

**DESENVOLVIMENTO PARA A INTERNET**

**E APLICAÇÕES MÓVEIS**



Docente:

Lourenço Correia da Silva

Estudantes:

Guilherme Gaspar nº122981

Lourenço Santos nº123028

Manuel Albuquerque nº123010

Lisboa, 18 de maio de 2025

Índice

[Introdução 2](#_Toc198490806)

[Pontos fortes e inovadores 2](#_Toc198490807)

[Recursos externos gratuitos 3](#_Toc198490808)

[Tipos de Utilizadores: 3](#_Toc198490809)

[Principais Funcionalidades: 3](#_Toc198490810)

[Diagrama de classes 4](#_Toc198490811)

[Bibliografia 4](#_Toc198490812)

[Artigo sobre Atomic Design Pattern: 4](#_Toc198490813)

[Fonte tipográfica Lilita One: 4](#_Toc198490814)

[Lista de palavras comuns em inglês: 4](#_Toc198490815)

[Biblioteca Luxon 4](#_Toc198490816)

## Introdução

Neste projeto, quisemos tornar o ato de aprender a escrever mais rápido no computador algo divertido. Para isso, desenvolvemos um jogo onde o utilizador pode acompanhar as suas estatísticas e progresso ao longo do tempo.  
Como a competitividade está no nosso ADN, criámos também uma secção de competições, onde os jogadores podem competir entre si para ver quem escreve mais depressa.  
Mas não é só sobre competição, a componente social também é essencial. Por isso, decidimos incluir a possibilidade de formar equipas, fazer amigos e, claro, trocar mensagens, tanto entre amigos ou equipas.

## Pontos fortes e inovadores

Conceber este projeto trouxe alguns desafios que procurámos resolver da melhor forma possível.

No lado do React, seguimos o padrão “Atomic Design”, procurando manter os componentes o mais simples e modulares possível, dividindo o código em subcomponentes sempre que fez sentido, de modo a conseguir reutilizá-los.

No Django, adotámos uma lógica semelhante: organizámos os *models* e *views* em diferentes aplicações, de forma a manter o projeto mais limpo, modular e fácil de manter.

Num aspeto mais específico, para a realização dos testes optámos por uma abordagem diferente do habitual. Em vez de um campo de *input* clássico, adicionámos um *listener* à janela para captar diretamente as teclas pressionadas pelo utilizador. À medida que o utilizador escreve, cada carácter é renderizado com uma cor específica, proporcionando um *feedback* instantâneo. Esta abordagem torna o processo mais envolvente e permite ao utilizador perceber rapidamente onde está a errar ou a acertar.

Tivemos também vários problemas com o sistema de login, relacionados com o manuseamento de *cookies* no browser. Os erros surgiam apenas ao usar localhost:3000 em vez de 127.0.0.1:3000. Para contornar isso, criámos uma função que deteta o uso de localhost e redireciona para 127.0.0.1, resolvendo assim um problema que o Django não nos permitiu ultrapassar diretamente, apesar de muita pesquisa.

De forma a simplificar os pedidos ao *backend*, centralizámos a configuração do Axios numa constante disponibilizada no ficheiro AuthContext.js que abrange toda a aplicação, garantindo o envio automático de credenciais e do CSRF token.

A aplicação foi desenvolvida para impedir o acesso direto a páginas específicas via URL. Em vez de expor rotas como “/competition/1”, usamos caminhos genéricos como “/competitions/”, sendo a própria componente responsável por apresentar o conteúdo adequado. Se um utilizador tentar aceder a páginas inexistentes ou sem permissões para tal, será redirecionado para a *NotFoundPage* ou verá uma mensagem de acesso negado.

## Recursos externos gratuitos

Durante o desenvolvimento deste projeto, recorremos a alguns recursos externos gratuitos que facilitaram a sua concretização. Utilizámos a fonte tipográfica gratuita **Lilita One**, disponível no Google Fonts, e um ficheiro com 1000 palavras comuns em inglês, encontrado online.

Para a formatação de datas, optámos pela biblioteca **luxon** como alternativa mais leve e atual, uma vez que a **moment** foi descontinuada.

