|  |
| --- |
| Análise e Projeto de Sistemas |
|  |
| *Instituto Politécnico Cávado e do Ave*  *Guimarães, Braga 17/2018*  *15256 - Diogo Araújo*  *15249 - Hugo Moura*  *13069 - Luís Amaro*  *15258 - Nuno Macedo*  *15255 - Pedro Macedo* |

Análise e Projeto de Sistemas

ROYAL PETS

Índice

[Instigadores do Projeto 3](#_Toc516559236)

[Propósito do Sistema 3](#_Toc516559237)

[Contextualização 3](#_Toc516559238)

[Visão do Produto 3](#_Toc516559239)

[Cliente, Comprador e stakeholders 4](#_Toc516559240)

[Cliente 4](#_Toc516559241)

[Compradores 4](#_Toc516559242)

[Stakeholders 4](#_Toc516559243)

[Utilizadores do Sistema 4](#_Toc516559244)

[Restrições do projeto 5](#_Toc516559245)

[Restrições Obrigatórias 5](#_Toc516559246)

[Factos e assunções relevantes 6](#_Toc516559247)

[Diagrama de contexto 7](#_Toc516559248)

[Processos de negócio 7](#_Toc516559249)

[Requisitos funcionais 8](#_Toc516559250)

[Tecnicas de levantamento de requisitos 8](#_Toc516559251)

[Requisitos funcionais de Utilizador 9](#_Toc516559252)

[Priorização de requisitos 11](#_Toc516559253)

[Técnica Utilizada 11](#_Toc516559254)

[Matriz de dependências 12](#_Toc516559255)

[Requisitos não funcionais 13](#_Toc516559256)

[Diagrama de casos de uso 14](#_Toc516559257)

[Diagrama de Classes 15](#_Toc516559258)

[Mockups 16](#_Toc516559259)

## Índice de Figuras

[Figura 1 - Diagrama de Contexto 7](file:///D:\\1%20Curso%20de%20redes\\2semestre\\analise%20e%20projeto%20de%20sistemas\\Royal%20Pets\\engrega%202\\APS.docx" \l "_Toc516559631)

[Figura 2 - Matriz de Dependencias 12](#_Toc516559632)

[Figura 3 - Diagrama Casos de Uso 14](#_Toc516559633)

[Figura 4 - Diagrama de classes 15](#_Toc516559634)

[Figura 5 - Mockup 1 16](#_Toc516559635)

[Figura 6 - Mockup 2 16](#_Toc516559636)

[Figura 7 - Mockup 3 17](#_Toc516559637)

[Figura 8 - Mockup 4 17](#_Toc516559638)

[Figura 9 - Mockup 5 18](#_Toc516559639)

[Figura 10 - Mockup 6 18](#_Toc516559640)

[Figura 11 - Mockup 7 19](#_Toc516559641)

# Instigadores do Projeto

## Propósito do Sistema

### Contextualização

Fundada em 2018, esta impressa tem como objetivo ajudar o dono com o seu animal.

Os seres humanos são diferentes assim como os animais, uns são gordos, outros altos e todos eles requerem uma rotina diferente, é nosso dever ajudá-lo na rotina diária do seu animal assim como estaremos ligados ao seu veterinário para que o seu animal tenha tudo em dia!

Se você estiver indeciso, nós estaremos dispostos em ajudar a encontrar o animal perfeito para fazer parte da sua vida, desde cães, gatos até a uma simples e pequena tartaruga.

### Visão do Produto

#### Descrição do Problema/Negócio

ROYAL PETS tem como objetivo ajudar o dono no dia-a-dia, não só terá capacidade de gerir o bem-estar do seu animal, assim como relembrar-lhe de todas as consultas, medicações e cirurgias. Esta terá uma panóplia de recursos ao seu dispor, como por exemplo, um orientador dedicado a si, tendo como principal objetivo ajudá-lo a fazer as melhores escolhas na alimentação, bem-estar e no seu animal.

#### Objetivos a Alcançar com o Produto

* Este projeto tem como objetivo a ajuda com os seus variados animais de estimação, ou seja, ajudar-lhe em vários aspetos ao decorrer do dia como por exemplo:
* Refeições diárias do animal;
* Alertas para consultas veterinárias;
* Aconselhamento de animal “perfeito” para determinado utilizador;
* Orientação para futuros regras animais;
* Consulta da documentação do seu animal de estimação;
* Localização do seu animal em tempo real.

## Cliente, Comprador e stakeholders

### Cliente

Destinado a todos os utilizadores que não pretendem pagar pela aplicação.

### Compradores

Destinado a todos os utilizadores que pretendem o obter todos os serviços disponíveis pela aplicação, sendo estes pagos.

### Stakeholders

A ROYAL PETS depara-se com três *stakeholders*, dos quais dois positivos e um negativo.

Um primeiro *stakeholder* são os clientes, pois existem dificuldades em controlar a saúde do seu animal de estimação, problema esse que será resolvido com a nossa aplicação.

