**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**ETEC DR EMÍLIO HERNANDEZ AGUILAR**

**KETLYN CESTARI CORREA**

**MATHEUS PEREIRA DOS SANTOS**

**MIGUEL HENRIQUE GUERREIRO SILVA**

**MYLENA KELLY DA SILVA SANTOS**

**NICOLAS LONGO**

**SOPHIA ROSARIO DE JESUS**

**TICYANNE SANTANA BARBOSA**

**DESENVOLVIMENTO DE SITE DE CADASTRO AO PROGRAMA DE CASTRAÇÃO DA PREFEITURA**

***PETCASTRA***

**Franco da Rocha, SP**

**2021**

**KETLYN CESTARI CORREA**

**MATHEUS PEREIRA DOS SANTOS**

**MIGUEL HENRIQUE GUERREIRO SILVA**

**MYLENA KELLY DA SILVA SANTOS**

**NICOLAS LONGO**

**SOPHIA ROSARIO DE JESUS**

**TICYANNE SANTANA BARBOSA**

**DESENVOLVIMENTO DE SITE DE SUPORTE AO TUTOR DO PET**

***PETCASTRA***

Trabalho apresentado ao Centro Estadual de Educação Tecnológica Paula Souza, unidade de Franco da Rocha, Etec Dr Emílio Hernandez Aguilar, como requisito parcial para obtenção do título de Técnico em Desenvolvimento de Sistemas.

Orientadora: Débora Vicente de Oliveira

Coorientador:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Franco da Rocha, SP**

**2021**

Dedicamos este trabalho aos nossos pais, irmãos, primos, e amigos, por terem nos apoiados desde o início desta jornada, e à nossa orientadora, Débora Vicente de Oliveira, que nos auxiliou na germinação das ideias, e durante todo o processo de desenvolvimento.

Agradecemos primeiro a Deus, por ter nos dado a vida. Segundo ao orientador educacional, José Carlos, por ter nos ajudado a conseguir as entrevistas. Aos entrevistados, pela colaboração. Aos nossos pais, por terem nos apoiado e nos dado todo o suporte necessário. Aos irmãos e amigos, por terem sido tão presentes e não nos terem deixado desistir. Também à nossa estimada professora Débora Vicente de Oliveira, por ter nos auxiliado e acreditado no nosso potencial e ao nosso professor Anderson Spera que nos auxiliou durante as reuniões. Aos nossos queridos professores, por nos proporcionarem todo o conhecimento necessário para o desenvolvimento deste projeto. E por fim, a cada um de nós, que nos dedicamos e trabalhamos em harmonia para conseguirmos finalizar este trabalho.

“Castrar é um ato de amor e responsabilidade de todos!”

- Projeto Castra Pet

# RESUMO

O presente trabalho visa apresentar um projeto que tem como foco a castração de animais domésticos; sua importância; e a participação atual da sociedade neste assunto. Através de pesquisas, foi obtido informações de castrações desde o período neolítico, onde a castração era realizada apenas em bovinos, para tranquilizá-los. Já no século XX, era praticada com a pura finalidade de “melhorar” suas carnes. Atualmente, a castração é feita de maneira simples e segura, tanto em machos quanto em fêmeas. Nas fêmeas, os ovários e útero são retirados. Nos machos, os testículos. Segundo dados coletados, a castração em animais domésticos vem crescendo de forma abundante nos últimos anos, e assim, aumentando a demanda de vagas em projetos sociais. Após 07 dias de pesquisa, certas inconsistências foram encontradas no atual sistema de castração oferecido pela prefeitura de Franco da Rocha, tais como: demora no atendimento; apenas um meio de comunicação entre o cliente e a entidade responsável pela castração; a falta de informação e conhecimento sobre o assunto também foi observado. Com isso em mente, este projeto tem como finalidade a construção de um site/aplicativo que facilite a comunicação entre donos de animais domésticos e tais entidades por meio da tecnologia, possibilitando um desempenho mais eficaz.

**PALAVRAS-CHAVES:** castração. animais. tecnologia. atualidade. comunicação**.**

# ABSTRACT

The following work aims to present a project that have focus on the castration of domestic animals; it's importance; and the current participation of society in this matter. Through research, we gathered information on castrations since the Neolithic period, where castration was performed only in cattle to tranquilize them. However, in the 20th century, it was practiced with the sole purpose of “improving” their meats. Presently, the castration is done in a simple and safe way, in both males and females. In females, the ovaries and uterus are removed. In males, the testicles. According to collected data, castration in domestic animals has been growing significantly in recent years, and thus, increasing the demand for vacancies in social projects. After 07 days of research, certain inconsistencies were found in the current castration system offered by the Franco da Rocha city hall, such as: delay in service; only one way of communication between the client and the entity responsible for the castration; the lack of information and knowledge about the subject was also observed. With that in mind, this project aims to build a website/application that improves communication between pet owners and such entities through technology, enabling a more effective performance.

**KEYWORDS**: castration. animals. technology. present. communication.

**INDICE DE FIGURAS**

[Figura 1: Página Inicial. 36](#_Toc106036143)

[Figura 2: Página de Cadastro. 37](#_Toc106036144)

[Figura 3: Página Meus Animais. 37](#_Toc106036145)

[Figura 4: Página Modal de Solicitar Castração. 38](#_Toc106036146)

[Figura 5: Página Principal Clínica. 39](#_Toc106036147)

[Figura 6: Página de Consultar Castrações - Clínica. 39](#_Toc106036148)

[Figura 7: Página Modal Editar Castração - Clínica. 40](#_Toc106036149)

[Figura 8: Página Principal - ADM. 41](#_Toc106036150)

[Figura 9: Página Solicitações - ADM. 41](#_Toc106036151)

[Figura 10: Página para Consultar Usuário - ADM. 42](#_Toc106036152)

[Figura 11: Estrutura Analítica do Projeto (EAP). 45](#_Toc106036153)

[Figura 12: Modelo de Prototipagem. 47](#_Toc106036154)

[Figura 13: Diagrama de Caso de Uso. 50](#_Toc106036155)

[Figura 14: Diagrama de Classes. 51](#_Toc106036156)

[Figura 15: Diagrama de Atividades. 52](#_Toc106036157)

[Figura 16: Diagrama de Sequência. 53](#_Toc106036158)

[Figura 17: Diagrama de Navegação de Telas Usuário. 54](#_Toc106036159)

[Figura 18: Diagrama de Navegação de Telas da Clínica. 54](#_Toc106036160)

[Figura 19: Diagrama de Navegação de Telas do Administrador. 55](#_Toc106036161)

[Figura 20: Diagrama Entidade e Relacionamento (DER). 58](#_Toc106036162)

[Figura 21: Visual Studio Code. 60](#_Toc106036163)

[Figura 22: USBWebserver. 61](#_Toc106036164)

[Figura 23: DIA. 62](#_Toc106036165)

[Figura 24: Draw.io. 63](#_Toc106036166)

[Figura 25: Photoshop. 64](#_Toc106036167)

[Figura 26: Google Forms. 65](#_Toc106036168)

[Figura 27: Trello. 66](#_Toc106036169)

[Figura 28: Excel. 67](#_Toc106036170)

[Figura 29: Powerpoint. 68](#_Toc106036171)

[Figura 30: HTML. 69](#_Toc106036172)

[Figura 31: CSS. 70](#_Toc106036173)

[Figura 32: PHP. 71](#_Toc106036174)

[Figura 33: GitHub. 72](#_Toc106036175)

[Figura 34: Canva. 73](#_Toc106036176)

[Figura 35: Index.php. 75](#_Toc106036177)

[Figura 36: Conexão.php 75](#_Toc106036178)

[Figura 37: Model e-mail 76](#_Toc106036179)

[Figura 38: Página Principal. 77](#_Toc106036180)

[Figura 39: Página Login. 78](#_Toc106036181)

[Figura 40: Página de Recuperação de Senha. 80](#_Toc106036182)

[Figura 41: Página de Código de Confirmação. 81](#_Toc106036183)

[Figura 42: Página Alterar Senha. 82](#_Toc106036184)

[Figura 43: Página de Cadastro. 83](#_Toc106036185)

[Figura 44: Página de Informação. 85](#_Toc106036186)

[Figura 45: Página Home do Usuário. 86](#_Toc106036187)

[Figura 46: Página Meus Animais. 87](#_Toc106036188)

[Figura 47: Página Perfil do Usuário. 89](#_Toc106036189)

[Figura 48: Página Modal de Solicitação. 91](#_Toc106036190)

[Figura 49: Página Modal de Editar o Animal do Usuário. 92](#_Toc106036191)

[Figura 50: Página Cadastrar Animal. 93](#_Toc106036192)

[Figura 51: ADM - Página Principal 95](#_Toc106036193)

[Figura 52: ADM - Modal Iniciar novo mês. 96](#_Toc106036194)

[Figura 53: ADM - Página Principal com dropdown cadastrar. 97](#_Toc106036195)

[Figura 54: ADM - Página principal com dropdown consulta. 98](#_Toc106036196)

[Figura 55: ADM - Página Cadastrar Clínica. 100](#_Toc106036197)

[Figura 56: ADM - Página Cadastrar Raça. 102](#_Toc106036198)

[Figura 57: ADM - Página Consultar Usuários. 103](#_Toc106036199)

[Figura 58: ADM - Página Modal Editar Usuário. 105](#_Toc106036200)

[Figura 59: ADM - Página Animais do Tutor. 107](#_Toc106036201)

[Figura 60: ADM - Página com Modal de Editar Animal 109](#_Toc106036202)

[Figura 61: ADM - Página Consultar Clínicas. 110](#_Toc106036203)

[Figura 62: ADM - Página com Modal de Editar Clínica 112](#_Toc106036204)

[Figura 63: ADM - Página Consultar Castrações. 114](#_Toc106036205)

[Figura 64: ADM - Página Consultar Raça. 116](#_Toc106036206)

[Figura 65: ADM - Página com Modal de Editar Raça. 117](#_Toc106036207)

[Figura 66: ADM - Página de Solicitações. 118](#_Toc106036208)

[Figura 67: ADM - Página Solicitação Clínica 120](#_Toc106036209)

[Figura 68: Clínica - Página Principal. 122](#_Toc106036210)

[Figura 69: Clínica - Página Consultar Castrações. 123](#_Toc106036211)

[Figura 70: Clínica - Página Modal Editar Castrações. 125](#_Toc106036212)

[Figura 71: Clínica - Página de Solicitações. 126](#_Toc106036213)

[Figura 72: Clínica - Página Solicitação de Data/Hora. 127](#_Toc106036214)

**INDICE DE GRÁFICOS**

[Gráfico 1: Gráfico da preferência de pet. 23](#_Toc106036215)

[Gráfico 2: Gráfico da quantidade de pet por pesquisado. 24](#_Toc106036216)

[Gráfico 3: Gráfico da média das idades dos pets dos pesquisados. 24](#_Toc106036217)

[Gráfico 4: Opinião dos pesquisados em relação à castração. 25](#_Toc106036218)

[Gráfico 5: Quantidade dos pesquisados que são de Franco da Rocha. 25](#_Toc106036219)

[Gráfico 6: Quantidade dos pesquisados que não são de Franco da Rocha. 26](#_Toc106036220)

[Gráfico 7: Motivação dos pesquisados sobre castrar o pet. 26](#_Toc106036221)

[Gráfico 8: Preferência de locais de castração. 27](#_Toc106036222)

[Gráfico 9: Contagem dos pesquisados que conhecem o programa de castração de Franco da Rocha. 27](#_Toc106036223)

[Gráfico 10: Opinião dos pesquisados sobre o programa de castração da prefeitura. 28](#_Toc106036224)

[Gráfico 11: Nível de satisfação com o programa da prefeitura. 28](#_Toc106036225)

[Gráfico 12: Problemas com o serviço de castração da prefeitura. 29](#_Toc106036226)

[Gráfico 13: Gráfico de Gantt. 49](#_Toc106036227)

**TABELAS**

[Tabela 1: Matriz de Responsabilidade. 48](#_Toc105853667)

# SUMÁRIO

[RESUMO 6](#_Toc106036228)

[ABSTRACT 7](#_Toc106036229)

[SUMÁRIO 13](#_Toc106036230)

[INTRODUÇÃO 18](#_Toc106036231)

[1 CASTRAÇÃO De Animais 20](#_Toc106036232)

[1.1 Benefícios 20](#_Toc106036233)

[1.1.1 Fêmeas 20](#_Toc106036234)

[1.1.2 Machos 20](#_Toc106036235)

[1.2 MALEFÍCIOS 21](#_Toc106036236)

[1.3 PROCEDIMENTO 21](#_Toc106036237)

[2 METODOLOGIA UTILIZADA 23](#_Toc106036238)

[2.1 LEVANTAMENTO DE INFORMAÇÕES 23](#_Toc106036239)

[2.2 DIFICULDADES ENCONTRADAS 29](#_Toc106036240)

[2.3 PROPOSTA DE SOLUÇÃO 30](#_Toc106036241)

[3 ENGENHARIA DE SOFTWARE 31](#_Toc106036242)

[3.1 ESTUDO DE VIABILIDADE 31](#_Toc106036243)

[3.2 REQUISITOS DO SISTEMA 31](#_Toc106036244)

[3.2.1 Funcionais 32](#_Toc106036245)

[3.2.2 Não Funcionais 33](#_Toc106036246)

[3.2.3 De Dados 34](#_Toc106036247)

[**3.2.3.1** Usuário: 34](#_Toc106036248)

[**3.2.3.2** Animal: 34](#_Toc106036249)

[3.2.4 De Segurança 35](#_Toc106036250)

[3.2.5 De Layout 35](#_Toc106036251)

[3.2.5.1 Página principal 36](#_Toc106036252)

[3.2.5.2 Página de cadastro 36](#_Toc106036253)

[3.2.5.3 Página meus animais 37](#_Toc106036254)

[3.2.5.4 Página modal de solicitação de castração 38](#_Toc106036255)

[3.2.5.5 Página principal do usuário clínica 38](#_Toc106036256)

[3.2.5.6 Página de consultar castrações do usuário clínica 39](#_Toc106036257)

[3.2.5.7 Página modal atualizar castração do usuário clínica 40](#_Toc106036258)

[3.2.5.8 Página principal do administrador 40](#_Toc106036259)

[3.2.5.9 Página de solicitações do administrador 41](#_Toc106036260)

[3.2.5.10 Página do administrador para consultar usuários 42](#_Toc106036261)

[3.3 ESCOPO 42](#_Toc106036262)

[3.3.1 Identificação 43](#_Toc106036263)

[3.3.2 Escopo do produto 43](#_Toc106036264)

[3.3.3 Requisitos de alto nível 43](#_Toc106036265)

[3.3.4 Entregas 43](#_Toc106036266)

[3.3.5 Critérios de aceitação: 44](#_Toc106036267)

[3.3.6 Não-escopo do projeto: 44](#_Toc106036268)

[3.3.7 Premissas: 44](#_Toc106036269)

[3.3.8 Restrições: 45](#_Toc106036270)

[3.4 ESTRUTURA ANALÍTICA DO PROJETO (EAP) 45](#_Toc106036271)

[3.5 CICLO DE VIDA DO SOFTWARE 46](#_Toc106036272)

[3.6 MATRIZ DE ATIVIDADES E RESPONSABILIDADES 47](#_Toc106036273)

[3.7 GRÁFICO DE GANTT 48](#_Toc106036274)

[3.8 DIAGRAMAS UML 49](#_Toc106036275)

[3.8.1 Caso de Uso 49](#_Toc106036276)

[3.8.2 Classes 51](#_Toc106036277)

[3.8.3 Atividades 51](#_Toc106036278)

[3.8.4 Sequência 52](#_Toc106036279)

[3.8.5 Navegação 53](#_Toc106036280)

[3.8.5.1 Diagrama de Navegação de Telas Usuário. 54](#_Toc106036281)

[3.8.5.2 Diagrama de Navegação de Telas da Clínica. 54](#_Toc106036282)

[3.8.5.3 Diagrama de Navegação de Telas do Administrador. 55](#_Toc106036283)

