



## Avaliação crítica de resultados

Tendo por base todos os requisitos levantados e consequente desenvolvimento de diagramas e estruturas lógicas, foram então implementadas as funcionalidades que tomaram parte no compromisso, já desenvolvido em fases anteriores. Tendo isto em conta, enumeram-se de seguida essas mesmas funcionalidades:

1. Visualizar apostas abertas;
2. Visualizar jogos disponíveis para aposta;
3. Efetuar pagamento;
4. Visualizar o histórico de apostas;
5. Receber pagamento;
6. Receber notificação de resultados;
7. Criar aposta;
8. Registar/Login;

Não foi possível implementar as funcionalidades dependentes da API previamente planeadas nomeadamente na primeira fase do projeto (consulta de estatísticas e envio de notificações relativamente a resultados e a jogos) uma vez que a equipa de trabalho foi incapaz de encontrar uma API *open source* com a confiabilidade necessária para o tipo de dados envolvidos.

Contudo, a omissão destas funcionalidades não tem consequências muito graves na aplicação como um todo, uma vez que os objetivos de maior importância são mesmo assim atingidos - a aplicação permite na mesma criar apostas e oferece a segurança prometida aos dados dos utilizadores. A mudança do estado das apostas é possível executando um `curl` externo ao sistema. Isto permitirá uma possível inserção de uma futura alternativa no lugar da API para obtenção de dados externos. Com isto podemos dizer que o produto construído tem capacidade de modificação, o que é crucial para uma boa manutenção e frequente atualização da aplicação, o que por sua vez aumenta a sua duração e utilização a um longo período de tempo.

Para além disto, apesar do planeado na segunda fase do projeto, não foi possível à equipa de trabalho construir o produto final na linguagem de programação Elixir. Isto deveu-se a um erro por parte da equipa de trabalho na estimativa do tempo necessário para a construção da aplicação, que foi mais que o previsto. Visto também que a equipa de trabalho não possuía no momento muita experiência em termos de utilização da linguagem de programação Elixir, decidiu-se utilizar a linguagem Java, muito também por causa das ferramentas que esta tem a oferecer que foram utilizadas neste trabalho, nomeadamente o `JavaSpring` (que se utilizou para a construção do modelo da aplicação) e o `ThymeLeaf` (que se utilizou para a interface gráfica do produto).

Por sua vez, como estas ferramentas possibilitam a construção de uma aplicação Web, o produto pode ser utilizável em todo o tipo de dispositivos, móveis ou não, o que leva a respeitar os requisitos de acessibilidades levantados na primeira fase do projeto. Para além disto, facilitaram o cumprimento de parte dos requisitos não funcionais, nomeadamente requisitos de manutenção (fácil modificação da aplicação), requisitos de aparência (o `ThymeLeaf` oferece uma interface gráfica amigável visualmente), requisitos de utilização (o `ThymeLeaf` oferece também

### **Autores:**

Ariana Lousada (PG47034), Carlos Gomes (PG47083),  
Márcia Teixeira (A80943), Tiago Sousa (PG47684)



uma interface intuitiva e de fácil utilização) e de segurança.

Em relação às restrições estipuladas também na primeira fase do projeto:

1. A aplicação deverá ter uma base de dados própria.
2. A aplicação deverá ser capaz de agregar dados de uma dada API.
3. A aplicação deverá disponibilizar apostas relativas a futebol, golfe e basquetebol.
4. A aplicação deverá apresentar mais do que uma opção possível para pagamento.

Com exceção da segunda restrição (por razões já mencionadas anteriormente) as restrições foram obedecidas: a aplicação contém uma base de dados própria com a utilização do H2; é possível fazer apostas relativas a futebol, golfe, basquetball e também ténis; a aplicação também oferece, como já mencionado em secções anteriores do documento, três tipos de pagamento distintos: Paypal, Cartão de Crédito e Criptomoeda *Cardano*.

Face aos requisitos novos relativos ao pagamento em vários tipos de moeda, a equipa de trabalho não se deparou com nenhum problema, visto que as funcionalidades estipuladas no início do projeto eram, na maior parte, de fácil modificação e aumento de complexidade.

Por fim, em relação aos possíveis riscos definidos na segunda fase do projeto:

1. A base de dados atinge a sua capacidade máxima;
2. API indisponível;
3. Serviço de pagamento indisponível;

A ferramenta utilizada para a gestão da base de dados (H2) oferece acima de tudo persistência dos dados e permite uma fácil gestão da mesma, através de uma estratégia de manutenção de dados ainda com a equipa técnica de informáticos futuramente responsável pela tarefa.

Relativamente ao terceiro risco, escolheram-se três tipos de pagamento que funcionam bem já em várias aplicações e *websites* existentes. Uma vez que estes tipos de pagamento são confiáveis, esta escolha poderá reduzir a probabilidade de acontecimento deste risco.

Em relação ao segundo risco, caso seja no futuro encontrada uma solução que substitua a utilização da API, terá de ser reavaliado de acordo com a ferramenta ou estrutura a usar, assim como a probabilidade e custo de potenciais riscos da utilização dessa ferramenta.

Em suma, o desenvolvimento deste projeto tornou evidente a importância de definição da arquitetura do produto assim como o levantamento dos seus requisitos. Também se verificou que reuniões com *stakeholders* do projeto podem aumentar significativamente a sua aceitação global, tanto no mercado como pelas entidades diretamente e indiretamente envolvidas.

Consideramos então que os principais objetivos do projeto como um todo foram atingidos, tendo culminado no desenvolvimento de uma aplicação útil, e com capacidade expansiva, tanto por parte do programador, como do utilizador.

#### **Autores:**

Ariana Lousada (PG47034), Carlos Gomes (PG47083),  
Márcia Teixeira (A80943), Tiago Sousa (PG47684)