IPV-ESTGL 2026

Desenvolvimento de um *Serious Game* para Treino de Pronúncia em Português Europeu

Simão Pinto Nascimento

Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Lamego

DESENVOLVIMENTO DE UM *SERIOUS GAME* PARA TREINO DE PRONÚNCIA EM PORTUGUÊS EUROPEU

Simão Pinto Nascimento

Janeiro, 2026

IPV-ESTGL 2026

Desenvolvimento de um *Serious Game* para Treino de Pronúncia em Português Europeu

Simão Pinto Nascimento

Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Lamego

DESENVOLVIMENTO DE UM *SERIOUS GAME* PARA TREINO DE PRONÚNCIA EM PORTUGUÊS EUROPEU

Simão Pinto Nascimento

Trabalho de Projeto
Mestrado de Tecnologias de Informação e Automação

Trabalho efetuado sob a orientação de
Professor Doutor Armando Cruz

Janeiro, 2026

Politécnico de Viseu
Tecnologia e Gestão Lamego

DESENVOLVIMENTO DE UM *SERIOUS GAME* PARA TREINO DE PRONÚNCIA EM PORTUGUÊS EUROPEU

Simão Pinto Nascimento

Trabalho de Projeto

Mestrado de Tecnologias de Informação e Automação

Trabalho efetuado sob a orientação de

Professor Doutor Armando Cruz

Lamego, 2026

AGRADECIMENTOS

A conclusão deste projeto representa o culminar de um percurso académico e pessoal que não teria sido possível sem o apoio de várias pessoas e instituições, às quais expresso, aqui, o meu sincero agradecimento.

Em primeiro lugar, agradeço ao meu orientador, **Armando Cruz**, pela disponibilidade, rigor e orientação ao longo de todas as fases deste trabalho. As sugestões, críticas construtivas e o equilíbrio entre exigência e confiança foram fundamentais para a concretização deste projeto.

À **Escola Superior de Tecnologia e Gestão de Lamego** e ao corpo docente do Mestrado em Tecnologias de Informação e Automação, agradeço os conhecimentos transmitidos, o acompanhamento prestado e o ambiente de trabalho que tornaram possível o desenvolvimento das competências necessárias para chegar a esta etapa.

Agradeço também aos colegas e amigos de curso, em especial **ao Rodrigo Magueja, ao Cristiano Santos e ao Pedro Saavedra**, pelo companheirismo, pela entreajuda e pelas conversas que tantas vezes ajudaram a clarificar ideias, encontrar soluções ou simplesmente a ganhar ânimo nos momentos mais exigentes.

Um agradecimento muito especial é devido à minha família. Aos meus pais, pela confiança incondicional, pelo apoio ao longo de todo o percurso académico e por nunca deixarem de acreditar em mim.

Não posso deixar de agradecer às pessoas que participaram, direta ou indiretamente, nos testes e na utilização da aplicação desenvolvida. A disponibilidade para experimentar o protótipo, dar opinião e apontar limitações contribuiu de forma decisiva para a melhoria da solução apresentada.

Por fim, agradeço a todos aqueles que, de forma menos visível, contribuíram para que este caminho fosse possível — seja através de uma palavra de motivação, de um conselho oportuno ou simplesmente pela presença nos momentos certos.

A todos, o meu sincero obrigado.

RESUMO

A pronúncia é um dos componentes mais críticos da competência oral, podendo condicionar a inteligibilidade da fala mesmo quando o vocabulário e a gramática são adequados. No caso do Português Europeu, a oferta de ferramentas específicas para treino de pronúncia com feedback automatizado e registo sistemático do desempenho é ainda limitada. Este relatório descreve o desenvolvimento e a avaliação de um protótipo de *serious game* para treino de pronúncia em Português Europeu, concebido para funcionar em modo offline e apoiar tanto a prática autónoma como a integração em contextos educativos.

O trabalho segue uma metodologia de natureza aplicada, combinando engenharia de software e prototipagem iterativa. O protótipo integra uma interface gráfica desenvolvida em Python com Pygame, um módulo de reconhecimento automático de fala baseado no modelo Whisper, um conjunto de frases organizadas em três níveis de dificuldade e um sistema de registo de dados suportado em ficheiros JSON e na geração de relatórios PDF. Em cada tentativa, o utilizador ouve uma frase em Português Europeu, repete-a e obtém uma transcrição automática, uma percentagem de acerto calculada por palavra e a identificação de determinados padrões de erro fonético.

A avaliação do protótipo, realizada em contexto de prova de conceito, mostra que o sistema funciona de forma estável, gera métricas de desempenho interpretáveis e produz relatórios estruturados que permitem acompanhar a evolução ao longo do tempo. São igualmente discutidas limitações importantes, relacionadas com a escala e diversidade reduzidas dos dados, a dependência de um modelo ASR genérico e a cobertura parcial dos fenómenos fonéticos, apontando-se linhas de trabalho futuro para aprofundar a análise fonética, especializar o reconhecimento de fala e alargar o estudo a populações de utilizadores mais variadas.

Palavras-chave: *serious games*; treino de pronúncia; Português Europeu; reconhecimento automático de fala; Whisper; Computer-Assisted Pronunciation Training.

ABSTRACT

Pronunciation is a critical component of oral proficiency and can strongly affect speech intelligibility, even when vocabulary and grammar are appropriate. In the case of European Portuguese, there is still a limited number of dedicated tools for pronunciation training that provide automatic feedback and systematic tracking of learner performance. This report presents the development and evaluation of a prototype *serious game* for pronunciation training in European Portuguese, designed to run offline and to support both individual practice and use in educational contexts.

The work follows an applied research methodology, combining software engineering with iterative prototyping. The prototype integrates a graphical interface developed in Python with Pygame, an automatic speech recognition module based on the Whisper model, a set of sentences organised into three difficulty levels, and a data logging subsystem based on JSON files and PDF report generation. In each attempt, the user listens to a reference sentence in European Portuguese, repeats it, and receives an automatic transcription, a word-level accuracy score and the identification of specific phonetic error patterns.

The proof-of-concept evaluation, based on a small exploratory dataset, indicates that the system operates stably, produces interpretable performance metrics and generates structured reports that enable monitoring of progress over time. Important limitations are also discussed, namely the limited scale and diversity of the data, the dependence on a generic ASR model and the partial coverage of phonetic phenomena. Future work directions include refining the phonetic analysis, specialising the speech recognition component and extending the empirical study to more diverse groups of users.

Keywords: *serious games*; pronunciation training; European Portuguese; automatic speech recognition; Whisper; Computer-Assisted Pronunciation Training.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE ABREVIATURAS

AFI – Alfabeto Fonético Internacional (International Phonetic Alphabet)

ASR – Reconhecimento Automático da Fala (Automatic Speech Recognition)

CALL – Aprendizagem de Línguas Assistida por Computador (Computer-Assisted Language Learning)

CAPT – Treino de Pronúncia Assistido por Computador (Computer-Assisted Pronunciation Training)

CSV – *Comma-Separated Values* (formato de ficheiro de texto com valores separados por vírgulas)

JSON – *JavaScript Object Notation* (formato de intercâmbio de dados em texto estruturado)

L2 – Língua segunda / língua alvo de aprendizagem

MALL – Aprendizagem de Línguas Assistida por Dispositivos Móveis (Mobile-Assisted Language Learning)

PE – Português Europeu

PDF – *Portable Document Format* (formato de documento portátil)

RGPD – Regulamento Geral sobre a Proteção de Dados

SDL – *Simple DirectMedia Layer* (biblioteca multimédia subjacente ao Pygame)

WER – *Word Error Rate* (taxa de erro de palavra)

INTRODUÇÃO

A aprendizagem de línguas mediada por tecnologia tem vindo a assumir um papel central num contexto social cada vez mais digital e globalizado. Entre as diferentes dimensões da competência comunicativa, a pronúncia destaca-se como um fator crítico para a inteligibilidade da fala e para a perceção de fluência por parte de interlocutores nativos ou proficientes. Vários autores sublinham que, mesmo quando o vocabulário e a gramática são adequados, uma pronúncia pouco clara pode comprometer a comunicação e limitar a participação em contextos académicos, profissionais e sociais (Celce-Murcia, Brinton & Goodwin, 2010; Derwing & Munro, 2015; Levis, 2005).

Na prática, muitos aprendentes têm acesso sobretudo a materiais focados em estrutura frásica e léxico, enquanto a componente fonética fica muitas vezes dependente de apoio presencial especializado ou de exposição informal à língua. Em contextos em que o acompanhamento por docentes ou terapeutas da fala não é facilmente acessível — seja por questões geográficas, económicas ou de horário — a pronúncia tende a ser trabalhada de forma pouco sistemática, sem feedback imediato e sem registo objetivo de progresso (Derwing & Munro, 2015).

Paralelamente, os *serious games* têm ganho espaço como abordagem promissora na educação, ao combinarem elementos típicos de jogo (objetivos, níveis, pontuações, recompensas) com metas pedagógicas explícitas. Diversos estudos apontam que este tipo de ambientes interativos pode aumentar a motivação, a persistência e o envolvimento dos utilizadores, enquanto promove prática repetida e estruturada de determinados conteúdos (Johnson, 2005; Wouters et al., 2013). No domínio das línguas estrangeiras, os *serious games* têm sido explorados para trabalhar vocabulário, gramática, compreensão auditiva e, mais recentemente, competências fonéticas, através de mecanismos de feedback imediato e de tarefas orais contextualizadas (Lago-Ferreiro et al., 2025; Sadigzade, 2025).

A integração de tecnologias de reconhecimento automático de fala em ambientes de aprendizagem cria condições para transformar a prática oral numa atividade mais monitorizada e mensurável. Em particular, a área de Computer-Assisted Pronunciation

Training (CAPT) tem vindo a demonstrar que é possível utilizar reconhecimento de fala para fornecer pistas sobre o desempenho do utilizador, melhorando a consciência fonética e a precisão articulatória ao longo do tempo (Neri, Cucchiarini, Strik & Boves, 2002; Amrate et al., 2023). No entanto, muitos dos sistemas descritos na literatura estão orientados para o inglês e para outros idiomas com maior peso global, permanecendo o Português Europeu relativamente menos explorado neste tipo de soluções.

É neste enquadramento que se insere o presente trabalho, desenvolvido no âmbito do Mestrado em Tecnologias de Informação e Automação da Escola Superior de Tecnologia e Gestão de Lamego. O projeto consiste no desenvolvimento de um *serious game* para treino de pronúncia em Português Europeu, recorrendo a tecnologias de reconhecimento de fala e a uma interface gráfica interativa. Pretende-se proporcionar ao utilizador um ambiente de prática orientada, com feedback imediato e registo de progresso ao longo do tempo, funcionando em modo offline para reduzir dependências de conectividade e aumentar a portabilidade em contextos educativos com infraestruturas heterogéneas.

A solução desenvolvida integra vários componentes tecnológicos: um motor de jogo implementado em Python, com recurso à biblioteca Pygame; um módulo de reconhecimento de fala baseado no modelo Whisper; um conjunto de áudios pré-gerados organizados por níveis de dificuldade; e um subsistema de recolha e análise de resultados suportado em ficheiros JSON e na geração de relatórios em formato PDF. O jogo está estruturado em três níveis de dificuldade (Fácil, Médio e Difícil), cada um com um conjunto de frases selecionadas para cobrir diferentes estruturas fonéticas e alguns padrões de erro típicos de aprendentes. Em cada sessão, o utilizador ouve uma frase em Português Europeu, repete-a e a aplicação procede à transcrição automática, à comparação com a frase de referência e ao cálculo de métricas de desempenho.

Para além de atribuir uma percentagem global de acerto, o sistema identifica padrões de dificuldades fonéticas, como redução da nasalização, trocas de consoantes (por exemplo, R→L), substituição de sons complexos (como CH→S) e omissão de consoantes finais. Estes padrões são registados de forma estruturada, permitindo alimentar o feedback apresentado ao utilizador e suportar análises agregadas de evolução ao longo do tempo. Os dados de cada sessão são armazenados num ficheiro de resultados, onde constam a frase esperada, a frase dita, o nível, a data, a percentagem de acerto e os padrões detetados.

Um dos contributos centrais deste trabalho consiste na capacidade de transformar esses registos em relatórios PDF completos, gerados automaticamente e utilizáveis em contexto offline. Foram implementados relatórios por nível e um relatório geral, que incluem um enquadramento textual sintético, indicadores estatísticos (por exemplo, média de acertos por nível), visualizações de evolução, tabelas com as sessões mais recentes e uma análise descritiva dos padrões fonéticos predominantes. Estes relatórios destinam-se tanto ao utilizador final como a docentes ou terapeutas da fala que pretendam acompanhar, de forma sistemática, o percurso de um aprendente.

Do ponto de vista científico e tecnológico, este trabalho procura responder a duas necessidades principais. Em primeiro lugar, a escassez de ferramentas dirigidas especificamente ao treino da pronúncia de Português Europeu com feedback automático baseado em reconhecimento de fala. Em segundo lugar, a necessidade de articular, de forma coerente, técnicas de reconhecimento de fala, princípios de design de jogos sérios e mecanismos de recolha e tratamento de dados, respeitando simultaneamente preocupações de ergonomia, usabilidade e privacidade da informação (Chapelle, 2009; Cucchiarini, 2017).

De forma sintética, estas lacunas podem ser condensadas na seguinte questão central:

É possível conceber e implementar um *serious game* offline para treino de pronúncia em Português Europeu que, combinando reconhecimento automático de fala, identificação de padrões de erro fonético e geração de relatórios estruturados de desempenho, apoie a prática autónoma e o acompanhamento sistemático da evolução do utilizador?

Para responder a esta questão, o objetivo geral deste projeto consiste em conceber, desenvolver e avaliar, em contexto de prova de conceito, um *serious game* para treino de pronúncia em Português Europeu que integre um motor de reconhecimento de fala, um módulo de análise de padrões de erro e um sistema de relatórios de desempenho. No Capítulo II apresentam-se em detalhe este objetivo geral e os objetivos específicos em que se desdobra, bem como a metodologia adotada para a sua concretização.

A metodologia adotada assume uma natureza essencialmente aplicada, combinando engenharia de software e prototipagem iterativa. O desenvolvimento da aplicação foi

orientado por ciclos sucessivos de conceção, implementação e teste funcional, com o objetivo de garantir estabilidade, tempos de resposta adequados em ambiente local e uma experiência de utilização coerente. Os dados gerados pelo jogo (sob a forma de registos de sessões, percentagens de acerto e padrões fonéticos) constituem também a base para a análise de métricas e para a discussão crítica da solução proposta.

A estrutura do relatório encontra-se organizada da seguinte forma. Após a presente Introdução, o capítulo seguinte apresenta o estado da arte, abordando conceitos de *serious games*, aprendizagem de línguas assistida por computador, tecnologias de reconhecimento de fala e trabalhos relacionados com treino de pronúncia.

Num capítulo dedicado à metodologia e ao enquadramento do desenvolvimento descrevem-se os procedimentos de recolha de requisitos, as principais opções tecnológicas e a lógica de prototipagem adotada. Seguidamente, apresenta-se a arquitetura da solução e detalham-se os principais módulos implementados, incluindo a interface de jogo, o motor de avaliação e o sistema de relatórios.

O capítulo de avaliação discute as métricas definidas, analisa os dados recolhidos e reflete sobre as potencialidades e limitações do protótipo. Por fim, o relatório encerra com as conclusões e a proposta de trabalho futuro, apontando caminhos para a evolução da aplicação e para investigações subsequentes na área do treino de pronúncia em Português Europeu suportado por jogos sérios.

CAPÍTULO I – ESTADO DA ARTE

I.1 - *Serious Games*: conceitos, evolução e aplicabilidade educativa

De forma simples, fala-se em *serious games* quando um jogo é desenhado com um propósito principal diferente do entretenimento, como formação, treino, educação ou sensibilização. Michael e Chen descrevem estes jogos como **experiências em que a educação, nas suas várias formas, é o objetivo central**, e não apenas o "divertimento pelo divertimento" (Michael & Chen, 2006). Ao conjugar regras, objetivos, feedback e narrativa com uma intenção pedagógica clara, o *serious game* passa a ser uma ferramenta de aprendizagem e não apenas um passatempo digital.

Nas últimas décadas, o conceito evoluiu bastante. A literatura mostra uma aplicação crescente dos *serious games* em domínios como a saúde, a defesa, a reabilitação motora, a formação profissional ou a educação formal e informal (Michael & Chen, 2006). Em contexto educativo, destacam-se sobretudo **três ideias** recorrentes na literatura (Johnson, 2005; Wouters et al., 2013):

- o jogo facilita o envolvimento emocional e cognitivo;
- permite repetir tarefas de forma estruturada, sem se tornar monótono tão depressa como uma ficha de exercícios tradicional;
- oferece feedback imediato, o que ajuda o utilizador a perceber rapidamente se está a melhorar ou não.

Revisões sistemáticas indicam que, quando bem desenhados, os *serious games* **podem ter impacto positivo na motivação, no envolvimento e, em muitos casos, em resultados de aprendizagem mensuráveis** (Wouters et al., 2013). No entanto, vários autores alertam que nem todo o "jogo educativo" é automaticamente eficaz. A qualidade do design pedagógico, a clareza dos objetivos e a forma como o feedback é apresentado fazem a

diferença entre um jogo interessante e uma verdadeira ferramenta de apoio à aprendizagem (Johnson, 2005).

Na aprendizagem de línguas, os *serious games* têm sido usados para trabalhar vocabulário, gramática, compreensão oral e produção escrita, mas também componentes mais específicas, como a pronúncia e a prosódia. Estudos recentes mostram que **ambientes de jogo podem apoiar tarefas orais contextualizadas**, recorrendo a desafios progressivos, recompensas e feedback imediato para sustentar a prática (Lago-Ferreiro et al., 2025; Sadigzade, 2025). A possibilidade de criar cenários de comunicação simulados, com personagens, objetivos e desafios progressivos, ajuda o aprendente a praticar num ambiente mais próximo de situações reais, mas com risco reduzido: pode errar, recomeçar e receber feedback, sem a pressão de estar a falar "ao vivo" com um nativo.

I.2 - Aprendizagem de Línguas Assistida por Computador

A sigla CALL (*Computer-Assisted Language Learning*) é hoje usada de forma relativamente ampla para designar o campo que estuda e explora aplicações do computador no ensino e na aprendizagem de línguas. Uma definição frequentemente citada, proposta por Levy, caracteriza CALL como a "procura e estudo das aplicações do computador no ensino e aprendizagem de línguas", sublinhando que o foco não está apenas nas ferramentas em si, mas também na forma como são concebidas, utilizadas e avaliadas (Levy, 1997).

Historicamente, costuma descrever-se a evolução de CALL em três grandes fases: **CALL behaviorista**, centrado em exercícios de repetição e reforço (drill-and-practice) inspirados em abordagens estruturalistas; **CALL comunicativo**, que procura promover uso mais significativo da língua, com ênfase em tarefas e atividades comunicativas; e **CALL integrativo**, que integra multimédia e comunicação em rede em ambientes mais ricos e flexíveis (Warschauer, 1996; Zhytska, 2012). Esta divisão em fases não é consensual, mas ajuda a perceber a passagem de usos predominantemente mecanicistas para abordagens mais centradas na interação, no significado e na integração de múltiplas tecnologias.

