|  |  |  |  |
| --- | --- | --- | --- |
|  | | Curso de TeSP em  Programação de Sistemas de Informação  Projeto em  Sistemas de Informação (PSI) | |
|  |  | |  |
| **Ano Letivo 2022/2023** |  | | **2º Ano, 1º Semestre** |
|  | | | |
| **Projeto em Sistema de Informação** | | | |

Icon

Description automatically generated

[substituir pelo logotipo do próprio projeto]

**Relatório de Especificação do SI**

***Cãopanhia***

|  |  |
| --- | --- |
|  |  |
| **Grupo:** PL1-X | **Data de entrega:** 2023-02-?? |
| **Nº** 2211881 | Leonel Diogo Ribeiro Freitas |
| **Nº** 2201120 | Tiago João Mota Vital |

ÍNDICE

[Índice de Figuras 4](#_Toc116994832)

[Índice de Tabelas 5](#_Toc116994833)

[1 Introdução 6](#_Toc116994834)

[1.1 Sumário executivo 6](#_Toc116994835)

[2 Definição da Lógica de Negócio 7](#_Toc116994836)

[3 Análise de Impacto 8](#_Toc116994837)

[4 Análise Concorrencial 9](#_Toc116994838)

[4.1.1 Kiwoko 9](#_Toc116994839)

[4.1.2 Goldpet 10](#_Toc116994840)

[4.1.3 Associação Zoófila de Leiria 11](#_Toc116994841)

[4.1.4 Comparação dos Sistemas 12](#_Toc116994842)

[4.1.5 Enquadramento da análise concorrencial no SI 12](#_Toc116994843)

[5 *Roles* e Requisitos do SI 13](#_Toc116994844)

[6 User Stories 19](#_Toc116994845)

[7 Diagrama de Entidade-Relacionamento 21](#_Toc116994846)

[8 Wireframes/Mockups 22](#_Toc116994847)

# Índice de Figuras

[Figura 1 – Diagrama de classes do SI 17](#_Toc83646898)

[Figura 2 – Wireframe/Mockup do ecrã principal (no exemplo: esq. Wireframe; dir. Mockup) 18](#_Toc83646899)

# Índice de Tabelas

[Tabela 1 – Descrição do Sistema 1 9](#_Toc83646982)

[Tabela 2 – Descrição do Sistema 1 10](#_Toc83646983)

[Tabela 3 – Descrição do Sistema 1 10](#_Toc83646984)

[Tabela 4 – Resumo das características/funcionalidades dos Sistemas concorrenciais 11](#_Toc83646985)

[Tabela 5 – Roles dos diferentes end-users do SI 12](#_Toc83646986)

[Tabela 6 – Requisitos funcionais Front-Office 12](#_Toc83646987)

[Tabela 7 – Requisitos funcionais Back-Office 13](#_Toc83646988)

[Tabela 8 – Requisitos funcionais App mobile 13](#_Toc83646989)

[Tabela 9 – Requisitos Não funcionais 14](#_Toc83646990)

# Introdução

Entre 2020 e 2021, o abandono de animais aumentou mais de 30%. Cerca de 43.600 cães e gatos foram abandonados e recolhidos por centros municipais em Portugal em 2021, de acordo com um novo relatório do Instituto da Conservação da Natureza e das Florestas.

De modo a tentar combater esses números, surgiu a Cãopanhia.

A Cãopanhia é um sistema on-line com a finalidade de facilitar a adoção de cães através de anúncios bem estruturados. Qualquer utilizador poderá criar anúncios e/ou responder a anúncios já criados por outro utilizador.

Para além de facilitar a adoção de cães, através deste sistema será possível comprar produtos como ração, medicação, brinquedos, etc de modo a faciliar o cliente no processo de adoção. Haverá ainda veterinários no sistema responsáveis por cuidar dos cães que mais necessitem.

