



App Coudelaria Militar

Projeto Final de Licenciatura

Elaborado por Miguel Bento
Discente nº 20151985

Orientador: Prof. Filipa Taborda

Barcarena
Novembro 2018

Lista de Abreviaturas

APP: Aplicação Móvel.

WEB: World Wide Web.

HTML: HyperText Markup Language.

CSS: Cascading Style Sheets

SQL: Structured Query Language

UML: Unified Modeling Language

UELN: Unique Equine Life Number

ID: Identificador

ER: Entidade Relação

PHP: Hypertext Preprocessor

Resumo

O acesso á informação hoje em dia é dado como garantido, no entanto, existem barreiras que podem colocar entraves á sua rápida e fácil disponibilização.

Este trabalho dá resposta ao desafio dos veterinários do Exército que pretendem ter acesso aos registos clínicos dos seus solípedes quando estão no terreno, da forma mais prática e funcional possível.

Existindo já uma aplicação Web desenvolvida que suporta toda a base de dados informativa dos solípedes, a App proposta focou-se assim na disponibilização da situação clínica em qualquer dispositivo móvel, usando técnicas responsivas, bem como o estudo de melhores interfaces gráficos sempre tendo em conta a usabilidade.

Desenvolveu-se um protótipo que estabelece a comunicação com uma Base de dados gémea da que se encontra em produção acedendo apenas aos dados necessários ao tratamento das situações clínicas associadas à observação de um solípede. Cria também uma experiência simples e funcional ao utilizador que não estará em condições ótimas de utilização de um dispositivo mesmo que móvel.

Palavras-chave: multiplataforma, mobile app, usabilidade, disponibilidade, adaptação.

Índice

1. INTRODUÇÃO	1
1.1O Problema	1
1.2Objetivos	2
2. ESTADO DA ARTE.....	2
2.1 Interfaces Gráficos	3
2.2 Ferramentas Tecnológicas	6
2.3 Arquitetura	7
3. ANÁLISE E DESENHO.....	7
3.1 Definição de Requisitos.....	8
3.1.1 Requisitos Funcionais.....	8
3.1.2 Requisitos Não Funcionais	8
3.2 Casos de Uso	8
3.3 Entidades.....	11
3.3.1 Esquema E/R	11
3.4 Base de Dados	11
3.4.1 Modelo Relacional.....	12
4. Desenvolvimento do Protótipo	13
4.1. Processos.....	13
4.2 <i>Mockups</i>	14
4.2.1 Login	15
4.2.2 Agenda	15
4.2.3 Gestão de Solípedes.....	16
4.2.4 Ficha Clínica.....	16
4.3 Implementação	17
4.3.1 1ª Camada: HTML.....	17
4.3.2 2ª Camada: PHP	20
4.3.3 3ª Camada: Base de Dados	20
5.Conclusão	21
Bibliografia	23
Anexo 1. Demo App Coudelaria Militar	24
Login & Registo.....	24
Perfil	24
Agenda	25
Token.....	26
Solípedes	26

Índice de Figuras

Figura 1 Exemplos de Inputs HTML.....	3
Figura 2 Exemplo de barras de navegação.....	4
Figura 3 Menu construído com CSS.	4
Figura 4 Exemplo de Menu de Paneis	5
Figura 5 Exemplo de um Menu Lateral	5
Figura 6 Caso de Uso: Registo de um Utilizador	9
Figura 7 Caso de Uso: Acesso ao Perfil de Utilizador	9
Figura 8 Caso de Uso: Agenda.....	10
Figura 9 Casos de Uso: Solípedes	10
Figura 10 Modelo Relacional.....	12
Figura 11 Alterar a Situação Clínica.....	13
Figura 12 Aprovar um Diagnóstico	14
Figura 13 Mockup do Ecrã Principal.....	14
Figura 14 Mockup do Ecrã de Login	15
Figura 15 Mockup do Ecrã de Agenda.....	15
Figura 16 Mockup do Ecrã de Solípedes	16
Figura 17. Mockup do Ecrã de Diagnóstico	17
Figura 18 Protótipo: Ecrã Principal.....	18
Figura 19 Exemplo de um Pop-Up.....	19
Figura 20 Protótipo: Login.....	24
Figura 21 Protótipo: Agenda	25
Figura 22 Protótipo: Adicionar Evento	25
Figura 23 Protótipo: Gestão de Tokens.....	26
Figura 24 Protótipo: Novo Token Gerado	26
Figura 25 Protótipo: Estado Geral	26
Figura 26 Protótipo: Alterar Estado	27

1. INTRODUÇÃO

A informação digital oferece bastantes benefícios para o dia-a-dia da nossa sociedade, no entanto o acesso à mesma pode ser feito de várias formas. Muitas vezes as aplicações desenhadas especificamente para uma necessidade, são feitas de forma a garantir que o utilizador tenha acesso rápido e fácil á informação a que está a tentar aceder.

A necessidade de agilizar processos importantes é cada vez mais necessária quando temos pouco tempo disponível e um grande volume de ações a realizar no dia-a-dia, se queremos apenas executar uma ação muito especifica não é necessário passar por menus e submenus para o fazermos.

Existindo já um sistema em que o utilizador tem acesso á informação de um modo uniforme, o desenho de uma segunda aplicação teria como finalidade o acesso a um conjunto de dados muito específicos de forma a disponibilizar o acesso aos mesmos a um conjunto de utilizadores que necessitem da informação no terreno.

Para atingir esse fim recorre-se ao desenvolvimento à media, a grande vantagem (opensoft.pt, s.d.) é a forma como reproduz os processos e fluxos da organização. O sistema é desenhado para se integrar de forma transparente com outras ferramentas de que a organização já disponha.

1.1 O Problema

Atualmente a veterinária do exército pondera a utilização de ferramentas para gestão dos dados clínicos dos solípedes. Neste contexto está em testes uma solução que lhes foi proposta num projeto integrado com toda a gestão cavalar do exército.

Os veterinários no terreno necessitam de ter acesso rápido e fácil á informação clínica dos seus solípedes. O sistema atual está montado de forma abrangente para toda a Gestão Cavalar, e disponibilizar informação muito especifica sobre as condições de tratamento de animais de grane porte, nomeadamente solípedes, passa a ser um problema que os veterinários encontram no terreno.

Impõem-se assim a necessidade de criar um interface simples e apelativo à consulta de dados pelos veterinários a partir de qualquer dispositivo móvel, disponibilizando essa mesma informação e interação para operações fundamentais como por exemplo alterar a situação clínica, “dar Baixa” a um solípede registar uma observação clínica ou agendar próxima consulta, agilizando assim a forma de como acedem á informação no terreno.

1.2 Objetivos

Objetivo geral: Criar um interface simples e apelativo à consulta/utilização de dados pelos veterinários a partir de qualquer dispositivo e em qualquer local.

Objetivos específicos:

- 1- Criação de um interface ágil e simples de utilizar.
- 2- Garantir que a aplicação assenta numa base de dados coerente com a já existente.
- 3- Controlar o acesso á aplicação.
- 4- Construir um protótipo que funcione em qualquer dispositivo móvel com acesso á internet.

Vantagens:

A criação de um protótipo visa demonstrar ao utilizador a facilidade do acesso á informação no terreno. Utilizando este exemplo pode-se adaptar facilmente a aplicação para outro âmbito ou objetivos mais específicos.

2. ESTADO DA ARTE

Para a elaboração do protótipo, dada a diversidade de dispositivos, cedo se percebeu que desenvolver em específico para uma plataforma móvel não seria a melhor solução. Em vez disso, o caminho escolhido seria o recurso à utilização de standards com Templates de design responsivo a fim da aplicação se adaptar a qualquer dispositivo móvel. Ao desenvolver-se em multiplataforma podemos usar as mesmas linguagens de programação aumentando assim a produtividade e compatibilidade, o que resulta num desenvolvimento mais rápido. (Ishwaran, 2017)

A usabilidade, considerada essencial neste projeto, é uma característica daquilo que é utilizável, funcional. É tornar a relação entre uma ferramenta e o seu utilizador o mais agradável e fácil possível. Para que uma determinada ferramenta seja eficiente, deve permitir que os seus utilizadores cumpram as suas tarefas da melhor forma possível. O mesmo princípio se aplica a computadores, websites, telemóveis e qualquer software aplicacional. Para que estes sistemas funcionem, os seus utilizadores devem conseguir utilizá-los eficazmente. (SAPO, s.d.)