Atualmente, quando se fala em CALL, o conceito abrange um leque muito diversificado de tecnologias e contextos: desde software instalado localmente até plataformas *web*; desde exercícios escritos até ambientes de comunicação síncrona e assíncrona; desde *drill-and-*

practice simples até mundos virtuais, corpora linguísticos, aprendizagem móvel (*Mobile-Assisted Language Learning*, MALL) e sistemas de tutor inteligente (Davies, 2003; Levy, 1997). O computador pode funcionar como tutor (apresentando e avaliando atividades), como ferramenta (por exemplo, para consulta de dicionários) ou como meio de comunicação entre aprendentes e falantes nativos.

Uma tendência importante na literatura recente é a deslocação do foco da tecnologia para a **pedagogia**. Em vez de discutir apenas "que software usar", discute-se em que medida uma determinada solução tecnológica cria condições favoráveis à aprendizagem: oportunidade de interação significativa, atenção à forma e ao significado, feedback útil, envolvimento ativo, entre outros aspetos. Obras como *CALL Environments* e *CALL Essentials* defendem que ambientes CALL eficazes são aqueles que conseguem alinhar princípios de aquisição de segunda língua com as potencialidades específicas das tecnologias usadas (Egbert & Hanson-Smith, 1999; Egbert, 2005).

Neste contexto, ganha relevância a questão de **como avaliar tarefas e materiais CALL**. Chapelle propõe um conjunto de critérios para avaliar a adequação de tarefas mediadas por computador, incluindo o seu potencial de aprendizagem linguística, o ajustamento ao perfil do aprendente, o foco no significado, o grau de autenticidade, o impacto (motivações, atitudes) e a viabilidade prática (Chapelle, 2001). Estes critérios lembram que não basta que uma atividade seja "interativa" ou "multimédia": é necessário que **ofereça oportunidades reais de aprendizagem e que se integre de forma coerente** em percursos didáticos mais amplos.

No caso específico da aprendizagem de línguas, CALL tem sido aplicado a uma grande variedade de competências: leitura, escrita, gramática, vocabulário, compreensão oral, interação síncrona e assíncrona, entre outras (Warschauer, 1996; Levy, 1997). A área do treino de pronúncia e da prosódia (CAPT) pode ser vista como um subcampo de CALL, em que se acrescenta uma camada de processamento de sinal e de análise fonética à lógica geral de conceção de tarefas e de feedback.

O protótipo desenvolvido neste trabalho insere-se precisamente nesta interseção: utiliza reconhecimento automático de fala e análise fonética para fornecer feedback imediato sobre

a produção oral, mas continua, em essência, a ser um sistema CALL, na medida em que organiza tarefas, gere conteúdos, apresenta feedback e regista a atividade do utilizador com objetivos de aprendizagem.

I.3 - Reconhecimento Automático da Fala: princípios e evolução

O reconhecimento automático da fala (*Automatic Speech Recognition*, ASR) designa o conjunto de técnicas que permitem a um sistema computacional transformar sinais de fala em texto. Em termos gerais, um sistema de ASR recebe como entrada um sinal de áudio, extrai descritores acústicos relevantes e, com base num modelo treinado previamente, produz uma sequência de palavras que procura corresponder ao enunciado original (Malik et al., 2021). Nos últimos anos, os avanços em aprendizagem profunda (*deep learning*) e a disponibilidade de grandes volumes de dados anotados transformaram profundamente este campo, permitindo **ganhos significativos de robustez e de desempenho** em contextos reais de utilização (Ahlawat et al., 2025).

De forma simplificada, muitos sistemas modernos de ASR seguem uma arquitetura de ponta a ponta. Em vez de separar rigidamente modelos acústicos, léxicos e linguísticos, uma única rede neuronal aprende a mapear diretamente o áudio para símbolos linguísticos (Ahlawat et al., 2025; Bhat & Bhattacharyya, 2023). Arquiteturas baseadas em *Transformers* e mecanismos de atenção tornaram-se particularmente relevantes, uma vez que conseguem modelar dependências de longo alcance na sequência de entrada e de saída, contribuindo para melhorias na taxa de erro de palavra (*Word Error Rate*, WER) em diversos idiomas e domínios (Ahlawat et al., 2025).

Em paralelo com estes desenvolvimentos técnicos, a literatura tem sublinhado a crescente presença de ASR em aplicações de uso quotidiano e em contextos educativos: assistentes de voz, legendagem automática, acessibilidade para utilizadores com dificuldades motoras ou visuais, apoio à escrita e, de forma muito relevante para este trabalho, aprendizagem de línguas (Shadiev & Yang, 2023). Revisões recentes mostram que tecnologias de reconhecimento de fala são usadas para apoiar tarefas de compreensão oral, para ditado e, com particular destaque, para treino de pronúncia, precisamente porque **permitem oferecer *feedback* imediato** baseado na comparação entre o que o aprendente disse e uma forma de referência (Shadiev & Yang, 2023; Mohsen et al., 2025).

No domínio específico do treino de pronúncia (CAPT), a integração de ASR permite automatizar a avaliação da produção oral, produzindo medidas de acerto e, em alguns casos, indicadores mais finos relacionados com segmentos ou padrões fonéticos (Rogerson-Revell, 2021). Revisões sistemáticas apontam para um **crescimento contínuo de trabalhos que combinam ASR com CAPT**, explorando diferentes formas de apresentar o *feedback* (numérico, textual, visual) e de o articular com objetivos pedagógicos, embora também identifiquem desafios importantes, como a sensibilidade a sotaques, ruído ou variação prosódica (Amrate & Tsai, 2024; Mohsen et al., 2025).

A escolha do motor de reconhecimento é, por isso, um elemento crítico em qualquer sistema que utilize ASR para fins educativos. Entre as soluções recentemente propostas, destaca-se o modelo Whisper, desenvolvido pela OpenAI, que consiste num sistema de reconhecimento de fala de ponta a ponta baseado em *Transformers*, treinado em aproximadamente 680 000 horas de áudio multilingue e multitarefa recolhido da *web* (Radford et al., 2022). Este modelo suporta não só transcrição em múltiplos idiomas, como também tradução de fala para inglês e identificação automática do idioma, apresentando níveis de robustez interessantes perante sotaques, ruído e variação de domínio. Estudos recentes que avaliam Whisper em diferentes cenários confirmam o seu potencial enquanto sistema genérico de ASR, embora apontem também **limitações e variação de desempenho consoante a língua** e o tipo de tarefa (Graham & Roll, 2024).

No contexto deste trabalho, Whisper é utilizado como componente central de reconhecimento de fala: a partir do áudio gravado pelo utilizador, o modelo produz uma transcrição textual que serve de base ao cálculo de percentagens de acerto e à identificação de padrões de erro fonético. O facto de se tratar de um modelo multilingue e relativamente robusto a ruído facilita a sua utilização em ambiente não controlado, ainda que a ausência de adaptação específica ao Português Europeu imponha limitações que serão discutidas na avaliação do protótipo. Este compromisso entre disponibilidade de um modelo pré-treinado de grande escala e ausência de *fine-tuning* para o contexto exato de utilização é típico de muitos projetos de prototipagem com ASR e deve ser tido em conta na interpretação dos resultados.

I.4 - Treino de Pronúncia e Análise Fonética em Ambientes Digitais

A pronúncia é um dos componentes mais visíveis – e muitas vezes mais sensíveis – da proficiência oral numa língua segunda. A literatura tem sublinhado, contudo, que o objetivo principal do ensino da pronúncia não deve ser a eliminação total do sotaque, mas antes a **promoção da inteligibilidade e da compreensibilidade**, isto é, a capacidade de o falante ser percebido e compreendido com esforço razoável pelo interlocutor (Derwing & Munro, 2005; Munro & Derwing, 2011). Esta mudança de foco, do "parecer nativo" para "ser entendível", tem implicações diretas na forma como se selecionam prioridades pedagógicas e se avaliam melhorias.

Revisões de investigação mostram que a instrução de pronúncia pode ter efeitos positivos significativos, embora os resultados variem consoante os objetivos, a duração das intervenções, os perfis dos aprendentes e as formas de avaliação usadas. Thomson e Derwing, numa revisão narrativa de 75 estudos, referem que uma larga maioria dos trabalhos reporta melhorias após intervenção explícita, mas também salientam a **necessidade de mais atenção a medidas de inteligibilidade e compreensibilidade**, para além de exames de precisão segmental isolada (Thomson & Derwing, 2015).

Em síntese, existe hoje evidência de que vale a pena ensinar pronúncia, mas continua a ser necessário alinhar melhor as práticas pedagógicas com os objetivos comunicativos e com o que se sabe sobre aquisição fonológica.

É neste contexto que se insere o subcampo de **Computer-Assisted Pronunciation Training (CAPT)**, geralmente entendido como a área de CALL dedicada a sistemas que apoiam o treino de pronúncia recorrendo a tecnologias de processamento de sinal, reconhecimento de fala e visualizações específicas (Vančová, 2023). Rogerson-Revell caracteriza CAPT como um conjunto de recursos e ambientes que exploram as potencialidades da tecnologia para **proporcionar prática intensiva, *feedback* imediato** e, idealmente, **representações visuais ou auditivas** que ajudem o aprendente a perceber a diferença entre a sua produção e um modelo-alvo (Rogerson-Revell, 2021). A promessa pedagógica de CAPT reside precisamente nesta capacidade de oferecer **prática individualizada, repetível e com retorno rápido**, complementando e não substituindo o trabalho realizado com docentes humanos (Pennington & Rogerson-Revell, 2019).

Nos últimos anos, têm surgido diversas revisões e estudos de síntese que procuram mapear o estado da arte em CAPT. Um estudo recente, baseado na análise de 403 resumos de artigos de investigação, mostra que se trata de um domínio em crescimento contínuo, com diversidade de línguas, populações e tecnologias, mas também com concentrações claras em certos contextos (por exemplo, inglês como L2) e em determinados tipos de tarefas (Mohsen et al., 2025). De forma complementar, uma revisão sistemática de estudos empíricos publicada em 2024 conclui que, em geral, intervenções baseadas em CAPT **tendem a produzir melhorias mensuráveis na pronúncia em comparação com grupos de controlo**, embora com variação de efeitos e com metodologias de avaliação nem sempre comparáveis (Amrate & Tsai, 2024).

Estudos empíricos que avaliam ferramentas CAPT específicas têm mostrado **melhorias mensuráveis na pronúncia** (por exemplo, Tejedor-García et al., 2020; Korzekwa et al., 2022), reforçando o potencial destas soluções.

De forma consistente, estes estudos apontam para o potencial de CAPT em apoiar o treino de aspetos segmentais (sons individuais, pares mínimos) e suprassegmentais (acentuação, ritmo, entoação), desde que as tarefas e o *feedback* sejam desenhados com cuidado pedagógico.

Ao mesmo tempo, a própria literatura de CAPT reconhece limitações e desafios. Entre os mais referidos contam-se a sensibilidade dos sistemas de reconhecimento a sotaques fortes, ruído e variação prosódica, a possibilidade de *feedback* enganador devido a classificações automáticas imprecisas, a tendência para focar sobretudo traços segmentais de baixa carga funcional e, em alguns casos, o risco de reforçar modelos de pronúncia excessivamente normativos ou "nativistas" (Rogerson-Revell, 2021; Derwing & Munro, 2005; Korzekwa et al., 2022). Revisões recentes recomendam, por isso, que sistemas CAPT sejam concebidos de forma a **privilegiar a inteligibilidade, a transparência do *feedback* e a possibilidade de integração em percursos mais amplos de ensino**, em vez de serem apresentados como soluções autónomas e infalíveis (Amrate & Tsai, 2024; Mohsen et al., 2025).

No caso específico do Português Europeu, a oferta de ferramentas CAPT com foco explícito em treino de pronúncia continua a ser **relativamente limitada** quando comparada com o que existe para inglês e outras línguas com maior peso no mercado de aprendizagem. A maioria das plataformas comerciais e de investigação concentra-se em inglês como L2, e quando incluem português, fazem-no muitas vezes sem distinção clara entre variedades europeia e brasileira. Nesta perspetiva, o protótipo desenvolvido neste trabalho situa-se numa área ainda pouco explorada: procura combinar princípios de CAPT com um foco explícito em padrões fonéticos do Português Europeu — por exemplo, nasalização, consoantes fricativas palato-alveolares ou realização de /ʁ/ e /ɾ/ —, aliando um motor de reconhecimento de fala genérico a mecanismos de análise e de visualização de *feedback* especialmente adaptados a estes fenómenos.

I.4.1 - Sons do Português Europeu relevantes para a análise fonética

A análise fonética usada neste trabalho não pretende cobrir toda a complexidade do sistema fonológico do Português Europeu, mas precisa de se apoiar em alguns contrastes fundamentais para que os padrões de erro registados façam sentido. Para isso, recorre-se à terminologia do Alfabeto Fonético Internacional (AFI), amplamente utilizado na descrição científica dos sons das línguas.

Um dos contrastes mais relevantes é o das consoantes róticas, habitualmente designadas como "R forte" e "r suave". O "R forte", frequentemente representado por /ʁ/, ocorre em posição inicial (por exemplo, *rato*, *rua*) ou em certas posições intervocálicas e finais, sendo produzido na parte posterior da cavidade oral. O "r suave", representado por /ɾ/, aparece tipicamente entre vogais (por exemplo, *cara*, *muro*) e corresponde a um toque rápido da língua na zona alveolar. A distinção entre estes dois sons é sistemática em Português Europeu e constitui um desafio frequente para aprendentes, o que justifica que o jogo registe trocas entre /ʁ/ e /ɾ/ como um padrão específico.

Outro conjunto importante de contrastes diz respeito aos fricativos. Em Português Europeu, distinguem-se fricativos como /s/ (como em *sapo*), /z/ (como em *casa*), /ʃ/ (o som típico de "ch", em *chave*) e /ʒ/ (som de "j", em *janela*). Trabalhos em fonética acústica mostram que estes sons, apesar de semelhantes, apresentam diferenças consistentes de lugar

de articulação, sonoridade e distribuição em contexto silábico. Para muitos aprendentes, a proximidade acústica destes fricativos leva a confusões sistemáticas, como produzir /s/ em vez de /ʃ/, ou não distinguir claramente /s/ de /z/.

Ao integrar estes contrastes na definição dos padrões de erro (por exemplo, registar substituições entre fricativos ou dificuldades recorrentes nas róticas), o *serious game* não se limita a dizer ao utilizador que a frase está "certa" ou "errada": **fornece uma leitura mais fina sobre *que tipo* de erro é que está a acontecer** e com que frequência, aproximando a análise automática de uma preocupação fonética mais próxima da prática de especialistas em Português Europeu.

I.5 - Jogos Educativos Desenvolvidos com Python e Pygame

Na parte tecnológica, este projeto assenta em Python e na biblioteca Pygame. Python é hoje uma das linguagens mais usadas tanto na indústria como na educação, tendo surgido de forma consistente nos primeiros lugares de rankings de popularidade e sendo amplamente adotada em unidades curriculares introdutórias de programação, em parte devido à sua **sintaxe simples, legibilidade e forte ecossistema de bibliotecas** (Dierbach, 2012; Kruglyk & Lvov, 2012; IEEE Spectrum, 2024). Esta popularidade facilita a manutenção e a extensão de protótipos como o jogo desenvolvido neste relatório, na medida em que reduz a barreira de entrada para outros programadores e se integra bem com ferramentas de análise de dados e de aprendizagem automática.

A biblioteca Pygame é uma camada em Python sobre a biblioteca SDL (*Simple DirectMedia Layer*), concebida especificamente para criar jogos e aplicações multimédia 2D. Fornece suporte para janelas, eventos, gráficos, som e gestão de *sprites*, funcionando em múltiplas plataformas e permitindo escrever jogos que correm de forma semelhante em diferentes sistemas operativos (Pygame, s.d.; Real Python, 2019). O facto de ser uma solução de código aberto e relativamente leve, sem impor um motor gráfico complexo, torna-a particularmente **adequada para projetos educativos e protótipos académicos** em que se pretende controlo direto sobre a lógica de jogo.

Há exemplos, tanto na literatura como na comunidade educativa, de jogos desenvolvidos com Python e Pygame para fins formativos, seja para ensinar matemática, conceitos de programação ou outras áreas STEM. Repositórios públicos e iniciativas de ensino de programação com jogos mostram a utilização de Pygame e de variantes como Pygame Zero em oficinas para crianças e jovens, em clubes de programação e em cursos introdutórios, precisamente pela combinação entre simplicidade de código e riqueza visual (Pygame, s.d.; Electronstudio, 2023; Codingal, 2025). Estes casos de estudo sugerem que Pygame é suficientemente **flexível para suportar jogos educativos relativamente completos**, mantendo uma curva de aprendizagem aceitável para quem já está familiarizado com o ecossistema Python.

Neste projeto, a combinação Python + Pygame + ASR (Whisper) permite implementar, no mesmo ambiente, a interface gráfica do jogo, a reprodução de estímulos auditivos, a captura da fala, a análise automática da pronúncia e o registo dos resultados. O facto de tudo funcionar localmente, sem necessidade de ligação contínua à internet, reforça também a adequação a contextos educativos com ligações instáveis ou restrições de acesso, enquanto facilita a preservação da privacidade dos dados de voz processados pelo sistema.

I.6 - Trabalhos Relacionados e Lacunas Identificadas

Existem atualmente várias aplicações móveis e plataformas web dedicadas ao treino de pronúncia em línguas como inglês, espanhol ou mandarim. Muitas destas soluções combinam ASR com atividades gamificadas, recompensas e estatísticas de progresso, o que as torna apelativas para utilizadores individuais. No entanto, a maioria destes produtos comerciais não foi desenhada especificamente para Português Europeu, e recorre muitas vezes a modelos ASR treinados sobretudo com dados de inglês ou de outras variantes do português, o que levanta dúvidas quanto à sensibilidade a detalhes fonéticos próprios do PE.

Por outro lado, vários trabalhos em CAPT focam-se mais na pontuação global da pronúncia (por exemplo, uma nota por frase ou por palavra) do que na análise estruturada de padrões de erro, como omissões, substituições específicas ou dificuldades persistentes em certos sons. Esta abordagem é útil para ter uma visão rápida do desempenho, mas menos

informativa para quem quer perceber exatamente onde estão as principais fragilidades de um aprendente.

Finalmente, é pouco comum encontrar sistemas que, além do treino interativo, integrem mecanismos de geração automática de relatórios estruturados em formato exportável (como PDF), pensados para acompanhamento ao longo do tempo por parte de docentes ou terapeutas da fala.

Entre os exemplos concretos descritos na literatura, destaca-se o *serious game* **e-*SoundWay***, concebido para o treino de fonética do inglês, que combina atividades lúdicas com exercícios de perceção e produção focados em contrastes segmentais específicos (como pares mínimos) e oferece **feedback imediato** sobre o desempenho do utilizador. Embora partilhe com o presente trabalho a lógica de *serious game* e a utilização de tarefas fonéticas estruturadas, o *e-SoundWay* está orientado para o inglês e não incorpora, tal como descrito, um mecanismo de análise sistemática de padrões de erro adaptados a fenómenos de uma língua concreta nem funcionalidades de geração de relatórios históricos em formato PDF.

