

# Introdução às Tecnologias Web

Enunciado do Projeto de 2020/2021

## Tema: Pokémon

O Pokémon é inspirado no passatempo de apanhar e colecionar insetos, mas, para tornar o conceito mais apelativo, em vez destes, consideram-se monstros. Existem vários jogos de vídeo e de cartas, livros, séries de televisão, e até filmes sobre o Pokémon, sendo que para este enunciado o foco está num **jogo de tabuleiro**. Assim, um jogador deve percorrer um caminho até chegar ao destino, apanhando os monstros que aparecerem. A apanha de um monstro acaba por ser uma conquista, pois envolve uma batalha que tanto o jogador como o monstro podem perder. Esta é a ideia base, sendo mais à frente detalhados os requisitos essenciais que devem ser cumpridos, bem como funcionalidades adicionais que os grupos podem explorar para ter melhor nota. Mais informações sobre o conceito de Pokémon podem ser obtidas na Wikipedia em <https://en.wikipedia.org/wiki/Pokémon>.

## Objetivos

1. Criação de um *site* com **informação sobre o jogo**, incluindo: a) conteúdos de *texto*, tais como regras e tutoriais, origem e curiosidades históricas, listas de opções e seus valores admissíveis, lista de campos necessários para o registo de jogadores, tabelas com pontuações e estatísticas, e ficha técnica com os autores e tecnologias usadas; b) conteúdos *gráficos*, como imagens dos avatares dos jogadores, dos personagens ou objetos que participam ativamente no jogo, e dos elementos decorativos do tabuleiro, bem como vídeos, por exemplo de demonstração de funcionalidades; e c) conteúdos *áudio*, como música e sons para ilustrar as ações durante o jogo.

Todos estes conteúdos devem fazer parte de páginas HTML, ligadas entre si de forma coerente, isto é, sem que o jogador se perca durante a aprendizagem e uso do jogo. A apresentação dos conteúdos deve ser especificada usando CSS.

2. Desenvolvimento das **funcionalidades do jogo**, incluindo: registo dos jogadores, inicialização do jogo conforme a configuração escolhida pelo jogador, eventualmente suportando modos com regras distintas, jogadas permitidas em cada momento, deteção do fim do jogo, e cálculo de pontuações e estatísticas.

A programação das funcionalidades deve ser feita em JavaScript e jQuery, e, no final, deve haver uma integração total com o *site* do objetivo 1.

## Requisitos essenciais do jogo

Os requisitos seguintes devem ser suportados pelos jogos de *todos* os grupos, servindo de base para a avaliação, e valem **15 valores** da nota no projeto.

1. O jogo tem de suportar o *modo individual*, isto é, com apenas um jogador a percorrer o caminho até ao destino. Não é necessário passar a vez a outro jogador, mas é preciso registar o tempo de cada jogo.
2. Antes de poder jogar, um jogador tem de se *identificar*, podendo ser necessário registar-se (*e-mail*, senha, faixa etária, e género), caso ainda não o tenha feito. Os registos de dados devem ficar guardados em *local storage* do *browser*.
3. De igual forma, só depois de se identificar é que um jogador pode ver *estatísticas* sobre o seu número de jogos e tempo total jogado.
4. No início de cada jogo, é mostrado um *tabuleiro* de dimensão 5x5, com um caminho fixo, com obstáculos em forma de árvores, pedras, e água. A largura e altura do tabuleiro devem ajustar-se automaticamente às dimensões e orientação do ecrã (*responsive web design*).
5. Existem três *pokémons* distintos, cada um com um determinado nível de força medido de 2 a 12. No início do jogo, o jogador escolhe um pokémon e os outros dois são oponentes, colocados em lugares fixos no caminho do tabuleiro.
6. O jogador começa sempre da mesma posição e o destino é sempre o mesmo. A *movimentação* deve ser feita com o teclado ou com o rato (um destes dispositivos), uma casa de cada vez, para cima, direita, baixo, ou esquerda.
7. Quando o jogador chega à posição de um pokémon oponente é iniciada uma *batalha*, durante a qual são lançados dois dados de seis lados. O primeiro a lançar os dados é quem tiver o maior nível de força; em caso de empate, dá-se preferência ao jogador. Se a soma dos dois dados for maior ou igual ao nível de força do oponente, então este perde um nível; caso contrário, fica com a mesma força. Depois, é a vez do outro interveniente na batalha lançar os dados, e assim sucessivamente, até um deles ter perdido todos os níveis de força.
8. O jogo *termina* se o jogador chegar ao destino ou se perder todos os níveis de força numa batalha, após o que deve ser mostrada uma tabela com os dez melhores tempos e respetivos jogadores, devendo essa informação ficar guardada em *local storage* do *browser*.
9. Para além dos conteúdos indicados no objetivo 1, devem ser geradas *animações* para a movimentação do jogador e outras ações do jogo.