O segundo *stakeholder* são os veterinários, pois a aplicação facilita o seu trabalho na verificação de vacinas e regulamentações apropriadas para o seu animal, notificando o utilizador da data. Guardando também um histórico de vacinas tomadas, consultas de rotina, situações de urgência, entre outros.

Deparamo-nos com o terceiro *stakeholder*, negativo, uma empresa concorrente que desenvolve uma aplicação idêntica, fazendo com que seja possível a perda de clientes, caso a aplicação da concorrência seja totalmente grátis.

## Utilizadores do Sistema

|  |  |
| --- | --- |
| Profissionais | Gratuitos |
| Clientes profissionais que terão acesso a informação mais aprofundada e acesso a informação a animais de criação (Ex: vacas, cabras, etc.). | Os clientes comuns os quais terão acesso a informações e dicas para o cuidado a ter com animais de estimação mais comuns (Ex: gatos, cães, etc.). |

Este sistema será dividido em dois tipos de utilizadores, os utilizadores *Premium* (Profissionais), que por sua vez terão um pagamento mensal para usufruir de todas as regalias disponíveis. E utilizadores *Free* (Gratuitos), onde estes terão um limite máximo de dois registos por utilizador sendo que cada animal terá um e um só *chip* dedicado a ele mesmo.

# Restrições do projeto

## Restrições Obrigatórias

O projeto vai ser baseado em ambiente web, uma vez que estes sistemas são conhecidos por serem dinâmicos e online, as informações podem ser acedidas em qualquer lugar e a qualquer momento. Web tem como vantagem a sua fácil manutenção e incidência de problemas técnicos. Royal Pets será criada numa plataforma denominada por BootStrap, uma framework web com código-fonte aberto, para o desenvolvimento de componentes de interface e front-end para sites e aplicações web usando HTML, CSS3 e JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo, possibilitando o visionamento no computador ou no smartphone.

Neste projeto será utilizada a linguagem PHP, Personal Home Page, uma multi-plataforma com fácil conectividade a sistemas Sybase, Mysql, Mssql e Oracle, compatíveis com o padrão ODBC, um padrão para acesso a sistemas gerenciadores de bancos de dados, usando drivers para operarem como uma camada de tradução entre a aplicação e o SGBD (sistemas gerenciadores de bancos de dados). PHP é rápido e robusto contendo um código aberto possibilitando a sua manutenção nas tarefas direcionadas.

Será usada também a linguagem HTML, uma linguagem combatível com todos os dispositivos móveis, na qual dispensa plug-ins, podendo aceder sem problemas ao site, mesmo com o flash desinstalado no PC. HTML possibilita criar plataformas acessíveis a pessoas com problemas de visão e alto contraste.

Css3, Cascading Style Sheets, tem com principal função abolir imagens de plano de fundo, bordas arredondadas, transições e efeitos para criar animações, dando assim, um aspeto atrativo e apresentável ao cliente.

Uma das principais linguagens para o projeto é o SQL, desenvolvido originalmente no inicio dos anos 70 nos laboratórios da IBM. Esta linguagem é um grande padrão de banco de dados. Isto advém da sua simplicidade e facilidade de uso, diferenciando-se assim das outas linguagens de consulta a base de dados no sentido em que uma consulta SQL especifica a forma do resultado e não do caminho para chegar a ele. Sendo esta uma linguagem declarativa em oposição a outras linguagens procedurais.

A linguagem server, fornece a lógica principal da aplicação, funcionando “por detrás dos panos”. Sempre que um usuário faz um HTTP *request*, o pedido é enviado para o servidor. A linguagem server-side, recebe o requerimento e faz o processamento. Depois, transforma o resultado em um XHTML e por fim envia ao navegador. Esta linguagem não só processa o requerimento como também verifica se o usuário está logado, buscando as informações ao banco de dados.

Atualmente a principal linguagem cliente-side em navegadores web, JavaScript, possui suporte á programação funcional e apresenta recursos como fechamentos e funções de alta ordem comumente indisponíveis em linguagens como Java e C++. Tendo como função interagir com o usuário, controlar o navegador, realizar comunicações assíncronas e alternar o conteúdo dos documentos exibidos.

Em síntese todos estas linguagens vão ser automatizadas por uma base de dados com os argumentos necessários, para uma usabilidade fácil, apelativa e intuitiva.

## Factos e assunções relevantes

Royal pets estará beneficiando de uma base de dados, promissora de fornecimento de raças de animais, estando estes em constante atualização.

A aplicação não só disponibilizará inquéritos desenvolvidos especialmente para futuros veterinários, de forma a satisfazer totalmente o utilizador da aplicação tornado esta mais fiável e flexível, como disponibilizará questionários ao utilizador da aplicação. Todas as opiniões são importantes para um melhor desempenho do sistema.

Serão recolhidas informações pertinentes de cada animal alocado no sistema, de forma a atuar em conformidade com o animal, estando este em constante atualização.