2.1 Interfaces Gráficos

Antecipar as necessidades do utilizador e assegurar que todos os elementos no ecrã são fáceis de entender, aceder e visualizar, é o princípio básico da usabilidade.

A escolha de que elementos colocar no ecrã é vital para o bom funcionamento da aplicação, todos eles têm de ser combinados de forma a que o utilizador consiga completar uma tarefa da forma mais eficiente possível. Os conteúdos são organizados e apresentados de forma a tirar partido dos espaços em branco, com o objetivo de simplificar a *interface*, melhorando a legibilidade e a procura de informação na página. (Usabilidade.gov, s.d.)

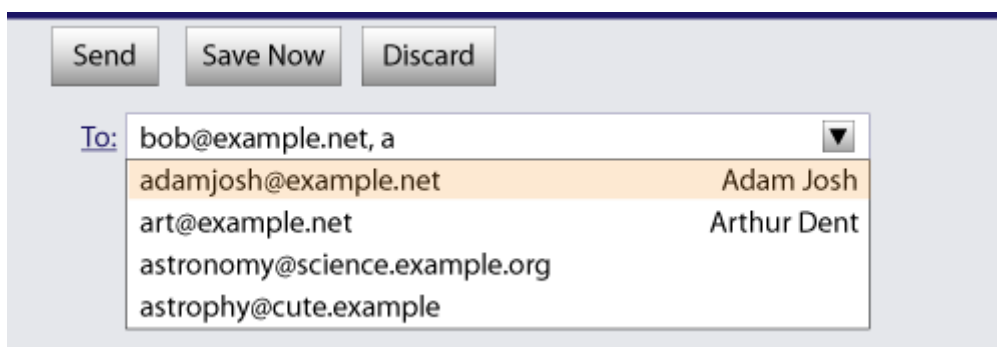


Figura 1 Exemplos de Inputs HTML

<https://www.w3.org/TR/2014/REC-html5-20141028/forms.html>

Ao desenhar a interface existem vários princípios (Waynext, 2017) que temos de ter em conta, por exemplo, manter a interface simples e mostrar apenas o necessário ao utilizador, manter a consistência entre os vários inputs de modo a que em toda a aplicação o utilizador não sinta que em cada página terá de aprender novos inputs, colocar em evidencia os elementos mais importantes para o utilizador deixando os menos significativos praticamente “invisíveis” e usar um layout que mantém uma sequencia lógica durante todo o processo.(Figura 1)

Menu de barras de navegação:

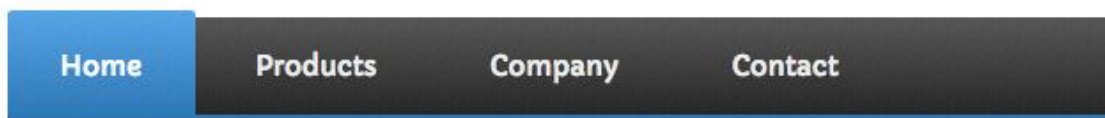


Figura 2 Exemplo de barras de navegação

<http://www.htmldrive.net/items/show/239/Horizontal-Subnav-w-CSS-jQuery>

Um estilo simples em que utilizador visualiza bem os botões, que pode conter ou não *drop-downs* mas simboliza um acesso á informação de forma rápida e livre de obstruções. (Figura 2)

Menu com *drop-downs*:



Figura 3 Menu construído com CSS.

<https://forum.webflow.com/t/styling-the-navbar-as-a-drop-down-menu/1538>

A evolução das barras de navegação com *drop-downs* a servir de submenus. Mantem a simplicidade, mas no entanto, dificulta a navegação em ecrãs com tamanho reduzido.

Cada submenu terá de ter um conjunto de seleções pré-definidas o que iria facilitar a aprendizagem do utilizador desde que a barra principal não contenha muitos botões. (Figura 3)

Menus de painéis:

Com a chegada do Windows 8 veio o novo paradigma de desenho de interfaces em painéis (Wikipedia, 2016), e como tal as páginas web também tentaram adaptar-se a essa mudança, hoje em dia existem menus bastantes simplistas e que funcionam bem numa aplicação web, exemplos:

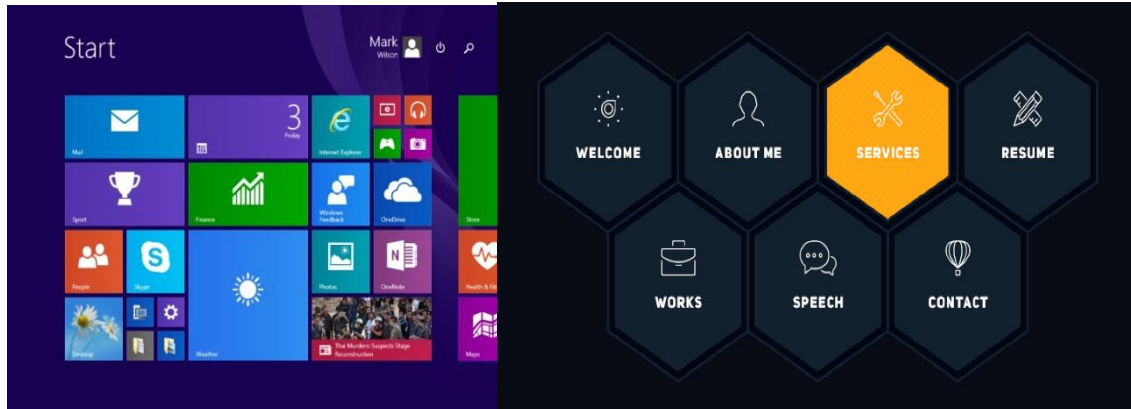


Figura 4 Exemplo de Menu de Painéis

<https://www.maketecheasier.com/customize-windows-10-start-menu/>

<https://stackoverflow.com/questions/42280954/hexagon-menu-made-in-css-and-html/>

A vantagem de usar um menu deste tipo em aplicações web seria que através da sua simplicidade o utilizador não teria qualquer problema em perceber qual a utilizada de cada botão no ecrã, no entanto após cada clique no ecrã, a navegação nunca iria ficar centralizada pois iria mover-se de um menu para o outro, outra desvantagem seria a visualização no ecrã pequeno pois diminuir o texto só iria dificultar o utilizador, deixando a única hipótese de remover botões no ecrã. (Figura 4)

Menu lateral:

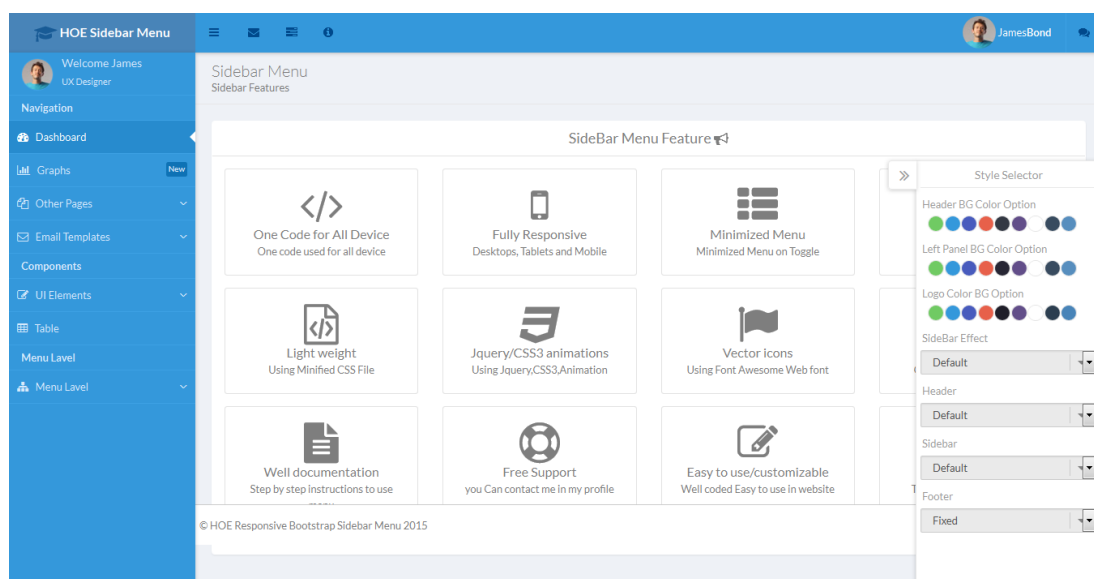


Figura 5 Exemplo de um Menu Lateral

<http://bluestreakstables.com/html-vertical-menu-bar-template/>

Uma navegação simples e responsiva em que toda a atenção encontra-se focada no centro do ecrã (Figura 5), altamente responsivo e pode combinar com elementos descritos acima. A barra lateral é usada somente para a navegação e o centro do ecrã para a visualização da informação e tomada de decisões. (Designrfix, s.d.)

