário ou ao uso da aplicação;
- **Manual:** Permite ditar o manual de instruções da aplicação ao utilizador.

A interpretação destes comandos de voz evoluiu ao longo do desenvolvimento da aplicação. Numa primeira fase, a interpretação destes comandos era realizada da mesma forma que o reconhecimento de voz, pois era feito um pedido à API para reconhecer aquele comando. Isto implicava certas limitações a nível de duração, pois implicava esperar pela interpretação do comando e consequente resposta da API.

Numa segunda e última fase, a interpretação dos comandos de voz passou a ser realizada de forma local, utilizando o serviço nativo da Google. Isto melhorou substancialmente o tempo necessário para o reconhecimento dos comandos, uma vez que já não é necessário efetuar pedidos a servidores externos para a interpretação dos comandos. Além disso, o objetivo desse componente é exclusivamente detetar e interpretar comandos simples de voz, como “próximo” ou “submeter”, para controlar a aplicação. Deste modo, o uso de um modelo leve e local é mais que suficiente para atender às necessidades específicas, garantindo uma solução eficiente e responsiva para o utilizador.

As restrições associadas a esta abordagem incluem a necessidade de o *smartphone* possuir um idioma definido na aplicação “Speech Recognition & Synthesis” do Google. Felizmente, esta funcionalidade está disponível a partir do momento em que o *smartphone* é configurado. Além disso, para garantir uma experiência ideal e o pleno funcionamento das funcionalidades da aplicação, é necessário que o dispositivo utilize o Android na versão 13 ou superior.

## 7.4 Funcionalidades Adicionais

Para melhorar a usabilidade e garantir uma interação mais intuitiva e eficiente, foram implementadas funcionalidades adicionais na aplicação. Estas funcionalidades complementam o sistema base, oferecendo suporte ao utilizador durante o preenchimento e submissão do formulário, com especial foco na acessibilidade e na identificação de erros. Abaixo, são descritas as principais melhorias introduzidas.

### ***Feedback auditivo***

A aplicação fornece *feedback* auditivo em vários momentos-chave para melhorar a experiência do utilizador. Ao iniciar a gravação de voz, é emitido um som que indica que o sistema está a gravar, e outro som é reproduzido no final da gravação, confirmando que esta foi concluída. Além disso, no momento da submissão do formulário, caso sejam detetados erros, a aplicação fornece *feedback* auditivo indicando os campos obrigatórios que não foram preenchidos. Esta funcionalidade tem como objetivo auxiliar o utilizador a identificar de forma rápida e clara as informações em falta, facilitando a correção e completude do formulário.

### **Scroll automático**

Quando o formulário é atualizado através de comandos de voz ou ocorre um erro na submissão, a aplicação realiza automaticamente um *scroll* até ao primeiro campo obrigatório que não tenha sido preenchido, colocando-o em destaque. Esta funcionalidade melhora a usabilidade, permitindo que o utilizador se concentre imediatamente no campo que necessita de atenção, otimizando a interação com a aplicação.

### **Assistente**

Um dos recursos disponíveis na nossa aplicação é o comando de “ajuda”. Este comando utiliza o *schema* de cada formulário para interpretar o seu estado atual, fornecendo informações relevantes que auxiliam o utilizador no preenchimento. Além disso, conta com o poder do GPT-4 para gerar respostas claras, estruturadas e úteis, otimizando a experiência do utilizador.

## **8 Resultados**

Nesta secção, são apresentados exemplos práticos que ilustram a interação e funcionalidade da aplicação. Os exemplos demonstram como o sistema processa e responde a *inputs* do utilizador, evidenciando a sua eficiência e utilidade.

Primeiramente, mostramos um caso de gravação de áudio processado pelo modelo Whisper, destacando a transcrição automática da fala e a resposta gerada pelo GPT para preencher os campos dum formulário.

**Exemplo de texto produzido pelo Whisper:** “Não é possível obter sintomas, o tempo estimado desde que a ferida apareceu é 9 meses e a ferida é recorrente.”

**Resposta gerada pelo GPT:**

```
1  {
2      "possivelObterEstimativaTempo": true,
3      "recorrente": "yes",
4      "sintomas": "no",
5      "tempoDesdeInicio": 9
6 }
```

Em seguida, apresentamos um exemplo do comando de “ajuda”, que também utiliza a gravação de áudio processado pelo modelo Whisper e a resposta do GPT aos pedidos feitos pelo utilizador.