Na versão *premium, aconselharemos o espaço indicado para cada animal de forma a obter a melhor produção/conforto para o animal.*

## Diagrama de contexto

Figura 1 - Diagrama de Contexto

## Processos de negócio

Este projeto promove o bom futuro para o cuido animal, com intuito de regularizar as atividades dos nos Animais de estimação;

O resultado da negociação pode ter um impacto significativo relativamente ao projeto de grande presença com o mesmo;

Os processos que vão ser feitos no projeto, iram ser divididos por fases e delimitados com um período de tempo para melhor regularização final do projeto;

O projeto tem como principal função promover um novo e grande futuro para nós portadores dos nossos animais assim como para todos os veterinários, tendo uma maior facilitação em termos de processos animais.

# Requisitos funcionais

## Tecnicas de levantamento de requisitos

Neste capítulo estão descritas as técnicas utilizadas no processo de levantamento de requisitos.

Usamos três técnicas de levantamento os inquéritos, brainstorming e as entrevistas.

Os inquéritos são focados em utilizadores comuns, desta forma, a usabilidade da aplicação ficará mis intuitiva e apelativa a todos os clientes, já as entrevistas, serão direcionadas a todos os médicos e enfermeiros veterinários de forma a avaliar e administrar os cuidados com o animal.

O processo de Brainstorming consistiu na divergência das opiniões. Cada elemento da equipa escreveu num documento o que tinha em mente sobre as funcionalidades que o sistema deveria disponibilizar. As ideias como era esperado foram divergentes e muitas delas irrealistas, mas assim foi possível obter uma grande diversidade de ideias.

Despois da divergência das opiniões, começou-se a excluir as ideias menos boas e modificar algumas para que fossem passíveis de ser implementadas. Em conclusão, finalizamos o processo com um conjunto de ideias possíveis de implementar e que todos os elementos da equipa concordam.

## Requisitos funcionais de Utilizador

Um requisito funcional descreve uma funcionalidade a ser disponibilizada aos utilizadores do sistema. Neste são especificado os requisitos funcionais do utilizador e do sistema. Os requisitos especificados, correspondem a uma primeira versão do sistema a desenvolver podendo ser alteradas ao longo do tempo.

Em seguida encontram-se enumeradas os requisitos de utilizador, que descrevem as expectativas e as necessidades dos utilizadores.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Descrição | Fonte | Data |
| RF1 | O Administrador de sistema pode criar perfis. | Brainstorming | 2018-04-04 |
| RF2 | O utilizador pode aceder a ficha médica do seu animal. | Inquérito cliente | 2018-04-09 |
| RF3 | O utilizador tem acesso aos dados do veterinário, ao qual o seu animal está associado. | Brainstorming | 2018-04-09 |
| RF4 | O utilizador deve estar autenticado no sistema. | Brainstorming | 2018-04-09 |
| RF5 | O utilizador, de acordo com a sua subscrição, deve registar os animais que pretender. | Brainstorming | 2018-05-10 |
| RF6 | O utilizador poderá alterar os dados do seu animal. | Brainstorming | 2018-05-10 |
| RF7 | O utilizador pode alterar os seus dados de perfil. | Brainstorming | 2018-05-10 |
| RF8 | O utilizador pode alterar a sua subscrição a qualquer altura. | Brainstorming | 2018-05-10 |
| RF9 | O utilizador pode remover animais a seu cuidado. | Brainstorming | 2018-05-19 |
| RF10 | O utilizador pode remover o seu perfil. | Brainstorming | 2018-05-19 |

Requisitos funcionais de Sistema

Os requisitos de sistemas estabelecem concordâncias entre o cliente e a equipa de desenvolvimento. Estes são enumerados na seguinte tabela.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Descrição | Fonte | Data |
| RF1 | O sistema deve permitir que o utilizador escolha o tipo de conta. | Inquérito cliente | 2018-05-20 |
| RF2 | O sistema deve disponibilizar uma ficha de informações pessoais para recolha de dados para o perfil. | Inquérito cliente | 2018-05-20 |
| RF3 | O sistema deve disponibilizar o preenchimento de um questionário, no fim de um registo, para recolha de informação sobre o animal. | Brainstorming | 2018-05-20 |
| RF4 | O sistema deve limitar o número de animais consoante o tipo de subscrição escolhida. | Brainstorming | 2018-05-21 |
| RF5 | O sistema contém uma base de dados para apoio a produtores. | Brainstorming | 2018-05-21 |
| RF6 | O sistema contém uma base de dados onde serão guardados os dados de cada animal. | Brainstorming | 2018-05-22 |
| RF7 | O sistema deve permitir o registo de produtores, separadamente dos utilizadores e veterinários. | Brainstorming | 2018-05-22 |
| RF8 | O sistema deve permitir o registo de veterinários, separadamente dos utilizadores e de produtores. | Brainstorming | 2018-05-22 |
| RF9 | O sistema emite alertas antecipados para consultas veterinárias, a animais que tenham um veterinário associado. | Inquérito cliente | 2018-05-22 |
| RF10 | O sistema disponibiliza funções de acordo a subscrição do utilizador. | Brainstorming | 2018-05-22 |

## Priorização de requisitos

Efetuamos priorização de requisitos e explicamos a técnica usada no seguinte processo.

