

ENGREQ

ISEP – Mestrado em Engenharia Informática

Engenharia de requisitos

1201000 BRUNO RIBEIRO 1201001 CARLOS RODRIGUES 1201007 JOSÉ PESSOA
1201477 FRANCISCO BORGES

Índice

1	Introdução	5
2	Funcionalidades do sistema e prioridades.....	6
3	Tecnologias utilizadas	7
3.1	Java	7
3.2	Spring Boot	7
3.3	React Native	7
3.4	Bcrypt	8
4	Sprint Planning	9
4.1	Sprint 1	9
4.2	Sprint 2.....	9
5	Regras de commit	10
6	Modelo de domínio da solução	11
6.1	Agregado dos utilizadores.....	11
6.2	Agregado dos Apiários.....	11
7	Diagramas de Componentes	13
8	User stories desenvolvidas.....	14
8.1	Como apicultor quero realizar a declaração anual de existências	14
a)	Responsável pelo desenvolvimento	14
b)	Aplicação desenvolvida.....	14
c)	Storymapping	16
8.2	Como apicultor quero efetuar um pedido de transumância.....	17
a)	Responsável pelo desenvolvimento	17
b)	Aplicação desenvolvida.....	17
c)	Storymapping	19
8.3	Como apicultor quero poder criar um apiário, efetuar o respectivo pedido de instalação	20
a)	Responsável pelo desenvolvimento	20
b)	Aplicação desenvolvida.....	20
c)	Storymapping	21
8.4	Registrar a constituição do apiario.....	22
a)	Responsável pelo desenvolvimento	22
b)	Aplicação desenvolvida.....	23

c)	Storymapping	24
8.5	Como apicultor quero realizar a inspeção do apiário e registrar a mesma.....	26
a)	Responsável pelo desenvolvimento	26
b)	Aplicação desenvolvida.....	26
c)	Storymapping	28
8.6	Como apicultor quero realizar um desdobramento e registrar a mesma	29
a)	Responsável pelo desenvolvimento	29
b)	Aplicação desenvolvida.....	29
c)	Storymapping	30
9	Requisitos não funcionais	32
9.1	Usabilidade	32
9.2	Performance.....	32
9.3	Segurança.....	32
9.4	Confiabilidade.....	32

Índice de Figuras

Figura 1 - Modelo de dominio.....	11
Figura 2 - Diagrama de componentes.....	13
Figura 3 - Página inicial	Figura 4 - Informação do Apicultor
Figura 5 - Submeter declaração	15
Figura 6 - MVP da declaração anual	16
Figura 7 - MVI,MVI2 e Backlog da declaração anual	16
Figura 8 - Página do Apiário	Figura 9 - Pedido de transumância
Figura 10 - MVP do pedido de transumância.....	19
Figura 11 - MVI, MVI2 e Backlog do pedido de transumância.....	19
Figura 12- Menu principal	20
Figura 13 Página criação apiários	21
Figura 14-MVP do registo dos apiario e o pedido de instalação.....	21
Figura 15-MVI, MVI2 e Backlog da criação de apiários	22
Figura 16 - página detalhe dos apiários	23
Figura 17 - Página de registo das colmeias	23
Figura 18 - Página de registo da colmeia	24
Figura 19 - Página de registo das colmeias com a colmeia adicionada	24
Figura 20-MVP do registo da constituição do apiário.....	25
Figura 21-MVI, MVI2 e Backlog do registo da constituição da colmeia.....	25
Figura 22 - Página detalhes colmeia.....	26
Figura 23 - Exemplo formulário inspeção	27
Figura 24 - Campo para adicionar observações	27
Figura 25 - MVP de realizar inspeção	28
Figura 26 - MVI, MVI2 e Backlog de realizar inspeção	28
Figura 27 - Página detalhes colmeia.....	29
Figura 28 - Página desdobramento.....	30
Figura 29 - MVP de realizar desdobramento.....	30
Figura 30 - MVI, MVI2 e Backlog de realizar desdobramento.....	31

1 Introdução

Com o intuito de apoiar as tarefas diárias dos apicultores em Portugal, surgiu a HapiBee, uma aplicação mobile de apoio à atividade apícola.

Neste relatório serão abordados alguns aspetos importantes para a organização do projeto e da equipa de desenvolvimento.

Durante o desenvolvimento do sistema, a equipa adotou uma metodologia scrum. O Scrum é uma estrutura ágil de gestão de projetos que tem como objetivo aumentar a eficiência no desenvolvimento de produtos complexos. Inicialmente criado para a gestão de projetos de software, o Scrum tem sido adotado em várias áreas devido à sua flexibilidade e à sua abordagem iterativa e incremental.

2 Funcionalidades do sistema e prioridades

Foram fornecidas à equipa de desenvolvimento uma lista de funcionalidades que deviam ser implementadas ao longo do sprint. Sendo estas:

- US1 - Como apicultor quero realizar a declaração anual de existências;
- US2 - Como apicultor quero efetuar pedido de transumância;
- US3 - Como apicultor quero poder criar um apiário, efetuar o respetivo pedido de instalação e registar a sua constituição/informação;
- US4 - Como apicultor quero realizar a inspeção do apiário e registar a mesma;
- US5 - Como apicultor quero realizar um desdobramento e registar a mesma.