[4 BANCO DE DADOS 56](#_Toc106036284)

[4.1 MODELAGEM CONCEITUAL 56](#_Toc106036285)

[4.1.1 Modelo Entidade e Relacionamento (MER) 57](#_Toc106036286)

[4.2 MODELAGEM LÓGICA 57](#_Toc106036287)

[4.2.1 Diagrama Entidade e Relacionamento (DER) 57](#_Toc106036288)

[4.3 NORMALIZAÇÃO 58](#_Toc106036289)

[4.4 MODELAGEM FÍSICA 59](#_Toc106036290)

[5 FERRAMENTAS UTILIZADAS 60](#_Toc106036291)

[5.1 VISUAL STUDIO CODE (VS CODE) 60](#_Toc106036292)

[5.2 USBWEBSERVER 61](#_Toc106036293)

[5.3 DIA 61](#_Toc106036294)

[5.4 DRAW.IO 62](#_Toc106036295)

[5.5 PHOTOSHOP CS6 63](#_Toc106036296)

[5.6 GOOGLE FORMS 64](#_Toc106036297)

[5.7 TRELLO 65](#_Toc106036298)

[5.8 EXCEL 66](#_Toc106036299)

[5.9 POWERPOINT 67](#_Toc106036300)

[5.10 HTML 68](#_Toc106036301)

[5.11 CSS 69](#_Toc106036302)

[5.12 PHP 70](#_Toc106036303)

[5.13 GITHUB 71](#_Toc106036304)

[5.14 CANVA 72](#_Toc106036305)

[6 MANUAL 74](#_Toc106036306)

[6.1 MANUAL TÉCNICO 74](#_Toc106036307)

[6.1.1 Index.php 74](#_Toc106036308)

[6.1.2 Conexao.php 75](#_Toc106036309)

[6.1.3 Model e-mail 76](#_Toc106036310)

[6.2 MANUAL DO USUÁRIO 76](#_Toc106036311)

[6.2.1 Usuário 77](#_Toc106036312)

[6.2.1.1 Página principal do site 77](#_Toc106036313)

[6.2.1.2 Página Login 78](#_Toc106036314)

[6.2.1.3 Página de recuperação de senha 79](#_Toc106036315)

[6.2.1.4 Página de Código de Confirmação 81](#_Toc106036316)

[6.2.1.5 Página de alteração de senha 82](#_Toc106036317)

[6.2.1.6 Página de cadastro 83](#_Toc106036318)

[6.2.1.7 Página de informação 85](#_Toc106036319)

[6.2.1.8 Página home do usuário 86](#_Toc106036320)

[6.2.1.9 Página meus animais 87](#_Toc106036321)

[6.2.1.10 Página perfil do usuário 89](#_Toc106036322)

[6.2.1.11 Página com a modal de solicitação 91](#_Toc106036323)

[6.2.1.12 Página modal de editar o animal do usuário 92](#_Toc106036324)

[6.2.1.13 Página de cadastro animal 93](#_Toc106036325)

[6.2.2 Administrador 95](#_Toc106036326)

[6.2.2.1 Página principal do administrador 95](#_Toc106036327)

[6.2.2.2 Página principal do administrador com modal início de mês 96](#_Toc106036328)

[6.2.2.3 Página principal do administrador com dropdown cadastrar. 97](#_Toc106036329)

[6.2.2.4 Página principal do administrador com dropdown consulta 98](#_Toc106036330)

[6.2.2.5 Página do administrador - cadastrar clínica 99](#_Toc106036331)

[6.2.2.6 Página do administrador - cadastrar raça do animal 101](#_Toc106036332)

[6.2.2.7 Página do administrador - consultar usuários 103](#_Toc106036333)

[6.2.2.8 Página do administrador – modal editar usuário 105](#_Toc106036334)

[6.2.2.9 Página do administrador – animais do tutor 106](#_Toc106036335)

[6.2.2.10 Página do administrador – modal editar animal 108](#_Toc106036336)

[6.2.2.11 Página do administrador - consultar clínicas 110](#_Toc106036337)

[6.2.2.12 Página do administrador – modal editar clínica 112](#_Toc106036338)

[6.2.2.13 Página do administrador - consultar castrações 114](#_Toc106036339)

[6.2.2.14 Página do administrador – consultar raça 116](#_Toc106036340)

[6.2.2.15 Página do administrador – modal editar raça 117](#_Toc106036341)

[6.2.2.16 Página do administrador - solicitações 118](#_Toc106036342)

[6.2.2.17 Página do administrador – solicitação clínica 119](#_Toc106036343)

[6.2.3 Clínica 121](#_Toc106036344)

[6.2.3.1 Página principal do usuário clínica 121](#_Toc106036345)

[6.2.3.2 Página do usuário clínica - consultar castrações 122](#_Toc106036346)

[6.2.3.3 Página do usuário clínica - modal editar castração 124](#_Toc106036347)

[6.2.3.4 Página do usuário clínica - solicitações 126](#_Toc106036348)

[6.2.3.5 Página do usuário clínica – solicitação data/hora 127](#_Toc106036349)

[7 CONSIDERAÇÕES FINAIS 129](#_Toc106036350)

[7.1 Comunicação entre usuário e prefeitura. 129](#_Toc106036351)

[7.2 Visibilidade. 129](#_Toc106036352)

[7.3 Informação sobre início e fim do programa. 130](#_Toc106036353)

[7.4 Punições em caso de faltas. 130](#_Toc106036354)

[7.5 Facilidade no transporte de informação para Excel. 130](#_Toc106036355)

[CONCLUSÃO 131](#_Toc106036356)

[APÊNDICE A 133](#_Toc106036357)

[APÊNDICE B 146](#_Toc106036358)

[REFERÊNCIAS 543](#_Toc106036359)

# INTRODUÇÃO

Desde os tempos antigos, a relação dos humanos com os animais é retratada de maneira marcante; tornando-se algo essencial, visto que é benéfica para ambos os lados.

  No período neolítico, a castração era muito comum em bovinos e tinha como propósito acalmar os machos - já que nas décadas 20 e 30 não existiam cercas definidas e os machos acabavam indo para uma área distante, porém na metade do século XX, tinham como objetivo melhorar a qualidade da carne.

Durante os últimos anos, a interação dos humanos com os animais aumentou de forma significativa. Apenas na pandemia, houve um aumento de 30% nas adoções de animais domésticos e de acordo com um artigo publicado no site Jovem Pan (2021), entre os felinos, 84% foram adquiridos. E entre os cães, 54%.

Ao entrar nesse assunto, é apenas natural o tópico “castração” surgir, afinal, é algo de extrema importância além de ser um ato de amor aos animais domésticos.

No caso dos animais domésticos, o objetivo da castração é oferecer um estilo de vida melhor, tornando-os mais dóceis e emocionalmente estáveis, ajudando também na diminuição das superpopulações de animais em situação de rua.

O medo da castração é algo recorrente que assola o coração de muitas pessoas, o que é algo normal, considerando o quão importantes os animais são para seus donos. Contudo esse medo se torna desnecessário quando a castração é realizada em clínicas especializadas e com profissionais capazes participando do procedimento. Infelizmente, há notícias falsas e fantasiosas circulando pela mídia, o que alimenta ainda mais o medo.

Existem muitos mitos sobre a castração de animais, tais como: “As fêmeas devem ter ao menos uma ninhada antes de serem castradas, para serem felizes e saudáveis”, a verdade é que os animais não precisam ter filhotes para viver uma vida plena e serem felizes.

A tecnologia veio inovando a castração através de novos recursos como os instrumentos, anestesias e técnicas cirúrgicas fazendo com que o animal sofra o mínimo possível nesse processo tão importante para o seu bem-estar.

Além dessas formas de inovação, a comunicação foi um meio que também recebeu grande ajuda da tecnologia, fazendo com que o tutor do animal seja informado do local da castração e seu método.

 Com a inovação deste método, surgiram programas gratuitos realizados pelas prefeituras, tornando a castração mais acessível, principalmente para pessoas de baixa renda. Mesmo com este grande avanço, muitas pessoas ainda não têm conhecimento sobre os programas, muito menos sobre a castração.

Na era digital, a melhor maneira para que a informação chegue nas pessoas é por meio das mídias; podendo ser transmitida em redes sociais, sites ou aplicativos de comunicação em massa para a divulgação das informações necessárias ao público.

Por fim, o projeto tem o objetivo não apenas de facilitar o trabalho do tutor do animal, mas também do servidor público, livrando-o de responder perguntas repetitivas e tediosas a todo momento enquanto o programa estiver ativo, podendo agilizar o processo de cadastro com o uso de um site próprio, que antes era feito em sua maior parte por telefone e verificar a disponibilidade de vagas no serviço. Além de informar sobre as datas de quando haverá a próxima campanha, o período que durará a mesma e informar os benefícios que o processo de castração pode trazer ao dono do pet, a cidade e ao animal.

Mesmo sendo algo essencial nas vidas dos animais, em alguns casos, a castração pode ter efeitos contrários. Este assunto será apresentado melhor a seguir.

# CASTRAÇÃO De Animais

Nos dias atuais, a castração é cada vez mais procurada por aqueles que possuem animais domésticos pois, além de ser um procedimento rápido e seguro para o animal, também evita a reprodução exagerada e dociliza seu comportamento, possuindo muitos outros benefícios.

Os malefícios não devem ser esquecidos e, dos casos em que a castração não é recomendada.

## Benefícios

De acordo com o Olhar Animal (2014), existem os seguintes benefícios:

### Fêmeas

* Evita infecção uterina (piometra – doença que atinge 60% das fêmeas);
* Se realizado antes do primeiro cio, diminui em até 95% as chances de tumor de mama;
* Evita gravidez indesejada, fugas de casa, e outros incômodos com o cio (como a “miação nervosa” das gatas);
* Evita o abandono de filhotes; inteiras, quando indesejadas.

### Machos

* Evita brigas por disputa territorial;
* Evita/diminui demarcação com urina em todos os lugares da casa;
* Diminui muito o cheiro forte da urina dos gatos;
* Previne tumores de próstata, e consequentemente hérnias Perineais;
* Evita que eles fujam de casa atrás de fêmeas no cio.

Nos animais em geral, evita que eles fujam de casa atrás de parceiros quando no cio; previne contra a “continuidade” de doenças hereditárias (tais como hérnias em geral, luxação de patela, displasia coxo-femoral…); além de aumentar sua expectativa de vida.

Aos cães que saem à rua: por não cruzarem, evita as chances de adquirir TVT (tumor venéreo transmissível);

## MALEFÍCIOS

A castração também possui suas desvantagens, sendo elas: propensão a dois tipos de câncer; crescimento elevado; complicações durante a operação, dentre outros.

Entretanto, de acordo com o veterinário Marco Antônio Gioso (2010), não existem malefícios na castração.

## PROCEDIMENTO

A realização desse procedimento em cães e gatos, tanto em fêmeas quanto em machos é muito simples, com poucos riscos à saúde do animal. A esterilização é feita depois da checagem da saúde do animal com a realização de uma bateria de exames. Nas fêmeas, a cirurgia é feita com a retirada do útero e dos ovários, e um dos maiores benefícios é a prevenção de tumores. Nos machos, a castração é feita com a retirada dos testículos.

A tecnologia veio inovando a castração através de novos recursos como os instrumentos, anestesias e técnicas cirúrgicas fazendo com que o animal sofra o mínimo possível nesse processo tão importante para o seu bem-estar.

Além dessas formas de inovação, a comunicação foi um meio que também recebeu grande ajuda da tecnologia, fazendo com que o tutor do animal seja informado do local da castração e seu método.

É recomendado que seja feita antes do primeiro ano de vida, aos cinco meses de idade ou até mesmo antes disso. Na chamada posse responsável os animais deveriam ser castrados sempre. Animais castrados não procriam, resultando em menos casos de abandono e uma vida mais saudável àqueles que passam pelo procedimento.

Ainda existem muitos mitos sobre a castração de animais, tais como: “As fêmeas devem ter ao menos uma ninhada antes de serem castradas, para serem felizes e saudáveis”, a verdade é que os animais não precisam ter filhotes para viver uma vida plena e serem felizes.

Atualmente a castração de pets é mais acessível, feita a partir de programas gratuitos, em clínicas veterinárias e em pet shops.

# METODOLOGIA UTILIZADA

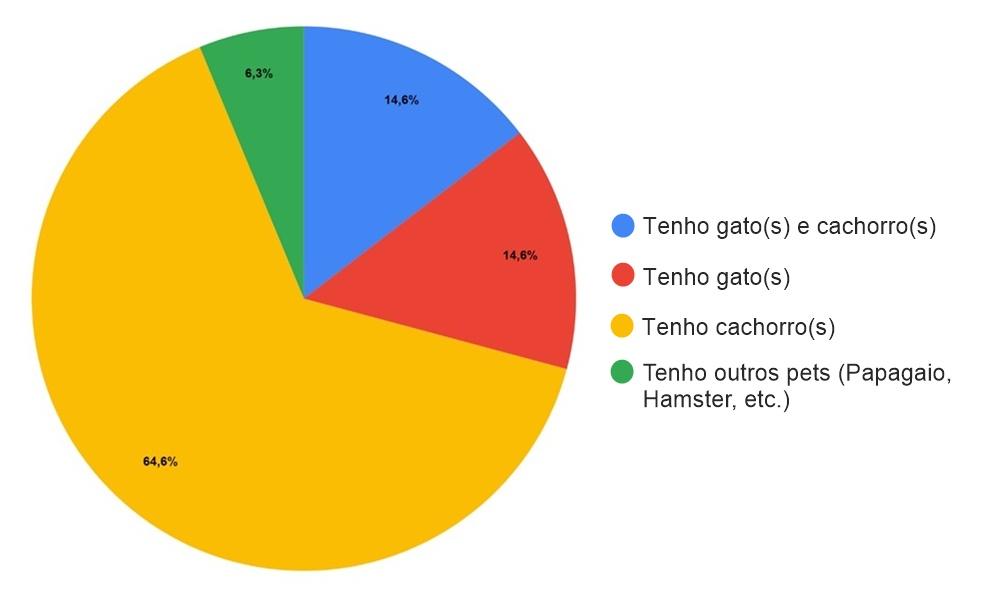
Metodologia é uma palavra derivada de “método”, vinda do latim “methodus” no qual seu significado é “caminho ou a via para a realização de algo”. Metodologia é o estudo dos melhores métodos para chegar a um determinado fim. Em um trabalho científico, a metodologia é a parte que é feita uma descrição minuciosa e rigorosa do objeto de estudo e dos métodos usados na pesquisa. (Significados, 2021).

Este projeto utilizou-se de pesquisas web gráficas que foi usada para um melhor entendimento do tema, aplicação de questionários para a população para maiores informações, e entrevista.

## LEVANTAMENTO DE INFORMAÇÕES

Para entender as dificuldades e saber opiniões referentes aos problemas, foi realizada uma pesquisa no dia 31 de agosto de 2021, através da ferramenta Google Forms. Com base na pesquisa foi possível coletar respostas de mais de 100 pessoas de cidades diferentes. Seguem informações coletadas.