### Técnica Utilizada

Para efetuamos esta operação, classificamos os mesmo com pontuações de 0 a 10 em duas categorias, importância e tempo de implementação. Um requisito classificado com elevada pontuação, é de alta importância, significa que é indispensável para a aplicação, tendo que aparecer obrigatoriamente no produto final. No entanto, um requisito com pontuação alta, exige mais dedicação e esforço a esse requisito. Após a classificação atribuída, será usada uma fórmula para obter a pontuação final do requisito:

De seguida, os requisitos são ordenados de acordo com a sua classificação. Porém, um requisito com alta pontuação será um requisito prioritário.

|  |  |
| --- | --- |
| # | Pontuação |
| RF1 | 9.4 |
| RF5 | 8.5 |
| RF7 | 7.1 |
| Rf4 | 6.7 |
| RF8 | 6.2 |
| RF6 | 6.2 |
| RF9 | 6.0 |
| RF2 | 4.6 |
| RF10 | 4.1 |
| RF3 | 3.6 |

### Matriz de dependências

De notar que, devido a alguns destes serem de difícil implementação e aparecerem com prioridade mais reduzida, alguns requisitos de maior prioridade são dependentes de outros requisitos de menor prioridade. Para resolver estes conflitos, foi construída uma matriz de dependências de modo a ordenar os requisitos consoante as suas dependências. Na Figura é apresentada a matriz desenvolvida, onde X indica o requisito que é dependente dos requisitos nas colunas.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  | X |  |  |  |  |  |  |  |
| 3 |  | X |  |  |  |  |  |  |  |  |
| 4 | X |  |  |  |  |  |  |  |  | X |
| 5 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  | X |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  | X |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  | X |
| 10 | X |  |  |  |  |  |  |  |  |  |

Figura 2 - Matriz de Dependencias

# Requisitos não funcionais

Um requisito não funcional é visto como um conjunto de restrições impostas ao sistema, desta forma é possível saber qual o mais útil, rápido, confiável, entre outros.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Tipo | Descrição | Fonte | Data |
|  | Usabilidade | O sistema deve conter uma interface intuitiva, para ser usada por utilizadores com poucos conhecimentos informáticos. | Inquérito cliente | 2018-04-09 |
|  | Operacional | O sistema está ambientado para dispositivos moveis e web. | Brainstorming | 2018-04-09 |
|  | Usabilidade | O administrador atribui perfis aos utilizadores consoante pedido. | Brainstorming | 2018-04-09 |
|  | Desempenho | O sistema deve funcionar em dispositivos móveis com baixa capacidade de desempenho. | Brainstorming | 2018-04-10 |
|  | Aparecia | Na pagina de utilizador, veterinário ou produtor, deve estar explicita a informação relativa ao próprio, bem como uma foto de perfil. | Inquérito cliente | 2018-04-10 |
|  | Segurança | Todos os utilizadores, veterinários e produtores tem as suas credenciais para entrar no sistema. | Brainstorming | 2018-04-10 |
|  | Legal | Todos os animais presentes na base de dados devem estar associados ao seu dono, sendo comprovado por documentação legal. | Inquérito cliente | 2018-04-10 |
|  | Aparência | O sistema deve apresentar pelo menos uma foto do animal pretendido. | Brainstorming | 2018-04-11 |
|  | Legal | Apenas utilizadores com idade igual ou superior a 18 poderão utilizar o sistema. | Brainstorming | 2018-04-11 |
|  | Usabilidade | O sistema deve estar bem estruturado e simples compreensão quando disponível em dispositivos móveis. | Inquérito cliente | 2018-04-11 |

# Diagrama de casos de uso

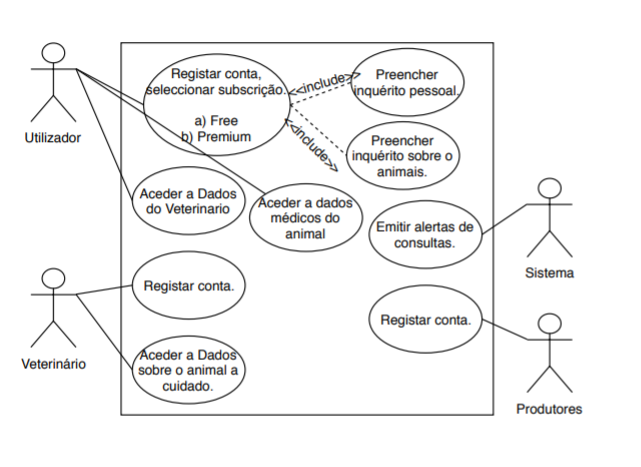


Figura 3 - Diagrama Casos de Uso

Este diagrama serve para identificar a fronteira entre o sistema e os atores e especifica as funcionalidades que o sistema disponibiliza ao utilizador.