A equipa analisou as funcionalidades apresentadas e definiu a prioridade das mesmas, de forma a garantir que esta se focasse nos requisitos mais importantes para o correto funcionamento da aplicação. Para tal, estas funcionalidades foram divididas por três grupos de prioridade, muito elevado, elevado e médio. Em seguida, será apresentada esta distribuição:

- Muito elevado:
 - US3: esta funcionalidade é fundamental para o sistema, visto que as restantes estão dependentes desta.
- Elevado:
 - US2, US4, US5: estas funcionalidades são importantes, porque representam as atividades diárias do apicultor.
- Médio:
 - US1: esta funcionalidade não é tão essencial para as atividades diárias do apicultor, já que existem outros métodos para realizar a declaração anual de existências e esta só ocorre uma vez por ano.

Para este projeto foi utilizado a framework Spring Boot , usando como linguagem o Java.

3 Tecnologias utilizadas

3.1 Java

Java é uma linguagem de programação de alto nível, orientada a objetos e multiplataforma. Ela foi projetada para ser uma linguagem simples, segura e portátil, permitindo que os desenvolvedores escrevam código uma vez e o executem em qualquer dispositivo ou sistema que tenha uma Máquina Virtual Java, independente do sistema operacional.

Java destaca-se como uma linguagem de programação orientada a objetos que se destaca pela ênfase em segurança, uma sintaxe simples e limpa, e um ecossistema robusto com uma ampla biblioteca padrão. Essas características fundamentais tornam o Java uma escolha prevalente em uma variedade de domínios, incluindo desenvolvimento empresarial, aplicações web, sistemas embarcados e aplicativos móveis. Sua relevância duradoura na indústria de software é impulsionada pela combinação de portabilidade, segurança e versatilidade oferecidas pela linguagem.

No caso do nosso sistema, esta linguagem foi utilizada para desenvolver o backend da aplicação mobile, tal como do portal.

3.2 Spring Boot

O Spring Boot é um framework Java que simplifica significativamente o desenvolvimento de aplicações. Ele oferece convenções e configurações padrão, eliminando grande parte da complexidade associada ao desenvolvimento Java tradicional. Uma das suas principais características é a inclusão de um servidor web embutido, o que facilita a implantação dos aplicativos.

O Spring Boot realiza automaticamente a configuração com base nas dependências do projeto, tornando o processo de configuração mais intuitivo e eficiente. Além disso, disponibiliza starters, que são conjuntos de dependências pré-configuradas, simplificando a inclusão de funcionalidades comuns.

Essas características combinadas fazem do Spring Boot uma boa escolha para o desenvolvimento eficiente e ágil de aplicações Java. Deste modo, a equipa decidiu utilizar o Spring boot em conjunto com as aplicações a serem desenvolvidas em java, isto é, o backend da aplicação mobile e do portal.

3.3 React Native

O React Native é um framework open-source para construir aplicações mobile com o uso de JavaScript e React. Permite aos developers utilizar React, uma biblioteca popular de JavaScript para construir interfaces de utilizador, para criar aplicações mobile nativas para as plataformas iOS e Android. A nossa equipa escolheu o React Native como a framework para

desenvolver a aplicação proposta devido às capacidades versáteis multiplataforma, à sua ampla popularidade e às recomendações sólidas na comunidade de desenvolvimento. Além disso, a experiência que a equipa possuía com o React Native reforçou ainda mais a escolha da plataforma.

3.4 Bcrypt

O bcrypt é uma função criptográfica de hash. Foi projetado para ser lento e intensivo em termos computacionais, tornando-o mais resistente a diversos ataques, principalmente brute-force.

O objetivo principal do bcrypt é fazer hash das passwords de forma segura antes de armazená-las dentro de uma base de dados. Ao contrário das funções de hash simples, como MD5 ou SHA-256, o bcrypt incorpora um "salt" - um valor aleatório único para cada password. Esse salt ajuda a proteger contra ataques, de modo a garantir que mesmo passwords idênticas tenham representações de hash diferentes.

Além disso, o bcrypt é intencionalmente lento, de modo a adicionar uma camada de segurança extra. À medida que a potência computacional aumenta ao longo do tempo, a lentidão do bcrypt permanece constante, tornando mais difícil para realizar ataques em grande escala.

Na nossa aplicação, o bcrypt é utilizado para verificar e encriptar as passwords do apicultor a realizar a autenticação.

4 Sprint Planning

4.1 Sprint 1

No nosso sprint planning do sprint 1, distribuimos as user stories entre os membros da equipa, tal como atribuímos tarefas específicas. O objetivo era que cada membro da equipa concluísse as suas respetivas tarefas antes de 8 de dezembro, de modo a equipa ter tempo suficiente para reunir e realizar uma revisão detalhada do trabalho realizado. A divisão de tarefas foi a seguinte:

Membro da equipa

Bruno Ribeiro (1201000)
Carlos Rodrigues (1201001)
José Pessoa (1201007)
Francisco Borges (1201477)

Tarefas

US5 ; Frontend Project Setup
US3 ; Backend Project Setup
US1 ; US2
US4

4.2 Sprint 2

No sprint planning do sprint 2 a equipa decidiu, devido ao facto de o MVP de todas as user stories estarem terminadas, que cada membro iria alterar o necessário em correspondência com o feedback recebido para a user storie que lhe tinha sido atribuída no sprint passado. Tal como no sprint passado, cada membro deve terminar as suas respetivas tarefas 2 dias antes da deadline, de modo a ser possível efetuar uma revisão detalhada do trabalho realizado.

5 Regras de commit

No Bitbucket, foram registrados os issues relacionadas ao desenvolvimento do projeto. Uma prática estabelecida é garantir que cada commit esteja associado à tag correspondente do issue, possibilitando uma vinculação eficiente entre o commit específico e a questão relacionada. Essa abordagem contribui para uma rastreabilidade clara e organizada, facilitando o acompanhamento do progresso do desenvolvimento e a compreensão das alterações feitas em relação às issues específicas.