2.2 Ferramentas Tecnológicas

A escolha das ferramentas tecnológicas tem como objetivo determinar que técnicas ou linguagens vamos utilizar para construir o protótipo. A escolha de desenvolvermos em multiplataforma leva a que o design responsivo seja utilizado em todo o desenvolvimento bem como a utilização de PHP para a lógica e SQL para a gestão dos dados.

Design Responsivo

Como o nome indica, é a utilização de técnicas de modo a que um *website* se adapte a qualquer dispositivo móvel (tablets, smartphones, etc.). No fundo, é a combinação de CSS e HTML para automaticamente redimensionar o tamanho do conteúdo que vemos no ecrã.

A construção assim, de toda a componente CSS seria necessária, no entanto optou-se por usar uma *framework* de modo a facilitar a implementação.

Uma *framework* CSS contém todas as ferramentas que necessitamos para a criação da página web, e hoje em dia a maior parte já se encontra virada para a responsividade. Após uma análise de algumas ferramentas, a escolha incidiu na W3.CSS, pois durante todo o percurso académico, a W3Schools foi sempre o repositório que qualquer duvidada relacionada com HTML ou CSS era esclarecida.

A *framework* W3.CSS suporta por defeito conteúdo responsivo, e foi desenhada a pensar em dispositivos móveis, é fácil de utilizar, gratuita, e extensamente documentada, o que no final se traduz em tempos mais baixos de implementação.

PHP

A linguagem escolhida para o protótipo foi PHP.

É *open source* e usada mundialmente pelas entidades mais conhecidos como o Facebook e Wordpress e normalmente é usada a nível académico o que facilita a pesquisa á resolução de problemas.

SQL

Visto que a base de dados do sistema original foi construída no SGBD Relacional MySQL e utiliza o SQL standard, infraestrutura idêntica foi utilizada para o protótipo.

2.3 Arquitetura

Para este projeto adotou-se o Modelo de Três Camadas (*Three Layer*) como arquitetura do protótipo.

Modelo de Três Camadas

Dividido em 3 camadas separadas em que cada camada apenas tem uma função, interagem com a camada seguinte e a alteração de uma não interfere com a outra. Trás benefícios como a manutenção do código ser feita em separado ou até a substituição de uma determinada tecnologia por outra ser feita sem grande impacto nas camadas adjacentes.

Camada 1 (Apresentação): Interação com o utilizador, inclui inputs e outputs, normalmente todo o conteúdo visual e estático é colocado nesta camada.

Camada 2 (Negócio): Contem toda a lógica do negócio, ou seja, todas as regras e procedimentos usadas para satisfazer as necessidades do negócio ou utilizador. São implementadas também as validações necessárias aos inputs vindos da camada de aplicação e obtém os dados vindos da camada de serviço de dados.

Camada 3 (Dados): Camada de dados ou serviço de dados, referente aos serviços que comunicam com a base de dados, normalmente todas as instruções SQL encontram-se nesta camada. (Kapahi, 2017)

3. ANÁLISE E DESENHO

A análise e desenho deste protótipo passa por primeiro percebermos quais as funções que o utilizador espere que o sistema realize, definindo-se assim requisitos através de um levantamento, depois com o auxílio dos caso de uso, consegue-se demonstrar como é que uma funcionalidade é desempenhada e por fim desenha-se um mapa onde fica demonstrado todas as entidades da aplicação e as suas relações.

3.1 Definição de Requisitos

Os requisitos estão divididos em funcionais e não funcionais.

3.1.1 Requisitos Funcionais

Um requisito que demonstra uma ação sobre o sistema ou uma funcionalidade pedida pelo utilizador.

- Controlar o acesso á aplicação.
- Permitir o registo de novos veterinários.
- Adicionar e consultar eventos.
- Adicionar novos solípedes bem como associar uma foto.
- Verificar a situação clínica, principalmente se está apto ou não.
- Adicionar e aprovar tratamentos bem como associar diagnósticos caso necessário.

3.1.2 Requisitos Não Funcionais

Requisitos em que o software a nível técnico deve suportar, ou em que linguagem deve ser desenvolvido ou que base de dados utilizar, etc.

- A aplicação deverá tentar usar um modelo de dados baseado na aplicação já existente.
- A base de dados do protótipo deverá ser desenvolvida em MySQL por razões de compatibilidade.
- Deverá ser possível usar a aplicação em qualquer plataforma que suporte um browser.

3.2 Casos de Uso

Uma análise ao problema revela que a aplicação irá ter apenas um tipo de utilizador em conta, o veterinário. A maior parte do *software* veterinário existente foca-se na gestão de um grande número de utilizadores como secretárias, rececionistas, técnicos, veterinários, etc., neste caso iria ter apenas um tipo de aplicação muito específico e apenas focar-se em dois tipos de perfis, o utilizador (veterinário) e o paciente (solípede).

Para demonstrar as principais funcionalidades da aplicação, desenharam-se os principais casos de uso em UML.

Registo de um novo utilizador

Para o veterinário se poder autenticar na aplicação o mesmo tem de estar registado, ao registar é necessário validar se o mesmo já não se encontra registado, em seguida validar se o *token* de registo é válido, caso não o seja, terá de se gerar um novo através de um outro utilizador já registado na aplicação, caso passe nas validações tem então acesso ao menu principal.

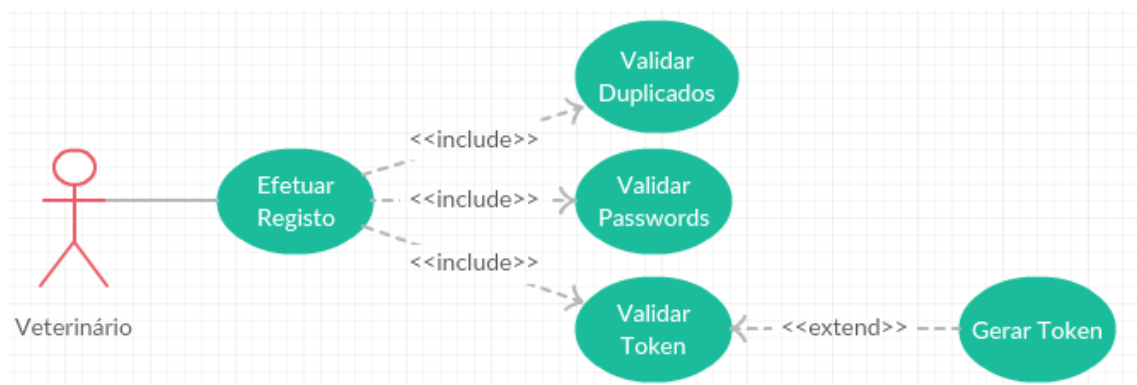


Figura 6 Caso de Uso: Registo de um Utilizador

Aceder ao perfil do utilizador

Neste caso pretendemos aceder ao perfil do veterinário, para isso o mesmo precisa de se autenticar, caso não o tenha feito é necessário o registo primeiro. Ao aceder ao perfil o mesmo pode visualizar o perfil ou editar o mesmo.