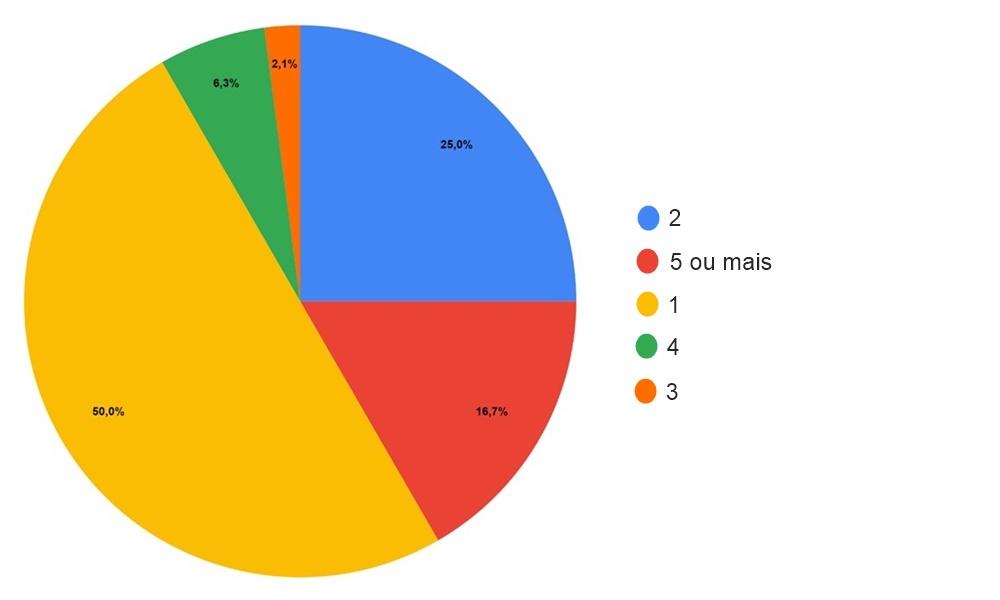
Gráfico : Gráfico da preferência de pet.



Fonte: Projeto PetCastra

93,8% da população possui gatos e cachorros. O restante (6,3%) possui outros pets.

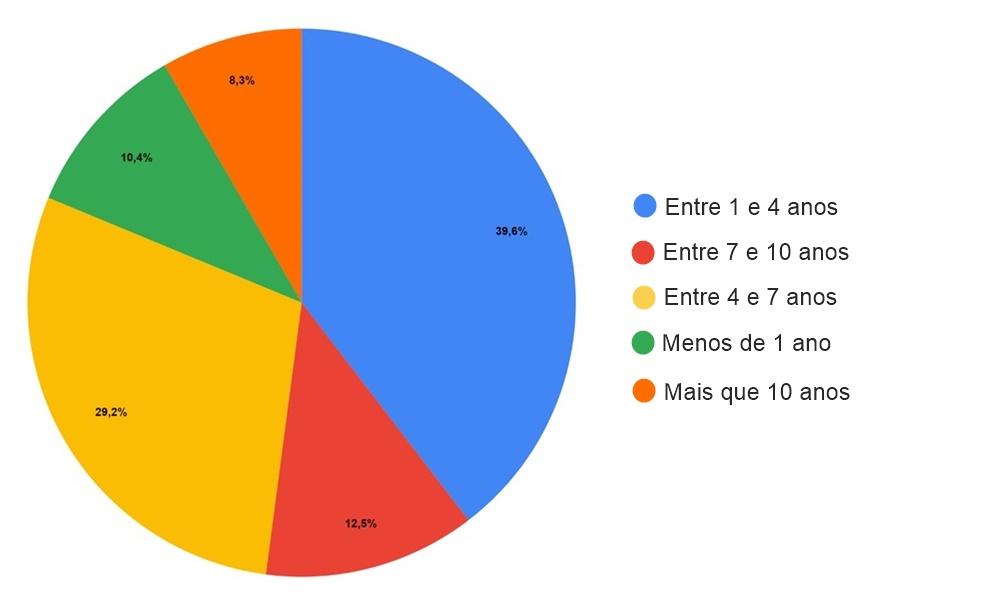
Gráfico : Gráfico da quantidade de pet por pesquisado.



Fonte: Projeto PetCastra

Metade dos entrevistados possuem apenas um animal, 25% possuem 2, 2,1% possuem 3, 6,3% possuem 4, e 16,7% dos entrevistados possuem 5 ou mais.

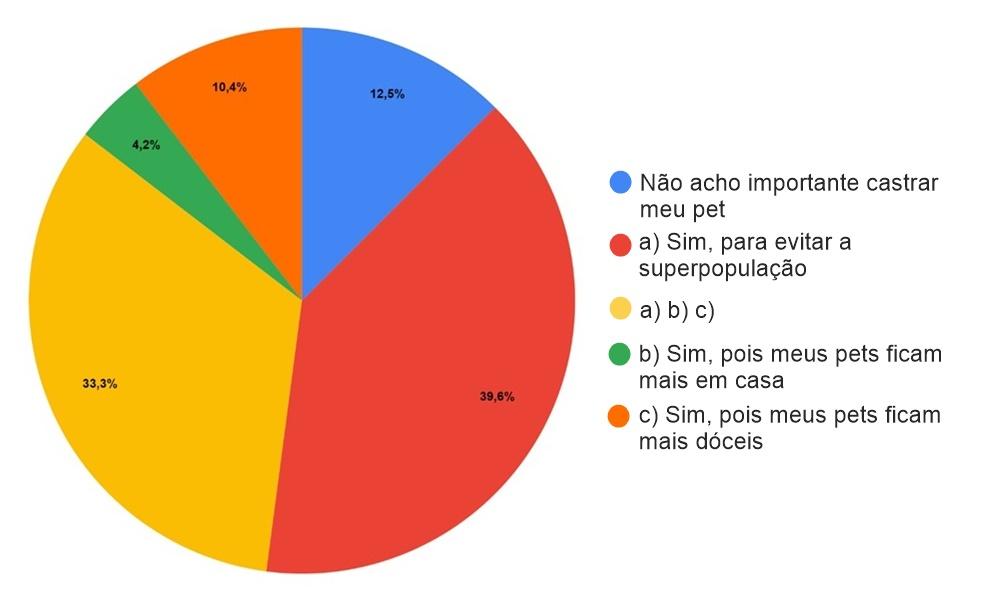
Gráfico 3: Gráfico da média das idades dos pets dos pesquisados.



Fonte: Projeto PetCastra

50% das pessoas têm animais de até 4 anos, 10,4% com menos de 1 ano. Os outros 50% são animais com mais de 4 anos de idade, o restante tem mais de 10 anos.

Gráfico 4: Opinião dos pesquisados em relação à castração.



Fonte: Projeto PetCastra

87,5% das pessoas acham muito importante a castração, sendo 39,6% para evitar a superpopulação.

Gráfico 5: Quantidade dos pesquisados que são de Franco da Rocha.

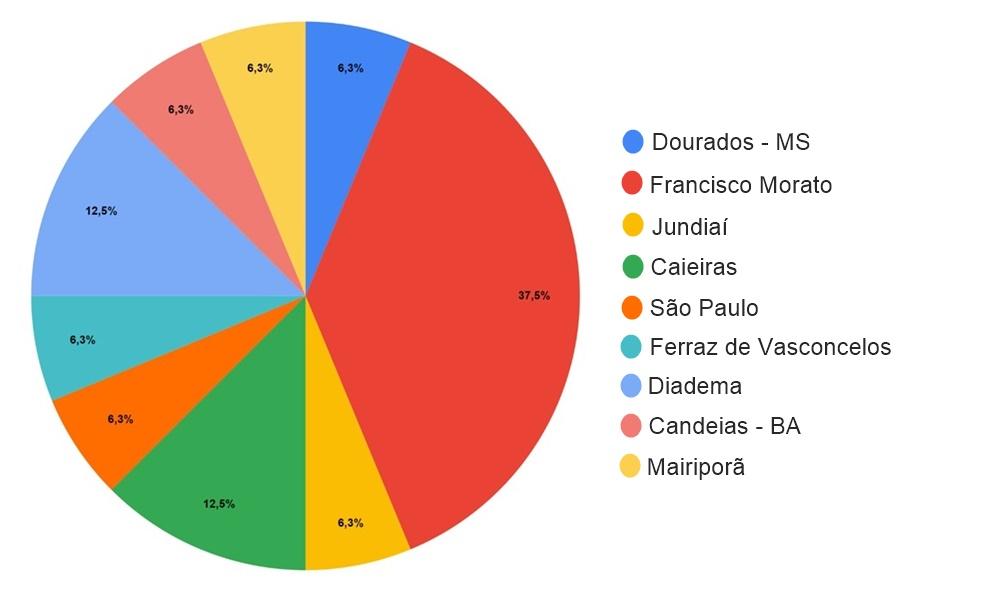
Gráfico, Gráfico de pizza

Descrição gerada automaticamente

Fonte: Projeto PetCastra

62,5% dos entrevistados moram em Franco da Rocha. O restante já morou ou conhece pessoas que moram.

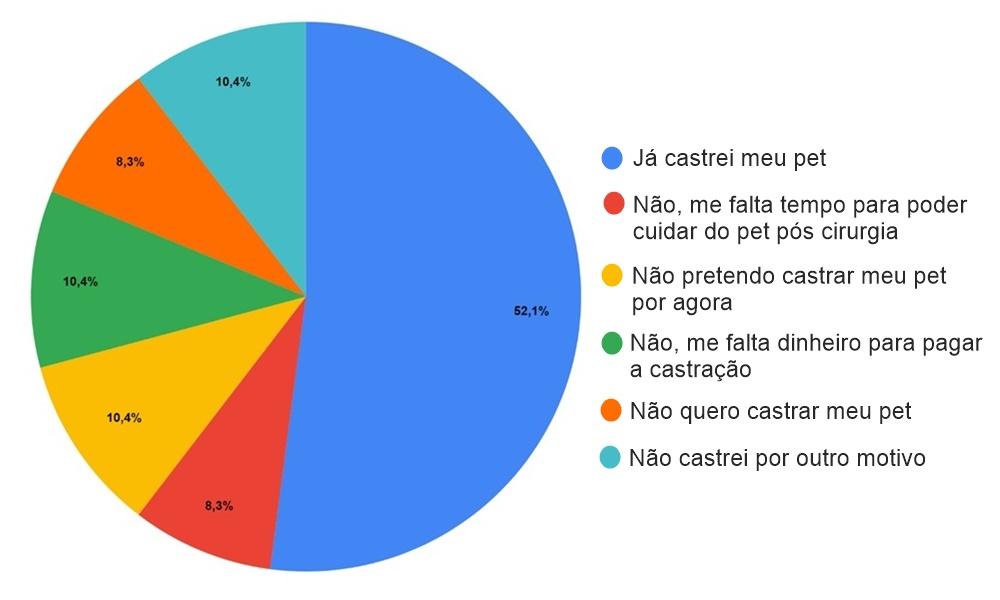
Gráfico 6: Quantidade dos pesquisados que não são de Franco da Rocha.



Fonte: Projeto PetCastra

A maioria dos entrevistados não Franco-rochenses são da região; 37,5% de Francisco Morato; 12,5% de Caieiras; 6,3% de Mairiporã; 6,3% de Jundiaí. Os outros 37,7% estão espalhados pelas cidades de Ferraz de Vasconcelos, São Paulo, Diadema, Candeias e Dourados.

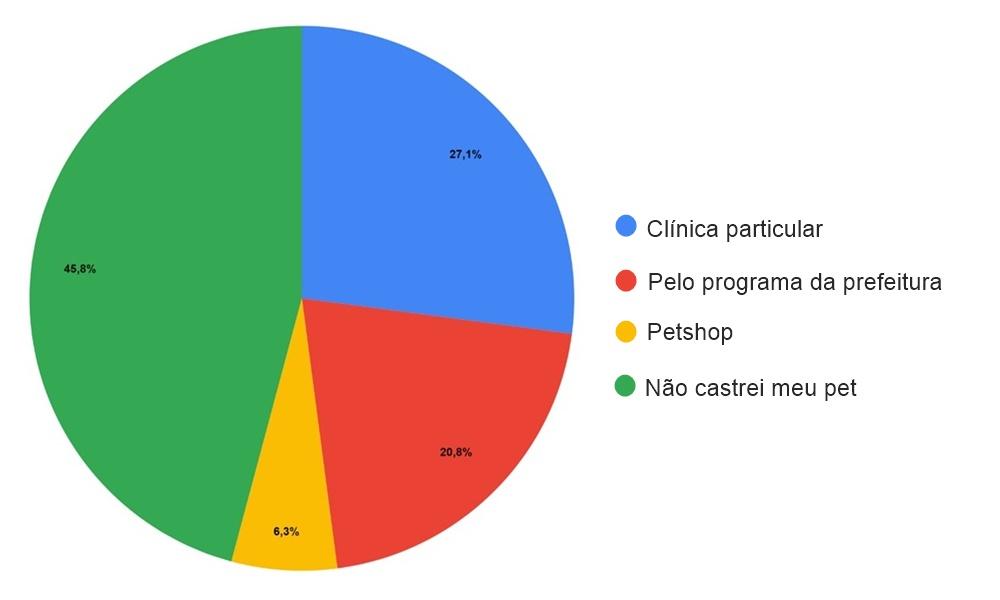
Gráfico 7: Motivação dos pesquisados sobre castrar o pet.



Fonte: Projeto PetCastra

52,1% da população já castrou seu animal e 10,4% não castraram por falta de dinheiro.

Gráfico 8: Preferência de locais de castração.



Fonte: Projeto PetCastra

54,2% das pessoas que responderam à pesquisa castraram seus animais; 27,1% em clínica particular; 20,8% pelo programa da prefeitura; 6,3% em Pet Shops. Restando 45,8% que não castraram.

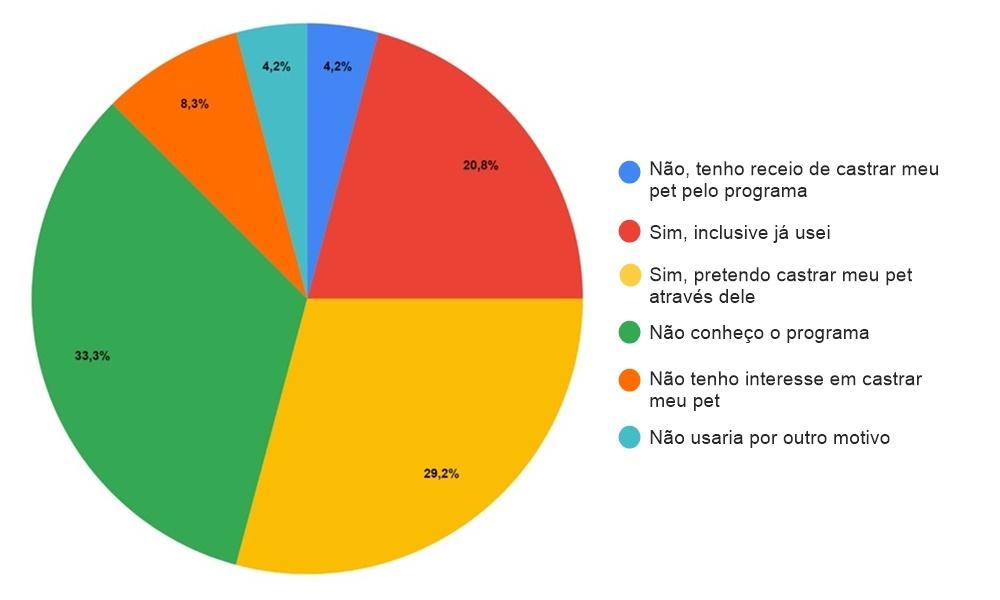
Gráfico 9: Contagem dos pesquisados que conhecem o programa de castração de Franco da Rocha.



Fonte: Projeto PetCastra

De acordo com o gráfico, 45,8% da população conhece a campanha de castração de Franco Rocha e 54,2% não conhecem, isto mostra que o projeto proposto pode também informar essa parte da população que não tem conhecimento sobre a campanha.