6 Modelo de domínio da solução

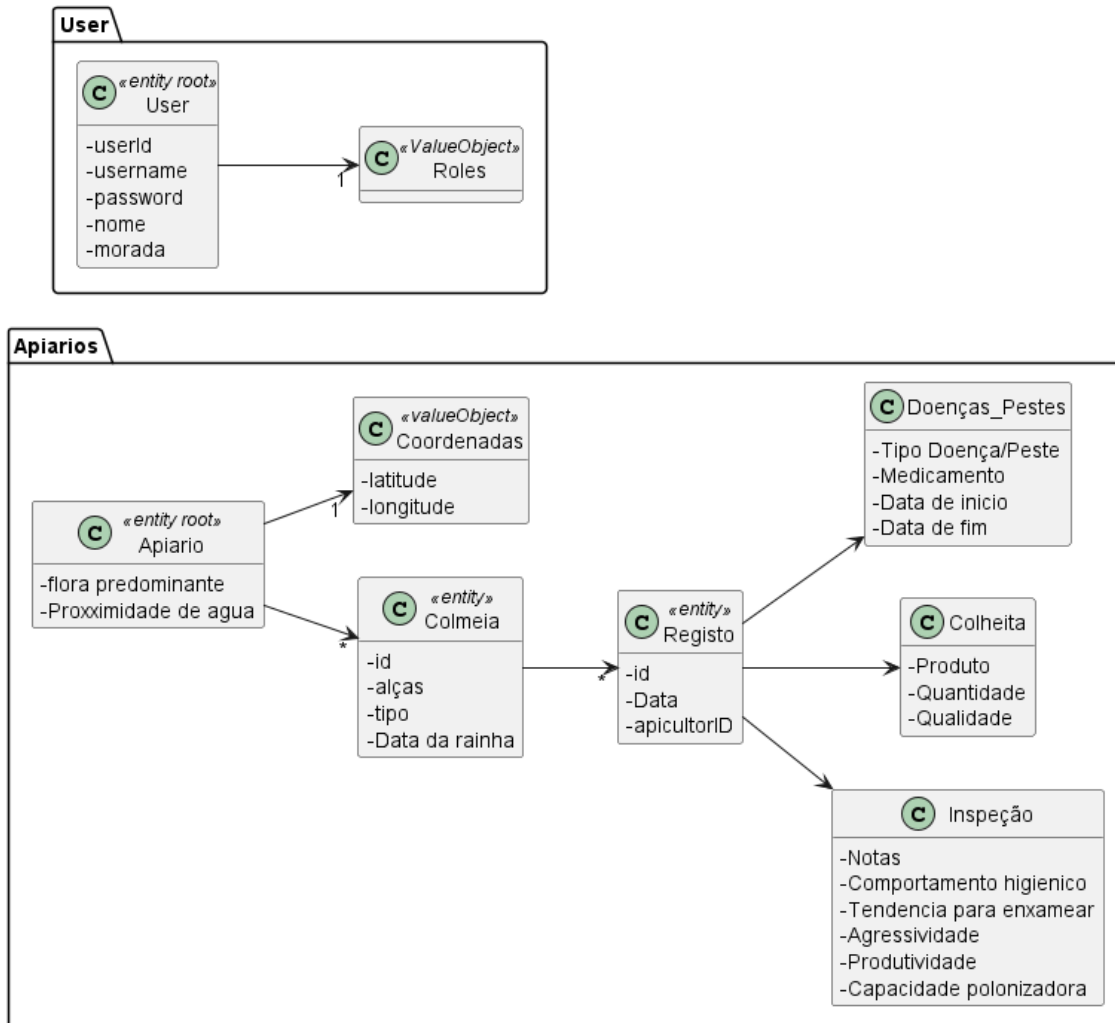


Figura 1 - Modelo de domínio

A partir do modelo de domínio acima, pode-se identificar dois agregados: o agregado dos utilizadores e o agregado dos apiários.

6.1 Agregado dos utilizadores

Este agregado engloba todos os usuários e suas funções correspondentes. Essas funções concedem permissões necessárias para executar ações específicas.

6.2 Agregado dos Apiários

Esse conjunto de dados abrange desde os apiários e suas colmeias, até os registros de trabalho realizados em cada colmeia. Esses conceitos estão intrinsecamente relacionados, uma vez que

os apiários representam a área onde as colmeias estão localizadas, sendo as colmeias elementos essenciais no processo da apicultura, e os registos de manutenção contêm informações diversas sobre o trabalho realizado em cada colmeia.

Apesar que os registos de colheita e registos de doenças e Pestes não estão presentes nas funcionalidades desenvolvidas, a estrutura já está preparada para as desenvolver mais tarde.