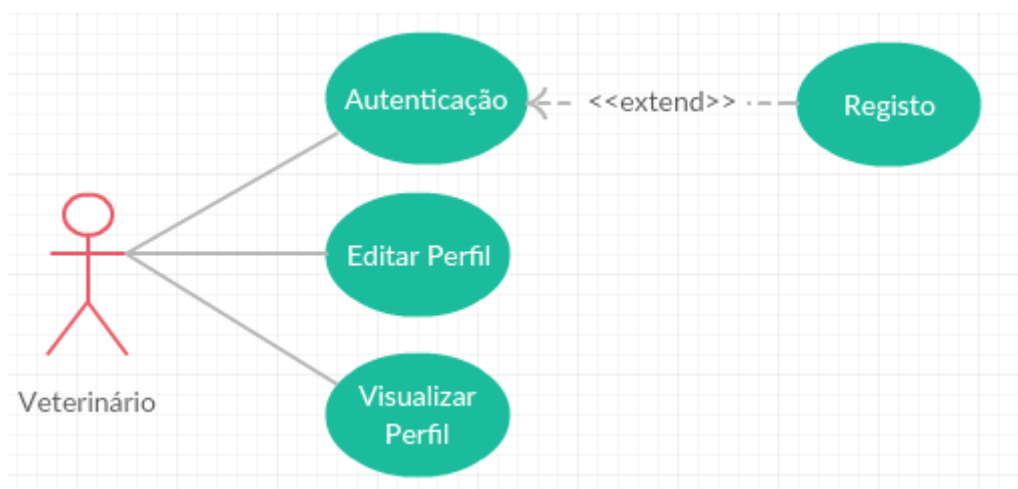


Figura 7 Caso de Uso: Acesso ao Perfil de Utilizador

Gestão de eventos através da Agenda

A agenda tem alguns casos de uso associados, o veterinário pode visualizar e apagar eventos caso esteja autenticado, além de visualizar podem também adicionar novos eventos ficando este sempre associado a um solípede previamente registado na aplicação.

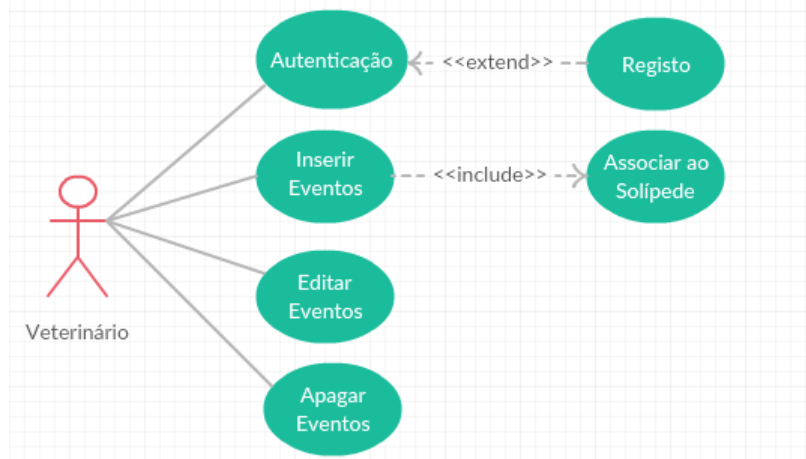


Figura 8 Caso de Uso: Agenda

Gestão de Solípedes

A gestão de solípedes por parte de um veterinário registado é feita através de o registo de um novo solípede, após o registo é aconselhável associar uma foto do mesmo, apesar de ser facultativo. Cada solípede tem uma situação clínica associada onde se pode adicionar e visualizar diagnósticos feitos pelos veterinários, cada diagnóstico pode ou não tem um ou vários tratamentos associados.

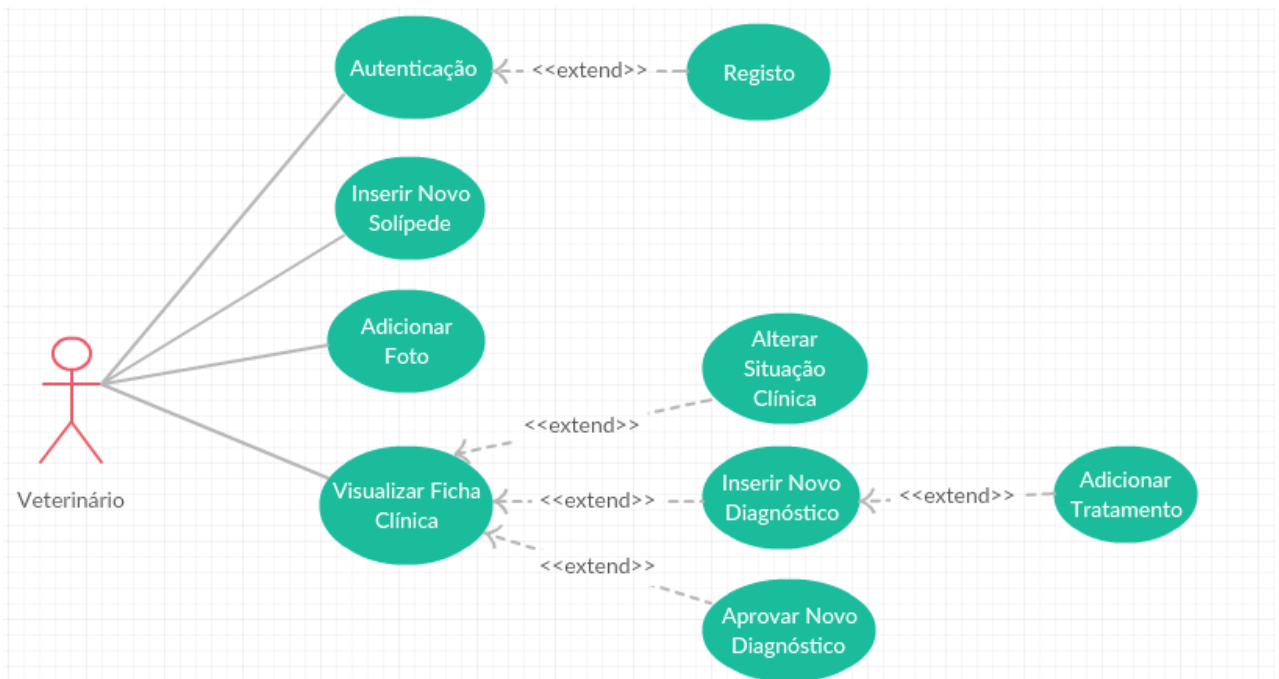


Figura 9 Casos de Uso: Solípedes

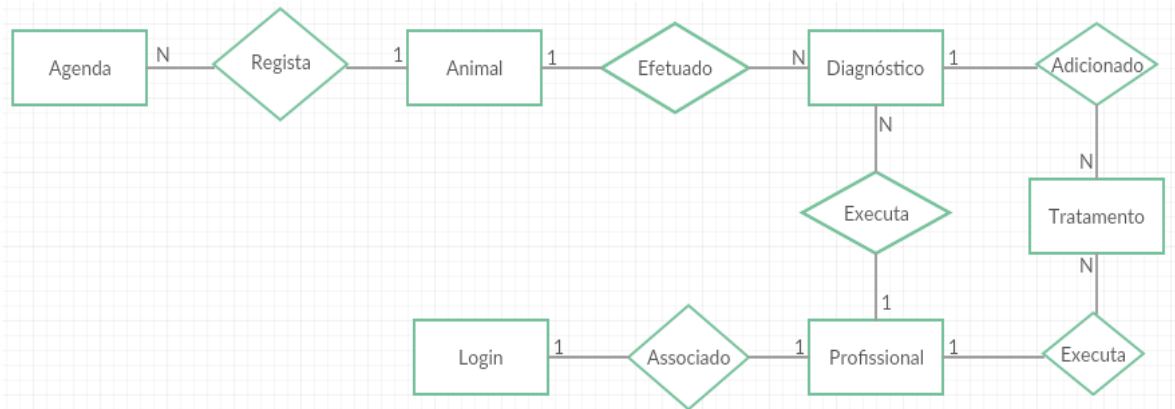
3.3 Entidades

É necessário saber quais as entidades que têm de ser representadas para suporte de dados na elaboração do protótipo. Além da representação dessas entidades também necessitamos de ter em conta a relação entre elas ou como interagem umas com as outras, neste caso as seguintes entidades foram identificadas como necessárias para o protótipo:

- Veterinário: o utilizador da aplicação.
- Agenda: local onde se regista eventos.
- Solípedes: o objeto que é gerido pelos utilizadores.
- Tratamentos: não existem sem associação a diagnósticos e são executados por um veterinário.
- Diagnóstico: associado a um solípede, podem ou não existir.

3.3.1 Esquema E/R

As entidades acima traduzem-se no seguinte esquema Entidade Relação:



3.4 Base de Dados

Tendo em conta que apenas uma parte da base de dados do sistema original irá ser usado, o desenho iria passar por uma análise das necessidades do utilizador e traduzir as mesmas para uma réplica em menor escala da base de dados do sistema original. O objetivo é criar um modelo simples, mas fiel ao original de forma a manter a coerência dos dados entre as duas bases de dados.

Foi utilizado o Esquema Entidade Relação para perceber como as várias entidades se relacionavam e também para visualizar a relação entre os vários atributos que as compõem.