De forma complementar, o estudo de Tejedor-García et al. (2020) avalia uma ferramenta CAPT para aprendentes de inglês centrada em **pares mínimos**, em que o sistema atribui pontuações automáticas à produção do utilizador e estas são comparadas com a avaliação de juízes humanos.

Os resultados mostram melhorias significativas na pronúncia e uma correlação razoável entre as pontuações automáticas e a avaliação humana, o que reforça o potencial destas soluções. Ainda assim, a ferramenta descrita funciona sobretudo como um módulo de treino e avaliação, sem uma componente de jogo completa e sem um foco explícito em relatórios agregados de evolução ao longo do tempo.

Outros trabalhos, como o de Korzekwa et al. (2022), exploram técnicas avançadas de síntese de fala e conversão de voz para gerar exemplos de pronúncias corretas e incorretas, aproximando-se de um laboratório fonético virtual, mas tendem igualmente a concentrar-se em inglês e em configurações de uso mais controladas do que num cenário de jogo offline para prática autónoma.

O *serious game* desenvolvido neste relatório procura responder a estas lacunas ao integrar, numa mesma solução:

- um motor ASR robusto (Whisper),
- a identificação explícita de padrões fonéticos de erro relevantes para o Português Europeu,
- uma progressão estruturada por níveis de dificuldade,
- feedback imediato para o utilizador após cada tentativa,
- e a geração automática de relatórios PDF detalhados, em modo offline.

Desta forma, o projeto contribui para um ecossistema de ferramentas de treino de pronúncia orientadas especificamente para o Português Europeu, com preocupações simultâneas de usabilidade, rigor fonético e capacidade de acompanhamento sistemático do progresso.

CAPÍTULO II – METODOLOGIA E OBJETIVOS

A metodologia seguida neste trabalho é de natureza essencialmente aplicada, centrada no desenvolvimento, validação técnica e análise funcional de um *serious game* para treino de pronúncia em Português Europeu. Mais do que produzir um modelo teórico abstrato, o objetivo foi **conceber, implementar e testar** um protótipo funcional que respondesse a um problema concreto identificado: **a escassez de ferramentas acessíveis, orientadas à pronúncia, com feedback automático e registo sistemático de desempenho.**

Neste sentido, adotou-se uma abordagem próxima da investigação aplicada em contextos reais (*real-world research*), em que a criação de um artefacto tecnológico é simultaneamente processo de investigação e resultado final (Robson & McCartan, 2016; Hevner et al., 2004). O ciclo de trabalho combinou práticas de engenharia de software com prototipagem iterativa, privilegiando versões sucessivas do jogo, testadas e refinadas à medida que novas necessidades iam surgindo.

II.1 – Objetivos do Trabalho

O objetivo geral deste trabalho consiste em conceber, desenvolver e avaliar, em contexto de prova de conceito, um protótipo funcional de *serious game* para treino de pronúncia em Português Europeu. Este protótipo deve combinar reconhecimento automático de fala, análise de padrões de erro fonético e geração de relatórios estruturados, de forma a apoiar a prática individual da pronúncia e a monitorização da evolução ao longo do tempo.

A partir deste objetivo geral, organizaram-se três eixos de **objetivos específicos**:

Eixo 1 – Conceção e desenvolvimento do protótipo

- **Levantar e formalizar** requisitos funcionais e não funcionais para um jogo de treino de pronúncia em Português Europeu, adequado a contextos de utilização autónoma e potenciando o uso em ambiente educativo.
- **Definir** a arquitetura do sistema, identificando os módulos de interface gráfica, processamento de áudio, reconhecimento de fala, análise fonética e geração de relatórios, bem como os fluxos de dados entre estes componentes.
- **Implementar** a aplicação em Python e Pygame, desenvolvendo uma interface gráfica em ecrã completo, menus de navegação por níveis de dificuldade e mecanismos de interação simples e consistentes.
- **Construir** um conjunto de frases organizadas por níveis de dificuldade, cobrindo estruturas fonéticas relevantes do Português Europeu e associando-lhes os respetivos recursos áudio.

Eixo 2 – Avaliação automática da pronúncia e análise fonética

- **Integrar** um modelo de reconhecimento automático de fala capaz de transcrever produções em Português Europeu e de alimentar métricas quantitativas de desempenho.
- **Definir e implementar** um algoritmo de comparação entre frase de referência e frase produzida, produzindo uma percentagem de acerto e identificando padrões de erro fonético previamente selecionados (por exemplo, nasalização reduzida, substituições consonânticas ou omissões).
- **Registar** de forma persistente, em formato estruturado (JSON), os resultados das sessões de jogo, incluindo informação sobre frases, percentagens de acerto, padrões detetados e contexto temporal.

Eixo 3 – Relatórios, análise de resultados e enquadramento ético

- **Conceber e gerar** relatórios PDF que sintetizem o desempenho do utilizador, apresentando estatísticas por nível, evolução temporal e distribuição de padrões fonéticos, em formatos adequados a consulta posterior.
- **Analisar** criticamente os dados recolhidos, discutindo o comportamento das métricas produzidas e a adequação dos padrões fonéticos selecionados face ao objetivo de treino.
- **Refletir** sobre implicações de privacidade e ética no registo e tratamento de dados de voz e desempenho, identificando cuidados a ter e limitações decorrentes dessas preocupações.

II.2 - Abordagem Geral da Investigação

O projeto enquadra-se numa lógica de **investigação aplicada e orientada a artefactos**, próxima da perspetiva de *design science*, em que se procura resolver um problema prático através do desenvolvimento de uma solução tecnológica concreta (Hevner et al., 2004).

No caso específico deste trabalho, o problema pode ser resumido da seguinte forma: **como disponibilizar uma ferramenta simples, offline e focada no Português Europeu que permita praticar pronúncia com feedback imediato e registo de evolução?**

Para responder a esta questão, a abordagem metodológica baseou-se em quatro ideias centrais:

- **Prototipagem incremental e iterativa**: em vez de tentar desenvolver o sistema completo de uma só vez, o jogo foi construído por versões sucessivas, começando por um núcleo funcional mínimo e adicionando gradualmente funcionalidades (Pressman & Maxim, 2014; Larman & Basili, 2003).

- **Validação contínua das funcionalidades**: cada nova versão foi testada de forma exploratória, procurando detetar erros, problemas de usabilidade e limitações técnicas antes de avançar para a fase seguinte.
- **Refinamento de requisitos à luz da prática**: alguns requisitos foram ajustados à medida que se percebia, na utilização real do protótipo, o que fazia sentido manter, simplificar ou reforçar (Sommerville, 2016).
- **Análise qualitativa e quantitativa dos dados gerados**: para além dos testes técnicos, os registos produzidos pelo jogo (percentagens de acerto, padrões fonéticos, evolução ao longo do tempo) foram usados como base para discutir o potencial pedagógico da solução.

Desta forma, a metodologia não se limita a descrever o processo de programação, mas procura enquadrar o desenvolvimento do jogo numa lógica de investigação estruturada, em que cada decisão técnica tem um motivo pedagógico ou funcional associado.

II.3 - Levantamento de Requisitos

O levantamento de requisitos foi um passo central, porque definiu desde cedo o que o jogo teria obrigatoriamente de fazer e em que condições o deveria fazer. A definição desses requisitos não resultou de um processo formal de engenharia de requisitos com utilizadores finais, mas de uma **análise gradual do problema**, apoiada na literatura disponível e na experiência prática do autor enquanto utilizador de ferramentas digitais de aprendizagem.

Em primeiro lugar, foi realizada uma leitura exploratória de literatura académica e técnica sobre *serious games*, CALL e sistemas de treino de pronúncia, recorrendo a artigos e relatórios acessíveis em bases de dados e repositórios institucionais. Essa leitura, embora não sistemática, permitiu identificar algumas características recorrentes em soluções deste tipo, como a organização em níveis, a importância do feedback imediato e o registo do desempenho ao longo do tempo. Em segundo lugar, foram observados, de forma informal, exemplos de aplicações de aprendizagem de línguas disponíveis publicamente, com especial atenção às formas de apresentar exercícios orais, utilizar áudio e sinalizar o progresso do utilizador.

Finalmente, o próprio autor sistematizou necessidades práticas que reconhece na utilização de aplicações educativas — simplicidade de navegação, funcionamento local sem dependência constante de internet, clareza do feedback e ausência de configurações complexas — e traduziu essas necessidades numa lista inicial de requisitos para o protótipo.

A partir destas fontes elaborou-se, então, uma lista inicial de requisitos, distinguindo a clássica separação entre requisitos funcionais (o que o sistema faz) e requisitos não funcionais (como o sistema se comporta). Nos primeiros incluíram-se, por exemplo, a possibilidade de selecionar níveis de dificuldade, ouvir a frase-modelo, gravar a produção, calcular uma percentagem de acerto e registar resultados em formato estruturado. Nos segundos destacaram-se aspetos como a execução local, a ausência de dependências externas complexas, a legibilidade da interface e o tempo de resposta aceitável após cada tentativa de pronúncia.

Este levantamento não correspondeu a um momento único, mas a um processo iterativo: à medida que os primeiros protótipos eram implementados e testados pelo autor, alguns requisitos foram sendo ajustados, refinados ou abandonados, em função da viabilidade técnica e da experiência de utilização observada na prática.

II.3.1 - Requisitos Funcionais

Entre os requisitos funcionais definidos para o protótipo destacam-se:

- Permitir a seleção de níveis organizados em três dificuldades (Fácil, Médio, Difícil).
- Apresentar ao utilizador uma frase em Português Europeu, acompanhada do respetivo áudio de referência.
- Gravar a fala do utilizador e processá-la localmente.
- Transcrever automaticamente o áudio com recurso ao modelo Whisper.
- Comparar a frase dita com a frase de referência, calculando uma percentagem de acerto.

- Identificar padrões fonéticos recorrentes (por exemplo, nasalização reduzida, trocas de consoantes ou omissões em final de palavra).
- Guardar os resultados num ficheiro estruturado (JSON), incluindo data, nível, frase de referência, transcrição e padrões detetados.
- Produzir relatórios PDF que agreguem desempenho, estatísticas e gráficos, em versão geral e por nível.

Estes requisitos foram pensados para garantir que o jogo não se limita a apresentar estímulos e recolher respostas, mas oferece também **feedback estruturado** e uma base de dados suficientemente rica para análise posterior.

II.3.2 - Requisitos Não Funcionais

Os requisitos não funcionais procuraram assegurar que a aplicação fosse utilizável em cenários reais, com recursos limitados:

- **Funcionamento offline**, evitando dependência de conectividade permanente.
- **Interface gráfica simples e intuitiva**, com navegação clara entre menus, níveis e relatórios.
- **Desempenho aceitável em computadores de características médias**, sem necessidade de hardware de topo.
- **Organização modular do código**, facilitando manutenção e futuras extensões.
- **Utilização de formatos de dados portáteis** (JSON e CSV), que possam ser analisados com ferramentas externas.
- **Geração de relatórios PDF com layout estável e legível**, evitando configurações complexas do lado do utilizador.

O levantamento de requisitos não foi um momento único, mas sim um processo iterativo: à medida que o jogo evoluía, alguns requisitos foram clarificados, outros simplificados e outros reforçados, em linha com o que a literatura descreve para projetos com forte componente exploratória (Pressman & Maxim, 2014; Robson & McCartan, 2016).

II.3.3 - Tecnologias e Ferramentas Utilizadas

A seleção das tecnologias teve em conta três critérios principais: **acessibilidade**, **robustez** e **facilidade de integração** entre componentes.

- **Python** foi escolhido como linguagem principal pela sua sintaxe simples, forte uso em contexto académico e vasto ecossistema de bibliotecas para processamento de áudio, aprendizagem automática e manipulação de ficheiros (Xinogalos, 2020).
- **Pygame** foi utilizado para a interface gráfica e gestão do jogo, permitindo criar um ambiente 2D interativo com controlo direto sobre o fluxo de ecrãs, eventos de teclado e reprodução de áudio.
- **Whisper** foi adotado como motor de reconhecimento automático de fala, pela sua capacidade de lidar com múltiplos idiomas e sotaques, mesmo em condições de gravação não profissionais.
- **ReportLab** foi utilizado para gerar relatórios em PDF, garantindo um layout consistente e totalmente offline.
- **JSON** serviu de base ao armazenamento de frases, resultados e estatísticas, pela sua simplicidade e fácil leitura tanto por humanos como por máquinas.
- **Áudios pré-gerados** foram organizados por nível e índice, garantindo correspondência clara entre cada frase textual e o respetivo estímulo sonoro.

Esta combinação permitiu construir um sistema onde **interface gráfica**, **reconhecimento de fala** e **análise fonética** comunicam entre si de forma relativamente simples, sem dependência de plataformas proprietárias ou serviços externos.

II.3.4 - Procedimento de Desenvolvimento

O desenvolvimento do *serious game* foi organizado em **quatro fases principais**, cada uma com ciclos internos de teste e revisão. Esta estrutura aproxima-se de modelos iterativos

de desenvolvimento de software, como os descritos por Pressman e Maxim (2014) e Sommerville (2016).

Fase 1 – Modelação e estruturação inicial

- Criação do *dataset* de frases, distribuídas por três níveis de dificuldade.
- Definição da estrutura de diretórios e organização dos módulos principais.
- Implementação do fluxo base do jogo: menu → seleção de nível → apresentação de frase → gravação.

Fase 2 – Integração do reconhecimento de fala

- Implementação do módulo de captura de áudio e gravação temporária.
- Integração do modelo Whisper e testes às transcrições obtidas.
- Criação do algoritmo de comparação entre frase esperada e frase dita, com cálculo de percentagens de acerto.

Fase 3 – Avaliação fonética e registo de dados

- Implementação das regras de identificação automática de padrões de erro (omissões, substituições, alterações de nasalização, etc.).
- Definição da estrutura do ficheiro resultados.json.
- Registo progressivo de todas as sessões de jogo, garantindo consistência e integridade dos dados.

Fase 4 – Relatórios e interface final

- Criação dos relatórios PDF por nível e relatório geral.
- Integração de gráficos, tabelas e estatísticas nos relatórios.
- Ajuste do design da interface gráfica, menus e navegação completa.
- Realização de testes completos ao fluxo do jogo, para verificar estabilidade e evitar falhas críticas.

Cada fase só foi considerada concluída depois de um conjunto mínimo de testes exploratórios, de forma a reduzir retrabalho nas fases posteriores e assegurar coerência global do sistema.

II.3.5 - Procedimento de Avaliação e Recolha de Dados

A avaliação do protótipo baseou-se nos **dados gerados automaticamente** durante a utilização do *serious game*. Cada interação produziu um registo contendo:

- data e hora da sessão,
- frase de referência,
- transcrição da fala do utilizador,
- nível selecionado,
- percentagem de acerto,
- padrões fonéticos detetados.

A partir destes registos foi possível:

- calcular métricas agregadas por nível (por exemplo, médias de acerto, número de sessões, distribuição de erros),
- analisar a evolução temporal das percentagens de acerto,
- identificar dificuldades sistemáticas associadas a determinados padrões fonéticos,
- gerar gráficos e tabelas que são usados tanto nos relatórios PDF como no próprio relatório.

Importa referir que esta avaliação tem um foco **funcional e exploratório**: pretende demonstrar que o sistema é capaz de reconhecer, registar e representar a evolução da pronúncia do utilizador, mais do que comparar de forma exaustiva a sua eficácia com outras abordagens de ensino. A discussão das limitações desta opção é retomada no capítulo de avaliação.

II.3.6 - Considerações Éticas e Proteção de Dados

Uma vez que o sistema trabalha com **dados de voz**, foram definidos princípios básicos para garantir **respeito pela privacidade e segurança da informação**, em linha com as boas práticas descritas na legislação europeia de proteção de dados, nomeadamente o Regulamento (UE) 2016/679 (Regulamento Geral sobre a Proteção de Dados – RGPD).

Foram adotadas as seguintes medidas:

- Todo o processamento de áudio é feito **localmente**, não sendo enviados ficheiros para servidores externos.
- As gravações são utilizadas apenas para efeitos de transcrição e são **eliminadas após o processamento**.
- Os ficheiros de resultados não contêm dados pessoais identificáveis (como nome, e-mail ou identificadores diretos).
- Os dados gerados destinam-se exclusivamente a fins de **investigação e treino de pronúncia**, não sendo usados para outros propósitos.

Desta forma, procura-se garantir que o protótipo cumpre princípios de minimização de dados, limitação da finalidade e segurança, o que é especialmente relevante em aplicações educativas que envolvem registo de voz.

CAPÍTULO III – ARQUITETURA DO SISTEMA

A arquitetura do sistema desenvolvido assenta numa **organização modular** que integra componentes de interface, processamento de áudio, reconhecimento automático de fala, análise fonética, persistência de dados e geração de relatórios. Esta divisão em módulos permite garantir clareza na estrutura, facilidade de manutenção e uma evolução futura simplificada.

O sistema foi concebido para funcionar de forma **totalmente local** (*standalone*), eliminando dependências de conectividade e assegurando estabilidade tanto em ambientes académicos como domésticos.

De forma resumida, o *serious game* é constituído por **seis blocos funcionais** principais:

- **Interface gráfica (Pygame)**, responsável por todos os ecrãs, menus e interações com o utilizador;
- **Gestão do dataset e dos níveis**, que organiza as frases por dificuldade, associa identificadores e controla a progressão entre níveis, lendo da estrutura frases_dataset.json;
- **Módulo de áudio e gravação**, encarregado de reproduzir os estímulos e capturar a fala do utilizador num formato adequado ao reconhecimento;
- **Motor de reconhecimento e avaliação fonética** (Whisper + algoritmo de comparação), que transforma o áudio em texto, calcula a percentagem de acerto e identifica padrões de erro;
- **Módulo de persistência de dados**, que regista em ficheiros JSON (resultados.json, progresso.json) a informação gerada em cada tentativa;
- **Módulo de geração de relatórios PDF** (ReportLab), que lê os ficheiros de resultados e produz relatórios estruturados sobre o desempenho e a evolução do utilizador.

Estes componentes comunicam entre si de forma sequencial e controlada, garantindo que cada interação do utilizador percorre um ciclo completo de estímulo → resposta → análise → registo e síntese em relatórios.

A **Figura 1** ilustra esta arquitetura global e os fluxos principais entre os módulos, desde a interação do utilizador com a interface gráfica até à geração de relatórios PDF.