**Exemplo de texto produzido ao invocar o comando de ajuda:** “Explica-me os campos presentes neste formulário.”

**Resposta ao pedido de ajuda solicitado:**

“ Claro, este formulário contém os seguintes campos:

1. “Indicate if the patient has symptoms” - Aqui, você deve indicar se o paciente apresenta sintomas. As opções são ‘sim’ ou ‘não’.

2. “The pain level” - Este campo é usado diversas vezes no formulário para indicar o nível de dor em várias situações, numa escala de 0 a 10.
3. “The estimated time since the injury started” - Aqui, você deve inserir o tempo estimado, em meses, desde o início da lesão ou ferida.
4. “If we can obtain the estimated time since the injury started” - Este campo indica se é possível obter uma estimativa de quando a lesão ou ferida começou. Responda com verdadeiro ou falso.
5. “Indicate if the wound is recurrent” - Este campo permite que você indique se a ferida é recorrente. As opções são ‘sim’, ‘não’ ou ‘não é possível obter’.”

Estes exemplos servem para validar a capacidade do sistema em lidar com situações reais e oferecem uma visão prática da aplicação no suporte ao utilizador.

## 9 Modelo de Negócio

### 9.1 Sistema Desenvolvido

#### 9.1.1 Mercado/Cliente

O sistema desenvolvido, proposto pela F3M, tem como principal objetivo ajudar os profissionais de saúde no preenchimento de formulários, através do uso da voz. De facto, o processo de preenchimento de formulários médicos foi relatado como cansativo e indesejável, dada a natureza extensiva dos mesmos. Numa primeira fase, o sistema será aplicado no mercado da saúde, inserindo-se na aplicação *Wounds*, desenvolvida pela F3M. Numa fase posterior, este poderá ser aplicado noutras áreas/mercados como a área social.

#### 9.1.2 Produto Proposto

O produto proposto é uma biblioteca, disponível para diferentes sistemas operativos, que torna possível o preenchimento de formulários através do uso da voz. Para além disso, é suportado o uso de comandos de voz para navegação entre componentes e comunicação com um assistente de voz.

O sistema desenvolvido torna a interação entre utilizador e sistema muito mais fluída e intuitiva, além de permitir ao cliente paralelizar as suas tarefas. Por exemplo, no caso dos formulários médicos, o enfermeiro pode ditar para o telemóvel o conteúdo do(s) formulário(s) e libertar as mãos para desempenhar outra tarefa. Além disso, este novo paradigma de interação liberta o profissional de saúde do uso de tecnologia durante a consulta, permitindo um maior foco nas necessidades do paciente, bem como evita que os profissionais precisem constantemente higienizar ou trocar luvas ao manipular dispositivos móveis, ou ainda anotar informações manualmente para posteriormente preencher os formulários, economizando tempo e esforço.

Por fim, o produto traz também valor às entidades patronais ao aumentar a eficiência na execução das tarefas. Isso permite que o tempo economizado seja dedicado a outras atividades, o que resulta em maior produtividade e, consequentemente, em maiores ganhos a longo prazo.

### **9.1.3 Tecnologia**

O sistema foi desenvolvido tendo em conta a boa manutenção e modularidade do mesmo com o objetivo de apresentar uma fácil instalação e integração noutras aplicações. O desenvolvimento do protótipo do sistema em React Native permite uma abordagem multiplataforma, o que permite o seu funcionamento em vários sistemas operativos de *smartphone*.

A nossa aplicação pode ser discriminada em 2 módulos principais: o reconhecimento de voz e o processamento de linguagem natural. Relativamente ao primeiro módulo, utilizamos o modelo de reconhecimento de voz Whisper da OpenAI, capaz de converter áudio em múltiplas linguagens para texto. O módulo de processamento de linguagem natural faz uso do modelo GPT da OpenAI, que interpreta o texto obtido do módulo do reconhecimento de voz e divide em conteúdo a ser preenchido no formulário.