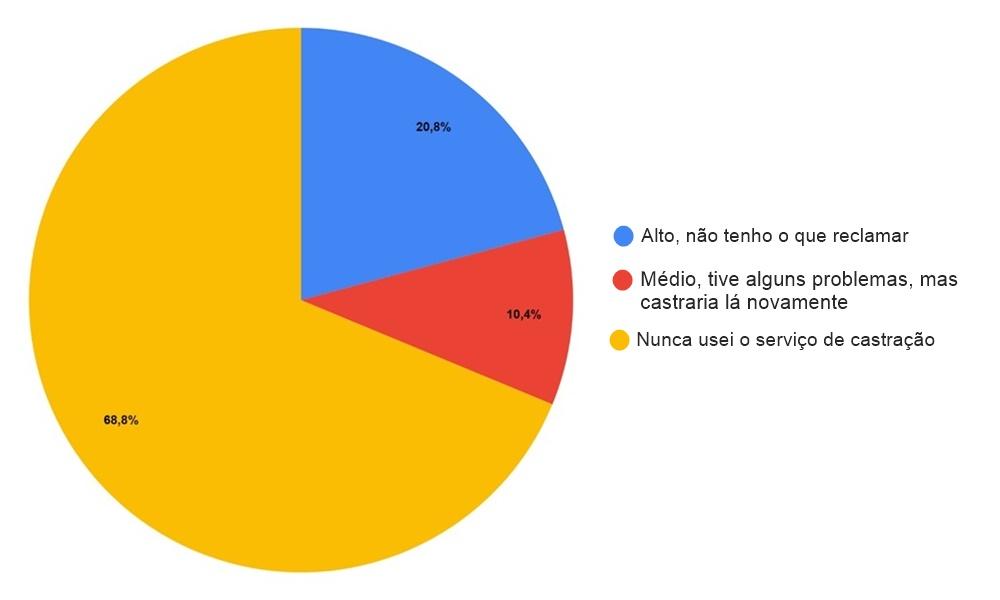
Gráfico 10: Opinião dos pesquisados sobre o programa de castração da prefeitura.



Fonte: Projeto PetCastra

8,2% das pessoas não usariam o programa gratuito por ter receio ou por outro motivo; 8,3% não demonstraram interesse em castrar seu animal; 33,3% não conheciam o programa de castração; 50% usariam o programa e 20,8% afirmaram que já usaram o programa.

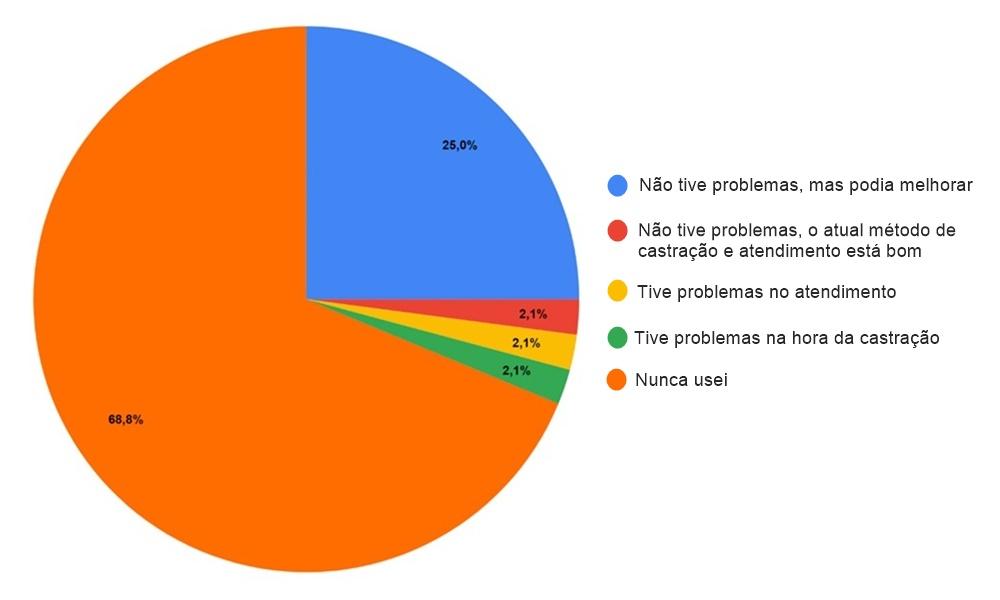
Gráfico 11: Nível de satisfação com o programa da prefeitura.



Fonte: Projeto PetCastra

68,8% nunca usaram o serviço de castração, um número bem alto e que poderia ser mudado. Dos 31,2% restantes, 20,8% não têm reclamações e os outros 10,4% tiveram alguns problemas.

Gráfico 12: Problemas com o serviço de castração da prefeitura.



Fonte: Projeto PetCastra

68,8% nunca usaram o programa de castração; 25% não tiveram problemas, mas acham que precisa melhorar; 2,1% não tiveram problemas e acham que o atual método está bom; 2,1% tiveram problemas com o atendimento e 2,1% tiveram problemas com a castração do animal.

## DIFICULDADES ENCONTRADAS

Com base nas informações coletadas, a proposta inicial é desenvolver um site que incentive e conscientize a população sobre a castração de animais domésticos. Durante a pesquisa, insatisfações com o atual sistema de cadastro foram encontradas, portanto, houve a ideia de conectar o site com a própria prefeitura e com as clínicas veterinárias vinculadas para um melhor atendimento, tanto para o animal quanto para o dono.

Dos problemas atuais relacionados com as tecnologias no processo de castração, há uma parte desse processo que é fundamental, sendo ela a etapa de cadastro do animal e do dono para a realização da cirurgia. De acordo com a pesquisa realizada para o projeto, 68,8% dos entrevistados nunca usaram o programa de castração gratuito da prefeitura, 25% falaram que o sistema podia melhorar, 4,2% tiveram algum problema no processo e somente 2,1% ficaram satisfeitos com a maneira atual do programa. Isso indica que o programa não está sendo uma escolha interessante para as pessoas na região e que o atual método pode evoluir para um mais eficaz utilizando da tecnologia, facilitando o processo todo e adicionando novidades.

## PROPOSTA DE SOLUÇÃO

Por fim, o projeto tem como principal objetivo facilitar o acesso ao cidadão que planeja fazer parte do serviço de castração. Originalmente planejado para ser entregue às mãos da prefeitura de Franco da Rocha, o projeto tem o objetivo não apenas de facilitar o trabalho do usuário, mas também do servidor público, livrando-o de responder perguntas repetitivas e tediosas a todo momento enquanto o programa estiver ativo, podendo agilizar o processo de cadastro com o uso de um site próprio que antes era feito em sua maior parte por telefone e verificar a disponibilidade de vagas no serviço; além de informar sobre datas de quando haverá a próxima campanha; o período que durará a mesma e informar os benefícios que o processo de castração pode trazer ao animal, seu dono e a cidade.

# ENGENHARIA DE SOFTWARE

Engenharia de Software é um conjunto de instruções para que o computador faça alguma tarefa desejada, sendo eles planejados e construídos de forma lógica pelos engenheiros de software (COSTA, 2015).

Engenharia de Software é a aplicação de um procedimento sistemático, metódico e quantificável em todas as fases do processo de software, sendo elas o planejamento, desenvolvimento e a manutenção do sistema (COSTA, 2015).

## ESTUDO DE VIABILIDADE

Esse estudo visa avaliar a função, desempenho e limitações que um software terá. Com isso é possível identificar se o sistema proposto atenderá ou não as necessidades do cliente (Fernando Godoy, 2012).

Este projeto propõe-se atender todas as necessidades do cliente e dos usuários, como por exemplo, seu acesso ao cadastro para efetuar a castração de seu animal. Facilitando assim, a comunicação de ambas as partes.

## REQUISITOS DO SISTEMA

Requisito é um conceito que tem a sua raiz etimológica no latim requīsitus, palavra esta que, por sua vez, deriva do verbo latino “requirere”, que se pode traduzir como “reclamar” ou “requerer”. (2016, CONCEITO DE).

De forma geral, um requisito é uma condição para se realizar determinada tarefa. Em um sistema, um requisito é uma condição para se fazer alguma ação.

### Funcionais

Requisito Funcional é a requisição de uma função que um software deverá atender/realizar. Ou seja, exigência, solicitação, desejo, necessidade, que um software deverá materializar (VENTURA, 2016).

* O site deve permitir aos administradores editarem informações sobre os animais cadastrados.
* Um e-mail de confirmação deve ser enviado para verificar se o usuário vai realmente comparecer na clínica.
* Um sistema de punição deve ser feito para restringir as pessoas que não compareceram no dia da castração e não possuem uma justificativa válida;
* O site tem que permitir os administradores a serem capazes de realizar uma busca no banco de dados para encontrar alguma pessoa específica, podendo usar como critério o nome, CPF ou RG;
* Deve ser feito um sistema que obriga o usuário a ler os termos de responsabilidade e orientações antes de finalizar o cadastro de castração;
* O site deve realizar o transporte de informações do banco de dados para um arquivo de Excel;
* No cadastro do usuário, deve utilizar um sistema que usa o CEP para pegar os dados do endereço e complementar de forma automática os campos relacionados;
* Uma filtragem entre os bairros deve ser feita para localizar a clínica mais próxima do usuário afiliada com o programa de castração da prefeitura para a realização da castração;
* Quando as vagas forem esgotadas, o site deverá bloquear o cadastro de novas castrações;
* O site deverá verificar quem possui benefício social ou é protetor de animais para disponibilização de vagas extras de castração.

### Não Funcionais

Os requisitos não funcionais são todos aqueles relacionados à forma como o software tornará realidade os que está sendo planejado. Ou seja, enquanto os requisitos funcionais estão focados no que será feito, os não funcionais descrevem como serão feitos (CUNHA, 2020).

* O site deve incluir as cores da prefeitura, sendo elas azul marinho e vermelho;
* A interface gráfica deverá seguir uma certa padronização do site da prefeitura;
* O site deve ser adaptável para ser acessado em dispositivos mobile;
* O site deverá ser intuitivo e fácil de usar, para que todas as pessoas possam entender como o site funciona independentemente da plataforma utilizada para realizar o cadastro;
* Ter uma página de confirmação das informações para finalizar o cadastro da castração, alertando que não será possível alterá-las posteriormente;
* Ter explicações sobre como é feito a castração, os benefícios de castrar o animal, sobre o microchip que será implantado e sobre a definição do que é porte pequeno, médio e grande;
* Colocar fotos de animais para a estilização do site;
* Realizar animações rápidas sobre a responsabilidade de ter um animal de estimação;
* Ter um sistema de segurança eficaz para prevenir o vazamento de dados sigilosos, DDoSe outros ataques de origem criminosa;
* Limitar 400 vagas para castração, com as seguintes restrições;
  + cinco vagas para protetores de animais;
  + duas vagas para quem possui benefício social;
  + uma vaga para o restante da população.

### De Dados

Os requisitos de dados são todos aqueles que devem ser recebidos e armazenados pelo sistema e que devem estar acessíveis ao cliente em qualquer momento.

#### Usuário:

* Nome;
* RG;
* CPF;
* Rua;
* Bairro;
* Número;
* CEP;
* Telefone;
* Celular;
* E-mail.

#### Animal:

* RGA;
* Sexo;
* Se é Canino ou Felino;
* Nome do animal;
* Cor;
* Raça;
* Porte pequeno, médio ou grande;
* Pelagem curta média ou alta;
* Idade.

### De Segurança

Os requisitos de segurança de software são o conjunto de necessidades de segurança que o software deve atender, sendo tais necessidades influenciadas fortemente pela política de segurança da organização, compreendendo aspectos funcionais e não-funcionais. (Fabrício Braz, 2012).

* Senha de no mínimo 8 caracteres no máximo 16;
* Captcha;
* Somente o administrador terá acesso a todos os cadastrados;
* Somente o administrador terá acesso às informações referentes aos dados do sistema;
* Não poderá ter o mesmo CPF ou RG em cadastros diferentes;
* Não poderá ter o mesmo RGA em cadastros diferentes;
* Punir o usuário caso ele não compareça a castração;
* Para a segurança do usuário verificar se a senha é fraca ou forte;
* Criptografar as senhas para enviar ao banco de dados;
* Ter controle sobre os administradores do sistema.

### De Layout

O projeto PetCastra utiliza-se das mesmas cores da prefeitura de Franco da Rocha: Azul, amarelo e vermelho. A cor azul para o usuário remete à ideia de delicadeza, estabilidade e tranquilidade. Amarelo para a clínica ajuda a ter foco e produtividade além de estimular o raciocínio e o intelecto. E vermelho para o administrador estimula a ação e é relacionada a força e poder.

#### Página principal

Ao entrar no website, o usuário conseguirá ter acesso a informações sobre o PetCastra e irá conseguir fazer login.

Figura 1: Página Inicial.

Interface gráfica do usuário, Texto, Aplicativo, Word

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página de cadastro

Na página cadastro o usuário poderá criar uma conta no sistema PetCastra de forma rápida, simples e segura.

Figura 2: Página de Cadastro.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página meus animais

Na página meus animais o usuário conseguirá cadastrar ou visualizar seus próprios animais, podendo também solicitar uma castração, editar ou excluir o animal.

Figura 3: Página Meus Animais.

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página modal de solicitação de castração

Nesta página o usuário poderá solicitar a castração daquele pet selecionado, conseguindo também colocar alguma observação, Ex: Alergias.

Figura 4: Página Modal de Solicitar Castração.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página principal do usuário clínica

Após fazer login como usuário clínica, a clínica conseguirá consultar castrações ou solicitações.

Figura 5: Página Principal Clínica.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página de consultar castrações do usuário clínica

Após clicar em consultar castrações, o usuário clínica irá parar nesta página onde ele obterá acesso a informações sobre possíveis castrações que ela terá que realizar, com data e horário.

Figura 6: Página de Consultar Castrações - Clínica.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página modal atualizar castração do usuário clínica

Após clicar no botão atualizar a clínica poderá editar a situação da castração.

Figura 7: Página Modal Editar Castração - Clínica.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página principal do administrador

Após fazer login como administrador, ele obterá acesso a cadastros de clínicas ou raças, a consultas de usuários, clínicas e castrações e acesso as solicitações.

Figura 8: Página Principal - ADM.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página de solicitações do administrador

Nesta página de solicitação, o administrador conseguirá ter acesso a todas as solicitações de castração feita por usuários, podendo assim escolher a clínica que realizará aquela castração.

Figura 9: Página Solicitações - ADM.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Página do administrador para consultar usuários

Na página consultar usuários, o administrador poderá realizar consultas em todos os usuários ou apenas um usuário específico cadastrado no sistema PetCastra, além de conseguir exportar o banco de dados para CSV ou Excel.

Figura 10: Página para Consultar Usuário - ADM.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

## ESCOPO

O escopo é o objetivo que se pretende atingir, é sinônimo de fim, propósito, desígnio ou o que vai atingir uma determinada operação e dentro de quais recursos de sistema podem ser utilizados. (Significados,2021).

Declaração de escopo do projeto

### Identificação

Projeto: Projeto Castra Pet

Equipe: Mylena, Miguel, Sophia, Matheus, Ticyanne, Nicolas e Ketlyn

### Escopo do produto

O produto tem como ideia principal facilitar no cadastro de tutores e animais, e agendamento de castração. Como também, conscientizar as pessoas sobre a importância da castração e informatizar a população sobre o programa gratuito da prefeitura.

### Requisitos de alto nível

* O site deve permitir os administradores editarem informações sobre os animais e donos cadastrados;
* O site tem que permitir os administradores a serem capazes de realizar uma busca no banco de dados;
* O site deverá verificar quem possui benefício social (NIS) ou é protetor de animais para disponibilização de vagas extras de castração;
* Quando as vagas estiverem esgotadas, o site deverá bloquear o cadastro de novas castrações.

### Entregas

* Plataforma de cadastramento e agendamento;
* Plataforma de informatização.

### Critérios de aceitação:

* O software deverá facilitar o cadastramento e agendamento;
* Os administradores devem ter facilidade no controle e organização do banco de dados.

### Não-escopo do projeto:

* Verificar o compartilhamento de informações do banco de dados de outro sistema de proteção social (CRAS).
* Desenvolver além do acordado entre ambas as partes;
* Manutenção pós entrega.

### Premissas:

* Adesão dos interessados ao Projeto Castra Pet;
* Excelência no servidor disponibilizado pelo cliente;
* Falta de comunicação entre os órgãos participantes do projeto;
* Os órgãos participantes que não colaborarem com o projeto, de forma que atrapalhem na execução dele.

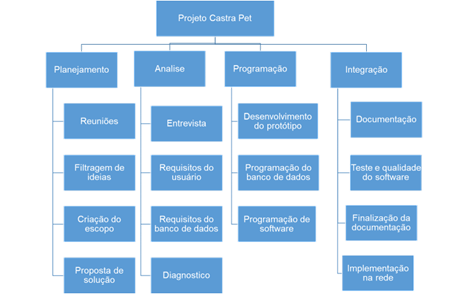
### Restrições:

* Orçamento limitado;
* Disponibilidade de acesso a dados;
* Projeto acadêmico, e com visibilidade para os desenvolvedores.
* Prazo estimado: 4 meses.

