

# UMinho - Mestrado em Engenharia Informática

## Requisitos e Arquiteturas de Software 2022/23

### Trabalho Prático RASBET

Uma dada empresa pretende criar o RASBET, um produto informático de suporte a apostas desportivas, para comercializar junto de casas de apostas. As funcionalidades do produto já foram definidas pelo Frederico Dias com algum detalhe no documento anexo. Contudo, o Frederico, o engenheiro de requisitos que estava a trabalhar nesse documento, foi-se embora e não o completou. Ele zangou-se com as pessoas da empresa, pelo que não vai ser possível contactá-lo para esclarecer alguma dúvida. Este trabalho prático deve usar esse documento como ponto de partida e tendo em conta o seguinte:

1. O professor Paulo Sousa funciona como elemento de apoio durante o semestre. É como se ele fosse o elemento da empresa responsável pelo desenvolvimento do produto. Qualquer dúvida, questão deve ser analisada com ele. As aulas PL também podem ser usadas para discutir aspetos do projeto.
2. As **funcionalidades indicadas** no documento devem ser consideradas como estando, em princípio, **corretamente identificadas**. Contudo, é expectável que cada grupo **enriqueça o produto com outras funcionalidades que considere importantes/prioritárias/críticas** para o sucesso do produto. Em particular, espera-se que o produto seja completo e integro.
3. O documento não inclui os requisitos não funcionais (atributos de qualidade), pois não foram tratados pelo Frederico. A empresa já decidiu que há três categorias que têm que ser consideradas: (1) **usabilidade**, (2) **desempenho**, (3) **escalabilidade**. Podem (e eventualmente devem) ser também tratadas outras categorias, **devendo a sua inclusão ser devidamente enquadrada e justificada**.
4. O projeto inclui três fases, sendo que cada uma delas pondera  $\frac{1}{3}$  na nota final coletiva de cada grupo:
  - a. Requisitos + implementação funcional (27/set-04/nov). Esta fase pressupõe a **alteração, a correção, o enriquecimento do documento de requisitos**. As partes do documento original que permanecem são marcadas com letra de cor cinza; ~~as partes antigas que são apagadas devem aparecer com letra traçada~~; as partes novas são escritas com letra de cor preta. Devem ser usadas técnicas de levantamento de requisitos adequadas ao contexto do projeto (e.g., entrevistas,

- comparação com a concorrência, personas). Espera-se que a documentação inclua um tratamento cuidadoso dos requisitos não funcionais. Adicionalmente, tem que ser entregue uma implementação das funcionalidades principais do produto, podendo os requisitos não funcionais ser ignorados nesta fase. As equipas entregam o documento de requisitos e disponibilizam o código desenvolvido.
- b. Requisitos não funcionais (04-nov-02/dez). A segunda fase implica o tratamento dos requisitos não funcionais na solução. Isso pressupõe que esses requisitos devem estar especificados de forma quantificável e que os mesmos estão devidamente prioritizados. Em especial, devem ser identificadas situações de interdependência entre eles. Antes desta fase se iniciar, a equipa de projeto poderá ser confrontada com a necessidade de refazer a solução por não ter tido a preocupação em tratar adequadamente os requisitos não funcionais. Serão valorizados os projetos que demonstrem que a base de código está preparada para acomodar alguns tipos de modificações. Em especial, espera-se que o código possa facilmente ser expandido para considerar mais tipos de desportos,<sup>1</sup> mais tipos de apostas, mais apostadores, mais línguas, mais moedas. As equipas entregam o documento de requisitos, destacando o que foi acrescentado/melhorado na parte relativa aos requisitos não funcionais, e a solução arquitetural. Voltam a disponibilizar a nova versão do código.
- c. Alteração funcional e otimização do sistema (02/dez-17/jan). Esta fase será iniciada com o anúncio de alterações funcionais. As equipas devem considerar essas alterações e incorporá-las no seu sistema. Esta fase pode igualmente ser usada para refabricar o código (*refactoring*), ou seja, manter a funcionalidade, mas melhorar os requisitos não funcionais. O uso de *design patterns* adequados aos problemas será altamente valorizado. As equipas entregam a nova solução arquitetural e uma análise (2 páginas no máximo) das alterações que foram realizadas no código. Voltam a disponibilizar a nova versão do código.

João M. Fernandes, André L. Ferreira, Manuel J. Alves, Paulo R. Sousa  
21/set/2022

---

<sup>1</sup> Por exemplo, futebol e andebol podem ser considerados da mesma tipologia, pois são modalidades desportivas coletivas, em que os jogos são disputados por duas equipas, baseados em golos e com apostas do tipo 1-X-2. O basquetebol poderá ser considerado de outra tipologia, pois não permite empates. De forma semelhante, corridas de fórmula 1 e de cavalos podem ser do mesmo tipo, se assumirmos que há uma série de competidores (individuais) à partida e que, no final da corrida, há uma ordem não total entre eles (i.e., pode haver dois competidores na mesma posição final).