## **9.2 Produtização**

### **9.2.1 Tecnologia**

A nível de tecnologia existem algumas diferenças entre o protótipo e o produto final. O protótipo utiliza serviços pagos da OpenAI, que foram pagos pela F3M para o desenvolvimento do mesmo. Para se conseguir produzir em massa, existem alterações a ser efetuadas, nomeadamente no serviço de subscrição, isto é, cada produto comercializado tem de possuir a sua própria chave no serviço de subscrição. É, portanto, necessário efetuar certas alterações na passagem do protótipo para o produto a ser comercializado.

### **9.2.2 Mercado**

O produto pode ser posicionado no mercado de uma forma bastante abrangente, pois pode ser aplicado aos mais diversos contextos. Por exemplo, este poderia ser aplicado em áreas como a educação, justiça, serviços públicos e logística. Para além disso, há uma preocupação cada vez maior com aspectos relacionados com a acessibilidade da tecnologia. O produto proposto simplificaria a interação dos utilizadores com deficiências visuais ou problemas de mobilidade com as aplicações que o implementassem.

Embora a procura por soluções de interação por voz em dispositivos móveis esteja a crescer significativamente, a oferta de produtos nesta área permanece limitada. A estratégia para que a biblioteca se destaque no mercado é garantir uma integração profunda com os utilizadores. O objetivo é proporcionar uma experiência tão eficiente e intuitiva que gere uma forte adesão por parte dos utilizadores, promovendo a sua utilização recorrente. Para além de fomentar o envolvimento dos utilizadores, propõe-se um processo contínuo de melhoria da biblioteca. Esta abordagem permitirá oferecer um serviço de qualidade crescente, acompanhado de funcionalidades adicionais que respondam às necessidades em constante evolução dos utilizadores.

O principal trunfo que o produto possui face a outras opções no mercado é a abrangência do mesmo, sendo aplicável em qualquer segmento do mercado, algo que não acontece com o resto das soluções existentes, que são direcionadas para um único segmento. No caso do concorrente *Speechly*, esta é uma solução direcionada para TI.

### 9.2.3 Produto *Mass Market*

O produto *mass market* será comercializado seguindo um modelo SaaS (*Software as a Service*). Neste modelo de comercialização, o *software* é comercializado como um serviço via internet, tal como acontece com a DropBox, SalesForce, entre outros. No âmbito do produto desenvolvido, o objetivo é permitir um número limitado de chamadas à API e caso, posteriormente, os clientes queiram continuar a utilizar o produto, estes devem pagar um serviço de subscrição.

Ainda assim, o objetivo não passa por seguir um modelo SaaS rígido e inflexível. Pelo contrário, o objetivo passa por possuir uma biblioteca base independente de qualquer tipo de cliente e, posteriormente, adaptar o serviço disponibilizado conforme o cliente.

## 9.3 Modelo e Estratégia de Negócio

### 9.3.1 Modelo de Negócio

O modelo de negócio proposto para o projeto é um *Software as a Service* (SaaS). A solução é focada principalmente no setor da saúde, mas também aplicável a outros setores. O produto oferece praticidade, eficiência e integração com sistemas já existentes. A aplicação utiliza tecnologias avançadas de reconhecimento de voz e processamento de linguagem natural para automatizar o registo de informações, reduzindo o tempo gasto em tarefas manuais e melhorando a produtividade.

O modelo de negócio é estruturado em planos de assinatura com o objetivo de proporcionar uma maior flexibilidade e escalabilidade para diferentes tipos de cliente. É de referir que todos os planos pagos terão acesso a um *free trial* de 7 dias. Deste modo, são apresentados os planos de assinatura:

- **Plano básico:** oferece um número de chamadas à API muito limitado, cujo principal objetivo é dar a conhecer a aplicação ao público;
- **Plano avançado:** tem uma maior capacidade em relação ao plano básico, sendo já indicado para clientes que queiram explorar ainda mais as funcionalidades;
- **Plano premium:** destinado a pequenas e médias empresas, que apresenta mais chamadas relativamente ao plano avançado;
- **Plano personalizado:** destinado a empresas grandes, onde a empresa entra em contacto direto com a equipa para definir os termos do plano, podendo existir um acompanhamento personalizado ao cliente.

O modelo de subscrições do produto foi desenvolvido com base em custos operacionais e em comparação com o concorrente *Speechly*, presente no mercado. Estes custos operacionais incluem a API do Whisper, ao custo de \$0,006 por minuto, e da API ChatGPT, com custo de \$0,005 por 1.000 caracteres. Para a precificação, foi considerado um lucro médio de 30%.