Este projeto surge no âmbito do curso de Programação de Sistemas de Informação do Instuto Politécnico de Leiria, e liga as unidades curriculares de Plataformas de Sistemas de Informação (onde aprendemos a utilizar a *framework* usada no projeto para a vertente Web, neste caso a “Yii2”), Serviços e Interoperabilidade de Sistemas (onde aprendemos a criar e trabalhar com uma API), Acesso Móvel a Sistemas de Informação (onde aprendemos a utilziar o software destinado à criação e desenvolvimento da aplicação android do projeto) e Projeto em Sistemas de Informação (onde somos orientados, não só no planeamento como no desenvolvimento do projeto).

## Sumário executivo

Este relatório está dividido em 7 pontos cruciais.

Começando pela Definição da Lógica de Negócio, onde é especificado o que nós pretendemos com este projeto, e pela Análise de Impacto, que descreve o impacto que nós acreditamos que este tipo de software vá ter na sociedade.

A Análise Concorrencial refere-se à análise de quatro *softwares* semelhantes e onde retiramos todos os aspetos positivos e negativos desses mesmos, no final fazemos ainda uma comparação de todos e uma breve descrição em como podemos enquadrar esses *softwares* no nosso projeto.

No ponto 5 (*Roles* e Requisitos do SI) são especificados os roles do nosso projeto, tal como todos os requisitos funcionais e não funcionais para o sistema, e os respetivos User Stories são também eles especificados no ponto 6.

Consta ainda o desenho de toda a base de dados no ponto 7 (Diagrama de Entidade-Relacionamento), e os mockups do sistema no ponto 8 (Wireframes/Mockups).

# Definição da Lógica de Negócio

Este sistema tem como grande objetivo facilitar a adoção de cães, possibilitando ao utilizador criar anúncios de adoção e/ou responder a anúncios já criados por outros utilizadores. O sistema contará também com uma loja de modo a auxiliar o cliente com produtos que o ajudaram no processo após a adoção de um cão.

Para além da role de cliente existem mais três roles:

* Administrador: responsável pela gestão de todos os utilizadores do sistema e da gestão do sistema em si;
* Gestor: responsável por toda a gestão e manutenção da loja;
* Veterinário: responsável pelo processo de tratamentos de animais, podendo marcar consultas com os clientes que possuem os animais para adoção que mais necessitam dessa ajuda.

Adicionando a tudo isto, haverá um questionário que auxiliará o cliente no processo de adoção de cães, indicando a raça de cão que mais se adequa ao cliente mediante um conjunto de perguntas realizadas pelo sistema.

Todas estas funcionalidades estarão disponíveis num sistema web, no entanto o sistema contará também com uma aplicação móvel.

A aplicação móvel foi pensada para este sistema como um elemento mais pessoal, ou seja, o grande objetivo da aplicação vai ser gerir os animais pessoais do utilizador, podendo adicionar novos registos, editar registos ou eliminar registos.

Para além da gestão dos animais pessoais, será também possível consultar encomendas já realizadas, visualizar se há consultas marcadas pelo veterinário a algum dos cães que o cliente tenha para adoção, e consultar um mapa com a indicação de parques e serviços na proximidade da localização do cliente.

Salvo o cliente, apenas o veterinário terá acesso à aplicação, podendo consultar todas as suas consultas marcadas e disponibilizando de um mapa com as coordenadas do local dessas mesmas consultas, facilitando assim a deslocação do veterinário.

# Análise de Impacto

O abandono de animais tem crescido a um ritmo estonteante nos últimos anos, muitas vezes devido a falta de responsabilidade ou falta de preparação para ter um animal de estimação.

Assim sendo, é fundamental existirem sistemas e associações que facilitem a adoção de animais e que auxiliem qualquer individuo na experiência de ter um animal de estimação.

O sistema “Cãopanhia” foca-se precisamente nesse contexto. Pretende-se que um cliente possa criar anúncios para a adoção de cães e/ou responder a anúncios já existentes da forma mais simples e intuitiva possível. Para além da adoção, o sistema contará também com uma loja com uma diversidade bastante extensa de produtos que facilitará um cliente inexperiente na adoção de um cão.

A necessidade de construir este sistema surge pela falta de sistemas para esta área. Uma situação um pouco preocupante tendo em conta o estado do mundo atual, onde tudo tem uma vertente digital, de fácil acesso e atualização de dados instantânea.