3.4.1 Modelo Relacional

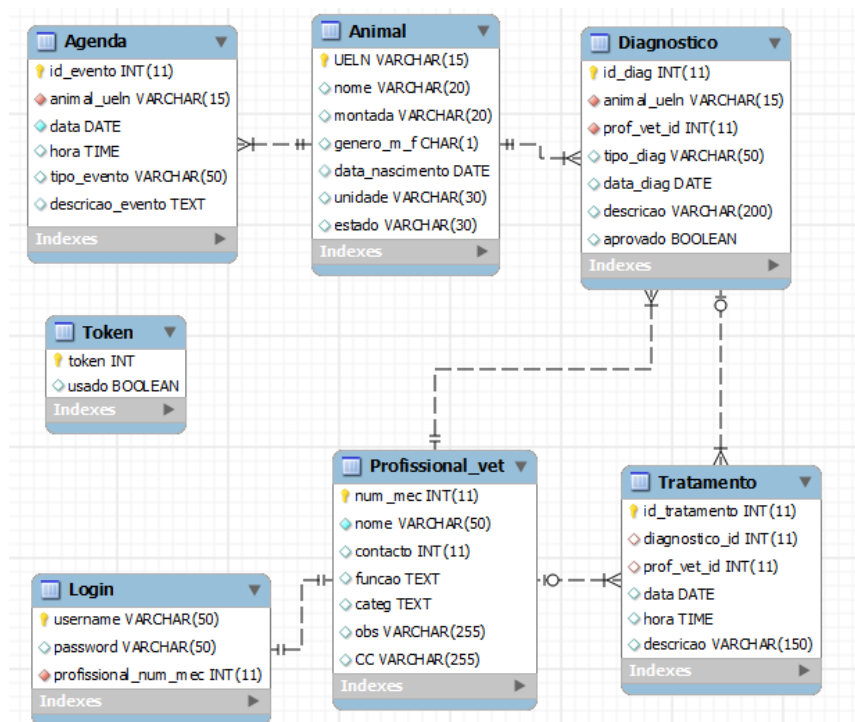


Figura 10 Modelo Relacional

O **Token** é uma cadeia de inteiros que é fornecido a um novo utilizador por alguém que já se encontra registado na aplicação. Tem o propósito de controlar o acesso á mesma e depois de usado o mesmo é dado como inutilizado colocando o booleano a *true*.

O **Login** apenas contem o *username* e *password* do utilizador, e tem de estar associado a um profissional usando assim como chave estrangeira o número mecanográfico.

Um **Diagnostico** tem de estar sempre associada a um animal (solípede e o número mecanográfico do veterinário que o efetuou).

Por último o **Tratamento** segue o mesmo princípio do diagnóstico onde só existe se for associada a um diagnóstico e a um profissional que efetuou esse tratamento.

É de notar que a figura acima demonstra o modelo relacional que evoluiu da associação de entidades e o seu esquema E/R apresentado no capítulo anterior, utilizando a aplicação MySQLWorkbench a partir do qual foi feito o transporte e consequente definição para a base de dados.

4. Desenvolvimento do Protótipo

Neste capítulo a partir dos requisitos que nos foram dados constrói-se um protótipo que respeite funcionalmente e não funcionalmente a expectativa dos utilizadores.

Numa primeira fase é necessário desenhar um esboço de como a aplicação em si irá funcionar, para isso recorreremos ao desenho de *Mockups* que nos possibilita ter uma visão de como os menus e algumas funcionalidades irão interagir com utilizador, posteriormente o desenho da base de dados onde será guardada a informação irá ser uma parte importante do protótipo pois terá de se assemelhar ao que já existe atualmente, por fim passamos para o desenvolvimento do protótipo em si usando as diferente tecnologias mencionadas no capítulo 2.

4.1. Processos

As figuras seguintes demonstram como exemplo dois casos da utilização da aplicação e a sequência de ações possíveis navegando pelos menus:

Alterar a Situação Clínica de um Solípede:

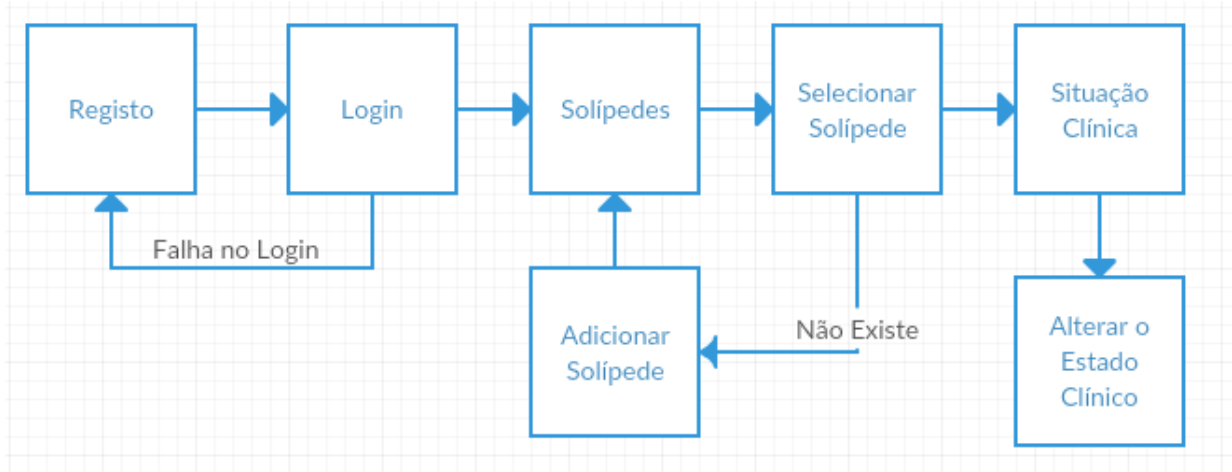


Figura 11 Alterar a Situação Clínica

Aprovar um Diagnóstico:

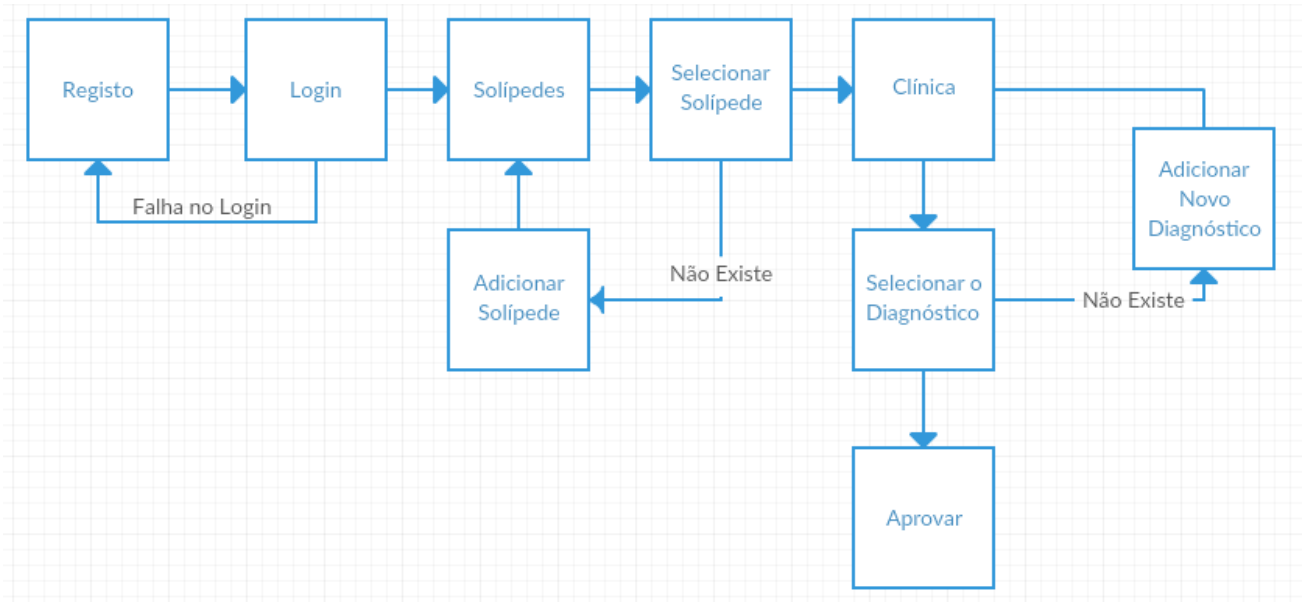


Figura 12 Aprovar um Diagnóstico

4.2 Mockups

No capítulo 2 foram ilustrados vários tipos de interfaces, cada um com as suas vantagens e desvantagens, o objetivo seria manter um sistema de menus coerente e a interface o mais simples possível durante todo o processo, foi escolhido então o sistema de menus laterais.