# Diagrama de Classes

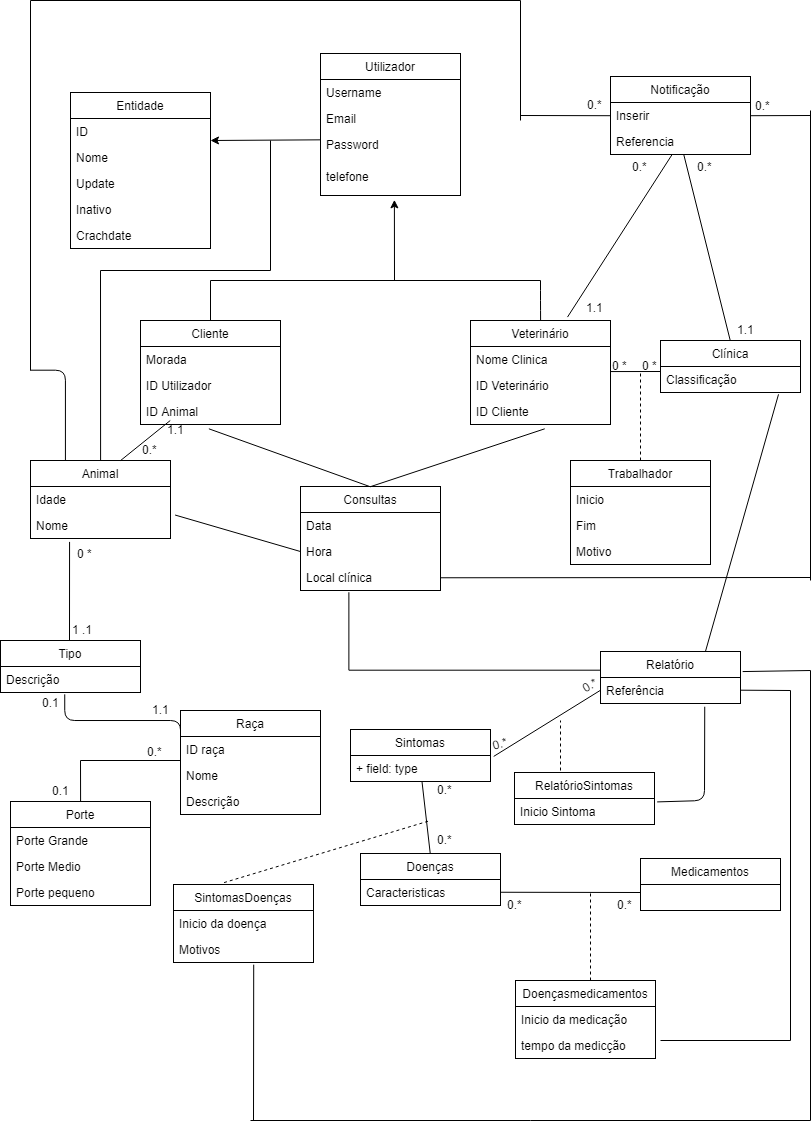


Figura 4 - Diagrama de classes

# Mockups

Neste sector é mostrado uma versão final ao usuário do software criado.