Básico	Avançado	Premium	Personalizado
<ul style="list-style-type: none"> <li>• 500 minutos de áudio;</li> <li>• 500000 caracteres de texto.</li> </ul>	<ul style="list-style-type: none"> <li>• 1200 minutos de áudio;</li> <li>• 1.5M caracteres de texto.</li> </ul>	<ul style="list-style-type: none"> <li>• 2500 minutos de áudio;</li> <li>• 4M caracteres de texto.</li> </ul>	<ul style="list-style-type: none"> <li>• Completamente personalizado de acordo com as necessidades do cliente;</li> <li>• Disponível apenas para plafonds superiores aos disponíveis.</li> </ul>
Free trial de 7 dias	Free trial de 7 dias	Free trial de 7 dias	-
8,90€	24,90€	47,90€	Sob consulta

Figura 3: Planos de subscrição.

O produto é distribuído principalmente por meio de uma plataforma *online* onde é possível proceder às assinaturas dos diversos planos referidos anteriormente, assim como contactar o suporte da equipa técnica. A receita do negócio é gerada principalmente através do pagamento recorrente dos planos mensais ou anuais.

A operação do serviço depende de uma infraestrutura *cloud* para garantir alta disponibilidade e desempenho, além de tecnologias disponibilizadas, como os modelos de reconhecimento de voz e processamento de linguagem natural da OpenAI. O desenvolvimento contínuo da solução é essencial para manter a competitividade em relação ao mercado, incluindo melhorias de *performance* e suporte a novas funcionalidades e idiomas. Numa primeira fase, os custos para manter a aplicação disponível estão relacionados essencialmente com utilização das APIs da OpenAI que disponibilizam os modelos anteriormente mencionados. Posteriormente, caso o negócio tivesse o sucesso esperado, teria de ser feito um estudo para averiguar se a utilização de uma infraestrutura própria da empresa para suportar os modelos utilizados seria mais viável financeiramente com o objetivo de reduzir os custos associados às APIs terceiras.

### 9.3.2 Viabilidade Financeira

De forma a garantir a inserção da empresa e do produto no mercado, foi calculado um fundo de maneio necessário para os primeiros 6 meses de operação. Foi considerada uma equipa de 7 engenheiros informáticos a trabalhar 4 horas por dia durante 5 dias por cada semana, a custo de 12€ por hora. Portanto, o investimento em ordenados soma aproximadamente 50.400€ em 6 meses. O modelo de trabalho *full remote* foi o escolhido, de modo a eliminar despesas como aluguer, água, luz, internet, gás, limpeza, entre outros custos associados a uma sede presencial da empresa.

Foi necessário também incluir um saldo inicial de forma a suportar a utilização da API da OpenAI e atender aos possíveis clientes no período inicial. Com base nesses fatores, o fundo de maneio estimado para garantir a operação sem interrupções nos primeiros 6 meses é de 55.000€. Para atingirmos este montante desejado, é importante procurar financiamento através de *business angels* e fundos de capital de risco, algo essencial para o desenvolvimento inicial do produto no mercado. Outra forma possível de obter esse investimento é através de incentivos europeus para a inovação tecnológica. Por fim, conclui-se que a solução é financeiramente viável e apresenta boas perspetivas para atrair investidores, dado que o investimento necessário não é excessivamente elevado.

### **9.3.3 Viabilidade Económica**

Com base nos custos operacionais estimados e no lucro médio inicial de 30%, o produto demonstra viabilidade económica, exigindo entre 228 a 1.227 clientes mensais, dependendo do plano de subscrição escolhido.

Para estudar o alcance das metas financeiras, foram analisadas 3 distribuições de clientes com diferentes focos. Na primeira, o plano básico foi o priorizado, com 1000 clientes, enquanto 350 escolheriam o plano avançado e 150 optariam pelo plano *premium*, sendo necessário um total de 1500 clientes com o lucro total de 9.542€. Na segunda distribuição, o foco foi no plano avançado com 550 clientes, 500 clientes no plano básico e 120 no *premium*, totalizando 1170 clientes e 9.461€ de lucro. Por fim, a distribuição com foco no plano *premium* com 400 clientes, 300 no plano avançado e 200 no plano básico, somando 900 clientes e 9.681€ de lucro.