**Vantagens:**

* Rápido acesso a informação;
* Não existe a necessidade de deslocamento a canis, correndo o risco de não encontrar o pretendido pelo cliente;
* Questionário online que auxiliará um individuo a encontrar a raça que mais se adequa a si mesmo;
* Criação de um anuncio para a adoção de um cão e possibilidade de adotar um cão de uma forma bastante intuitiva e simples;
* Disponibilização de uma loja que auxilia qualquer individuo a cuidar dos seus animais de estimação;
* Pagamentos digitais atempados e seguros;

**Desvantagens:**

* O sistema necessita de um dispositivo móvel e acesso à internet para funcionar perfeitamente;
* O sistema apenas engloba cães.

# Análise Concorrencial

Nesta secção iremos avaliar e comparar diferentes sistemas que já estão no mercado, tentaremos encontrar os seus defeitos e os seus pontos positivos e com esta pesquisa poderemos ter uma ideia de o que implementar no nosso sistema e/ou o que poderemos melhorar para conseguirmos ser inovadores e únicos para termos uma grande vantagem no mercado.

### Kiwoko

Nesta próxima tabela estará resumido as características do sistema feito pela Kiwoko

Tabela – Descrição do Sistema Kiwoko

|  |  |
| --- | --- |
|  | |
| **Nome:** | Kiwoko |
| **Site/App:** | Site: <https://www.kiwoko.pt/> // App: Kiwoko |
| **Descrição:** | A Kiwoko é uma cadeia de lojas, líder no setor de animais de estimação. Esta empresa é especialista em animais e têm uma ampla oferta de produtos e um preço acessível. No website é possível encontrar gamas completas de alimentação e tudo o que for necessário para o cuidado de cães, gatos, pássaros, roedores, répteis e peixes. Kiwoko tem mais de 5.000 produtos disponíveis em loja e 50.000 mediante encomenda. |
| **Vantagens:** | Website - Diferentes abas para quase todos os animais domésticos, categorias bem organizadas, botão para voltar ao início da página // Aplicação – as mesmas vantagens que o website e aba na parte debaixo para diferentes ações na aplicação |
| **Desvantagens:** | Website: Página sobrecarregada com informação/promoções, página demasiado grande, apresentação do website no final da página // App: A aplicação é praticamente uma copia do website |
| **O que falta:** | Um sistema para adoção e informação das diferentes raças de cães |

### Goldpet

Nesta próxima tabela estará resumido as características do sistema feito pela Goldpet

Tabela – Descrição do Goldpet

|  |  |
| --- | --- |
|  | |
| **Nome:** | Goldpet |
| **Site:** | https://goldpet.pt |
| **Descrição:** | 100% portuguesa, a Goldpet surgiu em 2004 com a missão de proporcionar felicidade e bem-estar a todos os animais de estimação. Hoje, é a loja favorita dos animais de estimação, que vibram a cada nova encomenda que recebem em casa, mas também dos tutores, que aqui encontram tudo o que procuram, ao melhor preço do mercado. Além de milhares de produtos para cães, gatos, aves, roedores, peixes e répteis, das principais marcas da área Pet, a Goldpet oferece, ainda, serviços dedicados ao cuidado animal. |
| **Vantagens:** | Website bastante simples com animações e página interativa para se tornar apelativo para o utilizador, página bem organizada, um blogue onde é publicado posts com informação sobre diferentes animais |
| **Desvantagens:** | Loja com produtos para diferentes animais não podendo focar-se num só animal |
| **O que falta:** | Um sistema para adoção e informação das diferentes raças de cães |

### Associação Zoófila de Leiria

Nesta próxima tabela estará resumido as características do sistema feito pela Associação Zoófila de Leiria