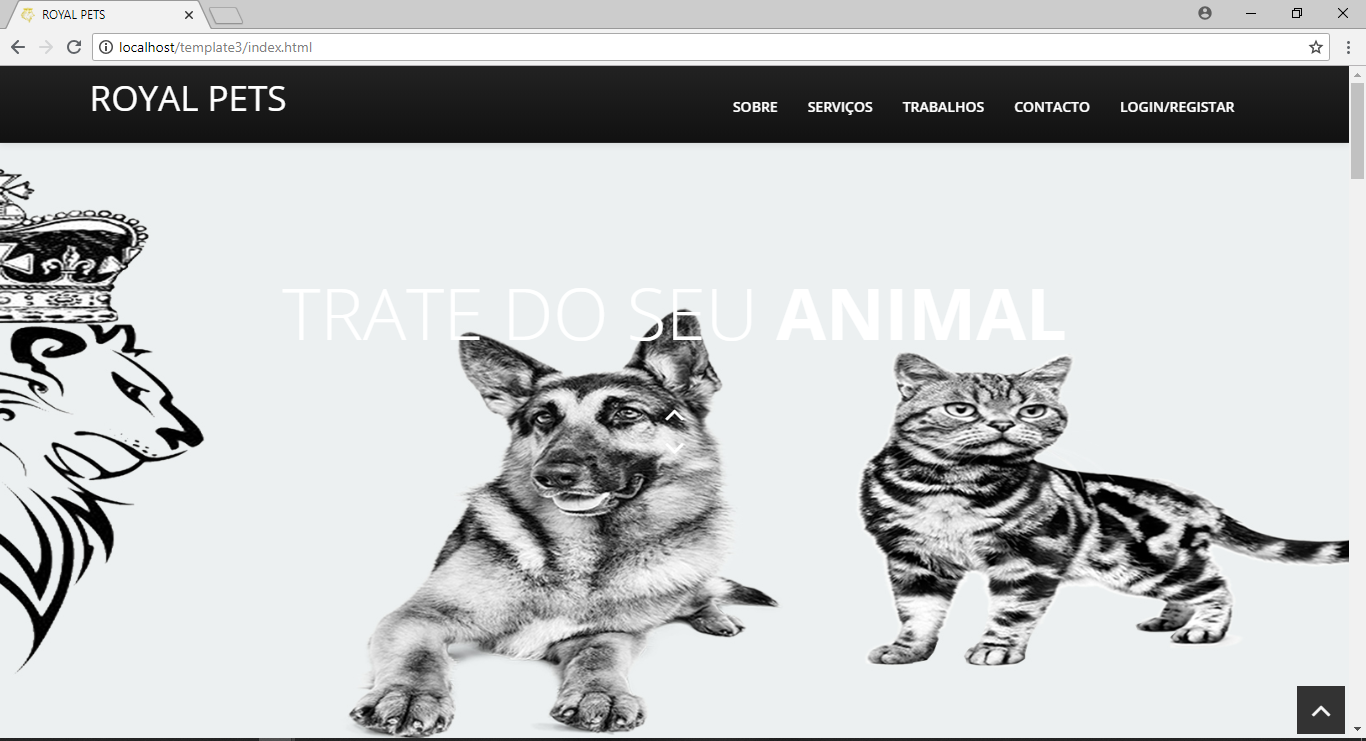


Figura 5 - Mockup 1



Figura 6 - Mockup 2

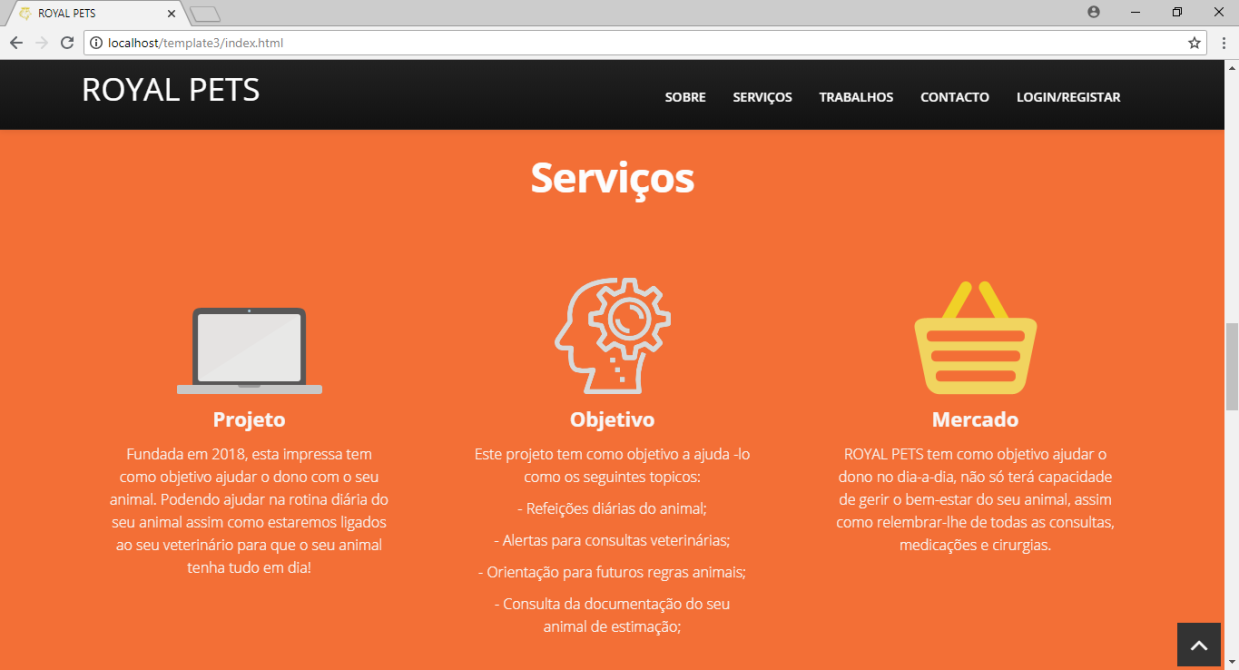