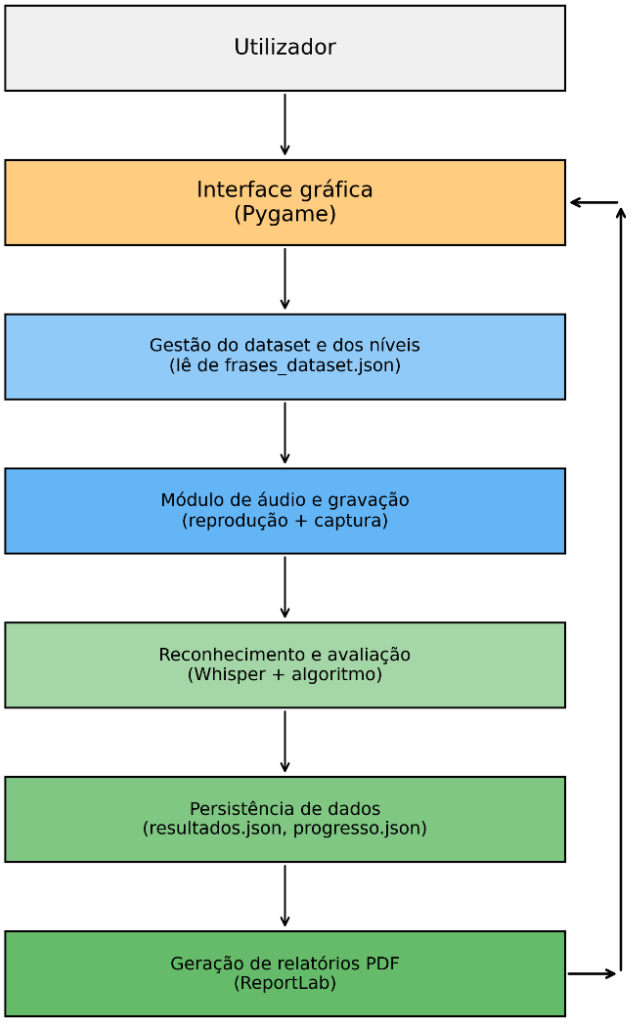


Figura 1 – Arquitetura global do sistema, com os principais módulos e fluxos de dados.

III.1 - Estrutura de Diretórios

O projeto encontra-se organizado em diretórios temáticos que se alinham com os módulos descritos anteriormente: uma pasta dedicada aos recursos multimédia (áudios e imagens), outra aos dados de entrada (dataset de frases) e outra aos dados de saída (resultados e relatórios).

Esta separação física acompanha a separação lógica da arquitetura, facilitando a manutenção, a leitura do projeto e uma futura migração para outros ambientes (por exemplo, uma versão web ou multiutilizador).

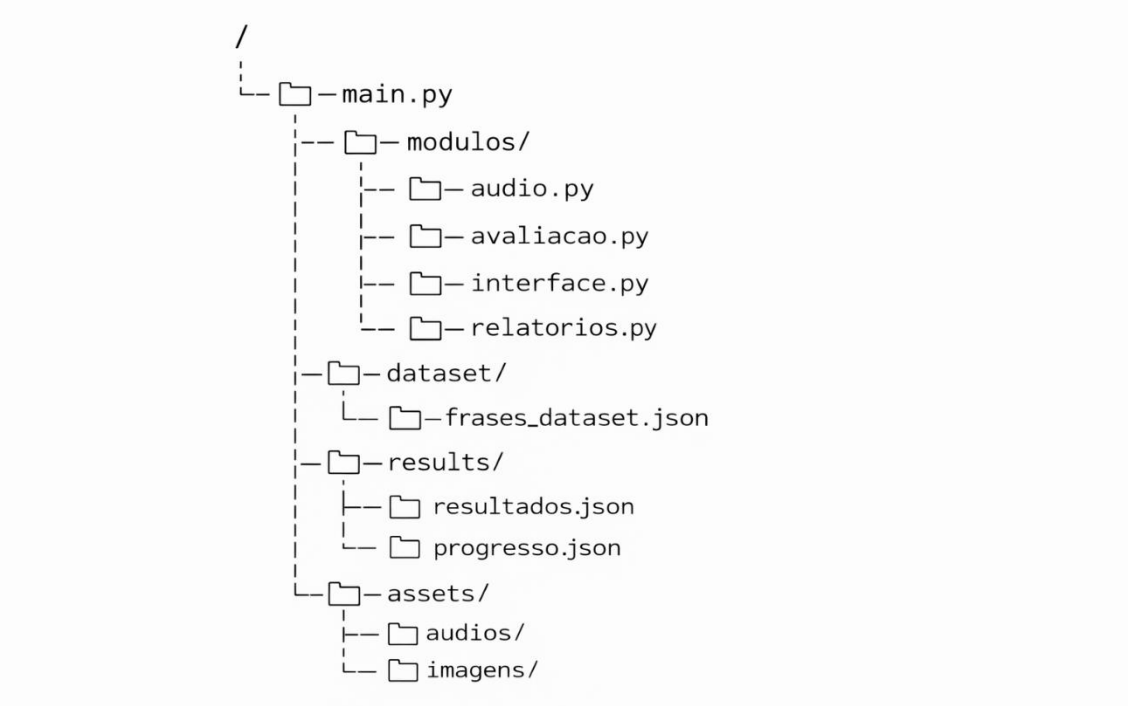


Figura 2 - Estrutura de diretórios do projeto, alinhada com os módulos funcionais da aplicação.

III.2 - Dataset de Frases e Organização dos Níveis

O dataset principal encontra-se armazenado em **frases_dataset.json**, contendo:

- texto da frase,
- lista de palavras,
- nível de dificuldade (Fácil, Médio, Difícil),
- identificador único,
- nome do ficheiro de áudio correspondente.

Cada nível contém 30 frases, totalizando 90 frases cuidadosamente distribuídas para cobrir:

- diversidade fonética,
- estruturas morfossintáticas variadas,
- padrões típicos de erro em aprendentes de PE.

A existência de um dataset estruturado permite escalabilidade futura — bastando adicionar novas entradas JSON para expandir o jogo.

A Tabela 1 apresenta um exemplo simplificado de entradas do ficheiro "frases_dataset.json", com o identificador, a frase, o nível de dificuldade e o nome do ficheiro áudio associado.

Tabela 1 - Amostra de dados contidos no dataset de frases.

Id	Frase	Nível	Ficheiro_audio
1	A água está fria.	Fácil	F_01.mp3
2	O sol brilha no céu.	Fácil	F_02.mp3
…	…	…	…

III.3 - Módulo de Áudio

O módulo de áudio é responsável por **gerir todo o fluxo sonoro da aplicação**, desde a reprodução dos estímulos até à preparação do sinal de voz para reconhecimento automático. Em termos arquitetónicos, este componente recebe uma frase do gestor de níveis, associa-lhe o respetivo ficheiro áudio de referência e devolve ao motor de reconhecimento um ficheiro intermédio num formato padronizado.

Não são guardados ficheiros de áudio permanentes do utilizador, garantindo que o sistema apenas trabalha com **dados temporários** e alinhados com os princípios de privacidade definidos.

III.4 - Módulo de Reconhecimento e Avaliação da Pronúncia

O motor de reconhecimento e avaliação agrega o modelo de reconhecimento automático de fala e o algoritmo de comparação. Na arquitetura global, este módulo recebe o áudio pré-processado, produz a transcrição da frase dita e gera um conjunto de indicadores: correspondência lexical, percentagem global de acerto e padrões fonéticos detetados. O objetivo arquitetónico não é detalhar o algoritmo, mas assegurar que existe **um único ponto responsável por transformar áudio em informação pedagógica** útil para os restantes componentes do sistema.

III.5 - Módulo de Interface Gráfica

O módulo de interface gráfica encapsula todos os ecrãs e interações com o utilizador. Este componente orquestra a navegação entre menus, seleção de níveis, apresentação de frases e visualização de feedback, comunicando com os módulos de áudio, reconhecimento e relatórios através de funções bem definidas. Do ponto de vista arquitetónico, a interface funciona como "frontend" do sistema, mantendo a lógica de negócio e o processamento de dados isolados em módulos internos, o que **facilita a substituição futura** da camada gráfica sem reescrever o núcleo da aplicação.

III.6 - Módulo de Armazenamento e Estrutura dos Resultados

O módulo de armazenamento é responsável por receber, a partir do motor de avaliação, os dados relevantes de cada tentativa e persistir essa informação em ficheiros estruturados. Ao nível da arquitetura, este componente abstrai o formato físico dos dados e expõe uma visão unificada de "registos de desempenho", passível de ser consumida tanto pela interface (para resumos imediatos) como pelo módulo de relatórios (para análises mais aprofundadas). Esta camada **garante consistência no modelo de dados** e simplifica a ligação a futuros sistemas externos.

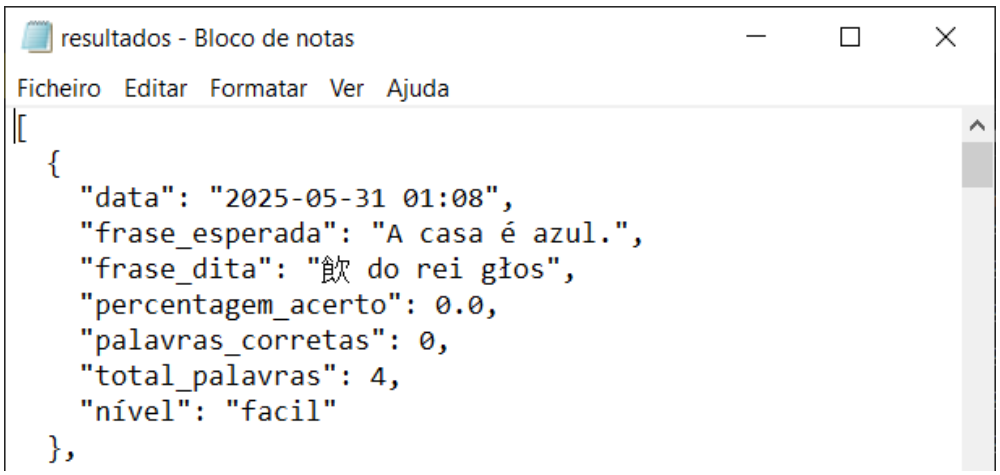


Figura 3 - Exemplo de registo no ficheiro resultados.json, com informação sobre frase, desempenho e padrões fonéticos.

III.7 - Geração de Relatórios PDF

O módulo de relatórios consome os registos de desempenho e agrega-os em documentos de síntese. Arquiteturalmente, este componente não altera os dados de base; limita-se a organizar estatísticas, gráficos e tabelas em formato PDF, funcionando como uma camada de apresentação avançada.

A sua existência separa claramente a recolha e armazenamento dos dados da forma como estes são comunicados ao utilizador ou ao investigador, reforçando a modularidade e a reutilização do núcleo de dados em diferentes contextos.

III.8 - Fluxo Funcional do Sistema

O fluxo principal da aplicação pode ser sintetizado em **seis etapas**:

1. Seleção da dificuldade e do nível
2. Apresentação da frase e reprodução do áudio
3. Gravação da fala do utilizador
4. Transcrição automática e comparação
5. Feedback imediato (percentagem + padrões)
6. Registo em JSON e atualização de relatórios

Este ciclo garante que cada interação resulta num conjunto completo de dados úteis para aprendizagem e análise posterior.

CAPÍTULO IV – IMPLEMENTAÇÃO

A implementação do *serious game* foi realizada de forma modular, seguindo princípios de clareza, separação de responsabilidades e reutilização de componentes. O código foi desenvolvido integralmente em Python, recorrendo a bibliotecas específicas para a interface gráfica, processamento de áudio, reconhecimento de fala e geração de relatórios.

Este capítulo descreve as principais decisões técnicas, a organização interna do código e o funcionamento detalhado dos módulos que compõem o sistema.

IV.1 - Estrutura Geral do Código

A implementação materializa os cinco pilares arquitetónicos descritos no Capítulo III em módulos de código independentes. Cada módulo corresponde a uma responsabilidade principal:

• **main.py** — ponto de entrada da aplicação, responsável pela inicialização, carregamento dos recursos e coordenação dos restantes módulos.

• **modulos/audio.py** — concretiza o módulo de áudio, implementando as rotinas de gravação, reprodução e pré-processamento do sinal sonoro.

• **modulos/avaliacao.py** — implementa o motor de reconhecimento e avaliação, integrando o modelo de reconhecimento automático de fala e o algoritmo de comparação responsável pelo cálculo da percentagem de acerto e pela deteção de padrões fonéticos.

• **modulos/interface.py** — concretiza o módulo de interface gráfica e o fluxo de interação, gerindo ecrãs, botões, eventos e transições entre estados do jogo.

• **modulos/relatorios.py** — implementa o módulo de relatórios, lendo os registos de desempenho e produzindo os documentos PDF com estatísticas, gráficos e tabelas.

Os diretórios **dataset/**, **results/** e **assets/** correspondem, respetivamente, ao repositório de frases, ao armazenamento persistente dos resultados e aos recursos multimédia utilizados pela interface.

Esta organização reforça a correspondência direta entre a visão arquitetónica e o código, o que simplifica a manutenção, a depuração e a extensibilidade futura do sistema.

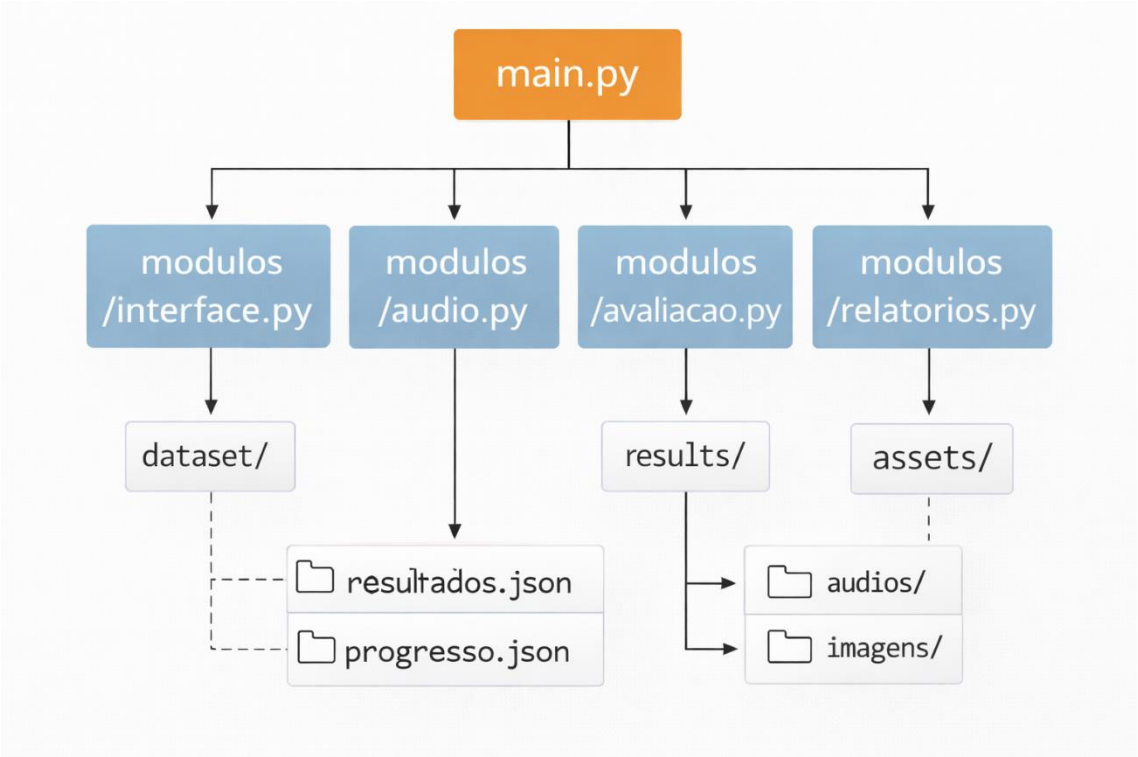


Figura 4 - Organização dos módulos de código e respetivas responsabilidades principais.

IV.2 - Interface Gráfica e Fluxo de Interação (Pygame)

A interface foi desenvolvida com a biblioteca **Pygame**, permitindo criar um ambiente responsivo e visualmente consistente.

IV.2.1 - Menus e Navegação

O jogo apresenta múltiplos ecrãs:

- ecrã inicial,

- seleção de dificuldade,
- seleção de nível,
- ecrã de treino (frase + gravação),
- ecrã de feedback,
- ecrã de relatórios.

A **Figura 5** mostra o ecrã inicial do jogo, com uma sala de aula ilustrada, o botão *Começar* para iniciar o treino e o botão *Relatório* para aceder aos dados de desempenho.

Figura 5 - Ecrã inicial do serious game de treino de pronúncia.

Após o ecrã inicial, o utilizador é encaminhado para o menu de seleção de dificuldade (Figura 6), onde pode optar entre os níveis *Fácil*, *Médio* e *Difícil*, representados por botões coloridos de elevada visibilidade.

Figura 6 - Ecrã de seleção de dificuldade (Fácil, Médio, Difícil).

Cada ecrã é desenhado em modo *full-screen*, garantindo imersão e evitando distrações visuais. A navegação é efetuada através de botões estilizados, criados com imagens vetoriais e fundos translúcidos, assegurando acessibilidade e clareza visual.

A Figura 7 ilustra o ecrã de seleção de nível dentro da dificuldade escolhida, em que cada ícone em forma de livro corresponde a uma frase e as estrelas representam o progresso.

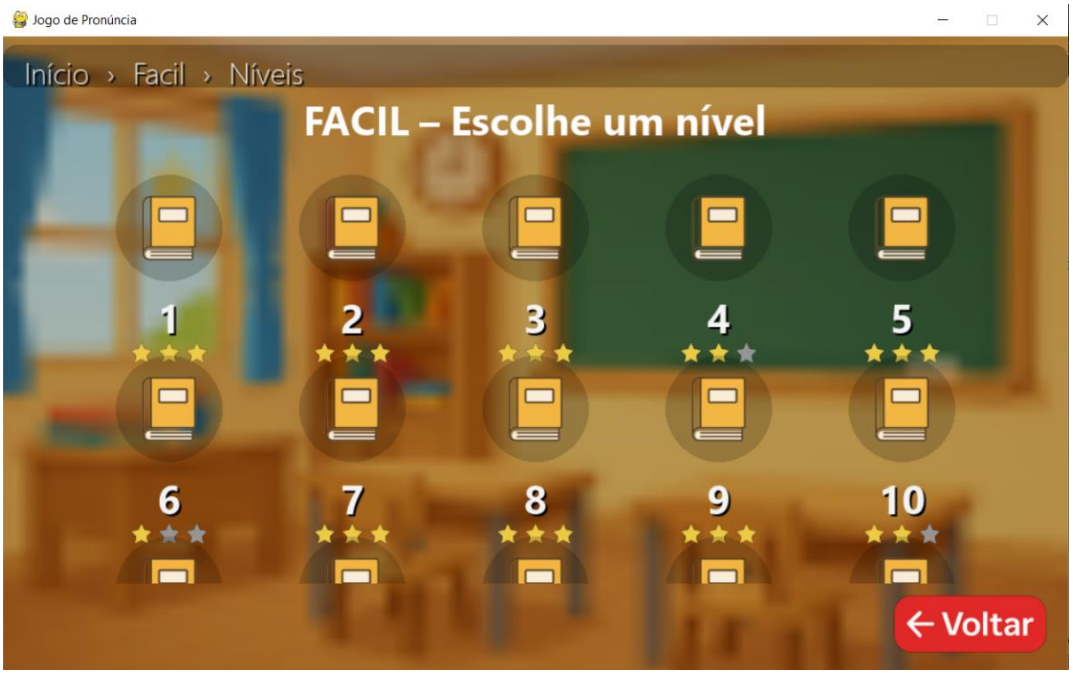


Figura 7 - Ecrã de seleção de nível, com livros e estrelas de progresso.