7 Diagramas de Componentes

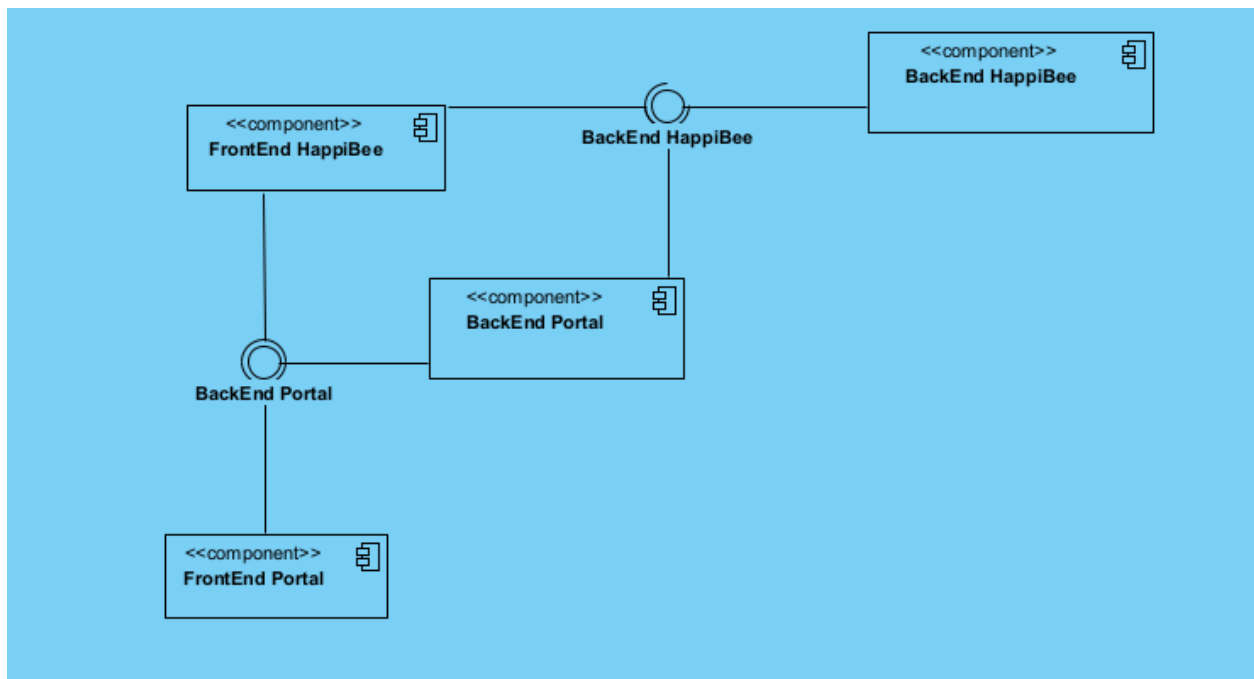


Figura 2 - Diagrama de componentes

A representação gráfica exibe nosso diagrama de componentes, revelando uma arquitetura modular. O diagrama é composto por quatro componentes distintos: HappiBee FrontEnd, HappiBee BackEnd, Portal FrontEnd e Portal BackEnd.

O HappiBee FrontEnd desempenha um papel crucial na interação do usuário, proporcionando uma interface clara e alinhada com suas expectativas. Este componente consome tanto a API do HappiBee BackEnd quanto a do Portal BackEnd.

8 User stories desenvolvidas

8.1 Como apicultor quero realizar a declaração anual de existências

a) Responsável pelo desenvolvimento

O responsável pelo desenvolvimento desta US foi o José Pessoa, número 1201007, com o apoio dos membros do grupo para analisar e tomar decisões importantes.

b) Aplicação desenvolvida

Para concretizar esta User Story, foi implementado um botão na página inicial que direciona para o registo da declaração anual, conforme representado na figura 3. Inicialmente, solicita a verificação das informações do utilizador e, após a validação, avança para a etapa seguinte, retratado na figura 4. De seguida, são exibidos os apiários do apicultor, juntamente com um botão que possibilita a submissão da declaração anual, como ilustrado na figura 5.



Figura 3 - Página inicial



Figura 4 - Informação do Apicultor



Figura 5 - Submeter declaração

c) Storymapping

Na figura 6 está representado o MVP (Minimum Viable Product) e na figura 7 o MVI, MVI2 e Backlog respetivamente.