Figura 7 - Mockup 3

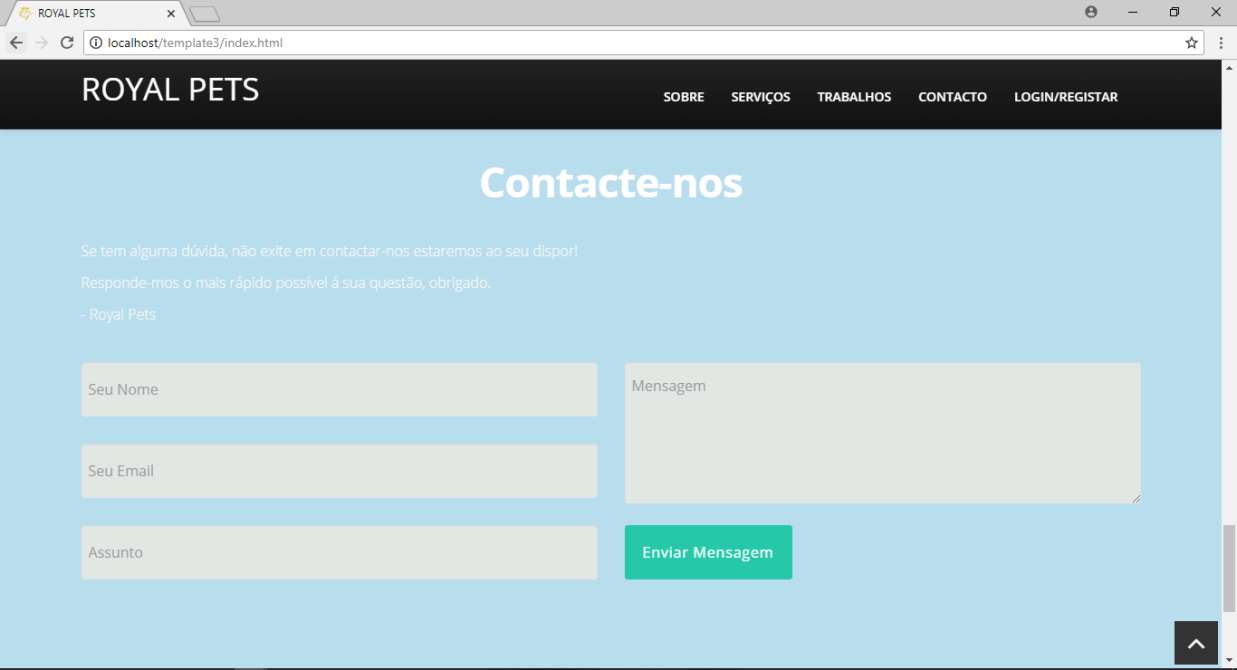


Figura 8 - Mockup 4

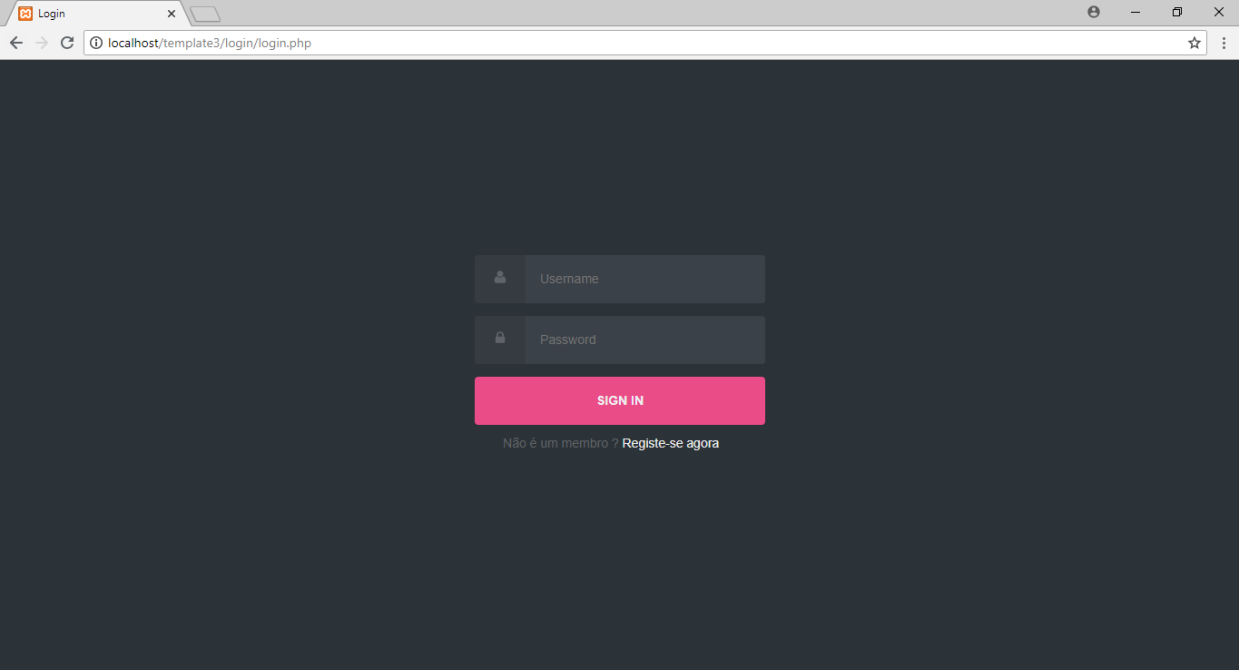


Figura 9 - Mockup 5

