

Engenharia de Requisitos

Edição 2023-24

versão 1.3

18 de dezembro de 2023

Resumo

Este documento consiste no enunciado do trabalho prático a desenvolver no contexto da disciplina de Engenharia de Requisitos. O trabalho consiste no desenvolvimento de uma solução informática que suporte a actividade apícola nas suas componentes de gestão diária, planeamento de operações e interacção com as entidades oficiais.

Tabela 1: Histórico de Versões

Versão	Descrição
1.0	Versão inicial
1.1	Definição de Papeis
1.2	Explicitação dos componentes da desenvolver Definição das US em função do tamanho do grupo Requisitos não funcionais
1.3	Adicionado o Sprint 2

1 Descrição

A DGADR seleccionou a SAS para o desenvolvimento da aplicação móvel HapiBee que interagirá com o Portal já existente (www.ifap.pt). A aplicação HapiBee pretende:

- facilitar a realização operações do dia-a-dia do apicultor, nomeadamente, gestão dos apiários, realização de inspeções, crestas e acções de profilaxia sanitária;
- planeamento optimizado de tarefas, como: gestão do calendário, geração de plano de visitas aos apiários, geração de listas de materiais necessários para a realização de tarefas apícolas;
- simplificação da interacção com entidades oficiais (e.g., DGAV), designadamente, Declaração Anual de Existências, Pedidos de Transumâncias, Pedido de Instalação de Apiário e comunicação de doenças.

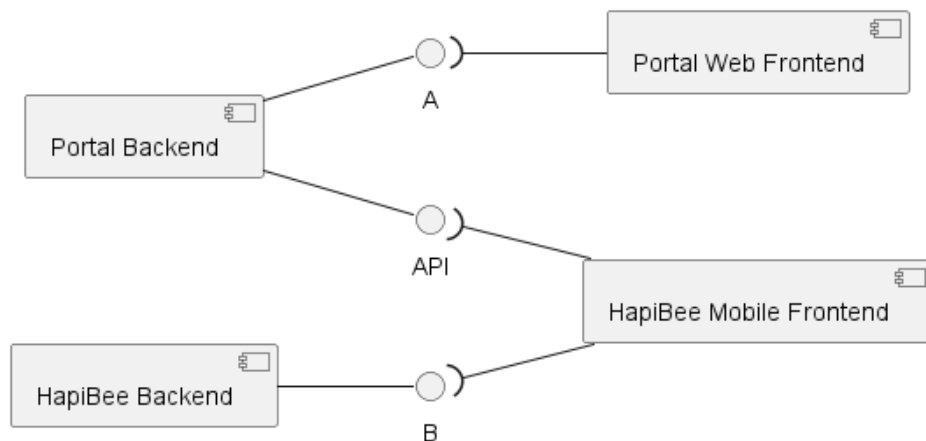


Figura 1: Arquitectura sugerida para solução a desenvolver

O projecto a desenvolver deverá ter em consideração (podendo ser alterada se não permitir responder aos requisitos especificados) a arquitectura apresentada no diagrama de componentes da 1. Em particular, a solução a desenvolver deverá considerar:

1. Desenvolvimento da aplicação HapiBee
 - (a) Aplicação móvel (HapiBee Mobile Frontend)
 - (b) Backend em sistema multi-tenant para guardar a informação de vários apicultores (HapiBee Backend)
2. Mock do lado do sistema de backend para providenciar as funcionalidade de execução da DGADR, DGAV e outras. (Portal Backend)
3. Desenvolvimento das interfaces para comunicar com o Portal (API)

Na solução a desenvolver não é necessário o desenvolvimento das funcionalidades do portal (Portal Web Frontend).

1.1 Método

O trabalho é realizado em grupos de 4 ou 5 alunos. O grupo deve decidir quais as tecnologias a usar tendo em atenção os recursos existentes. As escolhas tecnológicas, que devem ser justificadas, não devem limitar as boas práticas do processo de desenvolvimento, a arquitetura ou a solução final.

Os requisitos a implementar deverão ser revalidados com o cliente tendo em conta a limitação de recursos para cada iteração. Deve ser produzido inicialmente um product backlog.

Dentro do possível será usado scrum no processo de desenvolvimento. Serão produzidas duas iterações (este documento descreve as regras para a 1ª iteração).

As funcionalidades deverão ser priorizadas pela equipa, devendo ser atribuída uma ordem de prioridade para cada uma das mesmas e uma justificação pela decisão tomada. No início de cada iteração deverá ser feito um sprint planning.