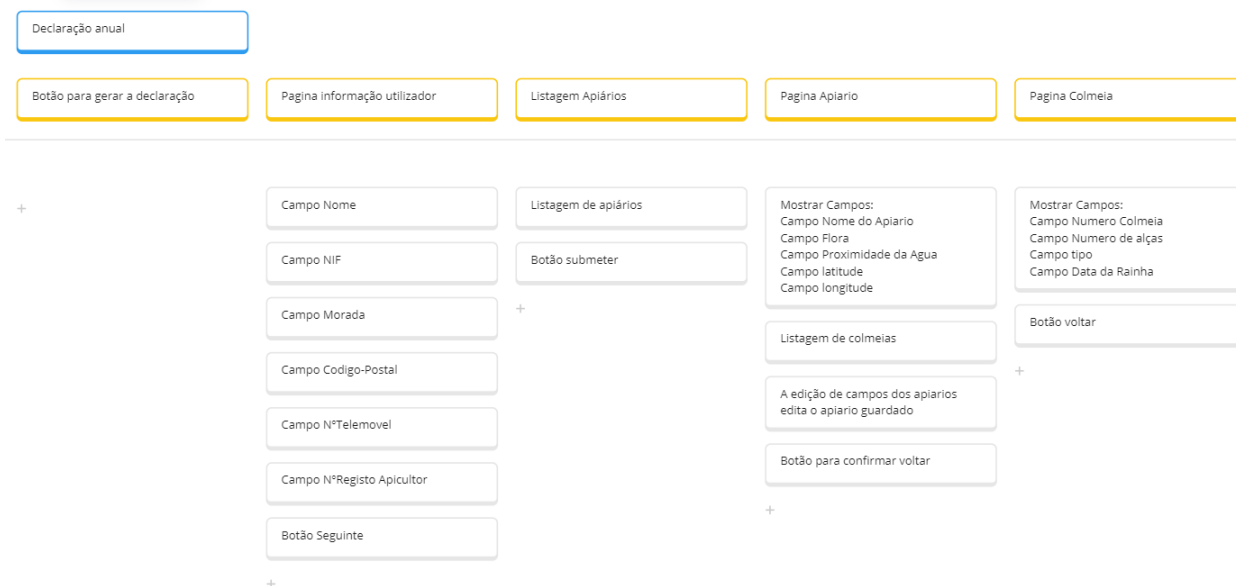


Figura 6 - MVP da declaração anual



Figura 7 - MVI,MVI2 e Backlog da declaração anual

Quanto a priorização, no MVP, estão representadas as funcionalidades que necessitam de estar implementadas para a declaração anual de existências funcionar, sendo as que necessitam de ser desenvolvidas primeiro. No MVI está presente a funcionalidade “Preenchimento automático da informação do utilizador”, que permite com que a declaração anual seja efetuada rapidamente e com menor chance de erro, pelo que foi desenvolvida de seguida. No MVI2 esta presente a funcionalidade “Atualização da informação do utilizador após o pedido” que permite o utilizador não ter que atualizar a sua informação posteriormente. Todas estas funcionalidades foram desenvolvidas no Sprint 1.

8.2 Como apicultor quero efetuar um pedido de transumância

a) Responsável pelo desenvolvimento

O responsável pelo desenvolvimento desta US foi o José Pessoa, número 1201007, com o apoio dos membros do grupo para analisar e tomar decisões importantes.

b) Aplicação desenvolvida

Para concretizar esta User Story, foi implementado um botão na página do apiário pretendido que direciona para o pedido de transumância, conforme representado na figura 8. De seguida, é exibido duas caixas de texto onde o apicultor pode preencher as coordenadas para onde pretende mover o apiário, e um botão para submeter o pedido, como ilustrado na figura 9.



Figura 8 - Página do Apiário

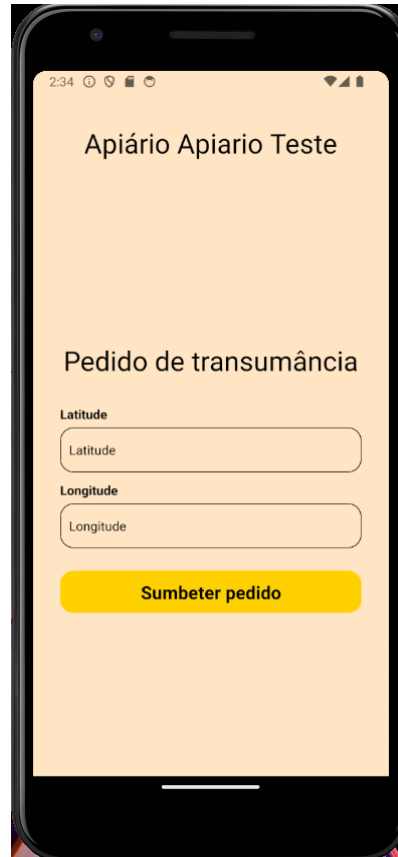


Figura 9 - Pedido de transumância

c) Storymapping

Na figura 10 está representado o MVP (Minimum Viable Product) e na figura 11 o MVI, MVI2 e Backlog respetivamente.



Figura 10 - MVP do pedido de transumância



Figura 11 - MVI, MVI2 e Backlog do pedido de transumância

Quanto a priorização, no MVP, estão representadas as funcionalidades que necessitam de estar implementadas para o pedido de transumância funcionar, sendo as que necessitam de ser desenvolvidas primeiro. No MVI está presente a funcionalidade "Adicionar estados do pedido de transumância" que permite o utilizador verificar o estado do seu pedido, pelo que foi desenvolvida de seguida. No MVI2 esta presente a funcionalidade "Filtrar por pedidos de transumância pendentes" que permite o utilizador identificar o apiário pretendido rapidamente. As funcionalidades apresentadas no MVP foram desenvolvidas no sprint 1, as do MVI no sprint 2 e a do MVI2 não foi desenvolvida.

8.3 Como apicultor quero poder criar um apiário, efetuar o respectivo pedido de instalação

a) Responsável pelo desenvolvimento

O responsável pelo desenvolvimento desta US foi o Carlos Rodrigues, número 1201001, com o apoio dos membros do grupo para analisar e tomar decisões importantes.

b) Aplicação desenvolvida

Para a realização desta User Story foi adicionada um botão a página principal para a criação de apiários, ao clicar neste botão irá ser redirecionado a página de registo de apiários. Nessa página estão os diversos campos a preencher e um botão para submeter o pedido de instalação.



Figura 12- Menu principal

The image shows a smartphone screen displaying a form titled "Pedido de instalação de apiário". Below the title is the subtitle "Informações gerais do apiário". The form contains five input fields, each with a label above it: "Nome do apiário", "Flora predominante", "Proximidade da água", "Latitude", and "Longitude". Each input field has a placeholder text matching its label. At the bottom of the form is a yellow button labeled "Submeter Pedido". The smartphone's status bar at the top shows the time 2:39 and various icons.

Figura 13 Página criação apiários

c) Storymapping

Nas figura 14 está representado o MVP (Minimum Viable Product) e na figura 15 o MVI, MVI2 e Backlog respetivamente.