Durante o treino, o utilizador vê no quadro a frase-alvo destacada e recebe instruções para manter a tecla ESPAÇO premida enquanto fala. A Figura 8 mostra este ecrã de apresentação da frase e preparação para a gravação.

Figura 8 - Ecrã de treino com apresentação da frase e instrução de gravação.

IV.2.2 - Carregamento de Recursos

Todos os recursos gráficos — ícones, fundos, botões e elementos decorativos — são carregados numa fase inicial, evitando tempo de espera durante o jogo e assegurando fluidez no desempenho.

IV.3 - Módulo de Áudio

O módulo de áudio foi implementado para garantir precisão na gravação e compatibilidade com o modelo Whisper.

IV.3.1 - Reprodução do Áudio de Referência

Cada frase possui um ficheiro .mp3. O sistema utiliza o mixer do Pygame para:

- carregar o ficheiro,
- assegurar volume adequado,
- impedir sobreposição entre sons,
- garantir sincronia entre reprodução e interface.

IV.3.2 - Gravação da Fala do Utilizador

A gravação é efetuada com:

- taxa de amostragem estável,
- formato .wav,
- duração controlada,
- guarda temporária.

IV.3.3 - Eliminação Segura

Após a transcrição:

- o ficheiro gravado é automaticamente removido,
- garantindo privacidade e reduzindo lixo digital.

IV.4 - Reconhecimento Automático de Fala (Whisper)

O Whisper foi incorporado como biblioteca local, permitindo transcrever áudio sem dependência de internet.

IV.4.1 - Processo de Transcrição

1. O ficheiro .wav é enviado para o modelo.
2. O modelo retorna texto normalizado.
3. A aplicação processa a transcrição, removendo marcas ou hesitações.
4. O texto é enviado para o motor de comparação.

O Whisper foi escolhido pela sua robustez em gravações amadoras, ruído ambiente e sotaques variados, o que o torna adequado para contextos educativos.

IV.5 - Algoritmo de Comparação da Pronúncia

O algoritmo de comparação é responsável por transformar a transcrição num indicador quantitativo da pronúncia do utilizador.

Visualmente, o resultado desta comparação é apresentado no ecrã de feedback (Figura 9), onde o utilizador vê a frase de referência e a sua produção, com realce a cores das diferenças mais relevantes.

Figura 9 - Ecrã de feedback da pronúncia, com comparação entre frase esperada e produzida.

IV.5.1 - Comparação Estrutural

O sistema verifica:

- palavras corretas,
- palavras omitidas,
- palavras alteradas,
- erros lexicais,
- divergências de ordem.

Utiliza listas tokenizadas para garantir que a comparação é realizada de forma granular.

IV.5.2 - Cálculo da Percentagem de Acerto

A percentagem final é calculada com base em:

- correspondência lexical,
- penalizações por omissões,
- penalizações por substituições,
- ajustes para frases curtas ou longas.

Este valor é apresentado de forma imediata ao utilizador e armazenado para análise posterior.

IV.6 - Identificação Automática de Padrões Fonéticos

Este módulo complementa a avaliação lexical ao procurar erros fonéticos comuns em aprendentes de Português Europeu.

São detetados, entre outros:

- nasalização reduzida,
- troca /ʁ/ ↔ /ɾ/,
- substituição CH→S,
- omissão de “s” final.

A deteção é realizada através de:

- padrões regulares,
- correspondência fonética aproximada inferida pelo Whisper,
- comparação direta entre frase dita e esperada.

Estes padrões são fundamentais para fornecer feedback orientado e gerar relatórios mais ricos.

IV.7 - Registo e Estruturação dos Resultados (JSON)

Após cada tentativa, o sistema cria um objeto JSON contendo:

- nível,
- frase original,
- transcrição da frase dita,
- percentagem de acerto,
- padrões detetados,
- data e hora.

O ficheiro **resultados.json** funciona como um diário de sessão, permitindo:

- análises estatísticas,
- construção de relatórios,
- avaliação temporal do desempenho.

IV.8 - Geração de Relatórios PDF (ReportLab)

A criação dos relatórios PDF é totalmente automatizada.

Os relatórios incluem:

- média de acerto por nível,
- evolução temporal,

- padrões fonéticos mais frequentes,
- tabela das últimas 40 tentativas,
- gráficos integrados,
- sínteses detalhadas.

Relatório – Fácil

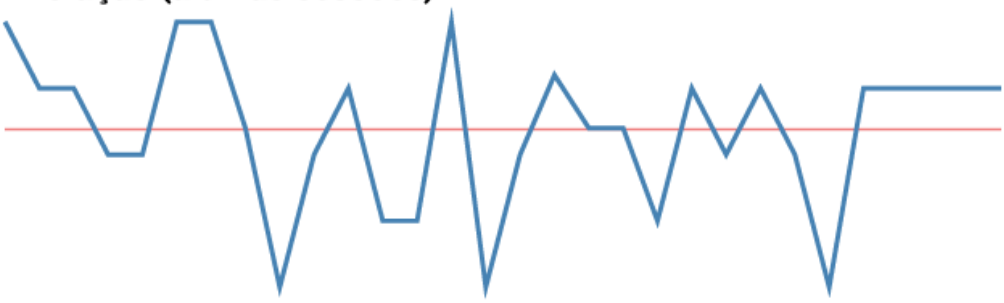
16/11/2025 17:18

Tentativas	33
Média de acerto	57.9%
Palavras corretas	82
Total de palavras	140

Padrões de pronúncia

- Nasalização reduzida: 0% das sessões (0 ocorrências)
- Troca R→L (líquidas): 0% das sessões (0 ocorrências)
- CH→S: 16% das sessões (3 ocorrências)
- Omissão de 'S' final: 0% das sessões (0 ocorrências)

Evolução (últimas sessões)



Média: 59.5%

Sessões (mais recentes)

Data	Acerto	Frase esperada	Tu disseste
2025-11-05 22:37	75.0%	O ninho está cheio.	O Ninho está saiu.
2025-11-05 22:37	75.0%	O ninho está cheio.	O ninho está seio.
2025-11-05 22:37	75.0%	O ninho está cheio.	O Ninho está aceio.
2025-11-05 22:37	75.0%	O ninho está cheio.	O Ninho está a ceio.
2025-11-05 22:37	75.0%	O ninho está cheio.	O ninho está seio.

Figura 10 - Exemplo de relatório PDF gerado automaticamente pelo sistema, com gráfico de evolução e tabela de sessões.

Na interface do jogo, o acesso a estes elementos é feito através do ecrã de relatórios (Figura 11), que permite filtrar por nível de dificuldade e consultar as sessões registadas antes de exportar o relatório em PDF.

Figura 11 - Ecrã de relatórios do jogo, com separadores por nível e tabela de sessões.

O layout foi desenhado para ser simples, direto e legível. A consistência visual foi considerada essencial para utilização pedagógica posterior.

IV.9 - Testes e Validação

Durante o desenvolvimento, foram realizados:

- **testes de funcionalidade** (menus, botões, fluxo),
- **testes de estabilidade** (transição entre ecrãs, reprodução de áudio),
- **testes de precisão** (validação do Whisper),
- **testes de carga mínima** (dataset completo de 90 frases),
- **testes de integridade de relatórios**.

Os testes permitiram resolver problemas de:

- formatação de áudio,
- atrasos na interface,
- inconsistências entre frases esperadas e áudios,
- formatação JSON,
- layouts PDF.

CAPÍTULO V – AVALIAÇÃO E RESULTADOS

V.1 - Objetivos da avaliação

A avaliação do *serious game* desenvolvido teve como propósito principal **verificar se a solução cumpre os objetivos definidos na fase de conceção**:

1. Proporcionar um ambiente de treino de pronúncia em Português Europeu com feedback imediato e mensurável,
2. Registar de forma sistemática o desempenho do utilizador ao longo do tempo,
3. Identificar padrões recorrentes de dificuldade fonética e
4. Sintetizar essa informação em relatórios claros, utilizáveis em contexto pedagógico ou terapêutico.

Mais concretamente, a avaliação procurou responder a **quatro questões centrais**:

- Em que medida o jogo consegue registar e sintetizar o desempenho do utilizador ao longo do tempo, de forma consistente e consultável?
- Até que ponto as métricas implementadas (percentagem de acerto, número de palavras corretas, identificação de padrões fonéticos) são coerentes com aquilo que se observa nas produções de fala?
- Em que medida a arquitetura técnica (integração de Pygame, Whisper, JSON e ReportLab) se revela estável e adequada a um cenário real de utilização?
- Que potencialidades pedagógicas o protótipo evidencia e que limitações se tornam visíveis à luz dos dados recolhidos?

Dado o enquadramento aplicado do projeto, a avaliação não pretende assumir o estatuto de ensaio clínico ou estudo de larga escala, mas sim de prova de conceito. Interessa, sobretudo, demonstrar que a aplicação é tecnicamente robusta, capaz de produzir métricas consistentes e de transformar a interação oral com o sistema em dados analisáveis, suportando decisões futuras de evolução e utilização em contextos reais.

V.2 - Conjunto de dados e procedimento de análise

A análise desenvolvida neste capítulo baseia-se nos registos automáticos gerados pelo *serious game* e já descritos no Capítulo II, em particular na estrutura dos ficheiros resultados.json e progresso.json. Em síntese, cada interação com o jogo corresponde a um registo que inclui informação sobre a frase de referência, a transcrição produzida pelo modelo de reconhecimento, a percentagem de acerto por palavras, o nível de dificuldade associado e os padrões fonéticos identificados, bem como o ponto de avanço do utilizador em cada nível.

O interesse desta base de dados não reside apenas no detalhe de cada tentativa, mas na sua **acumulação ao longo do tempo**: é essa sequência de registos que permite reconstruir o percurso de treino, observar tendências de evolução, comparar desempenhos em diferentes níveis de dificuldade e identificar padrões de erro recorrentes. É a partir deste conjunto de dados que são construídas as métricas, gráficos e tabelas apresentados nas secções seguintes.

Mesmo com um número limitado de sessões, **os dados funcionam**, assim, como uma "caixa negra" transparente em que cada tentativa fica documentada de forma estruturada e passível de consulta posterior.

Durante a fase de desenvolvimento e testes exploratórios foram realizadas diversas sessões de utilização do jogo, predominantemente ao nível Fácil, com repetição de um conjunto limitado de frases. Esta estratégia permitiu observar, para os mesmos estímulos, produções com graus de acerto muito distintos, o que é particularmente útil para validar a sensibilidade do sistema a variações na qualidade da pronúncia e da transcrição.

Durante o desenvolvimento da aplicação foram realizadas várias dezenas de testes exploratórios, em diferentes fases de implementação, com o objetivo de verificar a estabilidade do jogo, a integração entre módulos e o comportamento do reconhecimento de fala em condições diversas. No entanto, muitos desses testes iniciais **não ficaram registados** de forma estruturada, uma vez que o módulo de registo automático de resultados só foi introduzido numa fase mais avançada do projeto.

Para efeitos de análise sistemática neste relatório, consideram-se apenas os dados registados após a implementação desse módulo. No total, foram contabilizadas **trinta e oito tentativas de pronúncia**, distribuídas por **onze frases distintas** e pelos três níveis de dificuldade, com **trinta e três tentativas no nível Fácil, 4 no nível Médio e 1 no nível Difícil**. Estas tentativas foram recolhidas em vários momentos de utilização do jogo ao longo da fase de desenvolvimento, em contexto controlado de teste, e constituem um conjunto de dados de natureza exploratória, adequado para uma prova de conceito mas ainda insuficiente para generalizações estatísticas robustas.

A análise foi estruturada em três eixos:

1. **Métricas de desempenho global** – observação da distribuição das percentagens de acerto, identificação de tentativas com valores extremos (0% e 100%) e verificação de coerência entre o número de palavras corretas e a métrica global;
2. **Evolução temporal** – leitura dos registos por ordem cronológica e inspeção dos gráficos de tendência gerados automaticamente, com foco na estabilidade do cálculo e na legibilidade das curvas de desempenho;
3. **Padrões fonéticos** – exploração das contagens de padrões na estrutura analise_fonetica, bem como da associação entre determinados alvos (targets) e erros esperados, de forma a avaliar a utilidade pedagógica dos indicadores.

Importa sublinhar que, nesta fase, o conjunto de dados é **reduzido** e corresponde maioritariamente a **sessões de teste conduzidas pelo autor**.

Assim, os resultados devem ser interpretados como **evidência de funcionamento e potencial pedagógico**, e não como prova estatística da eficácia do jogo em populações alargadas.

V.3 Resultados quantitativos de desempenho

A principal métrica de desempenho adotada é a **percentagem de acerto**, calculada a partir da comparação entre a transcrição produzida pelo modelo Whisper e a frase-alvo armazenada no dataset.

O cálculo tem em conta:

• o número de palavras corretas;

• as omissões (palavras que não aparecem na produção);

• as substituições e alterações lexicais;

• as diferenças estruturais mais evidentes entre a frase dita e a frase esperada.

Esta percentagem cumpre um papel duplo:

- Por um lado, fornece ao utilizador um indicador imediato do seu desempenho numa dada tentativa;
- Por outro, serve de base para a construção de métricas agregadas, como médias por nível e tendências de evolução ao longo do tempo.

Em complemento, são igualmente relevantes o número total de tentativas por nível, o total de palavras corretas produzidas e a tendência recente (por exemplo, as últimas tentativas num determinado nível), que permite perceber se o desempenho está a estabilizar, a melhorar ou a oscilar.

Os registos analisados evidenciam uma ampla variabilidade de percentagens de acerto, com tentativas que vão do 0% (nenhuma palavra correta) ao 100% (correspondência integral entre frase dita e frase de referência). Esta amplitude é desejável num cenário de avaliação, pois demonstra que o sistema responde de forma diferenciada a produções muito afastadas do alvo e a produções praticamente nativas.

Como se observa na Figura 12, a distribuição das percentagens de acerto é heterogénea, com tentativas concentradas sobretudo nas classes de 50 - 75% e de 75 - 100%.

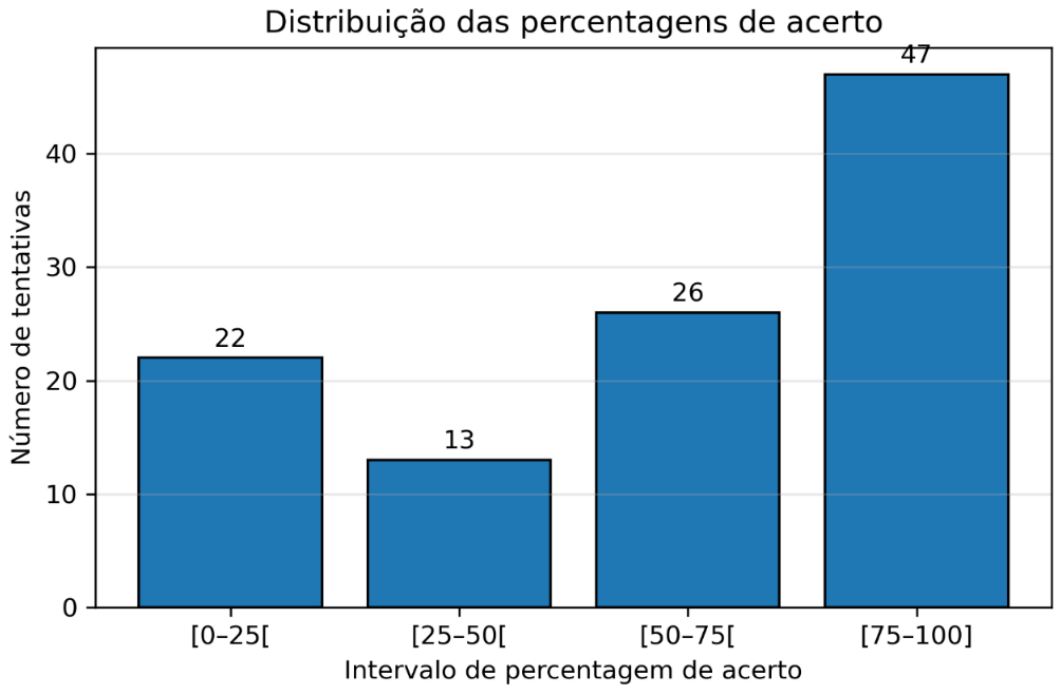


Figura 12 - Distribuição das percentagens de acerto nas tentativas registadas no serious game.

É possível identificar, para a mesma frase, séries de tentativas com desempenhos muito diferentes. Por exemplo, para frases como "A água está fria." ou "O pássaro canta alto." encontram-se registos com 0% de acerto (produção incompreensível ou muito distante do estímulo) e outros com valores intermédios (25%, 40%, 50%, 75%) e elevados (valores próximos de 100%).

Situações semelhantes surgem em frases como "O sol brilha no céu." ou "Tenho muito sono hoje.", em que a progressão de tentativas ilustra tanto casos de **reconhecimento perfeito** como casos em que o modelo regista apenas **uma ou duas palavras corretas**.

Esta distribuição heterogénea cumpre **dois objetivos:**

- **Validação técnica** – confirma que o cálculo da percentagem de acerto é sensível ao número de palavras corretamente produzidas e que a métrica varia de forma coerente com a inspeção qualitativa das transcrições;

- **Potencial pedagógico** – evidencia que o jogo consegue devolver ao utilizador uma noção clara de sucesso ou dificuldade numa dada frase, permitindo monitorizar melhorias tentativa a tentativa.

Os relatórios PDF gerados pela aplicação, a partir dos mesmos registos, **sintetizam estes resultados através de estatísticas agregadas** (número de tentativas, média de acerto, total de palavras corretas) e de um gráfico de evolução que representa as últimas sessões. Mesmo com um número limitado de interações, **o gráfico permite visualizar oscilações e eventuais tendências de melhoria**, contribuindo para uma leitura rápida do desempenho global.

Em termos de robustez, a geração de relatórios mostrou-se **estável**: as estatísticas são calculadas sem erros, o número de tentativas é corretamente contabilizado e as tabelas das sessões correspondem, linha a linha, ao conteúdo de resultados.json. Também o mecanismo de filtragem por nível, presente na interface de relatórios, produz subconjuntos coerentes de tentativas sempre que existam registos associados à dificuldade selecionada.