Figura 13 Mockup do Ecrã Principal

Do lado esquerdo encontram-se as possíveis ações sobre o menu atual, no topo a descrição do menu onde nos encontramos, e no meio fica a informação propriamente dita sobre qual a ação que o utilizador escolheu.

4.2.1 Login

O primeiro ecrã que o utilizador terá contacto é o Login, caso o utilizador não esteja registado poderá fazê-lo através do ecrã de registo.

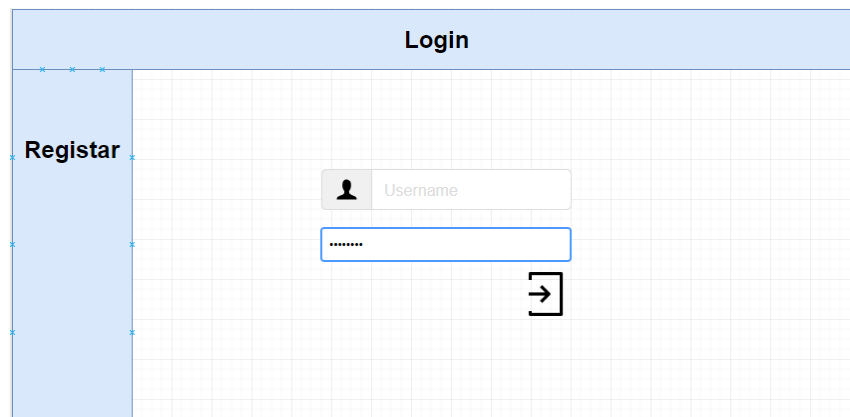


Figura 14 Mockup do Ecrã de Login

4.2.2 Agenda

A agenda contém uma lista de todos os eventos que foram adicionados organizados por data, também disponibiliza uma funcionalidade para apagar o evento em questão.

Caso o utilizador queira adicionar um novo evento basta carregar no botão de adicionar onde terá de indicar a data, hora, o tipo de evento, a descrição e associar o mesmo a um solípede.

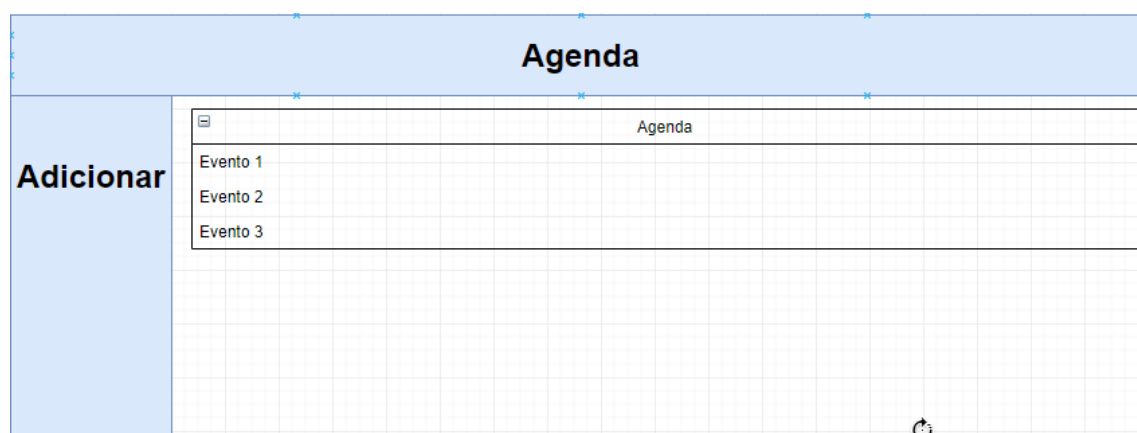


Figura 15 Mockup do Ecrã de Agenda

4.2.3 Gestão de Solípedes

O utilizador terá duas opções, ou adicionar um novo solípede ou escolher um solípede já registado e passar para o próximo ecrã.

Ao adicionar é necessário adicionar alguns dados básicos como o UELN, o seu nome, montada, data de nascimento, entre outros. Ao registar um novo solípede pode agora aceder á sua página de informação.



Figura 16 Mockup do Ecrã de Solípedes

Nesta página contem toda a informação disponível sobre o solípede, o utilizador pode visualizar a foto atual bem como atualizar a mesma, pode também alterar o estado clínico do solípede (representado por uma cor) onde verde significa apto e vermelho não apto.

4.2.4 Ficha Clínica

A página da ficha clínica é composta em duas partes, a primeira mostra uma lista de todos os diagnósticos feitos até ao momento em que cada um pode ser ou não aprovado, a outra parte é a adição de um tratamento a um diagnóstico(opcional) bastando selecionar a linha pretendida.

Diagnóstico				
Novo	ID	Data	Descrição	Aprovado
	1	2018-01-01	Teste Exemplo	Sim Tratamento >
	2	2018-02-09	Teste Exemplo	Não Tratamento >
	3	2018-01-20	Teste Exemplo	Sim Tratamento >

Figura 17. Mockup do Ecrã de Diagnóstico

4.3 Implementação

O código, para uma melhor manutenção e implementação, foi dividido em 3 camadas em cada ficheiro PHP como referenciado no capítulo 2. A primeira camada contém a parte visual ou a interface do programa onde todo o HTML estático se encontra, a segunda camada contém todo o código PHP necessário para o conteúdo dinâmico ou a lógica do negocio, e na terceira camada encontra-se algum JavaScript usado para funções muito específicas.

4.3.1 1ª Camada: HTML

Visto que o protótipo teria de ser altamente responsivo, foi usada uma *framework* CSS da W3.Schools denominada de W3.CSS.

Esta *framework* é aplicada em todo o HTML ao longo do ficheiro PHP, bastando adicionar o link necessário:

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
</head>
```

De seguida usa-se uma combinação de 3 elementos para gerar o Menu lateral, *banner* e o conteúdo da página propriamente dita:

App Coudelaria Militar

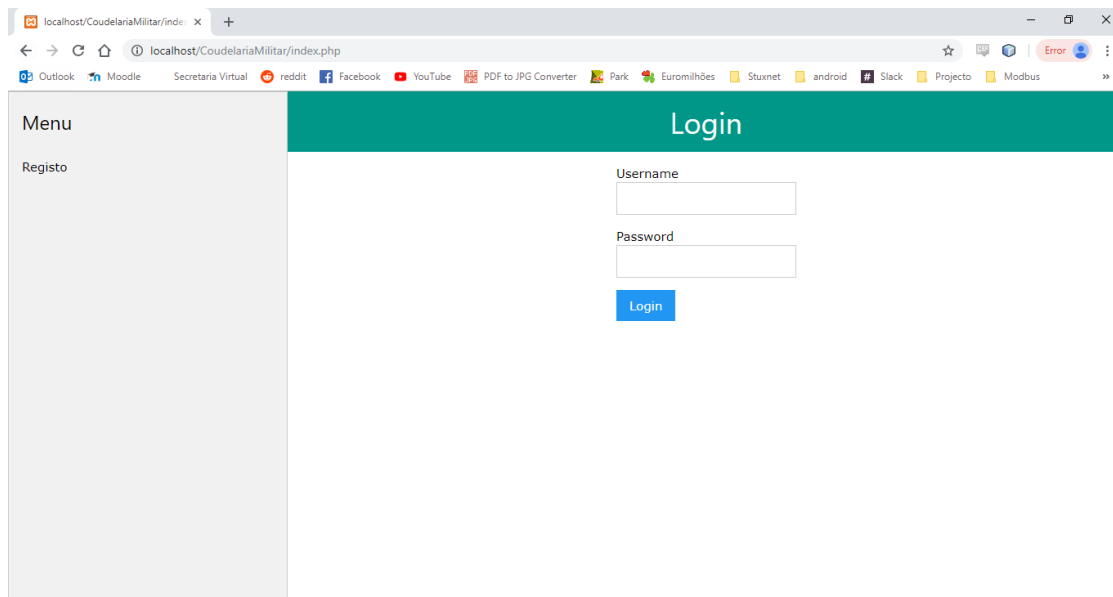


Figura 18 Protótipo: Ecrã Principal

O Menu encontra-se sempre do lado esquerdo e é usado para o utilizador navegar pelos vários ecrãs, é facilmente gerado via um elemento contido na *framework* denominado de *sidebar*:

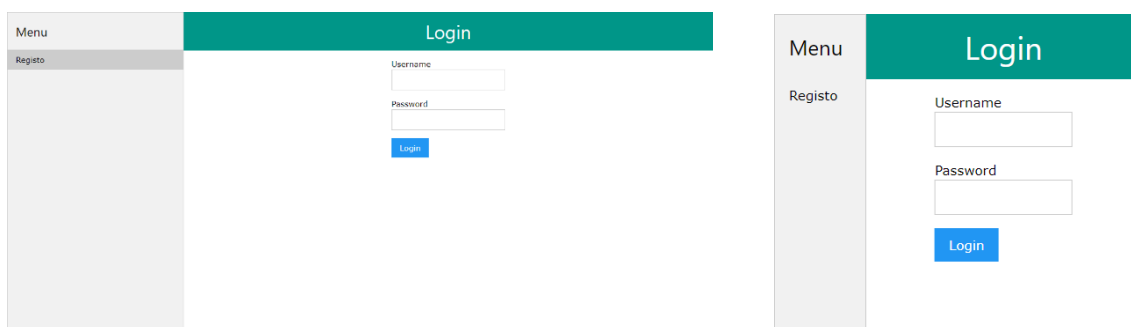
```
<div class="w3-sidebar w3-light-grey w3-bar-block w3-border" style="width:25%">  
  <h3 class="w3-bar-item">Menu</h3>  
  <a href="registo.php" class="w3-bar-item w3-button">Registo</a>  
</div>
```

De notar que tanto o menu como o conteúdo da página em si, têm o tamanho mínimo de 25% do tamanho total disponível, ou seja, caso os elementos sejam encolhidos até esse ponto, nunca vão passar este limite imposto pois o resultado seria uma página com texto impossível de se ler de tão pequeno que seria.

O *banner* ou título encontra-se num *container* à parte, mas visto estar incorporado no mesmo `<div>` que o conteúdo da página principal, também é imposto a largura mínima de 25%.

Por fim o conteúdo da página principal encontra-se todo ele dentro de um *container* onde todo o código PHP usado é automaticamente aplicado a responsividade da *framework* CSS.

As figuras seguintes demonstram como o protótipo se adapta a um ecrã de telemóvel e a um ecrã de portátil usando a responsividade da *framework*:



Pop-Ups

De modo a simular uma mensagem de informação para o utilizador para o utilizador foi utilizado um elemento da *framework* denominado de Modal.

O Modal simula uma pop-up com uma mensagem para o utilizador, essa mensagem pode desde um erro a uma informação de sucesso, visto estar integrada na *framework* também é responsiva:

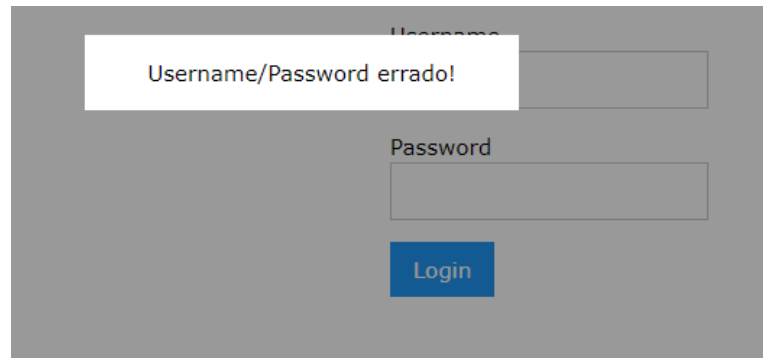


Figura 19 Exemplo de um Pop-Up

Os modais encontram-se no fim do ficheiro PHP e basta adicionar a mensagem que queremos e dar um ID:

```
<div id="id01" class="w3-modal">
  <div class="w3-modal-content w3-center" style="max-width:300px">
    <div class="w3-container">
      <p>Username/Password errado!</p>
    </div>
  </div>
</div>
```

Depois as mesmas são fechadas carregando fora da própria caixa de texto, no código é fechada também dinamicamente suando *Javascript*:

```
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
```


4.3.2 2ª Camada: PHP

Uma *Query* é uma instrução á base de dados, como por exemplo retornar um valor, inserir ou modificar uma linha, no caso do protótipo o mesmo usando *mysqli*, podemos efetuar *queries* “embebidas” no próprio código PHP. Todos os ficheiros PHP contidos no relatório usam sempre a mesma abordagem:

1. Obter os valos dos inputs do utilizador.
2. Efetuar uma *Query* á base de dados com os valores.
3. Obter o resultado.
4. Construir o HTML dinamicamente através de PHP.

Dando como exemplo o login, os passamos acima são demonstrados:

```
$myusername = mysqli_real_escape_string($conn, $_POST['login_username']);  
$mypassword = mysqli_real_escape_string($conn, $_POST['login_password']);
```

Obtemos os valores das caixas de texto ‘login_username’ e ‘login_password’ que vêm em POST e guardamos em duas variáveis. De seguida efetuamos uma Query á base de dados onde tentamos comprar o que o utilizador colocou com o que existe de momento registado:

```
$sql = "SELECT profissional_num_mec FROM trabalho_final.login WHERE username = "  
.$myusername . " and password = " . $mypassword . """;
```

Como podemos verificar podemos usar variáveis diretamente na instrução SQL o que facilita imenso as Queries á base de dados.

Caso não retorne erros obtemos o resultado com o auxílio de uma variável:

```
$result = mysqli_query($conn, $sql);
```

E depois, dependendo de como queremos manipular o resultado, basta gerar o HTML dinamicamente.

4.3.3 3ª Camada: Base de Dados

O acesso á base de dados é feito através da extensão *mysqli*.

Foi escolhida por ser bastante fácil e intuitiva de implementar e também já tinha sido utilizada em outras disciplinas para a ligação á base de dados.

Configuração

A configuração da base de dados foi feita localmente no PC pois para efeitos de protótipo não houve a necessidade de transportar a mesma para um servidor aplicacional na internet.

No ficheiro PHP é sempre colocado um include que contém os dados do acesso ao servidor local:

```
include 'includeDadosBD.php';
```

O mesmo é utilizado em todos os ficheiros PHP que necessitam de acesso á base de dados:

```
<?php
$servername = "127.0.0.1:3306";
$username = "root";
$password = "";
?>
```

Antes de se iniciar a ligação cria-se sempre uma nova conexão instanciando um objeto do tipo *mysqli*, depois usa-se o mesmo ao longo do ficheiro PHP para gestão da ligação. Caso exista um erro, “mata-se” a ligação com a instrução *die*:

```
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Caso esteja tudo ok, avança com as *queries* SQL.

5.Conclusão

Com este trabalho foi possível por em prática todo um conjunto de conhecimento que foi adquirido nestes 3 anos de curso, não serviu apenas para a consolidação de conhecimentos, mas também para tentar dar uma solução a um problema real que os veterinários da Coudelaria Militar estavam a enfrentar.

A escolha da linguagem de programação PHP foi um desafio pessoal imposto pois nunca tendo experiencia com a mesma foi necessário muito trabalho para ultrapassar vários obstáculos á sua implementação, no entanto, a utilização de outras tecnologias que foram apresentadas durante a duração do curso, serviram para consolidar conhecimentos.

Apesar de ser um protótipo, não deixa de ser uma solução viável para ser implementado no sistema que se encontra em produção caso os utilizadores finais assim o entenderem, o facto de ser construído em camadas pode ser utilizado para futuras implementação de outros projetos não tendo necessariamente de serem relacionados com a veterinária.

App Coudelaria Militar

Para melhorias fica a adição de vídeos ou outros objetos multimédia a alguns ecrãs, por exemplo, criar uma galeria de fotos para cada solípede ou adicionar um ou mais vídeos a cada tratamento efetuado.