Todas as sessões de trabalho da equipa de desenvolvimento devem iniciar com a reunião de *daily scrum* (o docente deve estar presente nas reuniões que decorrerão na aula prática).

No final de cada iteração deverá ser feito o *sprint review* onde a equipa apresentará também o trabalho ao *product owner* e cliente .

Pretende-se continuar neste trabalho a separar de forma clara o papel do docente do papel do cliente, pelo que as dúvidas sobre funcionalidades devem continuar a serem colocadas em reuniões marcadas com o cliente.

1.2 Papéis

- *Product Owner* – docente pode apoiar, mas a equipa é responsável pela priorização
- *Scrum Master* - a ser decidido pela equipa (pode mudar no segundo sprint)
- *Team Developers* – todos os elementos do grupo
- Outros – A definir pela equipa

O objectivo deste projecto é desenvolver um Produto Viável Mínimo de forma iterativa e incremental, desta forma, o trabalho será dividido em dois Sprints:

- Sprint 1 – semanas 8 a 11 – de 6/Novembro a 3/Dezembro
- Sprint 2 – semanas 12 a 15 – de 4/Dezembro a 7/Janeiro

Uma descrição do PVM é fornecida para cada sprint. As equipas devem seguir as USs fornecidas e ter em conta o seu encadeamento e respectivas dependências e no final de cada sprint cada equipa deve conseguir satisfazer os requisitos especificados. As equipas devem ser capazes de adicionar as USs ao backlog, dimensioná-las adequadamente e distribuí-las pelos membros da equipa.

1.3 Sprint 1

Na aplicação móvel Hapibee é pretendido:

1. Como apicultor quero realizar a declaração anual de existências;
2. Como apicultor quero efetuar pedido de transumância;

Alguns aspectos a ter consideração:

- (a) Autorização da DGAV
 - i. Condicionantes de localização previstas na lei (distancia a habitações, arruamentos, outros apiários, etc)
 - ii. Existência de fechos sanitários
- (b) Zonas Controladas e não controladas
- (c) Autorização da Entidade Gestora no caso de Zona Controlada
- (d) Notificações por email e/ou *Push Notification*

3. Como apicultor quero poder criar um apiário, efetuar o respectivo pedido de instalação e registar a sua constituição/informação;
4. Como apicultor quero realizar a inspeção do apiário e registar a mesma;
5. Como apicultor quero realizar um desdobramento e registar a mesma (apenas para grupos de 4 ou 5 elementos);
6. Como apicultor quero realizar a cresta (apenas para grupos de 5 elementos).

Assumimos que as operações de adição de apicultor, adição zona controladas, utilizadores e outras operações CRUD não *core* serão carregadas por ficheiro ou inseridas na Base de Dados. Nenhum desenvolvimento é necessário no portal, mas é necessário garantir que a informação pode ser consumida pelo mesmo.

1.4 Sprint 2

No Sprint 2 os alunos deverão considerar:

1. o feedback obtido na apresentação do TP1
2. os requisitos funcionais e não funcionais não terminados no TP1.
3. a visualização do histórico das diversas entidades (declaração anual de existências, apiários, colmeias, crestas, etc.).

Deverão também entregar um relatório constituído por:

1. Diagrama de componentes e de modelo de domínio da solução
2. Lista das tecnologias utilizadas e respetivo propósito
3. Um capítulo por User Story, onde deverá constar:
 - (a) elemento do grupo responsável por esse user story
 - (b) Storymapping para cada User Story com a priorização dividida em:
 - i. MVP (Minimum Viable Product)
 - ii. MVI1 (Minimum Viable Increment 1)
 - iii. MVI2 (Minimum Viable Increment 2)
 - iv. Backlog
 - (c) Identificação do que foi concluído no Sprint 1 e Sprint2 (por exemplo através de uma label)
 - (d) Justificação por escrito da razão porque foi priorizado dessa forma (com exceção do backlog)
 - (e) Ecrãs da aplicação desenvolvida (Screenshots)
4. Lista de requisitos não funcionais e breve descrição de como conseguiram resolver os mesmos. Exemplo: Para garantir o funcionamento da solução na ausência de uma ligação à internet foi (...). Quando a aplicação retomava o acesso à internet foi realizado (...) para garantir o sincronismo com o backend da aplicação.

1.5 Requisitos não funcionais

Os requisitos não funcionais (ou de qualidade) a considerar são aqueles que foram identificados no SRS submetido, complementados com os comentários recebidos nas respectivas apresentações e futuras interacções com os stakeholders. Em particular é necessário garantir o funcionamento da solução na ausência de uma ligação à internet.