## ESTRUTURA ANALÍTICA DO PROJETO (EAP)

A EAP (Estrutura Analítica do Projeto), do inglês Work Breakdown Structure (WBS), é uma subdivisão hierárquica do trabalho do projeto em partes menores, mais facilmente gerenciáveis. Seu objetivo primário é organizar o que deve ser feito para produzir as entregas do projeto, ela garante ao gerente de projetos a visibilidade das principais entregas, facilitando o controle de tempo e de custo. (Andreia Silva Justo, 2018).

Figura 11: Estrutura Analítica do Projeto (EAP).



Fonte: Projeto PetCastra.

## CICLO DE VIDA DO SOFTWARE

O ciclo de vida é uma estrutura que contém um processo reconhecido pela indústria como uma série de atividades, etapas e tarefas que são necessárias para o desenvolvimento do software, contendo em si a vida do sistema, desde a definição de seus requisitos até o seu uso.

A definição dessas etapas e atividades possibilitam prover marcos e pontos de controle para avaliação da qualidade e gerência do projeto (Park 91).

O modelo de ciclo de vida é o primeiro a ser escolhido no processo de criação do software. A partir desta escolha é possível definir com facilidade as opções mais adequadas às necessidades do cliente, até quando e como o usuário receberá a primeira versão operacional do sistema.

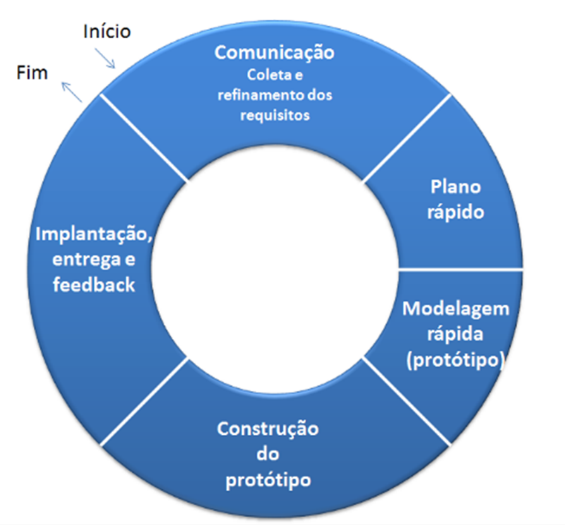
Processo de software é um conjunto de atividades que definem o desenvolvimento de um sistema computacional. As fases seguem uma maneira sequencial para facilitar no processo de criação, sendo elas: definição de requisitos, análise, projeto, desenvolvimento, teste e implantação, em cada uma delas são definidas, além das suas atividades, as funções e responsabilidades de cada membro da equipe, e como produto resultante, os artefatos.

Não existe um modelo ideal, o perfil é montado de acordo com as necessidades e desejos do cliente, porém a escolha do modelo adotado vai definir como o projeto será desenvolvido, aprimorado e finalizado, por isso tamanha importância nesse processo. (Devmedia.com, 2011).

Neste projeto será usado o modelo prototipagem. O paradigma de prototipagem começa com a definição dos requisitos. Um projeto rápido é realizado e concentra-se na representação daqueles aspectos que ficarão visíveis ao cliente. O protótipo é criado, avaliado e é ajustado para satisfazer as necessidades do cliente. (Diegomacedo.com, 2012).

Na imagem a seguir, há um modelo de prototipagem. Nesta imagem contém cada passo necessário e importante para a realização do projeto até a finalização e utilização do software.

Figura 12: Modelo de Prototipagem.



Fonte: Projeto PetCastra.

Este modelo foi escolhido pelo grupo, porque é o que mais se encaixa nos requisitos necessários e na ordem em que será realizada o projeto.

## MATRIZ DE ATIVIDADES E RESPONSABILIDADES

Matriz de responsabilidade é uma ferramenta que contribui para que os membros de uma equipe visualizem de forma mais clara os seus papéis e responsabilidades dentro de um determinado projeto. A ideia é evitar possíveis falhas na distribuição das tarefas ou dúvidas sobre a quem pertence cada atividade. Problemas como esses podem não só comprometer a gestão de pessoal, mas também impactar negativamente no desenvolvimento e na entrega dos resultados. (Silva, 2021).

Tabela 1: Matriz de Responsabilidade.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Miguel | Mylena | Ticyanne | Nicolas | Sophia | Matheus | Ketlyn |
| **Programação** | A/R | I/C | I | A/R | I | I | I/C |
| **Revisão** | R | R | R | R | R | R | R |
| **Pesquisas** | A/R | R | R | R | R | R | R |
| **Documentação** | R | R | A/R | R | A/R | R | R |
| **Entrevistas** | A/R | R | I/C | R | R | R | I/C |
| **Banco de dados** | C | A/R | I | C | R | I | R |
| **Diagramas** | R | R | I | R | R | I | I |
| **Design** | R | I | I | I | I | I | A/R |
| **Segurança** | R | C | C | R | C | A/R | C |

|  |
| --- |
| **Legenda:** |

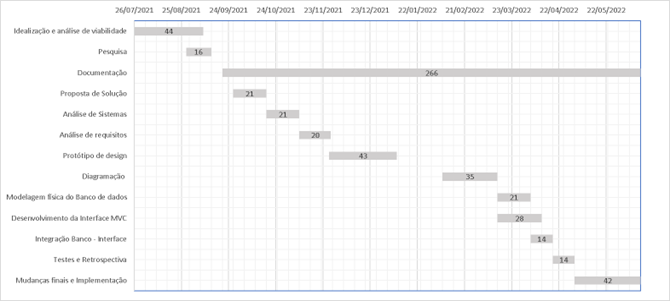
|  |  |  |  |
| --- | --- | --- | --- |
| **R**  **Responsável** | **A**  **Autoridade** | **C**  **Consultado** | **I**  **Informado** |
| Quem é designado para trabalhar nessa atividade. | Quem tem autoridade para tomar decisão. | Quem deve ser consultado e participar da decisão da atividade no momento que for executada. | Quem deve receber a informação de que uma atividade foi executada. |

Fonte: Projeto PetCastra.

## GRÁFICO DE GANTT

O gráfico de Gantt ou diagrama de Gantt é uma ferramenta visual utilizada para controlar e gerenciar o cronograma de atividades de um projeto. Com ele, é possível listar tudo que precisa ser feito para colocar o projeto em prática, dividir em atividades e estimar o tempo necessário para executá-las. (Espinha, 2021).

Gráfico 13: Gráfico de Gantt.



Fonte: Projeto PetCastra.

## DIAGRAMAS UML

Diagramas UML descrevem o limite, a estrutura e o comportamento do sistema e os objetos nele contidos. A Linguagem de modelagem unificada (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos. (Lucidchart, 2021).

### Caso de Uso

Um diagrama de caso de uso pode resumir os detalhes dos usuários do seu sistema (também conhecidos como atores) e suas interações com o sistema. Um diagrama de caso de uso eficaz pode ajudar sua equipe a discutir e representar: metas que seu sistema ou aplicativo ajuda essas entidades a atingir; o escopo do seu sistema. (Lucidchart, 2021).

Figura 13: Diagrama de Caso de Uso.

Diagrama, Desenho técnico, Esquemático

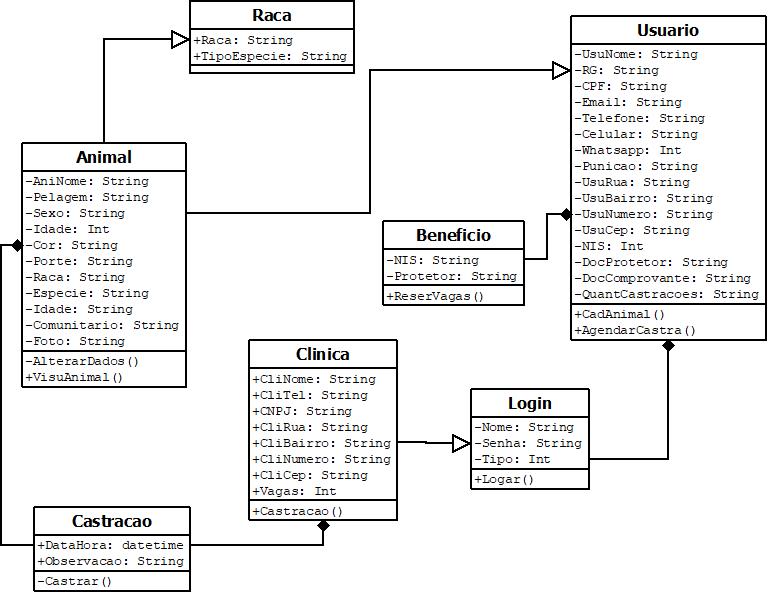
Descrição gerada automaticamente

Fonte: Projeto PetCastra.

### Classes

Diagramas de classes estão entre os tipos mais úteis de diagramas UML pois mapeiam de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos, operações e relações entre objetos. (Douglas, 2019).

Figura 14: Diagrama de Classes.



Fonte: Projeto PetCastra.

### Atividades

Os diagramas de atividades ajudam as pessoas dos lados dos negócios e do desenvolvimento de uma organização a se reunirem para compreender o mesmo processo e comportamento. É considerado um diagrama de comportamento porque descreve o que deve acontecer no sistema que está sendo modelado. (Lucidchart, 2021).

Figura 15: Diagrama de Atividades.

Diagrama

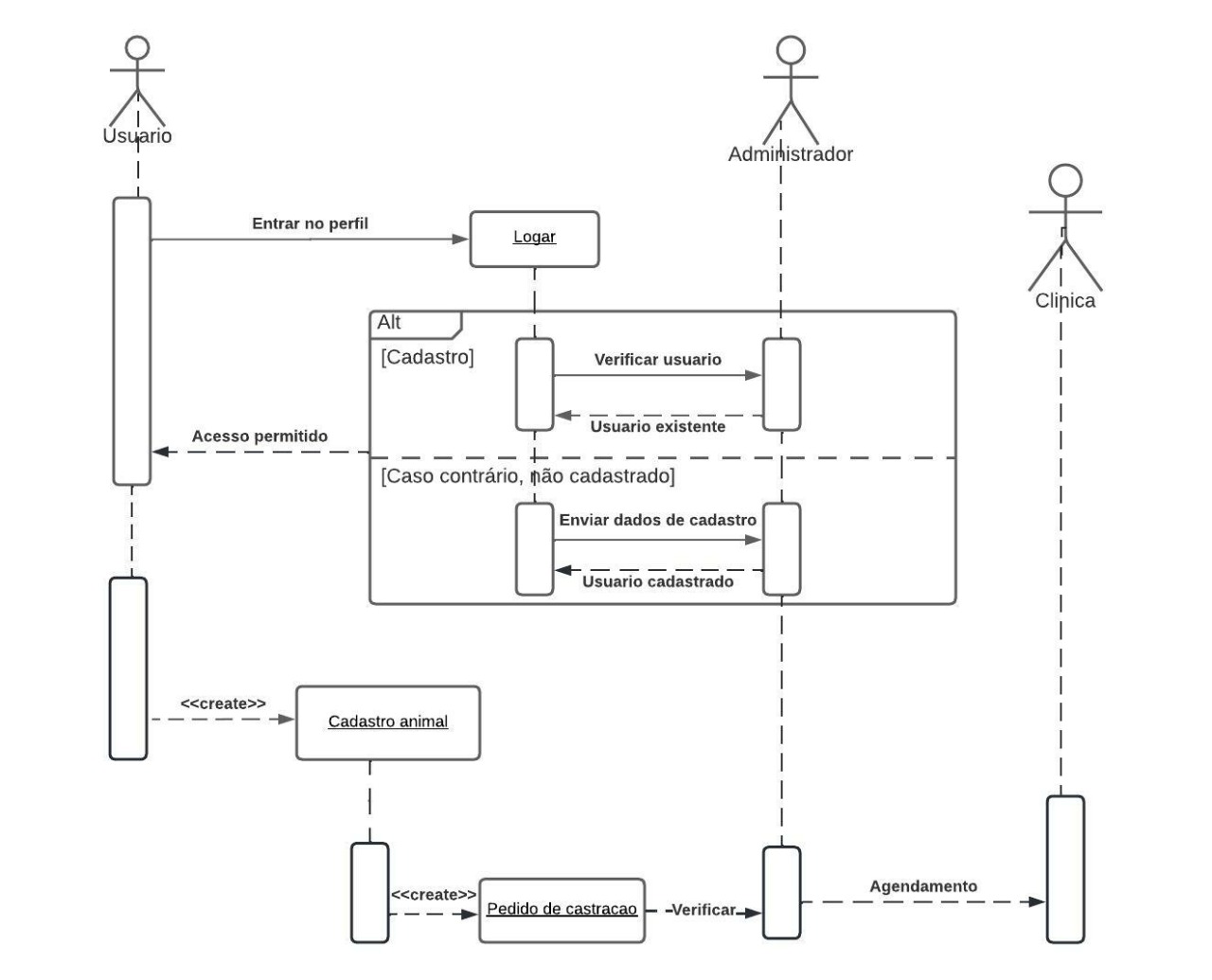
Descrição gerada automaticamente

Fonte: Projeto PetCastra.

### Sequência

Um diagrama de sequência é uma espécie de diagrama de interação, pois descreve como, e em qual ordem, um grupo de objetos trabalha em conjunto. (Lucidchart, 2021).

Figura 16: Diagrama de Sequência.



Fonte: Projeto PetCastra.

### Navegação

É um diagrama temporário não editável que mostra os resultados de uma consulta em um elemento do diagrama de contexto. (IBM, 2021).

O diagrama de Navegação foi usado com o objetivo de simplificar e facilitar o entendimento e ajudar a entender como funcionaria o acesso entre páginas.

#### Diagrama de Navegação de Telas Usuário.

Este diagrama tem como objetivo orientar o usuário, para que ele consiga navegar por o site de uma forma segura e rápida.

Figura 17: Diagrama de Navegação de Telas Usuário.

Diagrama

Descrição gerada automaticamente com confiança média

Fonte: Projeto PetCastra.

#### Diagrama de Navegação de Telas da Clínica.

Este diagrama tem como objetivo orientar o usuário clínica, a navegar por o site de uma forma segura e rápida.

Figura 18: Diagrama de Navegação de Telas da Clínica.

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

#### Diagrama de Navegação de Telas do Administrador.

Este diagrama tem como objetivo orientar o administrador a navegar e administrar o site de uma forma segura e rápida.

Figura 19: Diagrama de Navegação de Telas do Administrador.

Diagrama

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

# BANCO DE DADOS

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados (Oracle, 2022).

## MODELAGEM CONCEITUAL

Modelos conceituais de dados são usados nas fases iniciais da modelagem de dados para organizar e definir conceitos e regras baseados em requisitos de caso de uso. Eles são os menos detalhados dos três tipos de modelos de dados, mas de forma alguma isso os torna menos úteis. Na verdade, um dos principais benefícios dos modelos conceituais de dados é que eles podem ser rapidamente compreendidos e comunicados às partes interessadas fora da bolha "tecnológica".

Modelos conceituais de dados fornecem às organizações um ponto de partida que deve ser evoluído para diagramas mais ricos em contexto à medida que se movem através dos estágios dos modelos de dados. Através de análise de caso de uso, design de caso de uso e design de banco de dados, a complexidade e o nível de detalhe eventualmente atingirão o pico com modelos de dados físicos (Quest Software, 2022).

### Modelo Entidade e Relacionamento (MER)

O MER é utilizado para descrever os objetos do mundo real através de entidades, com suas propriedades que são os atributos e os seus relacionamentos (Alura, 2021) (não foi utilizado).