Tabela – Descrição do AZL

|  |  |
| --- | --- |
|  | |
| **Nome:** | Associação Zoófila de Leiria |
| **Site:** | <https://azlfa.com/> |
| **Descrição:** | Na AZL (Associação zoófila de Leiria) os valores são o compromisso, a integridade, a sensibilização e a responsabilidade, sendo uma preocupação constante no respeito pela vida e combate ao abandono de animais. A AZL tem por missão apoiar a defesa e proteção de todos os animais, em particular, os que se encontrem em situação de sofrimento e risco de vida, usando todos os recursos necessários e possíveis para o interesse e bem-estar animal. |
| **Vantagens:** | Redes sociais apresentadas no topo do website bastante simples e fácil de acessar a informação que se pretende, adoção com dados bem detalhados e descrição bastante apelativa para a adoção do animal |
| **Desvantagens:** | Website pouco apelativo. |
| **O que falta:** | Nada a apontar |

### Comparação dos Sistemas

Em seguida iremos fazer uma comparação dos sistemas anteriormente estudados para conseguirmos estruturar detalhadamente o nosso sistema.

Tabela – Resumo das características/funcionalidades dos Sistemas concorrenciais

|  |  |  |  |
| --- | --- | --- | --- |
| Funcionalidades | Kiwoko | Goldpet | AZL |
| Loja de Produtos | x | x | x |
| Adoção de Animais | - | - | x |
| Aplicação Mobile | x | - | - |
| Informação sobre os animais | - | x | x |
| Sistema de Anúncios | - | x | x |
| Veterinários prontos para acudir | - | - | - |
| Sistema de Doações | - | - | x |

### Enquadramento da análise concorrencial no SI

Estes 3 sistemas foram escolhidos pelo mesmo propósito que foi “Animais”.

Apesar de animais serem uma grande parte presente no nosso dia a dia existe pouca oportunidade no mercado online. Pois quando se fala do mercado na área dos animais, só é possível afirmar que a vasta maioria seja de produtos de animais, enquanto os canis não têm tanto impacto no mercado. Por isso o nosso grupo decidiu unir tanto um canil como uma loja para poder juntar estas áreas do mercado tão distintas.

A analise feita no capítulo acima ajudou-nos a perceber o mercado dos animais e como devemos atingir de certa forma esse mercado para que possamos esmagar a concorrência e poder fazer um sistema que seja acessível e simples de usar.

Esta análise ajudou bastante na ideia de estruturação e implementação do nosso sistema. Conseguimos identificar os pontos positivos e negativos nos diferentes sistemas e graças a isso podemos abordar as diferentes vantagens e adicioná-las e/ou implementar as desvantagens anteriormente identificadas, mas desta vez refeitas para melhor proveito.

# *Roles* e Requisitos do SI

Esta secção contém a descrição detalhada de todos os roles existentes no sistema, os requisitos funcionais da aplicação Web (front-office e back-office) e aplicação móvel, e os requisitos não funcionais de todo o sistema.

Na Tabela 9 é feito um cruzamento entre todos os requisitos funcionais e todos os roles do sistema.

Tabela – Roles dos diferentes end-users do SI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Descrição detalhada | Back-office | Front-office | App mobile |
| Administrador | Gestão de todos os utilizadores do sistema e gestão do sistema | x |  |  |
| Gestor | Gestão e manutenção da loja | X |  |  |
| Veterinário | Marcação de consultas com os clientes que possuem os animais para adoção que mais necessitam dessa ajuda. |  | x | X |
| Cliente | Possibilidade de criar e/ou responder a anúncios de adoção de cães |  | x | x |
| Role | Descrição detalhada | Back-office | Front-office | App mobile |