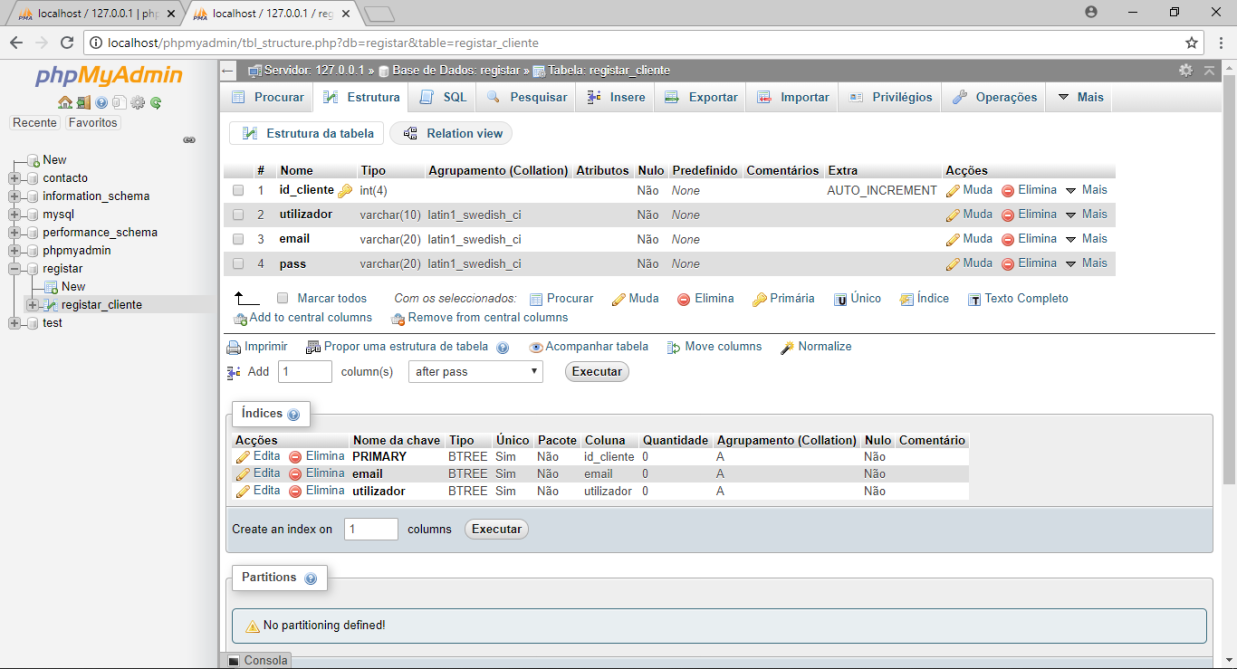


Figura 10 - Mockup 6

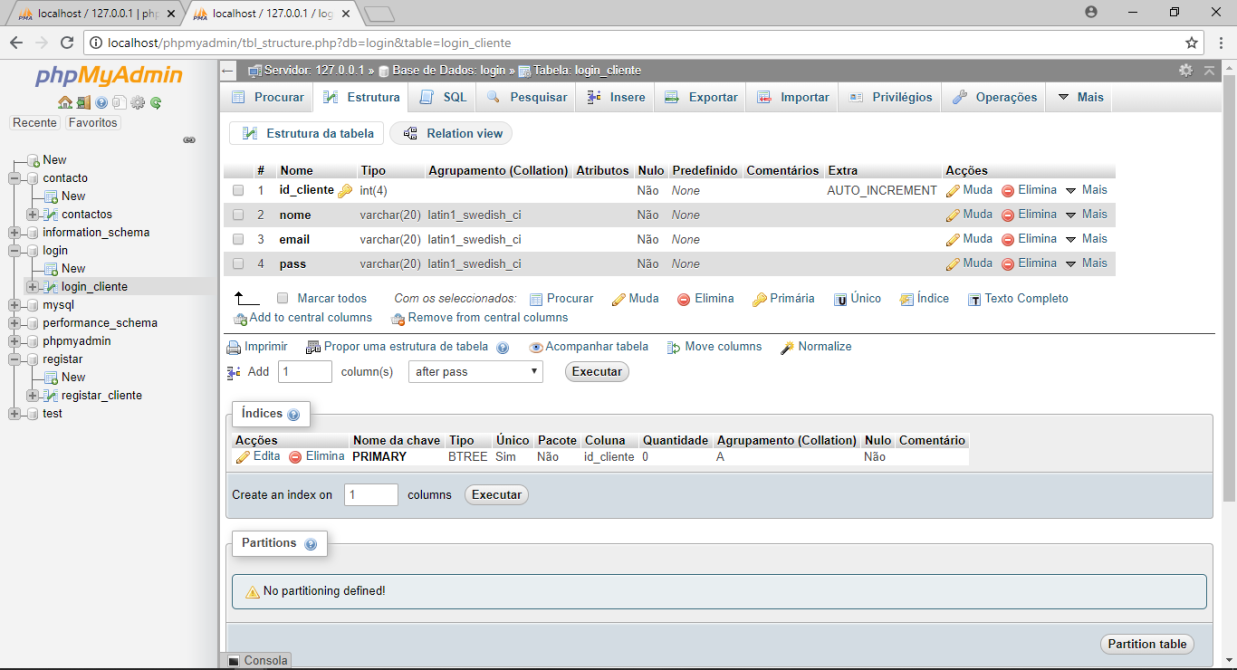


Figura 11 - Mockup 7