## MODELAGEM LÓGICA

A modelagem lógica consiste em determinar quais informações serão necessárias ao Banco. Estas informações serão divididas em Tabelas. Também serão definidos nesta fase os Campos das Tabelas, seus atributos e propriedades e ainda as Chaves Primárias e Secundárias, seu Índices e Relacionamentos (CPS Cetec).

### Diagrama Entidade e Relacionamento (DER)

O DER é utilizado para representar em forma gráfica o que foi descrito no MER - Modelo Entidade Relacionamento (Alura, 2021).

Figura 20: Diagrama Entidade e Relacionamento (DER).

Diagrama

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

## NORMALIZAÇÃO

Normalização é o processo de organização de dados em um banco de dados. Isso inclui a criação de tabelas e o estabelecimento de relações entre essas tabelas de acordo com as regras projetadas para proteger os dados e tornar o banco de dados mais flexível, eliminando a redundância e a dependência inconsistente (Microsoft, 2022).

## MODELAGEM FÍSICA

O Modelo de Entidades e Relacionamentos é um modelo abstrato cuja finalidade é descrever, de maneira conceitual, os dados a serem utilizados em um Sistema de Informações ou que pertencem a um domínio.

A modelagem física consiste na escolha de um SGBD e a criação do projeto (Modelagem Lógica) neste sistema (CPS Cetec).

SGBD é a sigla para Sistema Gerenciador de Banco de Dados – em inglês, Data Base Management System. Por meio desse recurso, operacionalizado na forma de software, sua empresa se habilita a realizar uma série de tarefas e atividades, tomando seus dados como referência (FiveActs, 2021).

# FERRAMENTAS UTILIZADAS

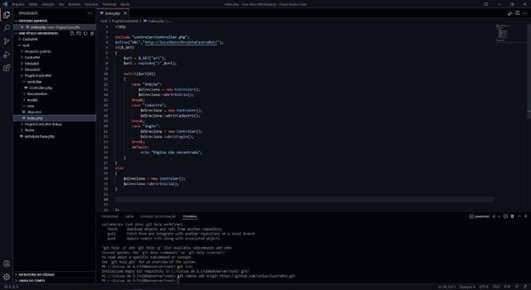
Este projeto de auxílio às prefeituras e clínicas veterinárias se utilizou de várias ferramentas que possibilitaram seu desenvolvimento. Dentre elas pode-se mencionar HTML, VS Code, PHP, CSS, Photoshop como as mais utilizadas. O projeto pôde ser concretizado e obter excelência através das diversas funcionalidades que serão expostas abaixo.

## VISUAL STUDIO CODE (VS CODE)

O Visual Studio Code é um software editor de código-fonte desenvolvido pela Microsoft para Windows e outras plataformas. Ele possui suporte para depuração, controle de versionamento, Git incorporado, realce de sintaxe, complementação inteligente de código e refatoração de código. (Hanashiro, 2021)

O Visual Studio Code foi utilizado no projeto PetCastra para construir o website.

Figura 21: Visual Studio Code.



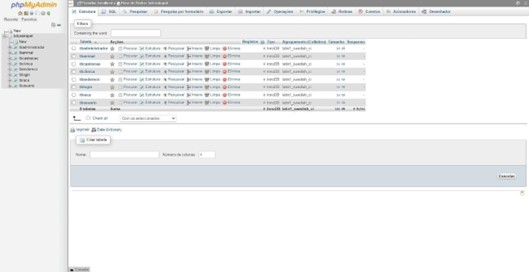
Fonte: Projeto PetCastra.

## USBWEBSERVER

USBWebserver pertence a Electronic Team Publishing, é uma combinação de software de servidor web popular: Apache, MySQL, PHP e phpMyAdmin. Com esse software é possível desenvolver e exibir seus sites PHP em qualquer lugar e a qualquer hora. (Eletronic, 2022)

O USBWebserver foi utilizado no projeto PetCastra para implementação do website.

Figura 22: USBWebserver.



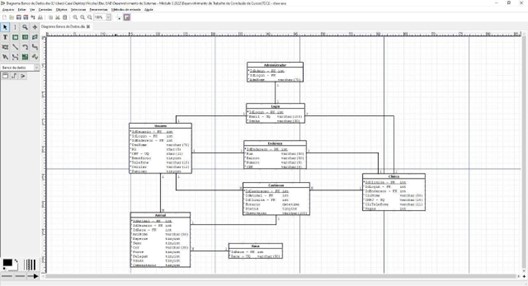
Fonte: Projeto PetCastra.

## DIA

O DIA foi desenvolvido por Alexander Larsson, e é um software de desenho de código aberto gratuito para Windows e outras plataformas. Ele suporta mais de 30 tipos de diagramas diferentes, como, fluxogramas, diagramas de rede, modelos de banco de dados, além de ter exportações de vários formatos. (zbrown,2020)

O DIA foi utilizado no projeto PetCastra para desenvolver diagramas UML.

Figura 23: DIA.



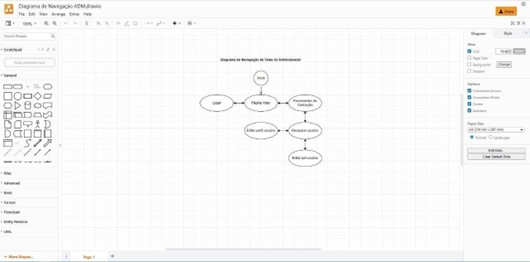
Fonte: Projeto PetCastra.

## DRAW.IO

Draw.io foi desenvolvido por Gaudenz Alder, é um editor gráfico no qual é possível desenvolver desenhos, gráficos e outros. Ele disponibiliza recursos para criação de qualquer tipo de desenho, com foco em arquitetura da informação. (Furtado, 2013)

Draw.io foi utilizado no projeto PetCastra para desenvolver diagramas UML.

Figura 24: Draw.io.



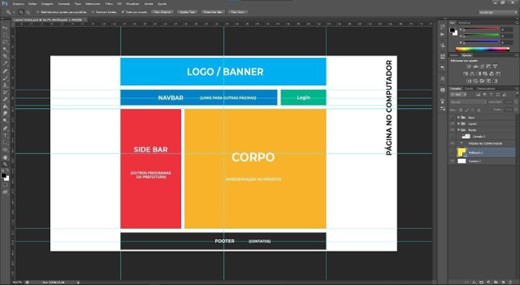
Fonte: Projeto PetCastra.

## PHOTOSHOP CS6

Photoshop CS6 pertence a Adobe Systems, é um software de uso profissional para correção de imagens. Com o programa é possível eliminar objetos, remover imperfeições, mudar cores, iluminação e entre outros ajustes. (Cruz,2019)

O Photoshop CS6 foi utilizado no projeto PetCastra para criação de Logotipos.

Figura 25: Photoshop.



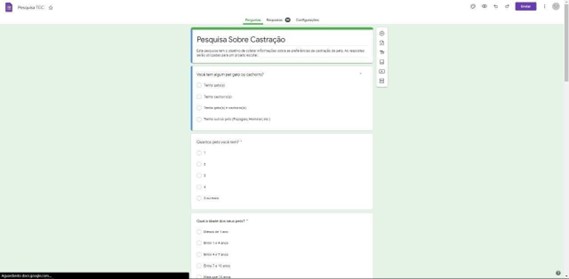
Fonte: Projeto PetCastra.

## GOOGLE FORMS

Google Forms é uma ferramenta da empresa Google com o objetivo de realizar pesquisas em formulários personalizados e questionários online em que as respostas são organizadas em gráficos e podem até ser exportadas para uma planilha. (Informática, 2021)

O Google Forms foi utilizado no projeto PetCastra para realizar pesquisas de campo.

Figura 26: Google Forms.



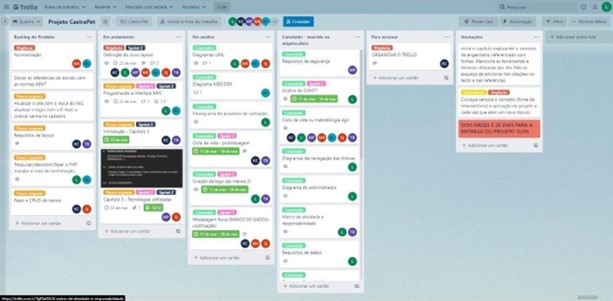
Fonte: Projeto PetCastra.

## TRELLO

Trello foi fundado pelo Fog Creek Software e em 2014 se separou e tornou-se a Trello.inc, é uma ferramenta para organização de tarefas através de cartões, sendo possível separá-los em escalas de concluído, pouco urgência e urgência. (Loubak, 2020)

O Trello foi usado no projeto PetCastra para auxiliar no desenvolvimento do projeto e na divisão de tarefas.

Figura 27: Trello.



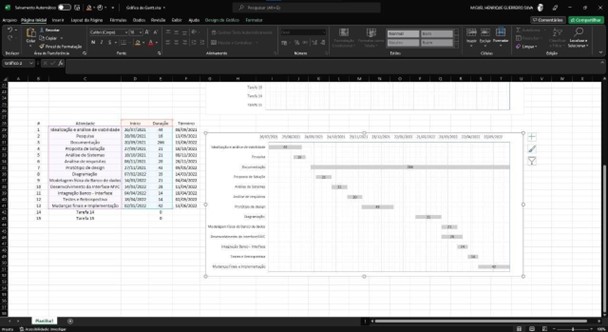
Fonte: Projeto PetCastra.

## EXCEL

Excel é um software da Microsoft, é utilizado para solucionar uma infinidade de expressões matemáticas e fórmulas prontas, também é usado para criação de tabelas de diferentes tipos de controle. (Adamin, 2010)

O Excel foi utilizado no projeto PetCastra para a criação do gráfico de Gantt e organizar as pesquisas de campo.

Figura 28: Excel.



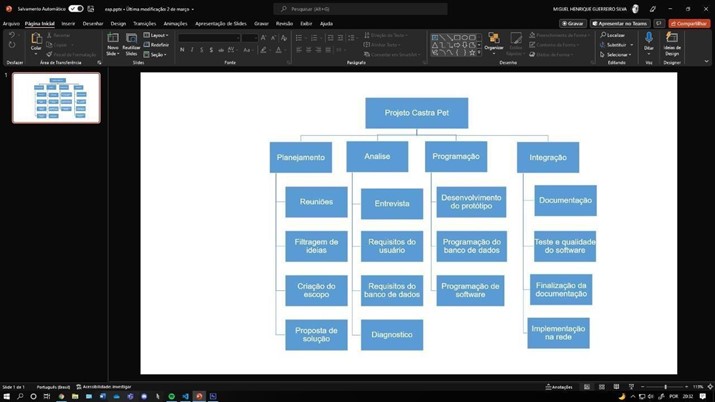
Fonte: Projeto PetCastra.

## POWERPOINT

PowerPoint é um software da Microsoft, é utilizado para criação de apresentações e criação de slides, o software possui diversas ferramentas e recursos disponíveis. (MGTEK, 2020)

O PowerPoint foi utilizado no projeto PetCastra para a criação da EAP, diagrama de atividades e ciclo de vida.

Figura 29: Powerpoint.



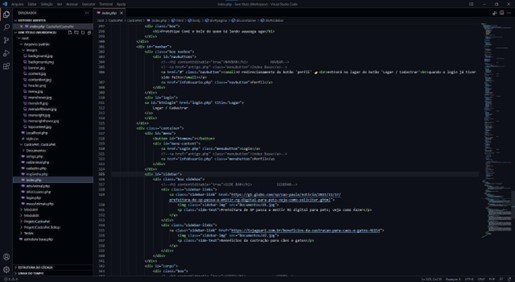
Fonte: Projeto PetCastra.

## HTML

O HTML é a abreviação de Linguagem de Marcação de Hipertexto, foi desenvolvido por W3C e WHATWG, é muito usado na construção de páginas web para adicionar novos conteúdos como por exemplo, imagens, vídeos e textos. (TOTOVS, 2020)

O HTML foi usado no projeto PetCastra para desenvolver a base do website.

Figura 30: HTML.



Fonte: Projeto PetCastra.

## CSS

CSS é a abreviação de Cascading Style Sheets, foi desenvolvida por World Wide Web Consortium, o CSS serve para formatar as páginas com mais detalhes e atributos específicos como cores, posicionamento, fontes, tamanho, animações etc. A ferramenta trabalha em conjunto com o HTML. (TOTVS, 2020)

O CSS foi usado no projeto PetCastra para dar cor e “vida” para a website.

Figura 31: CSS.



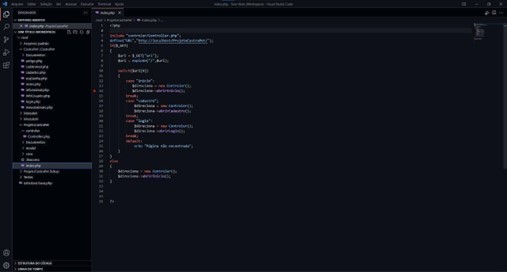
Fonte: Projeto PetCastra.

## PHP

O PHP é uma linguagem de script open source de uso geral, foi desenvolvido por Rasmus Lerdorf, e é especialmente adequada para o desenvolvimento web, também é capaz de gerar conteúdo dinâmico na World Wide Web. (php, 2001)

O PHP foi usado no projeto PetCastra para auxiliar na criação do banco de dados e incrementar ações ao website.

Figura 32: PHP.

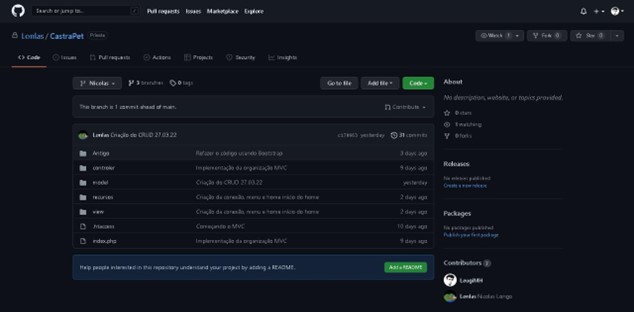


Fonte: Projeto PetCastra.

## GITHUB

O GitHub foi fundado por Tom Preston, Chris Wanstrath, Scott Chacon, P.J. Hyett. É uma plataforma de hospedagem de código fonte para controle de versão e colaboração. Permitindo que programadores possam desenvolver o mesmo código em diferentes lugares. O GitHub foi usado no projeto PetCastra para colaboração do código entre todos os desenvolvedores.

Figura 33: GitHub.

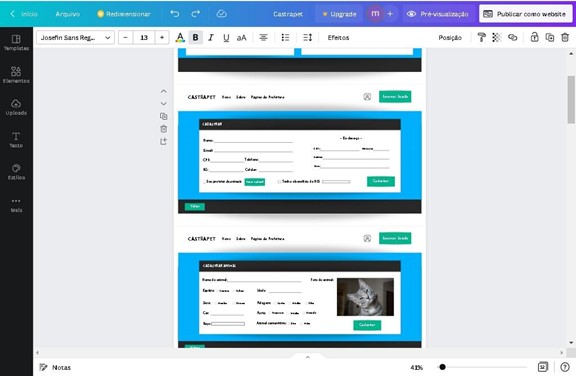


Fonte: Projeto PetCastra.

## CANVA

O Canva foi fundado por Cliff Obrecht, Melanie Perkins e Cameron Adams. O Canva foi lançado em 2013 e tem como principal função garantir que qualquer pessoa possa criar qualquer design para publicar em qualquer lugar. O Canva foi utilizado no projeto PetCastra para auxiliar na criação das telas do website.

Figura 34: Canva.



Fonte: Projeto PetCastra.

# MANUAL

O manual trata-se de um guia que ajuda a compreender o funcionamento de algo. Fornece instruções necessárias para que um utilizador possa operar determinado equipamento, objeto, software ou uma ferramenta. Normalmente o manual vem com imagens, para, não só ilustrá-lo, como ajudar na compreensão, o mesmo pode vir organizado em itens ou passos.