Tabela – Requisitos funcionais Front-Office

|  |  |  |
| --- | --- | --- |
| # | Requisito | Prioridade |
| RF-FO-01 | O utilizador não registado deve conseguir registar-se no sistema | Alta |
| RF-FO-02 | O utilizador deve conseguir realizar o login com as suas credenciais | Alta |
|  | O sistema deve apresentar mensagens de erro caso as credenciais inseridas no login não correspondam a um utilizador ou não sejam válidas | Médio |
|  | O utilizador deve conseguir terminar a sua sessão | Alta |
| RF-FO-0 | O utilizador deve conseguir criar um anúncio de adoção | Alta |
| RF-FO-0 | O utilizador deve conseguir editar um anúncio de adoção já criado por si mesmo | Alta |
| RF-FO-0 | O utilizador deve conseguir eliminar um anúncio de adoção já criado por si mesmo | Médio |
| RF-FO-0 | O utilizador deve conseguir procurar anúncios | Alta |
|  | O utilizador deve conseguir filtrar a sua pesquisa por anúncios por raça, idade, género, microship, castração e localidade | Médio |
|  | O utilizador deve conseguir adotar um cão de um anúncios de adoção anteriormente criado por outro utilizador | Alta |
|  | O utilizador deve conseguir consultar os produtos da loja | Alta |
|  | O utilizador deve conseguir filtrar a sua pesquisa por produtos da loja por tipo de produto | Médio |
|  | O utilizador deve conseguir adicionar os produtos e a quantidade de produtos que pretende a um carrinho de compras | Médio |
|  | O utilizador deve conseguir retirar os produtos que desejar do carrinho | Médio |
|  | O utilizador deve conseguir reduzir ou aumentar a quantidade de um determinado produto presente no carrinho | Médio |
|  | O sistema deve guardar o carrinho quando o utilizador estiver offline | Médio |
|  | O utilizador deve conseguir associar um método de expedição à sua encomenda | Médio |
|  | O utilizador deve conseguir realizar uma encomenda | Alta |
|  | O utilizador deve conseguir visualizar o seu histórico de encomendas | Médio |
|  | O utilizador deve conseguir responder a um questionário que o auxiliará na escolha da raça de cão que mais se adeque | Baixa |
|  | O utilizador deve conseguir aceder a uma página com informações uteis e importantes | Baixa |
|  | O utilizador deve conseguir aceder a uma página com possíveis dados para realizar uma doação | Baixa |
|  | O veterinário deve conseguir escolher um determinado cão e marcar uma consulta | Alta |  |
|  | O veterinário deve conseguir aceder ao seu histórico de consultas e alterar o estado de um dos cães que consultou | Médio |  |

Tabela – Requisitos funcionais Back-Office

|  |  |  |
| --- | --- | --- |
| # | Requisito | Prioridade |
| RF-BO-01 | O administrador deverá ter acesso aos dados dos utilizadores e veterinários | Alta |
| RF-BO-02 | O administrador deverá poder alterar qualquer dado previamente criado | Média |
| RF-BO-03 | O administrador deverá poder eliminar qualquer dado previamente criado | Baixa |
|  | O administrador deverá gerir as encomendas feitas pelos utilizadores |  |
|  | O administrador deverá gerir o stock da loja |  |
|  | O administrador deverá gerir pré-encomendas de utilizadores |  |
|  | O administrador deverá poder cancelar encomendas |  |
|  | O administrador deverá gerir a disponibilidade dos veterinários |  |
|  |  |  |

Tabela – Requisitos funcionais App mobile

|  |  |  |
| --- | --- | --- |
| # | Requisito | Prioridade |
| RF-AM-01 | O utilizador deve conseguir realizar o login com as suas credenciais | Alta |
| RF-AM-02 | O sistema deve apresentar mensagens de erro caso as credenciais colocadas no login sejam inválidas ou não correspondam ás de um utilizador já registado | Alta |
| RF-AM-03 | O sistema deve apresentar um menu com ligações a todas as atividades disponíveis ao utilizador | Alta |
|  | O utilizador deve conseguir terminar a sua sessão | Alta |
|  | O utilizador deve conseguir aceder a um mapa com parques e serviços diversificados próximos à sua localização | Baixa |
|  | O veterinário deve conseguir consultar todas as suas consultas marcadas | Alta |
|  | O veterinário deve conseguir visualizar os detalhes de qualquer uma das suas consultas | Média |
|  | O veterinário deve conseguir aceder a um mapa com as coordenadas da consulta selecionada por si | Baixa |
|  | O utilizador deve conseguir visualizar todos os seus animais pessoais | Alta |
|  | O utilizador deve conseguir visualizar os detalhes de qualquer um dos seus cães | Alta |
|  | O utilizador deve conseguir editar os dados de qualquer um dos seus cães | Alta |
|  | O utilizador deve conseguir criar um novo registo de um cão | Alta |
|  | O utilizador deve conseguir eliminar um dos seus cães | Média |
|  | O utilizador deve conseguir visualizar o seu histórico de encomendas | Média |
|  | O utilizador deve conseguir visualizar todas as consultas marcadas por um veterinário a um dos seus cães para adoção | Alta |