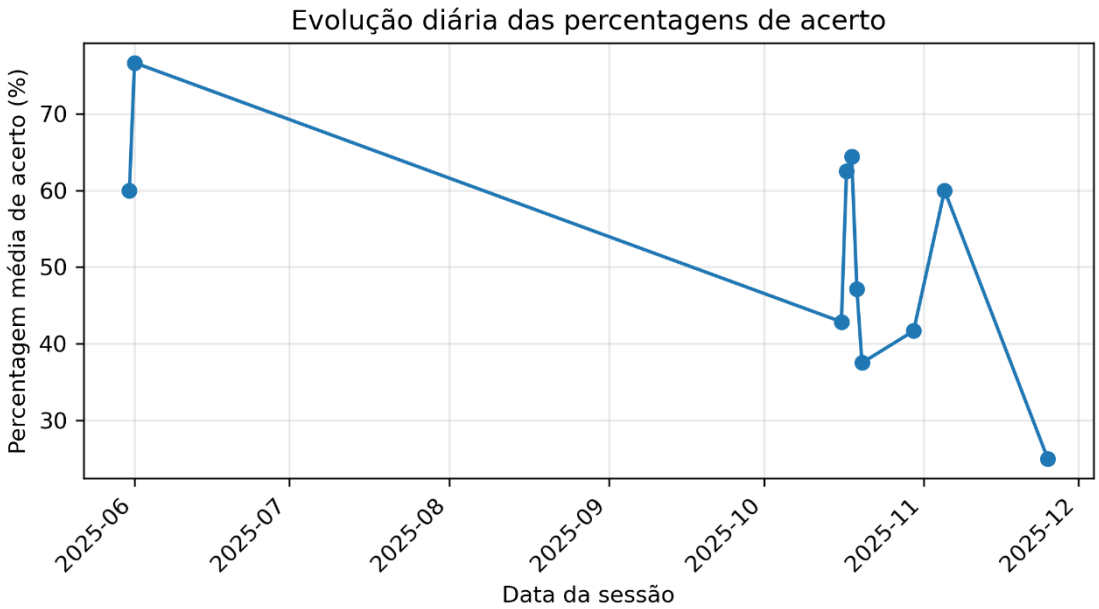


Figura 13 - Evolução temporal das percentagens de acerto ao longo das sessões de jogo.

Do ponto de vista técnico, os testes realizados confirmaram igualmente a **estabilidade do fluxo completo de utilização**, desde a seleção do nível até à geração dos relatórios.

A navegação entre ecrãs (menu inicial, seleção de níveis, ecrã de treino e ecrã de relatórios) ocorre de forma fluida, sem bloqueios ou encerramentos inesperados, a reprodução dos estímulos sonoros e a gravação da fala do utilizador decorrem dentro de tempos aceitáveis em ambiente local; e os testes de leitura e escrita em resultados.json e progresso.json, incluindo cenários de ficheiros inexistentes ou mal formatados, evidenciaram um comportamento robusto.

Ao longo do desenvolvimento foram detetados e corrigidos problemas pontuais de formatação de áudio, sincronização entre frase e ficheiro correspondente, tratamento de exceções na leitura de JSON e pequenos ajustes na disposição gráfica dos relatórios, não se verificando, na versão final, erros críticos durante a utilização prevista.

V.4 - Análise dos padrões fonéticos

Para além da percentagem global de acerto, o sistema implementa um módulo de análise fonética heurística, que procura **quatro tipos de padrão**:

- **nasalização reduzida** (dificuldades em vogais e segmentos nasais, como "ão", "õe", "am");
- **troca R→L** (substituição de consoantes vibrantes /ʁ/ ou /ɾ/ por laterais /l/);
- **substituição CH→S** (realização de /ʃ/ como /s/ em grafemas ch);
- **omissão de /s/ em posição final**, especialmente em plurais ou formas verbais terminadas em -s.

Nos registos analisados verificam-se, em particular, **ocorrências de substituições do tipo CH→S** em frases como "O ninho está cheio.", em que produções aproximadas a "seio" são detetadas e anotadas na estrutura analise fonética.

Noutros casos, apesar de a transcrição se afastar do alvo em vários pontos, as heurísticas não identificam um padrão dominante específico, sendo o campo dificuldade principal deixado a null.

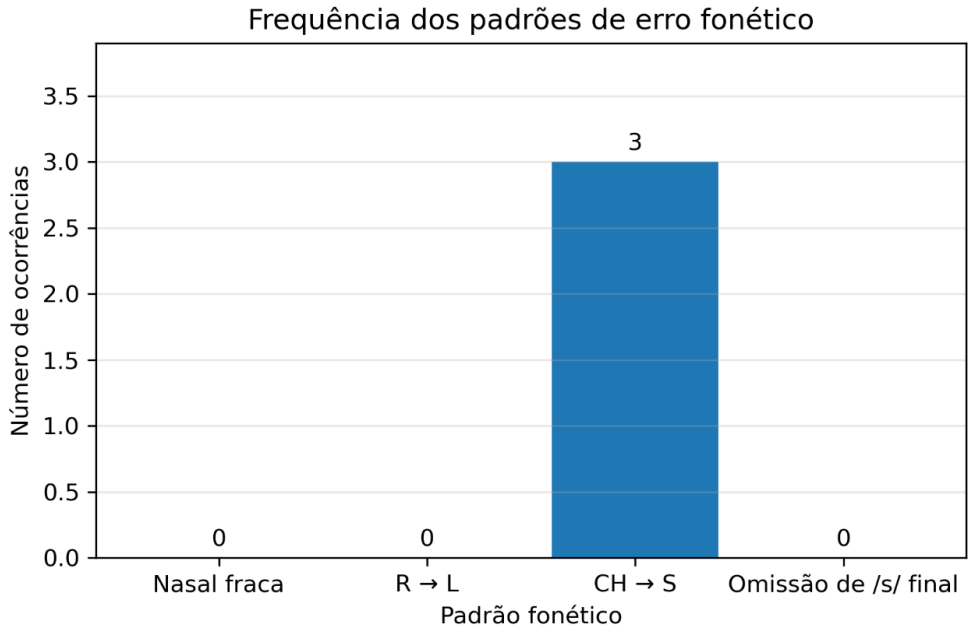


Figura 14 - Frequência relativa dos principais padrões de erro fonético detetados pelo sistema.

Do ponto de vista pedagógico, estes indicadores fonéticos têm duas funções complementares:

1. Enriquecer o feedback imediato, ajudando o utilizador a perceber, para além da percentagem global, em que tipo de sons tende a falhar;
2. Suportar a interpretação posterior, seja por parte de um docente ou terapeuta, ao oferecer uma síntese dos tipos de erro mais frequentes ao longo das sessões.

Embora o número de registos ainda não permita inferências fortes sobre a prevalência dos diferentes padrões, a avaliação demonstra que o mecanismo de análise é funcional, que as contagens são corretamente armazenadas em resultados.json e que os relatórios PDF conseguem integrar esta informação de forma textual e gráfica.

Esta componente aproxima o *serious game* de uma lógica de apoio à terapia da fala ou à prática fonética supervisionada, na medida em que ajuda a focalizar o treino em aspetos concretos da pronúncia.

V.5 - Discussão e limitações

A avaliação realizada permite afirmar que, dentro do âmbito definido para este trabalho e tendo em conta a natureza prototípica da solução, o *serious game* desenvolvido começa a cumprir os objetivos centrais propostos:

- **Funciona de forma estável** como *serious game* de treino de pronúncia em Português Europeu, integrando interface gráfica, captura de áudio, reconhecimento automático de fala, análise fonética e geração de relatórios;
- **Gera métricas de desempenho interpretáveis**, com percentagens de acerto que, na maioria dos casos analisados, se revelam coerentes com a observação qualitativa das transcrições e das produções de fala;
- **Regista e organiza dados de forma adequada à análise**, graças à estrutura JSON adotada e à articulação com relatórios PDF detalhados, permitindo reconstruir o percurso de treino do utilizador;
- **Oferece indicadores fonéticos adicionais** que, embora heurísticos e ainda exploratórios, constituem um primeiro passo no sentido de uma análise articulatória mais fina em contexto de *serious game*.

Não obstante estes resultados encorajadores, a avaliação apresenta **limitações importantes**, que condicionam o alcance das conclusões e devem ser explicitamente assumidas:

- **Volume e natureza dos dados**: O número de sessões registadas é reduzido e corresponde, em grande medida, a testes realizados pelo próprio autor em condições controladas. Isto significa que os dados não refletem ainda a diversidade de sotaques, ritmos de fala, idades e perfis de aprendizagem que se encontrariam num contexto real de utilização.

- **Dependência do desempenho do Whisper**: As métricas calculadas (percentagem de acerto e identificação de padrões fonéticos) dependem fortemente da qualidade da transcrição produzida pelo modelo Whisper. Ruído ambiente, microfones de baixa qualidade, variações prosódicas ou articulações muito rápidas podem introduzir erros de reconhecimento que o sistema interpreta como falhas de pronúncia, mesmo quando a produção do utilizador é, na prática, aceitável. Assim, parte dos desvios registados não resulta exclusivamente da pronúncia, mas também das limitações do próprio modelo de reconhecimento.

- **Condições de execução e capacidade de processamento**: O protótipo foi desenvolvido e testado em equipamento de características intermédias, sem recurso a aceleração dedicada (como GPU). Para garantir tempos de resposta aceitáveis, foi necessário optar por configurações de modelo mais leves e otimizar o código, o que implicou um compromisso entre precisão da transcrição, fluidez da interface e tempo de processamento, sobretudo quando se utiliza o dataset completo ou quando os relatórios agregam muitas tentativas.

- **Ausência de estudo controlado com utilizadores**: Não foi conduzido, nesta fase, um estudo experimental comparando, por exemplo, o desempenho de grupos com e sem acesso ao jogo, nem foram recolhidas medidas independentes de inteligibilidade junto de ouvintes humanos. Consequentemente, não é possível afirmar, com rigor estatístico, que o uso continuado do protótipo conduz a melhorias significativas na pronúncia; apenas se demonstra que a ferramenta é capaz de registar, medir e apresentar de forma organizada o desempenho do utilizador.

- **Cobertura parcial dos fenómenos fonéticos**: As heurísticas de análise fonética implementadas — centradas em fenómenos como nasalização reduzida, trocas /ʁ/–/ɾ/, substituição CH→S e omissão de "s" final — são úteis para sinalizar dificuldades frequentes, mas não substituem uma anotação fonética manual nem cobrem a totalidade dos contrastes relevantes do Português Europeu. Devem, por isso, ser

entendidas como um primeiro esboço de diagnóstico automático, passível de gerar falsos positivos e falsos negativos.

- **Interface ainda não adaptativa**: Embora a interface seja funcional, estável e adequada a um contexto de utilização individual, não inclui, nesta fase, mecanismos de adaptação automática — por exemplo, a seleção dinâmica de frases em função das dificuldades observadas, a personalização de níveis ou a recomendação de tarefas específicas para cada tipo de erro.

- **Questões éticas e de gestão de dados**: Do ponto de vista ético e de privacidade, o processamento local e a eliminação das gravações após a transcrição são medidas importantes. Contudo, em contextos multiutilizador ou institucionais, será necessário definir procedimentos formais de consentimento informado, políticas de retenção de dados e responsabilidades claras na sua gestão, sobretudo se os registos vierem a ser cruzados com informação identificável dos utilizadores.

Em síntese, a discussão dos resultados mostra que o protótipo é **tecnicamente viável e pedagogicamente promissor**, mas também que se encontra num estádio inicial de maturidade, condicionado por limitações de dados, de recursos computacionais e de profundidade da análise fonética.

Estas limitações não fragilizam o trabalho; pelo contrário, ajudam a que os resultados sejam lidos de forma **honesta e contextualizada**, delimitam o contexto em que as métricas devem ser interpretadas e abre caminho a desenvolvimentos futuros mais robustos, quer ao nível metodológico, quer ao nível tecnológico.

CAPÍTULO VI – CONCLUSÕES E TRABALHO FUTURO

VI.1 - Síntese do trabalho realizado

O trabalho desenvolvido teve como objetivo **conceber, implementar e avaliar** um protótipo funcional de *serious game* para treino de pronúncia em Português Europeu, dirigido a contextos educativos e de prática autónoma.

A solução proposta integra uma **interface gráfica** desenvolvida em Python com Pygame, um **módulo de reconhecimento automático de fala** baseado no modelo Whisper, **mecanismos de análise fonética heurística** e um sistema de **registo e tratamento de dados** suportado em ficheiros JSON e relatórios PDF gerados com a biblioteca ReportLab.

Ao longo do relatório foram descritas as principais etapas do processo:

1. o enquadramento teórico nos domínios dos *serious games*, da Aprendizagem de Línguas Assistida por Computador e do reconhecimento automático de fala;
2. a definição de requisitos funcionais e não funcionais; a construção do dataset de frases, organizado por níveis de dificuldade e alvos fonéticos;
3. o desenho da arquitetura modular da aplicação;
4. a implementação dos vários módulos;
5. e, por fim, a análise dos dados gerados durante as sessões experimentais de utilização.

O protótipo resultante permite ao utilizador:

- **selecionar níveis de dificuldade** organizados em três categorias (Fácil, Médio e Difícil);
- **ouvir um estímulo áudio** de referência em Português Europeu;

- **gravar** a sua própria produção;
- **obter uma transcrição automática** e uma percentagem de acerto calculada por palavras;
- **visualizar padrões de erro** fonético detetados;
- e **consultar relatórios PDF** que sintetizam, de forma gráfica e textual, o seu desempenho ao longo do tempo.

Embora se trate de um protótipo, e não de um produto final, o sistema demonstrou ser **tecnicamente estável e coerente com os objetivos definidos**, constituindo uma prova de conceito para a utilização articulada de ASR, *serious games* e análise fonética no treino da pronúncia de Português Europeu.

VI.2 - Principais contributos

Do ponto de vista científico e tecnológico, o trabalho desenvolvido oferece um conjunto de contributos que podem ser sintetizados em quatro eixos principais:

1. **Integração de ASR num *serious game* de pronúncia para Português Europeu**
   A aplicação concretiza a integração de um modelo de reconhecimento automático de fala robusto num contexto de *serious game* especificamente orientado para o Português Europeu. Esta integração permite transformar a prática oral, frequentemente pouco monitorizada, numa atividade acompanhada por métricas objetivas e registos historicamente consultáveis.
2. **Modelo de avaliação que combina métrica global e padrões fonéticos específicos**
   Para além de uma percentagem de acerto por palavras, o sistema incorpora heurísticas para identificar padrões de erro fonético recorrentes (por exemplo, nasalização reduzida, trocas entre /ʁ/ e /ɾ/, substituições CH→S, omissão de /s/ final). Esta camada adicional de análise aproxima o protótipo de ferramentas de apoio ao treino articulatório focalizado, indo além de uma simples comparação literal entre frases.

3. **Arquitetura modular, offline e baseada em tecnologias acessíveis**
   A arquitetura do sistema, organizada em módulos distintos para interface, áudio, reconhecimento de fala, análise fonética, persistência de dados e relatórios, demonstra que é possível construir um ambiente de treino sofisticado recorrendo a tecnologias abertas, multiplataforma e executáveis em modo totalmente local. Esta característica é particularmente relevante para contextos educativos com restrições de conectividade ou de políticas de dados.

4. **Sistema de relatórios orientado à monitorização pedagógica**
   A geração automática de relatórios PDF por nível e de um relatório geral, com estatísticas agregadas, gráficos de evolução e tabelas de sessões recentes, constitui um contributo concreto para a monitorização do progresso. Estes relatórios podem ser utilizados pelo próprio aprendente, por docentes ou por terapeutas da fala como suporte à análise das dificuldades e à tomada de decisões sobre estratégias de treino.

Em conjunto, estes contributos mostram que o *serious game* desenvolvido não é apenas um protótipo técnico, mas uma proposta de solução articulada para um problema real: a escassez de ferramentas específicas para treino de pronúncia em Português Europeu com feedback automatizado e registo estruturado de desempenho.

VI.3 - Limitações assumidas

Tal como discutido no capítulo de avaliação, o trabalho apresenta um conjunto de limitações que importa reafirmar de forma sintética, tanto por honestidade científica como para balizar o âmbito das conclusões:

- **Escala e diversidade reduzidas dos dados:**

  O volume de registos em resultados.json é ainda limitado e concentra-se sobretudo em sessões de teste realizadas pelo autor. A ausência de utilização extensiva por parte de diferentes perfis de aprendentes (diversas línguas maternas, faixas etárias,

contextos acústicos) impede generalizações fortes sobre o comportamento do sistema em cenário real.

- **Dependência de um modelo de reconhecimento genérico:**

  O desempenho das métricas calculadas está condicionado pela qualidade das transcrições produzidas pelo modelo Whisper, que não foi especificamente treinado para o conjunto fechado de frases do jogo nem para aprendentes de Português Europeu enquanto L2. Em determinados casos, erros de reconhecimento podem ser interpretados como erros de pronúncia, afetando a exatidão da avaliação.

- **Limitações de recursos computacionais e compromisso entre precisão e fluidez:**

  A opção por uma solução totalmente offline, executada em hardware de capacidade intermédia, exigiu compromissos entre precisão máxima do modelo, tempo de resposta e fluidez da interface. Modelos mais pesados, potencialmente mais rigorosos, não foram considerados viáveis no contexto de desenvolvimento e utilização previsto.

- **Cobertura parcial da dimensão fonética:**

  As heurísticas de análise fonética focam-se em um conjunto restrito de fenómenos segmentais. Aspetos como entoação, ritmo, acentuação frásica ou subtilezas vocálicas não são ainda captados, o que limita a abrangência do diagnóstico fonético.

- **Ausência de validação pedagógica sistemática:**

  Não foi realizado um estudo controlado com grupos de aprendentes, comparando, por exemplo, a evolução da pronúncia com e sem utilização do jogo ao longo de um período significativo. As conclusões apresentadas têm, por isso, um caráter funcional e exploratório, e não experimental no sentido estatisticamente robusto.

Estas limitações não invalidam os contributos alcançados, mas lembram que o protótipo se encontra numa fase inicial de maturidade, devendo ser encarado como base para desenvolvimentos subsequentes e para estudos empíricos mais alargados.

VI.4 - Linhas de trabalho futuro

As limitações identificadas e os resultados obtidos sugerem diversas direções promissoras para trabalho futuro, tanto em termos técnicos como pedagógicos:

1. **Alargamento da base de utilizadores e estudos em contexto real:**

   Uma primeira linha de continuidade consiste em disponibilizar o jogo em contextos reais de ensino (por exemplo, aulas de Português Língua Estrangeira, cursos de formação específica ou clínicas de terapia da fala) e recolher dados de um número alargado de utilizadores. Isto permitiria:

   - caracterizar a utilização do sistema em cenários heterogéneos;
   - avaliar a aceitação da interface e do tipo de feedback;
   - medir, com instrumentos externos, o impacto do jogo na melhoria efetiva da pronúncia.

2. **Especialização e afinação do reconhecimento de fala:**

   Uma segunda direção passa pela exploração de modelos ASR especializados, treinados ou ajustados para o domínio do Português Europeu e para o conjunto de frases do jogo. Poderá considerar-se:

   - o *fine-tuning* de modelos existentes com corpora focados em PE;
   - a utilização de técnicas de alinhamento forçado para introduzir informação fonética mais fina;
   - a combinação de modelos de diferente complexidade, adaptando-os à capacidade de processamento disponível.

3. **Expansão do modelo de análise fonética:**

   Outra linha de evolução consiste em alargar o conjunto de fenómenos fonéticos analisados, incluindo aspetos prosódicos (ritmo, entoação), contrastes vocálicos mais subtis e padrões específicos de determinados grupos de aprendentes. Uma possível abordagem será articular heurísticas baseadas em texto com informação adicional sobre a dinâmica temporal do sinal de áudio.