The diagram illustrates a storymap for the apiary registration process. It begins with a horizontal sequence of three cards: "Criar um apiario" (highlighted with a blue border), "Pagina criação do apiario", and "Pedido de instalação" (highlighted with a yellow border). Below this sequence, a vertical timeline is shown, starting with the label "MVP | 75". The timeline includes several cards: "Campo Nome do Apiario", "Email", "Campo Flora", "Campo Proximidade da Agua", "Campo latitude", "Campo longitude", and "Botão Submeter". A plus sign (+) is positioned to the right of the "Campo Flora" card, indicating further details or a next step in the process.

Figura 14-MVP do registo dos apiario e o pedido de instalação



Figura 15-MV1, MV12 e Backlog da criação de apiários

Quanto as priorizações, no MVP, estão representadas as funcionalidades que necessitam de estar implementadas para a criação de apiários funcionar, sendo as que necessitam de ser desenvolvidas primeiro. No MV1 está presente a funcionalidade "Permitir a visualização dos pedidos de instalação" que permite ao apicultor ver os pedidos de instalação que estão pendentes. No MV12 esta presente a funcionalidade "Adicionar estados ao pedido" que permite ao apicultor acompanhar melhor o processo. Para o backlog foram ainda identificadas funcionalidades sobre o sistema de comunicação do processo e notificações. As funcionalidades apresentadas no MVP foram desenvolvidas no sprint 1 e houve algumas modificações no sprint 2.

8.4 Registrar a constituição do apiario

a) Responsável pelo desenvolvimento

O responsável pelo desenvolvimento desta US foi o Bruno Ribeiro, número 1201001, com o apoio dos membros do grupo para analisar e tomar decisões importantes.

b) Aplicação desenvolvida

Para a realização desta User Story foi adicionado um botão na página do apiário para adicionar colmeias. Em seguida é apresentada uma página, que contém uma lista de colmeias a adicionar que inicialmente vai estar vazia, um botão para adicionar colmeia e outro para submeter o registo das colmeias. Ao clicar no botão “Adicionar colmeia” irá ser redirecionado para a página de registo da colmeia com os vários campos a preencher. Após o registo da colmeia irá ser redirecionado para a página anterior que agora aparecerá a colmeia que acabou de ser registada na lista das colmeias, quando quiser finalizar o registo das colmeias o utilizador deve clicar no botão “Submeter pedido” que irá submeter o registo das colmeias.

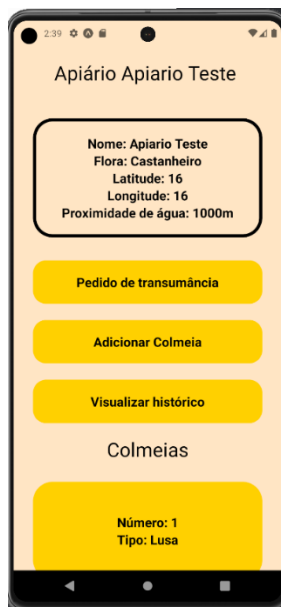


Figura 16 16- página detalhe dos apiários



Figura 17 17- Página de registo das colmeias

2:40

Pedido de instalação de apiário

Adicionar colmeia

Número da colmeia

Número da colmeia

Tipo da colmeia

Tipo da colmeia

Número de alças

Número de alças

Data da rainha

Data da rainha

Adicionar colmeia

Figura 18 18- Página de registo da colmeia

2:41

Instalação de colmeias

Colmeias

Nº	Tipo	Nº Alças
1	Lusitana	5

Adicionar colmeia

Submeter pedido

Figura 19 19- Página de registo das colmeias com a colmeia adicionada

c) Storymapping

Na figura 20 está representado o MVP (Minimum Viable Product) e na figura 21 o MVI, MVI2 e Backlog respetivamente.

