Para resolução desse desafio eu comecei usando um algoritmo que aprendi na época da faculdade de desenvolvi novamente há alguns meses. O algoritmo consiste em fazer uma busca de menor caminho em uma matriz e é baseado no algoritmo A\* (A star).

Esse projeto foi feito em python + pygame e pode ser acessado no github a partir do seguinte link: <https://github.com/daosantiago/path>

Para adaptar ao contexto de custo variável para cada passo, apenas adicionei mais um atributo à classe Tile e adicionei esse valor ao cálculo de custo de cada passo.

Trabalhando um pouco para adaptar de esse algoritmo, percebi que o código não estava muito bom e não ia atender corretamente ao que precisava ser feito. Então decidi refazer o algoritmo de busca, aplicando o clássico algoritmo de Dijkstra, onde o custo de acesso a todos os pontos de uma mapa é calculado a partir do ponto inicial fornecido.

O backend foi criado em python + Django, mantendo uma estrutura simples de API com uma chamada para os módulos de python puro que fazem a criação do tabuleiro, atribuição de custos e cálculos de rotas, devolvendo para a estrutura da API fazer a serialização e retornar para o frontend.

Para desenvolvimento do frontend decidi por aplicar React por ser uma tecnologia que já trabalhei em um projeto e que já estudei alguns cursos. O projeto em que trabalhei anteriormente era feito em uma versão mais antiga do React, ainda usando classes, então levei um tempo pra entender e adaptar alguns dos conceitos para o paradigma funcional.

Minha maior dificuldade no código até o ponto atual do projeto foi a correta orquestração da manipulação de estados usando useState e useEffect entre os diferentes componentes sem que gerasse algum resultado indesejado.

Por esse último ponto ter tomado bastante do tempo de desenvolvimento, o design do frontend ficou um pouco comprometido.