Todas as componentes da aplicação foram inteiramente desenvolvidas por nós, o que permitiu maior flexibilidade no design. Recorremos a inteligência artificial para auxiliar no refinamento e alinhamento do CSS já elaborado, garantindo maior fidelidade visual aos *mockups* concebidos.

## Tipos de Utilizadores:

* **Jogadores Não Registado: Utilizadores que ainda não fizeram login, podem apenas jogar individualmente.**
* **Jogadores Registado:** Utilizadores registados que participam em competições, acompanham as suas estatísticas e interagem com outros jogadores através de equipas ou amizades.
* **Administradores:** Responsáveis pela gestão de eventos e supervisão do bom funcionamento da plataforma, assim como acesso a todas as funcionalidades dos jogadores registados.

## Principais Funcionalidades:

|  |  |
| --- | --- |
| **Funcionalidade** | **Utilizador** |
| Participação em jogos de escrita com nº de palavras à escolha do utilizador | Jogador Não Registado e Registado |
| Registo resultados e estatísticas individuais | Jogador Registado |
| Sistema de amigos e mensagens | Jogador Registado |
| Criação e gestão de equipas, com hierarquias | Jogador Registado |
| Criação de competições | Administrador |
| Moderação de atividades e comportamentos dos jogadores (ex.: banir) | Administrador |

O nosso projeto apresenta várias funcionalidades que podem ser facilmente acedidas através do menu de navegação localizado no *header* da página.

Em primeiro lugar, destaca-se a **participação em jogos de escrita com número de palavras à escolha do utilizador**, onde cada jogador pode definir o número de palavras a escrever, adaptando assim a experiência às suas preferências. Esta funcionalidade está disponível para qualquer utilizador, esteja ou não registado.

Como mencionado na proposta do nosso projeto, os **utilizadores registados** têm acesso a um conjunto alargado de funcionalidades adicionais. A primeira é o **registo de resultados e estatísticas individuais**, que permite a cada jogador visualizar o seu desempenho em cada jogo e acompanhar a sua evolução ao longo do tempo. Adicionalmente, os jogadores registados podem usufruir do **sistema de amigos e mensagens**, no qual é possível adicionar e remover amigos, bem como trocar mensagens privadas com os mesmos, promovendo a interação entre utilizadores e fomentando uma vertente social dentro da plataforma. Outra funcionalidade disponível para utilizadores registados é a **criação e gestão de equipas com hierarquias**, que possibilita a qualquer jogador com conta criar uma equipa e assumir o papel de líder. Dentro de cada equipa existe também um sistema de mensagens exclusivo, sob a forma de um **chat de equipa**, que facilita a comunicação entre os membros.

Por fim, os **administradores** dispõem de ferramentas específicas que lhes permitem a **criação de competições**, podendo assim organizar eventos nos quais os utilizadores competem entre si. Estes administradores têm também acesso a funcionalidades de **moderação de atividades e comportamentos dos jogadores**, como, por exemplo, a capacidade de analisar *reports* efetuados contra utilizadores e aplicar penalizações, incluindo o banimento da plataforma, quando necessário.

## Diagrama de classes

A diagram of a company

AI-generated content may be incorrect.

## Bibliografia

### Artigo sobre Atomic Design Pattern:

Roopal, R. J. (2020). *Atomic Design Pattern: Structuring Your React Application*. Medium. Disponível em: <https://rjroopal.medium.com/atomic-design-pattern-structuring-your-react-application-970dd57520f8>

### Fonte tipográfica Lilita One:

Google Fonts. *Lilita One* [Fonte tipográfica]. Disponível em: <https://fonts.googleapis.com/css2?family=Lilita+One&display=swap>

### Lista de palavras comuns em inglês:

SMenigat. *Thousand Most Common Words* [Repositório GitHub]. Disponível em: <https://github.com/SMenigat/thousand-most-common-words/blob/master/words/en.json>

### Biblioteca Luxon

NPM. *Luxon* [Pacote JavaScript]. Disponível em: <https://www.npmjs.com/package/luxon>