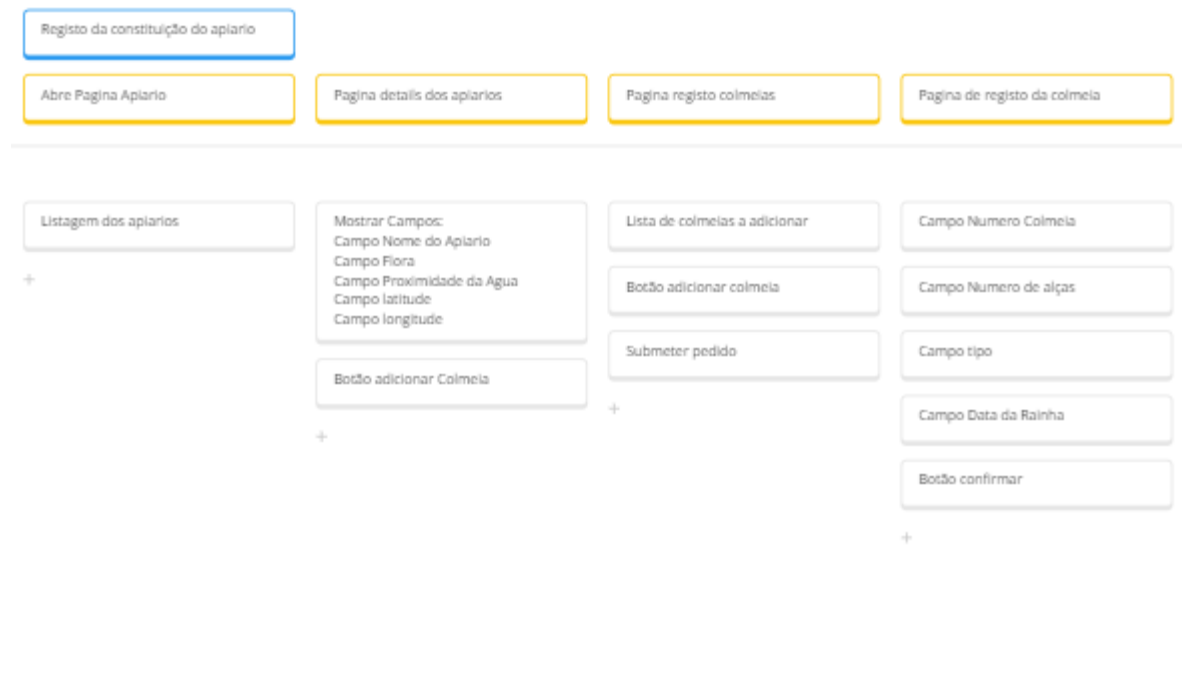


Figura 20-MVP do registo da constituição do apiário



Figura 21-MVI, MVI2 e Backlog do registo da constituição da colmeia

Quanto as priorizações, no MVP, estão representadas as funcionalidades que necessitam de estar implementadas para o registo da constituição de um apiário funcionar, sendo as que

necessitam de ser desenvolvidas primeiro. No MVI estão presentes funcionalidades de filtrar os apiários por nome e flora para o apicultor conseguir encontrar mais rapidamente o apiário. No MVI2 esta presente a funcionalidade de ordenar pela proximidade de água. As funcionalidades apresentadas no MVP foram totalmente desenvolvidas no sprint 1.

8.5 Como apicultor quero realizar a inspeção do apiário e registar a mesma

a) Responsável pelo desenvolvimento

O responsável pelo desenvolvimento desta US foi o Carlos Rodrigues, número 1201001, e o Bruno Ribeiro, número 1201000, com o apoio dos membros do grupo para analisar e tomar decisões importantes.

b) Aplicação desenvolvida

Para a realização desta User Story foi adicionado um botão na página da colmeia para adicionar um registo à colmeia. Em seguida é apresentada uma página, para cada campo a preencher durante a inspeção, com três opções, bom, médio e mau. Por fim foi adicionado um campo de escrita livre para que o apicultor consiga adicionar as observações desejadas.

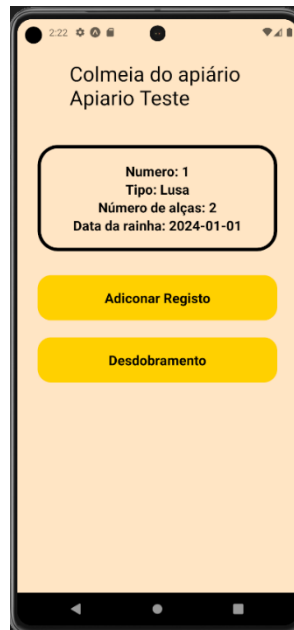


Figura 2222 - Página detalhes colmeia

Formulário de inspeção

Tendência para enxamear

☒ Bom

☐ Médio

☐ Mau

Anterior Seguinte

Figura 23 23 - Exemplo formulário inspeção

Formulário de inspeção

Observações

Observações da colmeia

Anterior Submeter

Figura 24 24 - Campo para adicionar observações

c) Storymapping

Nas figuras abaixo estão representados o MVP (Minimum Viable Product) e o MVI, MVI2 e Backlog respetivamente.

Figura 2525 - MVP de realizar inspeção

Figura 2626 - MVI, MVI2 e Backlog de realizar inspeção

No MVP estão presentes os aspetos necessários para que seja possível realizar uma inspeção. Sendo estes os formulários para introduzir o resultado das diferentes métricas. Para além disto, foram identificados alguns aspetos que adicionariam valor a esta funcionalidade e colocados no MVI1 e MVI2, sendo que as que estão no MVI1 foram consideradas mais importantes. No MVI1 foi identificado que seria interessante contar com opções de filtro e visualização de inspeções anteriores, assim como anexo de imagens à inspeção. Já para o MVI2 seria interessante adicionar gráficos de evolução da colmeia e, em cada campo do formulário, a opção selecionada na última inspeção. Para o backlog foram ainda identificadas duas funcionalidades, filtrar por opção selecionada e confirmação dos valores inseridos. Esta User Story foi totalmente desenvolvida no sprint 1.

8.6 Como apicultor quero realizar um desdobramento e registar a mesma

a) Responsável pelo desenvolvimento

O responsável pelo desenvolvimento desta US foi o Francisco Borges, número 1201477, com o apoio dos membros do grupo para analisar e tomar decisões importantes.

b) Aplicação desenvolvida

Para esta User Story foi necessário adicionar um botão na página de detalhes da colmeia. Na página do desdobramento é apresentado um campo de input onde é possível incrementar ou decrementar o número de alças para a nova colmeia.



Figura 2727 - Página detalhes colmeia

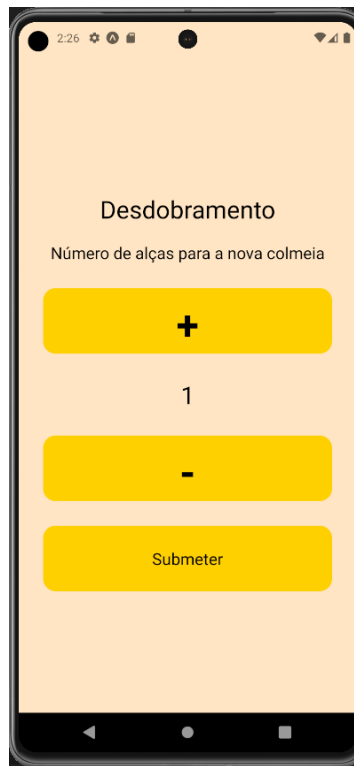


Figura 2828 - Página desdobramento

c) Storymapping

Nas figuras abaixo estão representados o MVP (Minimum Viable Product) e o MVI, MVI2 e Backlog respetivamente.