Bibliografia

(s.d.). Obtido de Usabilidade.gov: <https://usabilidade.gov.pt/conteudos>

(29 de Novembro de 2016). Obtido de Wikipedia:
[https://en.wikipedia.org/wiki/Metro_\(design_language\)](https://en.wikipedia.org/wiki/Metro_(design_language))

(18 de 09 de 2017). Obtido de Waynext: <https://www.waynext.com/waytrends/usabilidade-principios-para-marcas/>

Designrfix. (s.d.). Obtido de designrfix.com: <https://designrfix.com/web-design/advantages-of-using-vertical-navigation-menus>

Ishwaran, A. (25 de Janeiro de 2017). *www.tothenew.com*. Obtido de To The New:
<http://www.tothenew.com/blog/5-advantages-of-developing-a-cross-platform-mobile-app-and-some-of-its-popular-tools/>

Kapahi, M. (14 de Março de 2017). *Izenda*. Obtido de izenda.com:
<https://www.izenda.com/blog/5-benefits-3-tier-architecture/>

opensoft.pt. (s.d.). *OpenSoft*. Obtido de <https://www.opensoft.pt/desenvolvimento-medida/>

SAPO. (s.d.). Obtido de <https://ux.sapo.pt/usabilidade/>

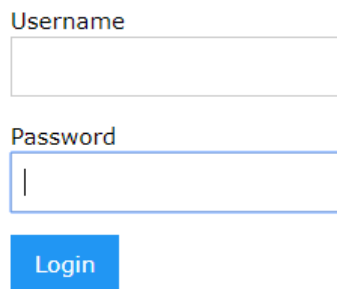
Wroblewski, L. (s.d.). *Mobile First*. Obtido em 18 de 11 de 2018, de
<http://www.lukew.com/ff/entry.asp?933>

Anexo 1. Demo App Coudelaria Militar

Pretende-se agora demonstrar o resultado final da elaboração do protótipo.

Login & Registo

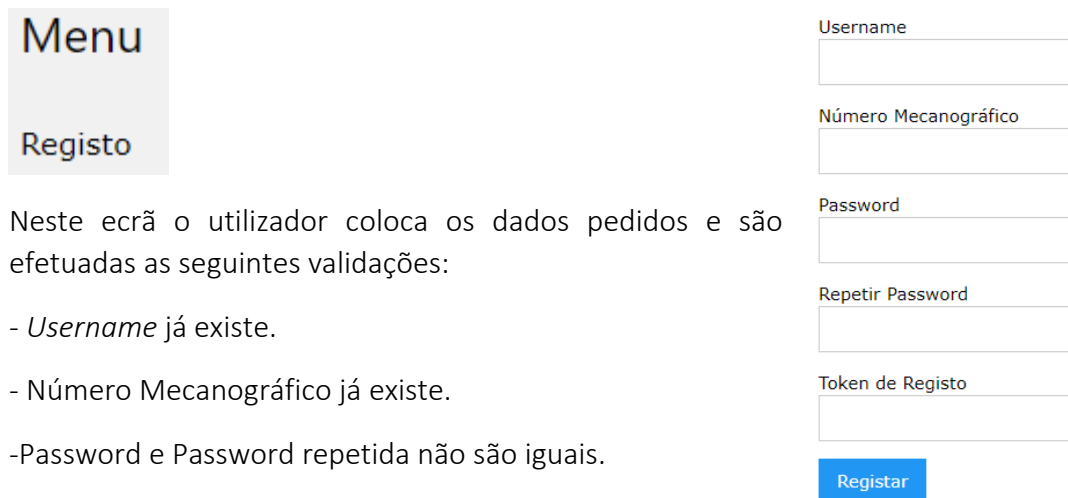
O primeiro ecrã que o utilizador encontra é o de Login.



A login form with two input fields: 'Username' and 'Password'. Below the fields is a blue button labeled 'Login'.

Figura 20 Protótipo: Login

Caso o utilizador falhe o login, tem a opção de se registar utilizando o botão para esse efeito:



A registration form with a menu on the left and input fields on the right. The menu has 'Menu' and 'Registo' options. The form fields are: 'Username', 'Número Mecanográfico', 'Password', 'Repetir Password', and 'Token de Registo'. A blue button labeled 'Registrar' is at the bottom.

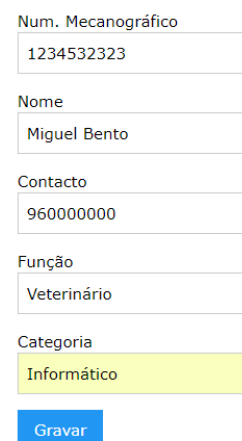
Neste ecrã o utilizador coloca os dados pedidos e são efetuadas as seguintes validações:

- *Username* já existe.
- Número Mecanográfico já existe.
- Password e Password repetida não são iguais.
- *Token* de registo é inválido.

Caso o utilizador passe nas validações surge uma mensagem a indicar que o registo foi efetuado.

Perfil

O ecrã de perfil não é de preenchimento obrigatório, mas, no entanto, é aconselhável o preenchimento do mesmo pois existem algumas funções que utilizam por exemplo a função e a categoria



A profile form with several input fields: 'Num. Mecanográfico' (1234532323), 'Nome' (Miguel Bento), 'Contacto' (960000000), 'Função' (Veterinário), and 'Categoria' (Informático). A blue button labeled 'Gravar' is at the bottom.

do utilizador, é de notar que o número mecanográfico está bloqueado o que não permite a edição do mesmo.

Agenda

O utilizador pode consultar a agenda seleccionando o botão correspondente no ecrã principal:

Agenda

Agenda					
ID	Cavalo	Data	Hora	Tipo	Descrição
2	56456456656	1989-06-06	13:01:00	Vacinação	Tudo ok com o solípede

Figura 21 Protótipo: Agenda

O ecrã principal da agenda contém todos os eventos que existem em sistema para todos os solípedes, o utilizador pode se assim o desejar, apagar qualquer evento da lista.

Ao adicionar um novo evento é encaminhado para outro ecrã:

Neste ecrã não são feitas validações de inserção de dados, apenas é verificado a formatação da data e hora e se existe um solípede selecionado.

Data

Hora

Tipo Evento

Descrição

Selecionar Solípede ▼

Figura 22 Protótipo: Adicionar Evento

App Coudelaria Militar

Qualquer utilizador pode inserir um evento, não havendo qualquer tipo de controlo de perfis.

Token

A gestão de registo de utilizadores é feita através de *tokens* de registo.

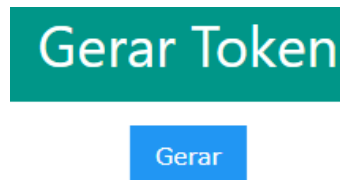


Figura 23 Protótipo: Gestão de Tokens

Os *tokens* gerados são constituídos por uma cadeia de número aleatórios e é feito o controlo de os *tokens* que já foram usados para um registo. Caso o *token* já tenha sido usado, é apresentado uma mensagem no ecrã de registo.

Quando o botão de geração é selecionado, é devolvido um novo *token*:

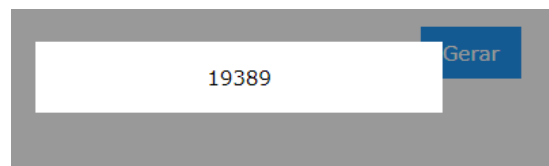


Figura 24 Protótipo: Novo Token Gerado

Solípedes



Figura 25 Protótipo: Estado Geral

App Coudelaria Militar

Neste ecrã o utilizador pode visualizar as informações básicas bem com o estado de cada solípede registado, pode também atualizar a foto e visualizar as informações clínicas do mesmo.

A situação clínica é representada por Apto ou Não Apto e o utilizador pode seleccionar o botão e alterar o estado clínico, ao seleccionar é apresentada a seguinte pop-up:

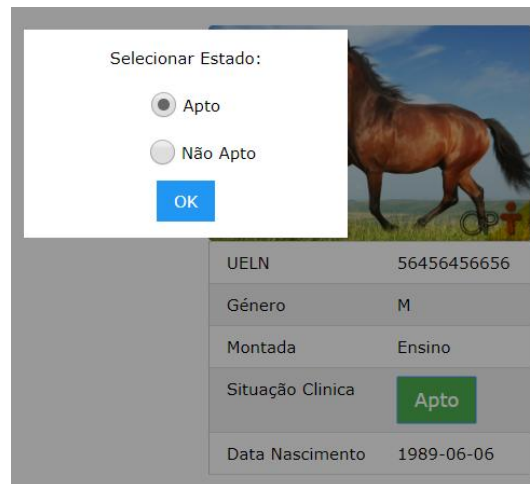


Figura 26 Protótipo: Alterar Estado

O estado também é representado no formato de cores onde vermelho é Não Apto e verde Apto.