4. **Mecanismos adaptativos de progressão e recomendação:**

   Em termos de design de jogo, seria pertinente introduzir mecanismos de adaptação automática que, com base nos registos de desempenho, selecionem frases mais adequadas ao perfil do utilizador, recomendem tarefas adicionais para padrões de erro frequentes ou ajustem a dificuldade de forma dinâmica. Tal permitiria evoluir de um percurso linear de níveis para uma experiência verdadeiramente personalizada.

5. **Interfaces de acompanhamento para docentes e terapeutas:**

   A informação hoje sintetizada em relatórios PDF poderia ser disponibilizada num painel de controlo específico para profissionais, com filtros por utilizador, por padrão fonético ou por período temporal. Esta camada adicional facilitaria a integração do jogo em planos de intervenção mais amplos e permitiria cruzar dados de sessão com observações qualitativas em sala de aula ou gabinete.

6. **Distribuição multiplataforma e refinamento da experiência de utilização:**

   Por fim, a evolução da aplicação poderá incluir o empacotamento do jogo sob a forma de executáveis simples de instalar, bem como a exploração de versões adaptadas a outros dispositivos (por exemplo, portáteis de baixo custo ou *tablets*) quando tal for tecnicamente viável, sempre garantindo a preservação dos princípios de privacidade dos dados de voz.

VI.5 - Considerações finais

Em síntese, o trabalho realizado demonstra que é possível conceber um *serious game* de treino de pronúncia em Português Europeu que combine reconhecimento automático de fala, análise fonética orientada e mecanismos de registo e visualização de desempenho. O protótipo desenvolvido **cumpre o papel de prova de conceito** e mostra a **viabilidade técnica e o potencial pedagógico** desta abordagem.

Ao mesmo tempo, o relatório assume explicitamente os **limites do caminho percorrido**: a escala reduzida da avaliação, a dependência de um modelo ASR genérico, as restrições de processamento e a cobertura parcial da complexidade fonética do Português Europeu. A consciência destas limitações é condição para que o trabalho possa ser interpretado de forma justa e servir de base sólida para investigações e desenvolvimentos futuros.

Desta forma, o projeto contribui, ainda que de modo inicial, para o ecossistema de ferramentas dedicadas ao ensino e treino do Português Europeu, apontando para um cenário em que *serious games* suportados por tecnologias de fala possam desempenhar um papel relevante na promoção de práticas de pronúncia mais frequentes, mais orientadas e melhor documentadas.

BIBLIOGRAFIA

Amrate, M. (2023). Computer-assisted pronunciation training: A systematic review. *ReCALL*. Disponível em

https://www.cambridge.org/core/journals/recall/article/computerassisted-pronunciation-training-a-systematic-review/71E786F7DFC99727837909FDED7A2320

Bhat, V., & Bhattacharyya, P. (2023). DISCO: A large scale human annotated corpus for disfluency correction in Indo-European languages. *Findings of the Association for Computational Linguistics: EMNLP 2023*. Disponível em https://aclanthology.org/2023.findings-emnlp.855/

Bhat, V., Darji, A., Chinthakindi, S., Vadehra, B., & Bhattacharyya, P. (2023). DisfluencyFixer: A tool to enhance language learning through speech-to-speech disfluency correction. *Proceedings of Interspeech 2023*. Disponível em https://www.isca-archive.org/interspeech_2023/bhat23_interspeech.pdf

Celce-Murcia, M., Brinton, D. M., & Goodwin, J. M. (2010). *Teaching pronunciation: A course book and reference guide* (2nd ed.). Cambridge University Press. Disponível em https://scispace.com/pdf/teaching-pronunciation-a-course-book-and-reference-guide-mx78rjrl5z.pdf

Chapelle, C. A. (2001). *Computer applications in second language acquisition: Foundations for teaching, testing and research*. Cambridge University Press. Disponível em https://catdir.loc.gov/catdir/samples/cam031/2001269254.pdf

Chapelle, C. A. (2009). The relationship between second language acquisition theory and computer-assisted language learning. In M. H. Long & C. J. Doughty (Eds.), *The handbook of language teaching*. Wiley-Blackwell. **Codingal.** (2023). Curso "Pygame course for kids". *Codingal*. Disponível em https://www.codingal.com/courses/pygame/

Codingal. (2023). The ultimate guide to Pygame. *Codingal Blog*. Disponível em https://www.codingal.com/coding-for-kids/blog/the-ultimate-guide-to-pygame/

Cucchiarini, C., & Strik, H. (2017). Automatic speech recognition for second language pronunciation training. In O. Kang, R. I. Thomson, & J. M. Murphy (Eds.), *The Routledge*

handbook of contemporary English pronunciation (pp. 556–571). Routledge. Disponível em https://www.taylorfrancis.com/chapters/10.4324/9781315145006-35/

Davies, G. (2003). Computer assisted language learning: Where are we now and where are we going? In J. Egbert & E. Hanson-Smith (Eds.), *CALL environments: Research, practice, and critical issues*. TESOL. Disponível em https://doi.org/10.1075/lllt.4

Derwing, T. M., & Munro, M. J. (2005). Second language accent and pronunciation teaching: A research-based approach. In J. Frodesen & C. Holten (Eds.), *The power of context in language teaching and learning* (pp. 179–191). Thomson Heinle. Disponível em https://www.researchgate.net/publication/231911026_Second_language_accent_and_pronunciation_teaching_A_research-based_approach

Derwing, T. M., & Munro, M. J. (2015). *Pronunciation fundamentals: Evidence-based perspectives for L2 teaching and research*. John Benjamins. Disponível em https://www.jbe-platform.com/content/books/9789027268716

Dierbach, C. (2012). *Introduction to computer science using Python: A computational problem-solving focus*. Wiley. Disponível em https://dl.acm.org/doi/10.5555/2591769

Egbert, J. (2005). *CALL essentials: Principles and practice in CALL classrooms*. TESOL. Disponível em https://www.amazon.com/CALL-Essentials-Principles-Practices-Classrooms/dp/1931185158

Egbert, J., & Hanson-Smith, E. (Eds.). (1999). *CALL environments: Research, practice, and critical issues*. TESOL. Disponível em https://benjamins.com/catalog/lllt.4

Electronstudio. (2023). *Coding games with Pygame Zero & Python*. Disponível em https://electronstudio.github.io/pygame-zero-book/

Graham, S. A., & Roll, K. (2024). Evaluating OpenAI's Whisper ASR: Performance analysis across diverse accents and speaker traits. *JASA Express Letters*. Disponível em https://pubmed.ncbi.nlm.nih.gov/38391582/

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. Disponível em https://aisel.aisnet.org/misq/vol28/iss1/6/

IEEE Spectrum. (2024). The top programming languages 2024. *IEEE Spectrum*. Disponível em https://spectrum.ieee.org/top-programming-languages-2024

Johnson, W. L. (2005). Lessons learned from games for education. *SIGGRAPH 2005 Educators Program*. Disponível em https://history.siggraph.org/wp-content/uploads/2022/07/2005-Educators-Forum-Johnson_Lessons-Learned.pdf

Kruglyk, V. A., & Lvov, M. S. (2012). Choosing the first educational programming language. In *ICT in Education, Research and Industrial Applications* (pp. 188–198). Disponível em https://ceur-ws.org/Vol-848/ICTERI-2012-CEUR-WS-paper-37-p-188-198.pdf

Lago-Ferreiro, A., Gómez-González, M. Á., & López-Ardao, J. C. (2025). A new serious game (e-SoundWay) for learning English phonetics. *Multimodal Technologies and Interaction*, 9(6), 54. Disponível em https://www.mdpi.com/2414-4088/9/6/54

Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, 36(6), 47–56. DOI: https://doi.org/10.1109/MC.2003.1204375

Levy, M. (1997). *Computer-assisted language learning: Context and conceptualization*. Oxford University Press. Disponível em https://www.routledge.com/Computer-Assisted-Language-Learning-Context-and-Conceptualization/Levy/p/book/9780194328145

Levis, J. M. (2007). Computer technology in teaching and researching pronunciation. *Annual Review of Applied Linguistics*, 27, 184–202. Disponível em https://jlevis.public.iastate.edu/ARAL2007.pdf

Lvov, M. S., & Kruglyk, V. A. (2014). Teaching algorithmization and programming using Python language. *Information Technologies and Learning Tools*. Disponível em https://www.researchgate.net/publication/287849240_Teaching_algorithmization_and_programming_using_Python_language

Malik, M. I., Ahmad, M., & Farooq, M. (2021). Automatic speech recognition: A survey. *International Journal of Computer Applications*. Disponível em https://ieeexplore.ieee.org/document/9418377

Michael, D., & Chen, S. (2006). *Serious games: Games that educate, train, and inform*. Thomson Course Technology. Disponível em https://books.google.com/books/about/Serious_Games.html?id=49kTAQAAIAAJ

Mohsen, M. A., Abdel Latif, M. M., & Hussein, R. (2025). A scientometric study of computer-assisted pronunciation training in second language acquisition: Technological affordances and research trends. *Humanities and Social Sciences Communications*. Disponível em https://www.nature.com/articles/s41599-025-04474-y

Munro, M. J., & Derwing, T. M. (1995). Foreign accent, comprehensibility, and intelligibility in the speech of second language learners. *Language Learning*, 45(1), 73–97. Disponível em https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9922.1995.tb00963.x

Neri, A., Cucchiarini, C., Strik, H., & Boves, L. (2002). Automatic speech recognition for second language learning: How and why it actually works. *Proceedings of the International Symposium on Applied Phonetics*. Disponível em https://repository.ubn.ru.nl/bitstream/handle/2066/76220/76220.pdf

Pennington, M. C., & Rogerson-Revell, P. (2019). *English pronunciation teaching and research: Contemporary perspectives*. Palgrave Macmillan. Disponível em https://books.google.com/books?id=1VptDwAAQBAJ

Pressman, R. S., & Maxim, B. R. (2014). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill. Disponível em https://www.mheducation.com/highered/product/software-engineering-pressman-maxim/M9780078022128.html

Pygame. (s.d.). Pygame [Website]. Disponível em https://www.pygame.org/

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. *Proceedings of the 40th International Conference on Machine Learning*. Disponível em https://arxiv.org/abs/2212.04356

Real Python. (2019). Pygame: A primer on game programming in Python. *Real Python*. Disponível em https://realpython.com/pygame-a-primer/

Regulamento (UE) 2016/679 do Parlamento Europeu e do Conselho, de 27 de abril de 2016, relativo à proteção das pessoas singulares no que diz respeito ao tratamento de dados pessoais e à livre circulação desses dados (Regulamento Geral sobre a Proteção de Dados – RGPD). *Jornal Oficial da União Europeia*, L 119, 1–88. Disponível em https://eur-lex.europa.eu/legal-content/PT/TXT/PDF/?uri=CELEX:32016R0679

Robson, C., & McCartan, K. (2016). *Real world research* (4th ed.). Wiley. Disponível em https://www.wiley.com/en-gb/Real%2BWorld%2BResearch%2C%2B4th%2BEdition-p-9781119144854

Rogerson-Revell, P. (2021). Teaching pronunciation. In S. Dekeyser & R. Ellis (Eds.), *The Routledge handbook of instructed second language acquisition* (2nd ed.). Routledge.

Sadigzade, Z. (2025). Dialect diversity and social change: New approaches in sociolinguistics. *Journal of Azerbaijan Language and Education Studies*, 2(3), 91–107. Disponível em https://portasapientia.com/index.php/JALES/article/view/27

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson. Disponível em https://software-engineering-book.com/

Tejedor-García, C., Aizawa, F., López, O., García-Serrano, A., & Riaza, M. (2020). Assessing pronunciation improvement in students of English using a controlled computer-assisted pronunciation tool. *IEEE Transactions on Learning Technologies*. DOI: https://doi.org/10.1109/TLT.2020.2980261

Thomson, R. I., & Derwing, T. M. (2015). The effectiveness of L2 pronunciation instruction: A narrative review. *Applied Linguistics*, 36(3), 326–344. DOI: https://doi.org/10.1093/applin/amu076

Vančová, H. (2023). AI and AI-powered tools for pronunciation training. *Journal of Language and Cultural Education*. Disponível em https://sciendo.com/2/v2/download/article/10.2478/jolace-2023-0022.pdf

Warschauer, M. (1996). Computer-assisted language learning: An introduction. In S. Fotos (Ed.), *Multimedia language teaching* (pp. 3–20). Logos. Disponível em https://www.researchgate.net/publication/247230653_Computer-Assisted_Language_Learning_An_Introduction

Wouters, P., van Nimwegen, C., van Oostendorp, H., & van der Spek, E. (2013). A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, 105(2), 249–265. Disponível em https://www.researchgate.net/publication/257719666_A_Meta-Analysis_of_the_Cognitive_and_Motivational_Effects_of_Serious_Games

Xinogalos, S. (2020). Learning programming using Python: The case of the DigiWorld educational game. *European Journal of Engineering and Formal Sciences*. Disponível em https://eu-opensci.org/index.php/ejeng/article/view/62750

APÊNDICES

Apêndice I – Dataset de frases do jogo de pronúncia

Este apêndice apresenta a organização do dataset de frases utilizado no protótipo de *serious game* desenvolvido neste trabalho. As frases encontram-se armazenadas, em formato digital, no ficheiro dataset_frases.json, localizado na pasta assets/dataset/ do projeto. Cada entrada do dataset inclui o texto da frase, o nível de dificuldade (Fácil, Médio ou Difícil), um identificador único e o nome do ficheiro de áudio associado. No corpo do relatório foi apresentada uma amostra deste dataset (Tabela 1); nas tabelas seguintes disponibilizam-se exemplos adicionais por nível de dificuldade, ilustrando a forma como os dados estão estruturados.

Tabela 2 - Exemplos de frases do nível Fácil

Id	Frase	Nível	Ficheiro_áudio
1	A água está fria.	Fácil	F_01.mp3
2	Tenho muito sono hoje.	Fácil	F_02.mp3
3	Bebi leite ao almoço.	Fácil	F_03.mp3
4	O ouro é teu.	Fácil	F_04.mp3
5	Para ali, por favor.	Fácil	F_05.mp3

Nota: o dataset completo do nível Fácil encontra-se no ficheiro dataset_frases.json.

Tabela 3 - Exemplos de frases do nível Médio.

Id	Frase	Nível	Ficheiro_áudio
1	Quero pão quente agora.	Médio	M_01.mp3
2	Os cães estão a dormir.	Médio	M_02.mp3
3	Ele põe o casaco.	Médio	M_03.mp3
4	Comem pão com manteiga.	Médio	M_04.mp3
5	As mesas estão limpas.	Médio	M_05.mp3

Nota: o dataset completo do nível Médio encontra-se no ficheiro dataset_frases.json.

Tabela 4 - Exemplos de frases do nível Difícil.

Id	Frase	Nível	Ficheiro_áudio
1	Dá-me um segundo, por favor.	Avançado	A_01.mp3
2	Vou-te ligar mais tarde?	Avançado	A_02.mp3
3	Os olhos dela brilham.	Avançado	A_03.mp3
4	O carro vermelho ruiu.	Avançado	A_04.mp3
5	Rua larga, ruído forte.	Avançado	A_05.mp3

Nota: o dataset completo do nível Difícil encontra-se no ficheiro dataset_frases.json.

APÊNDICE II - ESTRUTURA DOS FICHEIROS DE RESULTADOS (JSON)

Este apêndice descreve a estrutura dos principais ficheiros de dados gerados automaticamente pelo protótipo, em formato JSON. Estes ficheiros são utilizados para registar as tentativas de pronúncia realizadas no jogo e para guardar o estado de progresso do utilizador ao longo dos diferentes níveis. A informação aqui apresentada complementa a descrição do módulo de armazenamento feita no Capítulo III e serve de referência para futuros desenvolvimentos ou integrações com outras aplicações.

II.1 – resultados.json

O ficheiro resultados.json contém uma lista de registos, em que cada elemento corresponde a uma tentativa de pronúncia realizada no jogo. Cada registo é um objeto JSON com os campos apresentados na Tabela 5.

Tabela 5 - Estrutura de cada registo no ficheiro resultados.json.

Campo	Tipo	Descrição
data	string	Data e hora em que a tentativa foi registada (AAAA-MM-DD HH:MM).
frase_esperada	string	Frase apresentada ao utilizador como modelo de pronúncia.
frase_dita	string	Transcrição da pronúncia do utilizador obtida pelo modelo Whisper.
percentagem_acerto	número	Percentagem de acerto (0–100) calculada a partir de palavras_corretas / total_palavras.
palavras_corretas	número	Número de palavras da frase de referência consideradas corretas.
total_palavras	número	Número total de palavras da frase de referência.

nível	string	Nível de dificuldade em que a tentativa foi realizada (facil, medio, avancado).
item_id	string	Identificador da frase no dataset (por exemplo "F-001", "M-002"). *(opcional)*
targets	lista	Lista de alvos fonéticos associados à frase (ex.: "lh", "nh", "ditongo_ei"). *(opcional)*
analise_fonetica	objeto	Estrutura com contagens dos padrões fonéticos detetados e indicação da dificuldade principal. *(opcional)*

A Listagem que se segue apresenta um exemplo simplificado de registo contido em resultados.json (valores ilustrativos).

```
{
  "data": "2025-11-05 22:37",
  "frase_esperada": "O ninho está cheio.",
  "frase_dita": "O ninho está seio.",
  "percentagem_acerto": 75.0,
  "palavras_corretas": 3,
  "total_palavras": 4,
  "nível": "facil",
  "item_id": "F-008",
  "targets": ["nh"],
  "analise_fonetica": {
    "contagens": {
      "nasal_fraco": 0,
      "R->L": 0,
      "CH->S": 1,
      "omissao_S_final": 0
```

```
    },
    "total_tokens_ref": 4,
    "dificuldade_principal": "CH->S"
  }
}
```

II.2 – progresso.json

O ficheiro progresso.json organiza-se em três blocos principais ("facil", "medio" e "avancado"). Em cada bloco, as chaves "0", "1", "2", … representam os índices dos níveis, e os respetivos valores inteiros variam entre 0 e 3, correspondendo ao número de "estrelas" atribuídas a esse nível.

A Listagem que se segue apresenta um exemplo simplificado da estrutura de progresso.json.

```
{
  "facil": {
    "0": 3,
    "1": 2,
    "2": 1
  },
  "medio": {
    "0": 1
  },
  "avancado": {
    "0": 2
  }
}
```

Neste exemplo, o utilizador obteve três estrelas nos níveis 0, 1 e 2 do modo Fácil, uma estrela no nível 0 do modo Médio e duas estrelas no nível 0 do modo Avançado, refletindo o desempenho alcançado em cada um desses níveis.

APÊNDICE III – EXCERTOS DE CÓDIGO DA APLICAÇÃO DE TREINO DE PRONÚNCIA

Neste apêndice apresentam-se alguns excertos de código-fonte da aplicação desenvolvida. Os trechos seguintes estão escritos em Python e correspondem às funções centrais de avaliação da pronúncia, análise fonética, registo de resultados e preparação dos relatórios automáticos referidos no Capítulo III.