Figura 2929 - MVP de realizar desdobramento

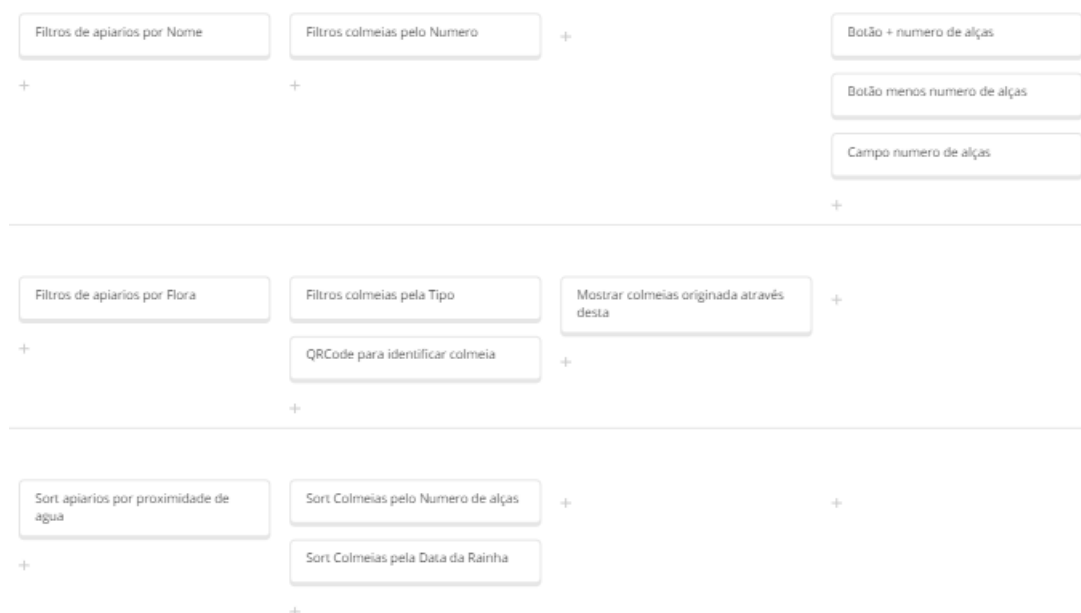


Figura 3030 - MVI, MVI2 e Backlog de realizar desdobramento

No MVP estão presentes os aspetos necessários para que seja possível realizar um desdobramento. Sendo estes a página dos apiários, os detalhes da colmeia e a página de desdobramento com um input para intruduzir o número de alças para a nova colmeia. Para além disto, foram identificados alguns aspetos que adicionariam valor a esta funcionalidade e colocados no MVI1 e MVI2, sendo que as que estão no MVI1 foram consideradas mais importantes. No MVI1 foi identificado que seria interessante contar com opções de filtro dos apiários e colmeias, assim como dois botões para adicionar e decrementar o número de alças da nova colmeia. Já para o MVI2 seria interessante adicionar diferentes filtros, QR code para identificação da colmeia e mostrar as colmeias originadas através da selecionada. Para o backlog foi identificado funcionalidade de ordenação de apiários e colmeias. Esta User Story foi totalmente desenvolvida no sprint 1.

9 Requisitos não funcionais

9.1 Usabilidade

Em termos de usabilidade, a aplicação mobile deve ser intuitiva e incluir mecanismos que facilitem a interação com o apicultor, tendo em conta as limitações de movimento causadas pela roupa própria da atividade. Para alcançar este objetivo, a equipa optou por desenvolver páginas na aplicação, com botões grandes, destinadas a melhorar a acessibilidade nas situações em que o apicultor enfrenta dificuldades de movimento.

9.2 Performance

Em setembro de cada ano, todos os apicultores do país devem preencher a declaração anual de existências. Em Portugal existem cerca de 25 mil apicultores e é esperado que preencham a declaração na última semana de setembro. Por este motivo, nessa semana, são esperados cerca de 5 mil utilizadores diários. Como o número de pedidos não é muito elevado, a equipa conseguiu atingir este objetivo através de uma aplicação bem estruturada.

9.3 Segurança

O sistema deve incorporar um controlo de acesso às funcionalidades, atribuindo diferentes privilégios aos vários utilizadores, limitando o acesso apenas ao estritamente necessário. Além disso, todos os dados pessoais recolhidos dos utilizadores devem ser encriptados antes de serem armazenados. Para alcançar este propósito, a aplicação requer que o utilizador se autentique na aplicação. As informações do utilizador são encriptadas na base de dados, e a ferramenta bcrypt é utilizada tanto para encriptar como para desencriptar, verificando assim a correção das informações inseridas pelo utilizador.

9.4 Confiabilidade

Como a apicultura é normalmente praticada em zonas florestais, onde o acesso à internet pode ser limitado, a aplicação deve ser capaz de armazenar localmente as atividades do apicultor quando a conexão à internet não estiver disponível. Posteriormente, esses dados locais são usados para atualizar a base de dados do sistema. Com esse propósito, a aplicação guarda uma cópia dos dados na local storage sempre que o utilizador a utiliza com acesso à internet. Se não houver acesso à internet, a aplicação utiliza os dados mais recentes disponíveis localmente em vez de os buscar à base de dados, e atualiza as alterações assim que o utilizador utiliza a aplicação com acesso à internet.