## MANUAL TÉCNICO

Um manual técnico é uma espécie de guia que transmite informações instrutivas para utilização de um dispositivo e correção de problemas, incluindo os procedimentos necessários para o correto funcionamento, manutenção e preservação. (pjufmg, 2022)

O manual técnico foi elaborado no intuito de orientar sobre o uso do software Castra Pet. Este manual apresenta os aspectos operacionais necessários para o técnico realizar eventuais mudanças, controle de erros e implementações ao sistema.

### Index.php

Cria uma URL global e direciona os endereço URL.

Figura 35: Index.php.

Texto

Descrição gerada automaticamente

Fonte: Visual Studio Code.

### Conexao.php

Função conectar.

Cria uma conexão direta com o banco de dados.

Figura 36: Conexão.php

Texto

Descrição gerada automaticamente

Fonte: Visual Studio Code.

### Model e-mail

Envia um e-mail de confirmação da castração, redefinição de senha e de falta de presença na data agendada.

Figura 37: Model e-mail

Texto

Descrição gerada automaticamente

Fonte: Visual Studio Code.

## MANUAL DO USUÁRIO

O manual do usuário é essencial para a aprendizagem, tanto das técnicas básicas quanto das mais avançadas de um programa ou aplicativo. O comprimento do manual irá depender unicamente do tipo de software, ele é detalhado com informações concisas e fáceis de encontrar.

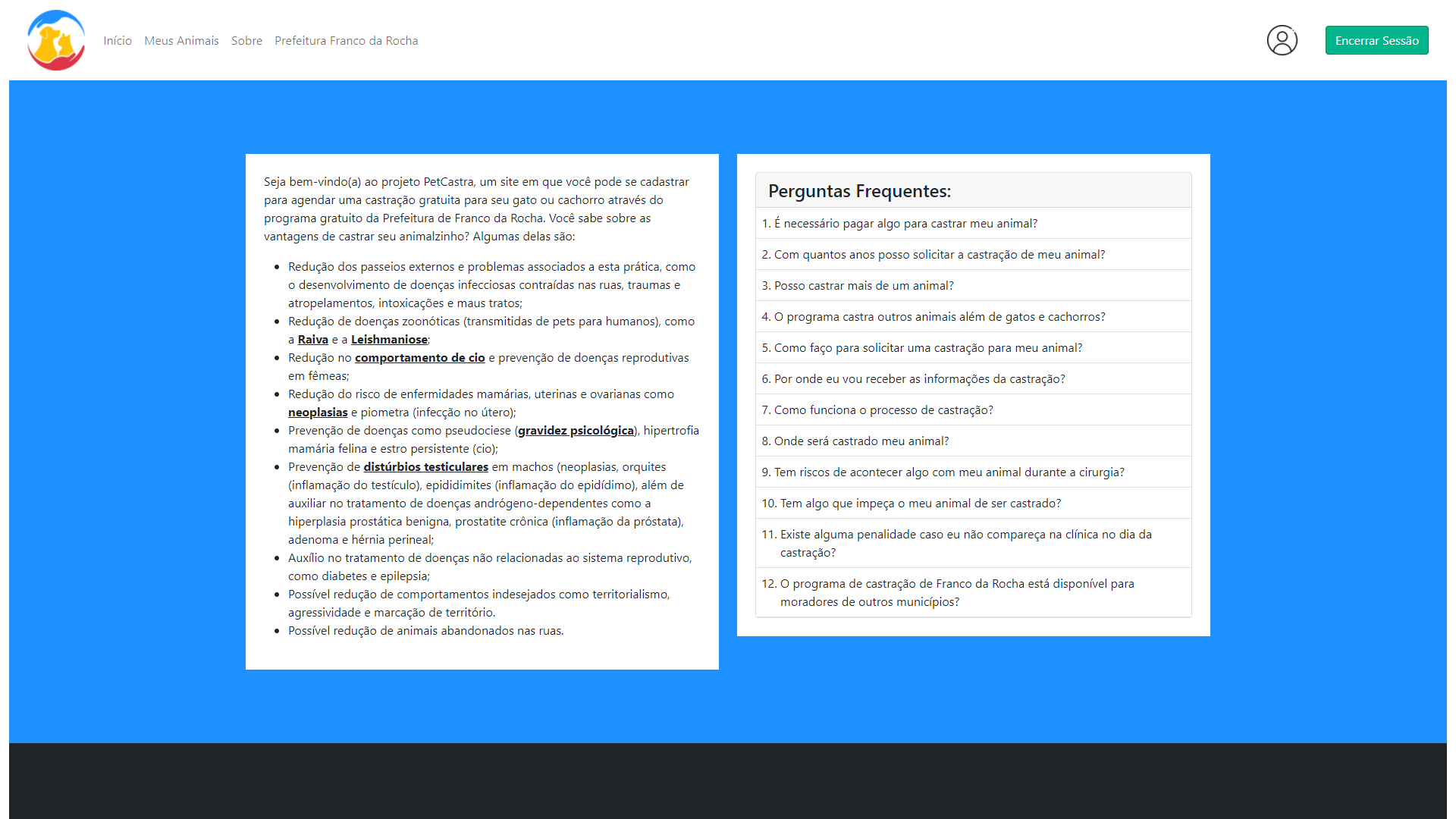
### Usuário

Descrição de como o usuário pode usar o software e quais são as funções disponíveis para ele.

#### Página principal do site

Ao iniciar o website o usuário conseguirá ter acesso a informações sobre o PetCastra e ao login.

Figura 38: Página Principal.



Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

3. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

4. **Botão Login –** Direciona o usuário para a página de login onde ele poderá acessar sua conta a partir de seu e-mail e senha.

5. **Campo para texto –** Campo deixado para inserção de texto.

6. **Campo para texto -** Campo deixado para inserção de texto.

#### Página Login

Na página login o usuário poderá fazer login no sistema ou caso não possua uma conta conseguirá criar uma.

Figura 39: Página Login.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

3. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

4. **Botão Login –** Direciona o usuário para a página de login onde ele poderá acessar sua conta a partir de seu e-mail e senha.

5. **Campo E-mail –** Campo para o usuário inserir seu e-mail.

6. **Campo Senha –** Campo para o usuário inserir sua senha.

7. **Botão Entrar –** Concede acesso ao sistema com base nas informações inseridas pelo usuário.

8. **Botão Cadastrar-se –** Direciona o usuário para a página de cadastro onde ele poderá se cadastrar através de um formulário.

9. **Botão Esqueci minha senha –** Direciona o usuário para a página de recuperação de senha.

10. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página de recuperação de senha

Na página recuperação de senha, após o usuário informar o seu e-mail o sistema PetCastra enviará informações para o usuário conseguir fazer a alteração de senha.

Figura 40: Página de Recuperação de Senha.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

3. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

4. **Botão Login –** Direciona o usuário para a página de login onde ele poderá acessar sua conta a partir de seu e-mail e senha.

5. **Campo E-mail –** Campo para o usuário inserir seu e-mail para recuperação de senha.

6. **Botão Enviar –** Envia um e-mail para o usuário com as instruções de recuperação de senha.

7. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página de Código de Confirmação

Após informar seu e-mail o sistema PetCastra enviará um código de recuperação para o usuário conseguir alterar sua senha.

Figura 41: Página de Código de Confirmação.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

3. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

4. **Botão Login –** Direciona o usuário para a página de login onde ele poderá acessar sua conta a partir de seu e-mail e senha.

5. **Campo Código de Confirmação –** Campo para o usuário inserir o código de confirmação que ele recebeu por e-mail.

6. **Botão Confirma –** Confirma as informações inseridas.

7. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página de alteração de senha

Após a confirmação do código de recuperação, o usuário poderá alterar sua própria senha.

Figura 42: Página Alterar Senha.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

3. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

4. **Botão Login –** Direciona o usuário para a página de login onde ele poderá acessar sua conta a partir de seu e-mail e senha.

5. **Campo Insira uma nova senha –** Campo para o usuário inserir sua nova senha.

6. **Campo Confirme sua senha –** Campo para o usuário confirmar sua senha.

7. **Botão Enviar –** Envia um e-mail para o usuário com as instruções de recuperação de senha.

8. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página de cadastro

Na página cadastro o usuário poderá criar uma conta no sistema PetCastra de forma rápida, simples e segura.

Figura 43: Página de Cadastro.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

3. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

4. **Botão Login –** Direciona o usuário para a página de login onde ele poderá acessar sua conta a partir de seu e-mail e senha.

5. **Campo Nome –** Campo para o usuário inserir seu nome.

6. **Campo E-mail –** Campo para o usuário inserir seu e-mail.

7. **Campo C PF –** Campo para o usuário inserir seu CPF.

8. **Campo RG –** Campo para o usuário inserir seu RG.

9. **Campo Comprovante de Residência –** Sinaliza que o usuário precisa de comprovante de residência para fazer o cadastro.

10. **Checkbox NIS –** Caso o usuário tenha o benefício NIS ele deve selecionar este campo. Após a seleção, o campo para inserir o número do NIS será liberado.

11. **Campo Telefone –** Campo para o usuário inserir seu telefone.

12. **Campo Celular -** Campo para o usuário inserir seu celular.

13. **Checkbox WhatsApp –** Caso o usuário tenha WhatsApp ele deve selecionar este campo.

14. **Botão Enviar Comprovante de Residência –** Botão que permite o usuário inserir o comprovante de residência.

15. **Campo Número NIS –** Campo para o usuário inserir o número do NIS.

16. **Campo CEP –** Campo para o usuário inserir seu CEP.

17. **Campo Bairro –** Campo para o usuário inserir seu bairro.

18. **Campo Rua -** Campo para o usuário inserir sua rua.

19. **Campo Crie uma senha –** Campo para o usuário criar sua senha.

20. **Campo Confirme sua senha –** Campo para o usuário confirmar sua senha.

21. **Campo Número -** Campo para o usuário inserir o número de sua residência.

22. **Botão Cadastrar -** Após ser pressionado todos os dados do usuário serão enviados e validados pelo banco de dados do PetCastra.

23. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página de informação

Na página informação o usuário conseguirá ter acesso a informações sobre a castração em Franco da Rocha, como funciona e quais benefícios.

Figura 44: Página de Informação.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

3. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

4. **Botão Login –** Direciona o usuário para a página de login onde ele poderá acessar sua conta a partir de seu e-mail e senha.

5. **Campo para texto –** Campo deixado para inserção de texto.

#### Página home do usuário

Na página home usuário, ele poderá ter acesso a seus próprios animais cadastrados no sistema ou até mesmo cadastrar seus animais, além de ter acesso as suas próprias informações.

Figura 45: Página Home do Usuário.

Interface gráfica do usuário, Texto, Aplicativo, Word

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Meus Animais –** Direciona o usuário para a página de seus animais cadastrados.

3. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

4. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

5. **Botão Perfil –** Direciona o usuário para seu perfil onde ele encontrará todas as suas informações.

6. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

7. **Campo para texto –** Campo deixado para inserção de texto.

#### Página meus animais

Na página meus animais o usuário conseguirá cadastrar ou visualizar seus próprios animais, podendo também solicitar uma castração, editar ou excluir o animal.

Figura 46: Página Meus Animais.

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Meus Animais –** Direciona o usuário para a página de seus animais cadastrados.

3. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

4. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

5. **Botão Perfil –** Direciona o usuário para seu perfil onde ele encontrará todas as suas informações.

6. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

7. **Foto do Animal –** Exibe a foto que o usuário fez upload na hora do cadastro do animal.

8. **Campo Nome –** Exibe o nome do animal.

9. **Campo Espécie –** Exibe a espécie do animal.

10. **Campo Sexo –** Exibe o sexo do animal.

11. **Campo Pelagem –** Exibe a pelagem do animal.

12. **Campo Porte –** Exibe o porte do animal.

13. **Campo Animal Comunitário –** Mostra se o animal possui um tutor ou é comunitário.

14. **Campo Idade –** Exibe a idade do animal.

15. **Campo Cor –** Exibe a cor do animal.

16. **Campo Raça –** Exibe a raça do animal.

17. **Solicitação em Análise –** Status da castração.

18. **Animal Castrado –** Status da castração.

19. **Botão Solicitar Castração –** Ao clicar neste botão o Pop-up Solicitar Castração aparecerá.

20. **Botão Editar –** Ao clicar neste botãoo usuário pode editar as informações do animal cadastrado.

21. **Botão Excluir –** Ao clicar neste botão o usuário pode excluir um animal cadastrado.

22. **Botão Cadastrar Animal –** Direciona o usuário para a página de cadastro de animais.

23. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página perfil do usuário

Após clicar no perfil, o usuário obterá acesso as suas próprias informações cadastradas.

Figura 47: Página Perfil do Usuário.

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Meus Animais –** Direciona o usuário para a página de seus animais cadastrados.

3. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

4. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

5. **Botão Perfil –** Direciona o usuário para seu perfil onde ele encontrará todas as suas informações.

6. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

7. **Campo Nome –** Exibe o nome do usuário.

8. **Campo E-mail –** Exibe o e-mail do usuário.

9. **Campo CPF –** Exibe o CPF do usuário.

10. **Campo RG –** Exibe o RG do usuário.

11. **Campo Telefone –** Exibe o telefone do usuário.

12. **Campo Celular –** Exibe o celular do usuário.

13. **Campo CEP –** Exibe o CEP do usuário.

14. **Campo Bairro –** Exibe o bairro do usuário.

15. **Campo Rua –** Exibe a rua do usuário.

16. **Campo Benefício NIS –** Exibe se o usuário possui algum benefício.

17. **Campo Protetor de Animais –** Exibe se o usuário é protetor de animais.

18. **Campo Número –** Exibe o número da residência do usuário.

19. **Botão Alterar minha Senha –** Ao clicar nesse botão o usuário será direcionado para uma outra página onde poderá trocar sua senha.

20. **Botão Meus Animais –** Ao clicar nesse botão o usuário será direcionado a página Meus Animais onde ele poderá ver todos os seus animais cadastrados.

21. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página com a modal de solicitação

Após clicar em solicitar castração o usuário poderá solicitar a castração daquele pet selecionado, conseguindo também colocar alguma observação, Ex: Alergias.

Figura 48: Página Modal de Solicitação.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Solicitar Castração –** Após clicar no botão Solicitar Castração este Pop-up aparecerá.

2. **Campo Observação –** Caso o usuário precise fazer alguma observação em relação ao animal.

3. **Botão Enviar Solicitação –** Ao clicar neste botão a solicitação de castração será enviada ao administrador.

4. **Botão Cancelar –** Cancela o envio da solicitação.

#### Página modal de editar o animal do usuário

Após clicar em editar animal o usuário poderá editar informações do animal selecionado, caso haja algum erro.

Figura 49: Página Modal de Editar o Animal do Usuário.

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Editar –** Após clicar no botão Editar este Pop-up aparecerá.

2. **Campo Nome do Animal –** Campo para o usuário inserir o nome do animal.

3. **Campo Espécie –** Campo para o usuário inserir a espécie do animal.

4. **Campo Sexo –** Campo para o usuário inserir o sexo do animal.

5. **Campo Cor –** Campo para o usuário inserir a cor do animal.

6. **Campo Raça –** Campo para o usuário inserir a raça do animal.

7. **Foto do Animal –** Exibe a foto que o usuário fez upload na hora do cadastro do animal.

8. **Campo Idade –** Campo para o usuário inserir a idade do animal.

9. **Campo Pelagem –** Campo para o usuário escolher a pelagem do animal.

10. **Campo Porte –** Campo para o usuário escolher o porte do animal.

11. **Campo Animal Comunitário –** Campo onde o usuário irá informar se o animal possui apenas um tutor ou é comunitário.

12. **Botão Escolher Arquivo –** Ao clicar neste botão o usuário poderá substituir a imagem por outra.

13. **Botão Editar –** Atualiza as informações do animal no banco de dados.

14. **Botão Cancelar –** Cancela a edição.

#### Página de cadastro animal

Após clicar em cadastrar animal, o usuário conseguirá cadastrar seu próprio animal no sistema PetCastra.

Figura 50: Página Cadastrar Animal.

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Início –** Direciona o usuário para a página inicial.

2. **Botão Meus Animais –** Direciona o usuário para a página de seus animais cadastrados.