III.1 – Avaliação da pronúncia

A função avaliar_pronuncia recebe a frase de referência e o texto reconhecido pelo motor Whisper, calcula métricas de desempenho (WER, CER e percentagem de acerto), marca as palavras corretas/incorretas para feedback visual e devolve um resumo em texto e um número de estrelas (0–3).

```python
def avaliar_pronuncia(frase_esperada, resultado_fala):
    """
    Avaliação robusta:
      - normaliza acentos/caixa, calcula WER/CER (%ERRO) e converte em %ACERTO, marca palavras corretas/erradas para colorir no UI
    Retorna: (lista_palavras, feedback_str, estrelas_int)
    """
    # alinhamento por palavras (mantém a ordem do que foi dito)
    lista_palavras, corretas, total = word_alignment_correct(frase_esperada, resultado_fala)

    # métricas
    wer = wer_percent(frase_esperada, resultado_fala)     # %ERRO
    cer = cer_percent(frase_esperada, resultado_fala)     # %ERRO
    acerto = max(0.0, 100.0 - wer)                        # %ACERTO por palavras

    # estrelas com base no ACERTO
    if acerto >= 95:
        estrelas = 3
    elif acerto >= 70:
        estrelas = 2
    elif acerto >= 40:
        estrelas = 1
    else:
        estrelas = 0

    sugest = top_palavras_a_melhorar(frase_esperada, resultado_fala, k=2)
    dicas_txt = f" • Foca em: {', '.join(sugest)}" if sugest else ""

    feedback = (
        f"Acerto: {acerto:.1f}%  •  WER: {wer:.1f}%  •  CER: {cer:.1f}%  "
        f"•  Palavras corretas: {corretas}/{total}{dicas_txt}"
    )

    return lista_palavras, feedback, estrelas
```

Figura 15 - Função de avaliação da pronúncia (percentagens, estrelas e feedback textual).

III.2 – Helpers de normalização e alinhamento (Funções Auxiliares)

Para que a avaliação seja consistente, o sistema normaliza os textos (remoção de acentos, caixa, espaços), calcula distâncias de Levenshtein e implementa funções de WER/CER e de alinhamento palavra a palavra, usadas na função anterior.

```python
# Palavras de função com peso reduzido na avaliação (português)
STOPWORDS_PT = {
    "o","a","os","as","um","uma","uns","umas",
    "de","do","da","dos","das","em","no","na","nos","nas",
    "e","ou","mas","que","se","com","por","para","ao","à","aos","às",
    "é","foi","era","ser","estar","está","são","era","era","há",
    "um","uma","me","te","se","lhe","nos","vos","lhes"
}

def strip_accents(s: str) -> str:
    """remove acentos/diacríticos e normaliza espaços."""
    nfkd = unicodedata.normalize("NFKD", s)
    s2 = "".join(ch for ch in nfkd if not unicodedata.combining(ch))
    return " ".join(s2.lower().strip().split())

def tokenize_words(s: str):
    return [t for t in strip_accents(s).split(" ") if t]

def levenshtein(a, b):
    """distância de Levenshtein para listas OU strings."""
    # transforma em listas de unidades
    if isinstance(a, str) and isinstance(b, str):
        a, b = list(a), list(b)
    n, m = len(a), len(b)
    if n == 0: return m
    if m == 0: return n
    dp = [list(range(m+1))]
    for i in range(1, n+1):
        dp.append([i] + [0]*m)
    for i in range(1, n+1):
        for j in range(1, m+1):
            cost = 0 if a[i-1] == b[j-1] else 1
            dp[i][j] = min(
                dp[i-1][j] + 1,        # remoção
                dp[i][j-1] + 1,        # inserção
                dp[i-1][j-1] + cost    # substituição
            )
    return dp[n][m]
```

Figura 16 - Funções de normalização, cálculo de WER/CER e alinhamento palavra a palavra. (Pt. 1)

```python
def cer_percent(ref: str, hyp: str) -> float:
    """Character Error Rate → devolve %ERRO (0..100)."""
    ref_c = strip_accents(ref).replace(" ", "")
    hyp_c = strip_accents(hyp).replace(" ", "")
    if len(ref_c) == 0:
        return 0.0 if len(hyp_c) == 0 else 100.0
    d = levenshtein(ref_c, hyp_c)
    return 100.0 * d / len(ref_c)

def wer_percent(ref: str, hyp: str) -> float:
    """Word Error Rate → devolve %ERRO (0..100)."""
    r = tokenize_words(ref)
    h = tokenize_words(hyp)
    if len(r) == 0:
        return 0.0 if len(h) == 0 else 100.0
    d = levenshtein(r, h)
    return 100.0 * d / len(r)

def word_alignment_correct(ref: str, hyp: str):
    """
    Alinha por palavras (após normalização) e devolve:
    - lista [(palavra_ouvida, correta_bool)], nº corretas, total de palavras de referência
    """
    import difflib
    r = tokenize_words(ref)
    h = tokenize_words(hyp)
    seq = difflib.SequenceMatcher(None, r, h)
    saida = []
    corretas = 0
    for tag, i1, i2, j1, j2 in seq.get_opcodes():
        if tag == "equal":
            for w in h[j1:j2]:
                saida.append((w, True))
                corretas += 1
        else:
            for w in h[j1:j2]:
                saida.append((w, False))
    total = len(r)
    return saida, corretas, total
```

Figura 17 - Funções de normalização, cálculo de WER/CER e alinhamento palavra a palavra. (Pt. 2)

```python
def top_palavras_a_melhorar(ref: str, hyp: str, k: int = 2):
    """
    Heurística: palavras de conteúdo da referência não observadas/corretas.
    Devolve até k sugestões.
    """
    r = tokenize_words(ref)
    h = tokenize_words(hyp)
    faltas = []
    for w in r:
        if w in STOPWORDS_PT:  # menor prioridade
            continue
        if w not in h:
            faltas.append(w)
    # se vazio, procura palavras com mais diferença de letras vs hipótese
    if not faltas:
        candidatos = []
        for w in r:
            best = min((levenshtein(w, x) for x in h), default=len(w))
            candidatos.append((best, w))
        candidatos.sort(reverse=True)
        faltas = [w for _, w in candidatos]
    return faltas[:k]
```

Figura 18 - Funções de normalização, cálculo de WER/CER e alinhamento palavra a palavra. (Pt. 3)

III.3 – Análise fonética heurística (Função de análise de padrões fonéticos)

A função analisar_fonetica implementa um conjunto de heurísticas simples orientadas ao Português Europeu, procurando padrões de erro frequentes (nasalização reduzida, troca R→L, CH→S e omissão de /s/ final). O resultado é um dicionário com contagens por padrão e a dificuldade dominante usada nos relatórios.

```python
NASAL_MARKERS = ("ã", "õ", "ãe", "õe", "ão")

NASAL_ENDINGS = ("am","an","em","en","im","in","om","on","um","un","ão","õe")

def _is_nasal_token(tok: str) -> bool:
    s = strip_accents(tok)
    if any(x in tok for x in NASAL_MARKERS):
        return True
    if any(s.endswith(e) for e in NASAL_ENDINGS):
        return True
    return False

def _r_to_l_variant(w: str) -> str:
    # troca r/rr por l (líquidas)
    return w.replace("rr","l").replace("r","l")

def _ch_to_s_variant(w: str) -> str:
    return w.replace("ch","s")

def _final_s_omission(w: str) -> str:
    return w[:-1] if w.endswith("s") else w

def _similar(a: str, b: str, max_ed=1) -> bool:
    return levenshtein(a, b) <= max_ed

def analisar_fonetica(ref: str, hyp: str) -> dict:
    """
    Devolve um dicionário com contagens por padrão e um campo 'dificuldade_principal'.
    Formato:
      {
        "contagens": {"nasal_fraco": 3, "R->L": 2, "CH->S": 1, "omissao_S_final": 4},
        "total_tokens_ref": N,
        "dificuldade_principal": "nasal_fraco"
      }
    """
    r = tokenize_words(ref)
    h = tokenize_words(hyp)

    H = set(h)
```

Figura 19 - Implementação das heurísticas de análise fonética para identificação de padrões de erro. (Pt. 1)

```
    counts = {"nasal_fraco":0, "R->L":0, "CH->S":0, "omissao_S_final":0}
    # 1) Nasalização:
    for w in r:
        if _is_nasal_token(w):
            base = strip_accents(w).replace("m","").replace("n","")
            if any(_similar(strip_accents(x), base, max_ed=1) for x in h):
                counts["nasal_fraco"] += 1

    # 2) R -> L
    for w in r:
        if "r" in w:
            var = _r_to_l_variant(w)
            if var in H or any(_similar(x, var, 1) for x in h):
                counts["R->L"] += 1

    # 3) CH -> S
    for w in r:
        if "ch" in w:
            var = _ch_to_s_variant(w)
            if var in H or any(_similar(x, var, 1) for x in h):
                counts["CH->S"] += 1

    # 4) Omissão de 'S' final
    for w in r:
        if w.endswith("s"):
            var = _final_s_omission(w)
            if var in H or any(_similar(x, var, 0) for x in h):
                counts["omissao_S_final"] += 1

    # dificuldade principal = o padrão com maior contagem (se >0)
    maior = max(counts, key=counts.get)
    dificuldade = maior if counts[maior] > 0 else None

    return {
        "contagens": counts,
        "total_tokens_ref": len(r),
        "dificuldade_principal": dificuldade
    }
```

Figura 20 - Implementação das heurísticas de análise fonética para identificação de padrões de erro. (Pt. 2)

III.4 – Registo de resultados e integração com relatórios

Após cada tentativa, o sistema regista num ficheiro JSON e num ficheiro CSV os principais dados da sessão (data, frase esperada, frase dita, percentagem de acerto, nível, identificadores do item e análise fonética). Estes registos são a base dos relatórios PDF descritos no Capítulo III.

```python
def guardar_resultado(frase_esperada, frase_dita, percentagem,
                      palavras_certas, total_palavras, nivel,
                      item_id=None, item_targets=None, analise_fonetica=None):
    resultado = {
        "data": datetime.now().strftime("%Y-%m-%d %H:%M"),
        "frase_esperada": frase_esperada,
        "frase_dita": frase_dita,
        "percentagem_acerto": round(percentagem, 2),
        "palavras_corretas": palavras_certas,
        "total_palavras": total_palavras,
        "nível": nivel
    }
    if item_id is not None:
        resultado["item_id"] = item_id
    if item_targets:
        resultado["targets"] = list(item_targets)
    if analise_fonetica:
        resultado["analise_fonetica"] = analise_fonetica

    # Guarda no ficheiro JSON
    ficheiro_json = "resultados.json"
    if not os.path.exists(ficheiro_json):
        with open(ficheiro_json, "w", encoding="utf-8") as f:
            json.dump([], f, indent=2)
    with open(ficheiro_json, "r", encoding="utf-8") as f:
        dados = json.load(f)
    dados.append(resultado)
    with open(ficheiro_json, "w", encoding="utf-8") as f:
        json.dump(dados, f, indent=2, ensure_ascii=False)

    # Também guarda em CSV
    ficheiro_csv = "resultados.csv"
    novo_ficheiro = not os.path.exists(ficheiro_csv)
    with open(ficheiro_csv, mode='a', newline='', encoding='utf-8') as f:
        writer = csv.writer(f)
        if novo_ficheiro:
            writer.writerow(["Data", "Nível", "Frase Esperada", "Frase Dita", "Percentagem Acerto"])
        writer.writerow([
            resultado["data"],
            nivel,
            frase_esperada,
            frase_dita,
            f"{percentagem:.2f}"
        ])
```

Figura 21 - Função responsável pelo registo persistente dos resultados (JSON e CSV).

APÊNDICE IV – GUIA DE INSTALAÇÃO E UTILIZAÇÃO DO PROTÓTIPO

IV.1 - Requisitos de execução

O protótipo desenvolvido foi concebido para ser executado em ambiente local, sem necessidade de ligação à Internet após a instalação inicial das dependências. Os requisitos mínimos recomendados são:

- **Sistema operativo:** Windows 10 ou superior;
- **Python:** versão 3.x instalada no sistema;
- **Bibliotecas principais:**
  - ❖ pygame – motor responsável pela interface gráfica e gestão de eventos do jogo;
  - ❖ whisper – modelo de reconhecimento automático de fala utilizado para transcrever a pronúncia do utilizador;
  - ❖ reportlab – geração automática dos relatórios em formato PDF;
  - ❖ outras bibliotecas de apoio indicadas no ficheiro requirements.txt (por exemplo, numpy, matplotlib, sounddevice, entre outras).

Recomenda-se, ainda, que o computador disponha de **microfone funcional** e **colunas ou auscultadores**, de modo a permitir a reprodução dos áudios de referência e a gravação da voz do utilizador.

IV.2 - Estrutura de ficheiros do projeto

A aplicação é distribuída numa única pasta principal, dentro da qual se encontram o código, os ficheiros de dados, os relatórios gerados e os recursos multimédia. A estrutura pode ser resumida da seguinte forma:

- **Ficheiros principais de código**

- ❖ main.py – ponto de entrada da aplicação; contém a lógica da interface do jogo, integração com o motor de reconhecimento de fala e chamadas aos restantes módulos.
- ❖ analise_resultados.py – script utilizado para gerar as figuras de análise (por exemplo, distribuição de percentagens, padrões fonéticos e evolução temporal) a partir do ficheiro resultados.json.
- ❖ gerar_audios.py e outros ficheiros mainv1.py, mainv2.py, mainv3.py, main backup.py, etc. – versões intermédias e scripts auxiliares mantidos por razões de desenvolvimento e histórico do projeto.

- **Ficheiros de dados**
  - ❖ resultados.json – ficheiro onde são registadas todas as tentativas de pronúncia efetuadas no jogo.
  - ❖ progresso.json – ficheiro que guarda o número de estrelas atribuídas a cada nível, por dificuldade.
  - ❖ map_seeds.json – ficheiro auxiliar com informação de configuração (por exemplo, mapeamento interno de frases/níveis).

- **Relatórios gerados**
  - ❖ Conjunto de ficheiros PDF produzidos automaticamente pela aplicação, com nomes do tipo:
    - relatorio_facil_YYYYMMDD_HHMM.pdf;
    - relatorio_medio_YYYYMMDD_HHMM.pdf;
    - relatorio_completo_YYYYMMDD_HHMM.pdf;
    - relatorio_geral_YYYYMMDD_HHMM.pdf.

  Estes relatórios são guardados na própria pasta do projeto, facilitando o acesso imediato aos resultados das sessões.

- **Figuras de análise**

    - ❖ Ficheiros de imagem como fig_padroes_foneticos.png, fig_distribuicao_percentagens.png e fig_evolucao_temporal.png, gerados pelo script de análise e reutilizados no relatório para ilustrar os resultados.

- **Recursos multimédia**

    - ❖ Pasta assets/, contendo os recursos necessários à execução do jogo:

        - assets/audios/ – ficheiros de áudio das frases de referência, organizados por nível de dificuldade;
        - assets/imagens/ – ícones, fundos e outros elementos gráficos usados na interface.

Esta organização mantém todos os elementos relevantes do protótipo concentrados numa única pasta, simplificando o processo de cópia para novos computadores e a execução do jogo em modo local.

IV.3 - Procedimento de instalação

Para instalar e preparar o protótipo para execução num novo computador, recomenda-se o seguinte procedimento:

1. **Cópia da pasta do projeto:**

   Copiar toda a pasta do projeto para o disco local do computador, incluindo:

   - ❖ o ficheiro main.py (ficheiro principal de arranque);
   - ❖ os restantes ficheiros de código (analise_resultados.py, versões intermédias do main, etc.);
   - ❖ os ficheiros de dados (resultados.json, progresso.json, map_seeds.json);
   - ❖ a subpasta assets/, com os áudios e as imagens utilizadas pela aplicação;
   - ❖ os relatórios PDF já gerados, caso se pretenda preservá-los.

2. **Verificação do Python instalado:**

   Confirmar que o computador possui uma versão recente do Python 3 instalada. Esta verificação pode ser feita através de um terminal, usando, por exemplo, um dos seguintes comandos:

   `python --version` ou `py --version`

3. **Instalação das bibliotecas necessárias:**

   As principais bibliotecas utilizadas pelo protótipo são:

   - pygame – interface gráfica e gestão do jogo;
   - openai-whisper – reconhecimento automático de fala;
   - sounddevice e scipy – captura e processamento de áudio;
   - numpy – operações numéricas de apoio;
   - reportlab – geração de relatórios em PDF.

   Com o Python instalado, as bibliotecas podem ser adicionadas através do terminal, na pasta do projeto, com o comando:

   ```
   pip install pygame openai-whisper sounddevice numpy scipy reportlab
   ```

   Caso alguma biblioteca adicional seja necessária, poderá ser instalada da mesma forma.

4. **Verificação dos recursos locais:**

   Confirmar que a subpasta assets/ contém:

   - assets/audios/ – ficheiros de áudio das frases de referência;
   - assets/imagens/ – ícones, fundos e outros elementos gráficos da interface.

Estes recursos são indispensáveis para o funcionamento correto da aplicação.

Após a conclusão destes passos, o protótipo encontra-se preparado para ser executado conforme descrito na secção D.4 (Execução do *serious game*).

IV.4 - Execução do *serious game*

Depois de concluída a instalação, o protótipo pode ser executado através dos seguintes passos:

1. Abrir uma janela de terminal na pasta raiz do projeto.
2. Executar o ficheiro principal:

   ```
   python main.py
   ```

3. A janela do jogo será aberta em modo fullscreen ou numa janela com resolução fixa, apresentando o ecrã inicial com o menu principal.

IV.5 - Fluxo típico de utilização

O fluxo de utilização previsto para o protótipo é o seguinte:

1. **Seleção da dificuldade e do nível:**

   No menu principal, o utilizador escolhe entre os níveis de dificuldade *Fácil*, *Médio* ou *Difícil*. Em seguida, escolhe um dos 30 níveis disponíveis dentro da dificuldade selecionada. Cada nível corresponde a uma frase específica do dataset.

2. **Escuta da frase de referência:**

   No ecrã de treino, o utilizador pode ouvir o áudio da frase de referência quantas vezes desejar, através do botão respetivo.

3. **Gravação da pronúncia:**

   Quando se sentir preparado, o utilizador clica no botão de gravação, pronuncia a frase em voz alta e encerra a gravação. O áudio é processado pelo modelo Whisper, que produz uma transcrição textual da pronúncia.

4. **Apresentação do feedback:**

   Após o reconhecimento, o sistema compara a frase dita com a frase de referência e apresenta:

   - a **percentagem de acerto** global;
   - o número de **palavras corretas / total**;
   - o **texto colorido**, destacando palavras bem produzidas e palavras problemáticas;
   - os **padrões fonéticos detetados** (por exemplo, nasalização reduzida, troca R→L, CH→S, omissão de "s" final);
   - o número de **estrelas** atribuídas ao nível (0–3), com base no desempenho.

5. **Registo automático dos dados:**

   Cada tentativa é registada no ficheiro resultados.json e o número de estrelas obtidas é atualizado em progresso.json, permitindo acompanhar a evolução ao longo do tempo.

6. **Consulta e geração de relatórios:**

   A partir do menu principal, o utilizador pode aceder à área de relatórios, onde é possível gerar:

   - relatórios **por nível de dificuldade**;

- um relatório **global**, incluindo todas as tentativas registadas. Os relatórios são guardados em formato PDF na pasta jogo_pronuncia/ e incluem resumos estatísticos, gráficos e tabelas de desempenho.