Tabela – Cruzamento dos requisitos funcionais e respetivos roles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Requisito | Role1 | Role2 | Role3 | Role4 |
| RF-FO-01 | .... | x |  |  |  |
| RF-FO-02 |  |  | x |  |  |
| RF-FO-03 |  |  |  | x |  |
| ... |  |  |  |  |  |
| RF-BO-01 |  | x | x | x |  |
| RF-BO-02 |  |  |  |  |  |
| RF-BO-03 |  |  |  |  |  |
| ... |  |  |  |  |  |
| RF-AM-01 |  |  |  |  |  |
| RF-AM-02 |  |  |  |  |  |
| RF-AM-03 |  |  |  |  |  |
| ... |  |  |  |  |  |

<Devem ser descritos todos os requisitos não funcionais que o site deve apresentar (ponto de vista do que sistema deve ser). Uma tabela para cada tipo de RNF.

**ID**: Nomenclatura exemplo

**Descrição**: Sujeito + Ação (com o verbo “deve”). Ex: O sistema deve utilizar a API xpto.

**Prioridade**: Alta, Média, Baixa (diz respeito à sua implementação, se é fulcral para o objetivo do sistema)

Usabilidade (Usability) – foca-se na perspetiva da interação com o utilizador (facilidade de uso). Ex: A interface do website deve ser user-friendly e fácil de usar.

Fiabilidade (Reliability) – habilidade do software se comportar de forma consistente e aceitável para o utilizador; é a probabilidade e percentagem do software funcionar sem falhas para um período específico de tempo ou uso. Ex: Os candidatos devem aceder aos seus currículos 98% das vezes, sem falhas.

Segurança (Security) – define como o sistema se encontra protegido ou como deve ser efetuado o acesso ao sistema. Ex: Devem ser efetuadas 2 cópias de segurança dos dados do sistema diariamente, sendo que 1 delas deve encontrar-se em local externo.

Eficiência (Performance) – definem o quão bem o sistema lida com capacidade, taxa de transferência e tempo de resposta (desempenho, espaço, tempo) Ex: O tempo de carregamento do website não deve superior a 1 segundo.

Disponibilidade (Availability) – define o tempo de funcionamento do sistema, o tempo que leva para reparar uma falha e o tempo entre ciclos. Ex: O diretor de recursos humanos deve poder publicar empregos no site durante a semana e ao fim-de-semana, a qualquer hora do dia (disponibilidade 24x7).

Ambiente (Portability) – especificam o ambiente operacional do sistema (capacidade de adaptação). Ex: O sistema deve ser capaz de correr no SO Windows e Linux.

Desenvolvimento (Implementation) – indicam restrições ao uso de ferramentas ou linguagens de programação; ou do desenvolvimento do projeto em si; standards. Ex: O sistema deve ser desenvolvido utilizando a linguagem de programação Java.