Considerando que o produto possui uma tecnologia que está cada vez mais em alta, é abrangente e pode atender a diversos setores, bem como pode ser adaptado para versão *web* além da *mobile*, a projeção de alcançar este número de clientes é realista em médio a longo prazo.

### **9.3.4 Estratégia de Entrada no Mercado**

A entrada do produto no mercado deve ser diferenciadora, de modo a conseguir obter uma base de clientes sólida para alavancar o crescimento do negócio. Assim, a estratégia inicial deverá focar-se em setores de mercado com maior necessidade, como saúde e logística.

Primeiramente, serão exploradas parcerias estratégicas com empresas de TI e distribuidoras de *software*, de modo a fazer uma avaliação inicial de produto e obter *feedback*. Da mesma forma, também serão realizados testes piloto com um grupo selecionado de profissionais de saúde para recolha de *feedback* detalhado sobre a usabilidade e eficácia do sistema. Todos estes *feedbacks* recolhidos serão, por sua vez, utilizados para aperfeiçoar o produto antes do lançamento comercial.

Para o produto se manter competitivo no mercado, será adotada uma estratégia inicial de precificação acessível, visando tornar o produto acessível e de fácil integração com as soluções já existentes atualmente. Para além disso, tal como referido anteriormente, a existência de planos com *free trial* permite a utilização do produto de forma gratuita, servindo como uma porta de entrada a novos utilizadores que poderão posteriormente recorrer a um plano pago.

Nesta entrada no mercado, também é importante dar a conhecer o produto ao mercado, porque não é suficiente apenas criar e disponibilizar o produto. Deste modo, é necessário chamar a atenção para o novo produto existente, adotando, por isso, uma estratégia de *marketing* para atrair as atenções. A ideia consiste no planeamento de campanhas de *marketing* digital, focadas nas redes sociais, bem como na participação em conferências e feiras relativas ao setor, por forma a dar a conhecer o produto e atrair potenciais clientes.

### **9.3.5 Estratégia de Crescimento**

Uma das estratégias de crescimento está relacionada com a expansão do negócio para novos mercados a nível geográfico, de modo a abranger utilizadores por todo o mundo. Para além disso, abranger novos setores, que envolvam o preenchimento de formulários, é também um dos objetivos do negócio. Será elaborado um plano de lançamento de novas versões do produto, com inclusão de novas funcionalidades com foco na customização e na experiência do utilizador. A constante

procura pela inovação trará ao produto um caráter atual e acima da concorrência, o que deverá ser responsável por manter clientes atuais e integrar cada vez mais clientes.

Com o aumento do número de clientes e das áreas de aplicação, será fundamental a criação de modelos próprios, ajustados às necessidades específicas de cada cliente (abandonando a subscrição de serviços externos). A transição para uma infraestrutura própria trará um maior controlo das operações e tornará o produto menos dependente de terceiros. Esta transição terá de ser devidamente estudada para averiguar se é rentável financeiramente a nível de custos. A infraestrutura própria implica custos adicionais como a compra de servidores, eletricidade e manutenção dos servidores. Para além disso, teria de ser efetuado um estudo para averiguar a melhor forma de balancear a carga dos servidores, para rentabilizar ao máximo o uso dos mesmos. Teriam de ser estudadas também técnicas de otimização dos custos dos servidores, como por exemplo averiguar as alturas do dia em que existe maior sobrecarga dos mesmos, permitindo assim desligar os servidores nesses momentos.

## 10 Conclusão

Em resumo, neste trabalho abordamos o problema do preenchimento de formulários por voz na área médica. O trabalho foi realizado em conjunto com a empresa F3M e integrou diversas fases, desde a investigação do problema em mãos até ao teste da solução final e à exploração de estratégias para viabilizar a sua comercialização. Como efeito secundário, a equipa teve a oportunidade de experimentar novas tecnologias emergentes como *Speech-Recognition* e *Natural Language Processing*. Os objetivos definidos foram validados pelo cliente (a empresa F3M), que vê utilidade no produto desenvolvido. Para além disso, o produto é suficientemente generalizável e viável financeiramente, podendo ser aplicado facilmente noutras contextos. Estamos, por isso, bastante satisfeitos com o resultado final alcançado.