## Funcionalidades adicionais do jogo

Seguem-se algumas funcionalidades que enriquecem o jogo, tornando-o mais desafiante e apelativo, e que permitem aumentar até **5 valores** a nota no projeto, sendo incentivada a criatividade para além das sugestões dadas.

- *Mais pokémons*, eventualmente com características para além do nível de força, permitindo ações mais complexas durante uma batalha. Por exemplo, cada pokémon pode ter um elemento da natureza (como fogo, água, ou terra), que pode ser, ou não, eficaz para conquistar o pokémon oponente.
- *Vários tabuleiros*, com diferentes dimensões, cada um com um caminho fixo. Nas posições dos obstáculos de um tabuleiro, o tipo de obstáculo pode ir mudando, para tornar o jogo menos repetitivo. O número de pokémons ao longo do caminho deve variar com o tabuleiro, podendo a sua posição ser fixa ou aleatória. Cada tabuleiro deve ter a sua tabela dos dez melhores tempos.
- *Tabuleiros mais ricos*, com novos tipos de obstáculos, como gelo ou arbustos, e de passagens ao longo do caminho, por exemplo pontes por cima de água.
- *Outros modos de jogo*, para além da contagem de tempo, tais como fazer o caminho com o mínimo de movimentos ou conquistar o máximo de pokémons e chegar ao destino dentro de um limite de tempo. Cada modo de jogo deve ter a sua tabela das dez melhores classificações.
- *Jogar com um pokémon conquistado numa batalha*, ou seja, permitir que o jogador colecione os pokémons conquistados ao longo do caminho e, aquando de uma nova batalha, escolha um desses pokémons para além do que selecionou no início do jogo. Para isto ser possível, o nível de força de um pokémon conquistado deve poder ser renovado gradualmente com a passagem do tempo.
- *Penalizações por o jogador ir contra obstáculos*, por exemplo, fazendo com que mais pokémons oponentes apareçam no caminho ou acrescentando segundos ao tempo real de jogo.
- *Modo multijogador*, com vários jogadores no mesmo jogo fazendo, alternadamente, um movimento de cada vez. Os jogadores têm de se inscrever no início, devendo cada um ter um pokémon distinto. Quantos mais jogadores se inscreverem, mais pokémons oponentes devem existir no caminho, podendo os jogadores batalhar entre si. Neste modo, ganha quem chegar primeiro ao destino. Deve existir uma tabela com as dez melhores classificações por tabuleiro, com indicação dos jogadores inscritos, do jogador vencedor, e do seu tempo.

- *Segundo dispositivo de movimentação do jogador*, nomeadamente, dos dois indicados no requisito essencial 6, o que não tiver sido suportado inicialmente.
- *Mais estatísticas*, agrupadas por tabuleiro e por modo de jogo, que complementem (e expliquem) os totais mencionados no requisito essencial 3.

### A ter em conta

- No início de todos os ficheiros de código (HTML, CSS, e JavaScript) deve estar, em comentário, a sigla de unidade curricular e o ano letivo, o número do grupo, e o nome, número, e turma dos alunos.
- O código JavaScript deve estar comentado, as variáveis devem ter *nomes inteligíveis*, e devem ser respeitadas regras de alinhamento dos blocos de código.
- Os ficheiros devem estar organizados em pastas, sendo sugerida a estrutura no guia “Getting started with the Web: Dealing with files,” do MDN Web Docs,<sup>1</sup> compreendendo *images*, *styles*, e *scripts*, ou o equivalente em português.
- É bem conhecido da indústria do *software* que é gasto mais esforço a corrigir e atualizar código existente do que a produzir código novo.<sup>2</sup> Apesar de poderem existir jogos semelhantes na Internet, quem se inspirar nos mesmos deve ter em conta que o esforço de interpretação do código e de adaptação das funcionalidades pode não compensar, especialmente se for reconhecido que a aprendizagem passa muito pela experiência de aplicação de conhecimentos.

### Entregas

- Cada objetivo do projeto tem uma entrega associada, realizada através de uma atividade na página de entrada do Moodle da unidade curricular.
- Para o objetivo 1, a cumprir até ao dia **21 de março** (domingo), deve ser entregue um ficheiro com nome ITW-2021-GXX-E1.ZIP, onde XX representa o número do grupo, com dois dígitos (ex. 01).
- Para o objetivo 2, realizável até ao dia **2 de maio** (domingo), o ficheiro deve chamar-se ITW-2021-GXX-E2.ZIP.

Bom trabalho no projeto!

---

<sup>1</sup> [https://developer.mozilla.org/docs/Learn/Getting\\_started\\_with\\_the\\_web/Dealing\\_with\\_files](https://developer.mozilla.org/docs/Learn/Getting_started_with_the_web/Dealing_with_files).

<sup>2</sup> Ver páginas 1 a 3, sobre a natureza do *software*, no livro Roger S. Pressman e Bruce R. Maxim, *Software Engineering: A Practitioner's Approach*, 8ª edição, McGraw-Hill, 2015.