>

Tabela – Requisitos Não funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| # | Requisito | Tipo | Prioridade |
| RNF-01 | A formatação do texto deve ser organizada com os devidos espaçamentos entre linhas e palavras. | Usabilidade | Baixa |
| RNF-02 | A aplicação deve ter uma navbar para fácil acesso as diferentes categorias da aplicação | Usabilidade | Alta |
|  | A aplicação deve ser apelativa e user friendly | Usabilidade | Media |
|  | O utilizador devera ter acesso aos seus animais mesmo sem internet | Fiabilidade |  |
|  | O utilizador devera ter acesso aos números dos veterinários/empresa mesmo sem internet | Fiabilidade |  |
|  | O sistema devera apresentar se um veterinário esta disponivel | Fiabilidade |  |
|  | O utilizador deverá ter acesso aos dados em diferentes dispositivos que esteja iniciada a sessão |  |  |
|  | O sistema devera ter um sistema de login para garantir segurança dos dados do utilizador | Segurança |  |
|  | Devem ser efetuadas 2 cópias de segurança dos dados do sistema semanalmente, sendo que 1 delas deve encontrar-se em local externo. |  |  |
|  | Deve estar otimizado para hardwares menos potentes. | Eficiência |  |
|  | Na criação do animal de estimação o servidor não deverá demorar mais do que 3 segundos para adicionar os dados na base de dados |  |  |
|  | A aplicação e os dados essenciais deverão estar disponíveis para o utilizador 24/7 |  |  |
|  | Deve ser atualizado com frequência. | Desenvolvimento | Baixa |
|  | O código deve ser bem definido, simples e esclarecedor para facilitar a leitura. | Desenvolvimento |  |
|  | Na aplicação deve ser usada a linguagem de programação Java para aplicação mobile e utilização da framework Yii 2 para o website. | Desenvolvimento |  |

# User Stories

<Devem ser especificados os requisitos funcionais do ponto de vista do utilizador sob forma de User Stories As *issues* devem ser estimadas em Story Points utilizando a sequência de Fibonacci: 1, 2, 3, 5, 8, 13, 20 (máx).>

As a <role>, I want <goal/desire> so that <benefit>

(Como um <função>, eu quero <objetivo/desejo> para que <benefício>)

Exemplo: **Como** Administrador de Bases de Dados, **eu devo** ser capaz de reverter um número selecionado de dados **para que** a versão anterior a eles seja restaurada.

Focar nestes 3 e evitar o Como (excluir aspetos de desenho, interface design statements).

“Como [persona],” – Para quem estamos a construir isto? Não estamos atrás de um título de trabalho, estamos atrás da persona da pessoa. Entendemos como a pessoa funciona, como pensa e como se sente.

“eu [quero/gostaria que],” – Aqui descrevemos o intuito, mas não as funcionalidades que eles usam. O que é que eles estão a tentar atingir? Deve ser livre da implementação – se estivermos a descrever algo específico da UI, e não o objetivo do utilizador, estamos a perder o objetivo.

“[para quê].” – como é que o desejo imediato deles para fazer algo se integra na big picture? Qual o objetivo geral que estão a tentar atingir? Qual o principal problema que precisa de ser resolvido?

|  |  |
| --- | --- |
|  |  |
| Título: US1 – Efetuar lançamento de dados | **SP**: 3 |
| Descrição: Como jogador quero ser capaz de lançar os dados de forma a poder fazer a jogada  Critérios de Aceitação:   * Só é possível efetuar uma jogada de cada vez * As faces dos dados têm de ser visíveis após o lançamento | |
| Título: US2 – Adicionar cliente | **SP**: 5 |
| Descrição: Como utilizador quero poder adicionar um cliente de modo a ficar registado  Critérios de Aceitação:   * Os campos nome, morada, contato e NIF têm de ser obrigatoriamente preenchidos * Não pode haver um cliente com o mesmo NIF | |
| Título: US – | **SP**: |
| Descrição:  Critérios de Aceitação: | |
| Título: US | **SP**: |
| Descrição:  Critérios de Aceitação: | |
| Título: US | **SP**: |
| Descrição:  Critérios de Aceitação: | |
| Título: US | **SP**: |
| Descrição:  Critérios de Aceitação: |  |
|  |  |
|  |  |

# Diagrama de Entidade-Relacionamento

<Modelo de dados do projeto e respetiva explicação>

Diagram

Description automatically generated

Figura – Diagrama de Entidade-Relacionamento

# Wireframes/Mockups

<Desenho dos wireframes já a aproximarem-se de mockups (mais detalhados). 1 para cada ecrã principal tanto para a app móvel como para o website>

Graphical user interface, application

Description automatically generated

Figura – Wireframe/Mockup do ecrã principal (no exemplo: esq. Wireframe; dir. Mockup)