3. **Botão Sobre –** Direciona o usuário para a página sobre onde ele encontrará informações sobre o programa/projeto.

4. **Botão Prefeitura de Franco da Rocha –** Direciona o usuário para o site oficial da Prefeitura de Franco da Rocha.

5. **Botão Perfil –** Direciona o usuário para seu perfil onde ele encontrará todas as suas informações.

6. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

7. **Campo Nome do Animal –** Campo para o usuário inserir o nome do animal.

8. **Campo Espécie –** Campo para o usuário inserir a espécie do animal.

9. **Campo Sexo –** Campo para o usuário inserir o sexo do animal.

10. **Campo Cor –** Campo para o usuário inserir a cor do animal.

11. **Campo Raça –** Campo para o usuário inserir a raça do animal.

12. **Campo Idade –** Campo para o usuário inserir a idade do animal.

13. **Campo Pelagem –** Campo para o usuário escolher a pelagem do animal.

14. **Campo Porte –** Campo para o usuário escolher o porte do animal.

15. **Campo Animal Comunitário –** Campo onde o usuário irá informar se o animal possui apenas um tutor ou é comunitário.

16. **Foto do Animal –** Campo para o usuário fazer upload da foto do animal.

17. **Botão Cadastrar -** Após ser pressionado todos os dados do animal serão enviados e validados pelo banco de dados do PetCastra.

18. **Botão Voltar –** Direciona o usuário para uma página anterior.

### Administrador

Descrição de como o usuário administrador pode utilizar o software e quais são as funções disponíveis para ele.

#### Página principal do administrador

Após logar como administrador, ele receberá orientações sobre privilégios que o administrador é capaz de fazer.

Figura 51: ADM - Página Principal

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Campo para texto –** Campo deixado para inserção de texto.

7. **Botão Iniciar Novo Mês** - Ao clicar neste botão o Pop-up Iniciar Novo Mês aparecerá.

#### Página principal do administrador com modal início de mês

Após Clicar em iniciar novo mês, o administrador conseguirá reiniciar todo o processo de castração gratuita.

Figura 52: ADM - Modal Iniciar novo mês.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Modal Iniciar Novo Mês** - Após clicar no botão Iniciar Novo Mês este Pop-up aparecerá.

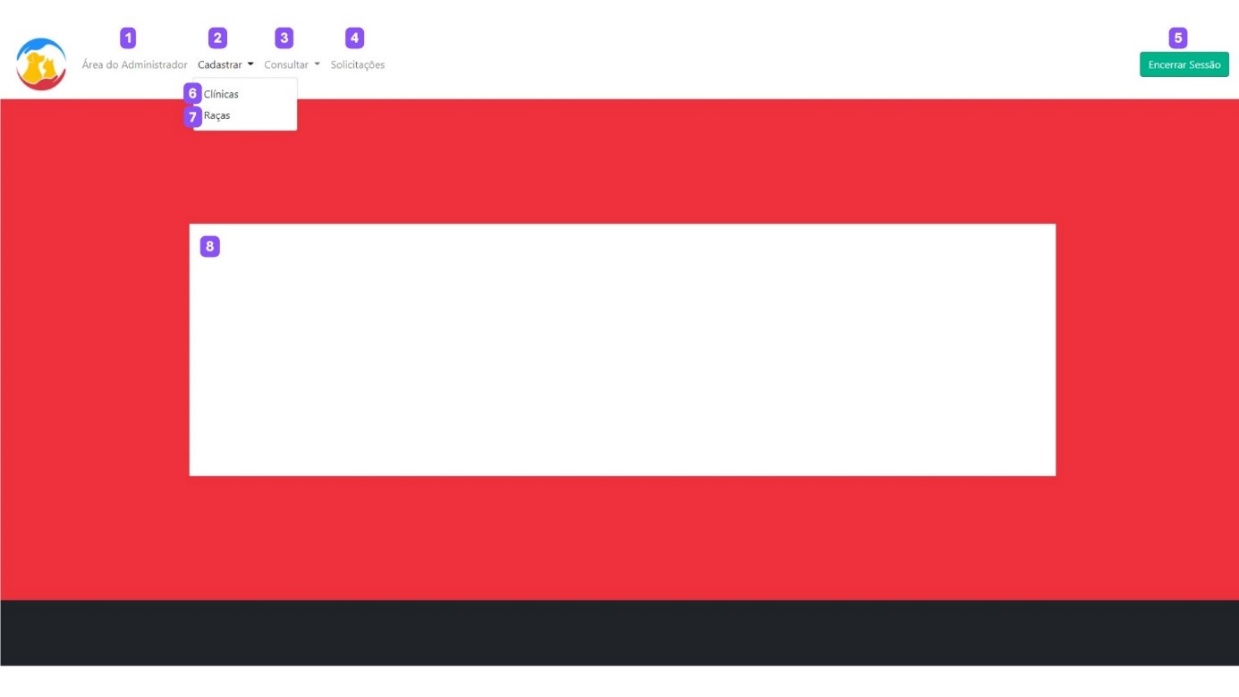
2. **Botão Iniciar** - Inicia um novo mês possibilitando novas castrações.

3. **Botão Cancelar** - Cancela a ação.

#### Página principal do administrador com dropdown cadastrar.

Após fazer login como administrador, ele obterá acesso a cadastros (clínica ou raças), consultas ou solicitações.

Figura 53: ADM - Página Principal com dropdown cadastrar.



Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Botão Clínicas –** Direciona o usuário administrador para a página de cadastro de clínicas.

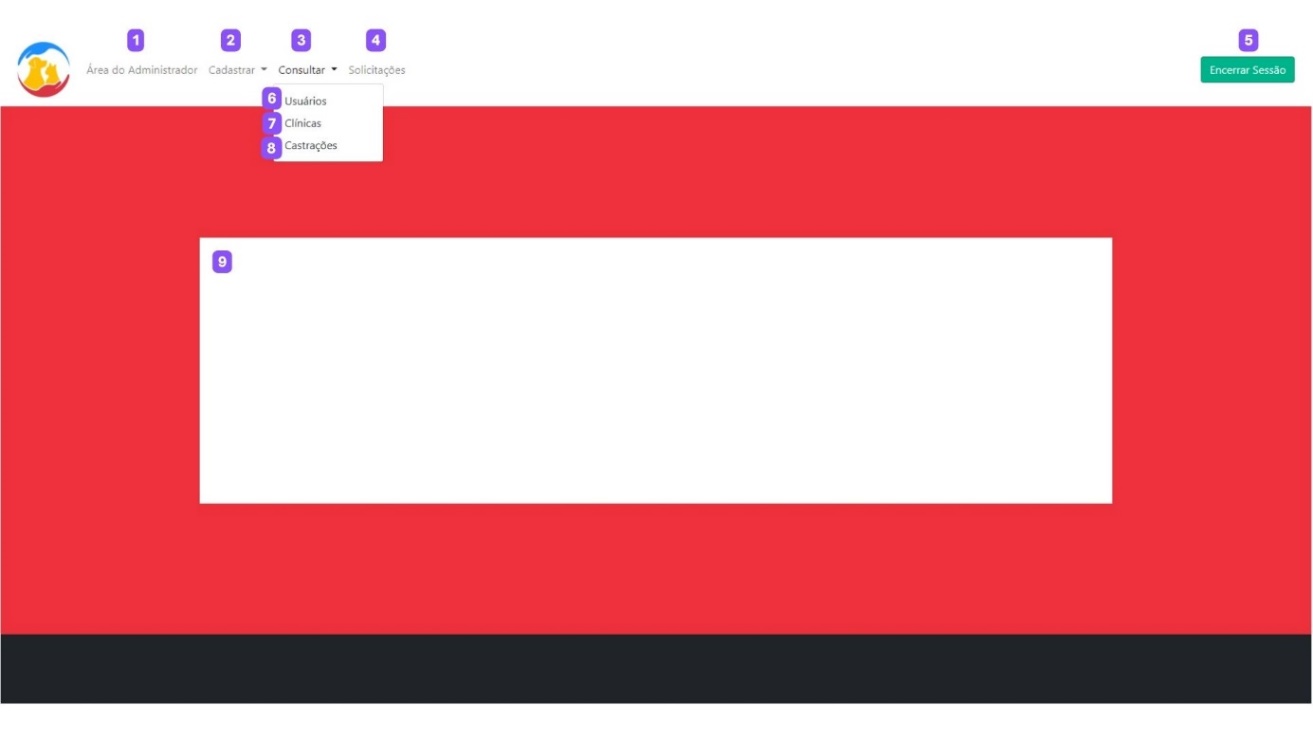
7. **Botão Raças –** Direciona o usuário administrador para a página de cadastro de raças.

8. **Campo para texto –** Campo deixado para inserção de texto.

#### Página principal do administrador com dropdown consulta

Continuando na página inicial do administrador, na aba de consultar ele conseguirá realizar consultas em usuários, clínicas e castrações cadastradas no sistema PetCastra.

Figura 54: ADM - Página principal com dropdown consulta.



Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Botão Usuários –** Direciona o usuário administrador para a página de consulta de usuários.

7. **Botão Clínicas –** Direciona o usuário administrador para a página de consulta de clínicas.

8. **Botão Castrações –** Direciona o usuário administrador para a página de consulta de castrações.

9. **Campo para texto –** Campo deixado para inserção de texto.

#### Página do administrador - cadastrar clínica

Após clicar em cadastrar clínica, o administrador conseguirá cadastrar uma clínica no sistema PetCastra.

Figura 55: ADM - Página Cadastrar Clínica.

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Campo Nome –** Campo onde o nome da clínica será inserido.

7. **Campo E-mail –** Campo onde o e-mail da clínica será inserido.

8. **Campo CNPJ –** Campo onde o CNPJ da clínica será inserido.

9. **Campo Telefone –** Campo onde o telefone da clínica será inserido.

10. **Campo Vagas –** Campo onde o número de vagas disponíveis na clínica será inserido.

11. **Campo CEP –** Campo onde o CEP da clínica será inserido.

12. **Campo Bairro –** Campo onde o bairro da clínica será inserido.

13. **Campo Rua –** Campo onde a rua será inserida.

14. **Campo Crie uma senha -** Campo para o usuário administrador criar a senha de login da clínica.

15. **Campo Confirme a senha –** Campo para o usuário administrador confirmar a senha de login da clínica.

16. **Campo Número –** Campo onde o número da clínica será inserido.

17. **Botão Cadastrar -** Após ser pressionado todos os dados da clínica serão enviados e validados pelo banco de dados do PetCastra.

18. **Botão Voltar –** Direciona o usuário administrador para uma página anterior.

#### Página do administrador - cadastrar raça do animal

Após clicar em cadastrar raça, o administrador poderá cadastrar uma raça no sistema PetCastra.

Figura 56: ADM - Página Cadastrar Raça.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Campo Espécie –** Campo onde o nome da espécie será inserido.

7. **Campo Raça –** Campo onde o nome da raça será inserido.

8. **Botão Cadastrar -** Após ser pressionado todos os dados da raça serão enviados e validados pelo banco de dados do PetCastra.

9. **Botão Voltar –** Direciona o usuário administrador para uma página anterior.

#### Página do administrador - consultar usuários

Após clicar em consultar usuários, o administrador poderá realizar consultas em todos os usuários ou apenas um usuário específico cadastrado no sistema PetCastra, além de conseguir exportar o banco de dados para CSV ou Excel.

Figura 57: ADM - Página Consultar Usuários.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Botão Visibilidade da Coluna –** Exibe as opções de visibilidade.

7. **Botão CSV –** Exporta as informações para um arquivo CSV.

8. **Botão Excel –** Exporta as informações para um arquivo Excel.

9. **Botão Imprimir –** Imprime as informações.

10. **Campo Pesquisar –** Campo para inserir informações para pesquisa.

11. **Campo ID –** Exibe o ID da castração.

12. **Campo Comprovante de Residência –** Exibe o comprovante de residência do usuário.

13. **Campo Nome –** Exibe o nome do responsável.

14. **Campo CPF –** Exibe o CPF do responsável pelo animal.

15. **Campo RG –** Exibe o RG do responsável pelo animal.

16. **Campo Benefício –** Exibe se o usuário possui algum benefício.

17. **Campo NIS –** Exibe o NIS do usuário (se ele possuir).

18. **Campo E-mail –** Exibe o e-mail do usuário.

19. **Campo Telefone –** Exibe o telefone do responsável.

20. **Campo Celular –** Exibe o celular do responsável.

21. **Campo Punição –** Exibe se o usuário possui alguma punição/restrição.

22. **Campo Animais –** Exibe o botão **Meus Animais.**

23. **Campo Ações –** Exibe ações que o administrador tem.

24. **Botão Animais –** Ao clicar neste botão o usuário administrador será direcionado a outra página onde verá os animais do usuário.

25. **Botão Editar –** Ao clicar neste botãoo usuário administrador poderá editar as informações do usuário.

26. **Botão Excluir –** Ao clicar neste botão o usuário administrador pode excluir um usuário.

27. **Campo Punido –** Exibe se o usuário está punido.

28. **Botão Anterior –** Mostra a página de registros anterior.

29. **Botão Próximo –** Mostra a página de registros seguinte.

30. **Botão Voltar –** Direciona o usuário administrador para uma página anterior.

#### Página do administrador – modal editar usuário

Após clicar em editar, o administrador poderá editar informações de usuários cadastrados no sistema caso haja erro.

Figura 58: ADM - Página Modal Editar Usuário.

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Editar –** Após clicar no botão Editar este Pop-up aparecerá.

2. **Campo Nome –** Campo para o usuário administrador inserir o nome do usuário.

3. **Campo E-mail –** Campo para o usuário administrador inserir o e-mail do usuário.

4. **Campo CPF –** Campo para o usuário administrador inserir o CPF do usuário.

5. **Campo RG –** Campo para o usuário administrador inserir o RG do usuário.

6. **Campo Punição –** Campo para o usuário administrador confirmar se o usuário possui ou não uma punição.

7. **Campo Telefone –** Campo para o usuário administrador inserir o telefone do usuário.

8. **Campo Celular –** Campo para o usuário administrador inserir o celular do usuário.

9. **Campo CEP –** Campo para o usuário administrador inserir o CEP do usuário.

10. **Campo Bairro –** Campo para o usuário administrador inserir o bairro do usuário.

11. **Campo Rua –** Campo para o usuário administrador inserir a rua do usuário.

12. **Campo Benefício NIS –** Campo para o usuário administrador confirmar se o usuário possui algum benefício.

13. **Campo Protetor de Animais –** Campo para o usuário administrador confirmar se o usuário possui algum benefício.

14. **Campo Número –** Campo para o usuário administrador inserir o número do usuário.

15. **Campo Número do NIS –** Campo para o usuário administrador inserir o número do NIS do usuário.

16. **Botão Editar –** Atualiza as informações do usuário no banco de dados.

17. **Botão Cancelar –** Cancela a edição.

#### Página do administrador – animais do tutor

Após clicar no botão verde (com desenho de pata) o administrador conseguirá visualizar animais cadastrados do usuário.

Figura 59: ADM - Página Animais do Tutor.

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Foto do Animal –** Exibe a foto que o usuário fez upload na hora do cadastro do animal.

7. **Campo Nome –** Exibe o nome do animal.

8. **Campo Espécie –** Exibe a espécie do animal.

9. **Campo Sexo –** Exibe o sexo do animal.

10. **Campo Pelagem –** Exibe a pelagem do animal.

11. **Campo Porte –** Exibe o porte do animal.

12. **Campo Animal Comunitário –** Mostra se o animal possui um tutor ou é comunitário.

13. **Campo Idade –** Exibe a idade do animal.

14. **Campo Cor –** Exibe a cor do animal.

15. **Campo Raça –** Exibe a raça do animal.

16. **Botão Editar –** Ao clicar neste botão um Pop-up aparecerá eo usuário administrador poderá editar as informações do animal cadastrado.

17. **Botão Excluir –** Ao clicar neste botão o usuário administrador pode excluir um animal cadastrado.

18. **Botão Voltar –** Direciona o usuário administrador para uma página anterior.

#### Página do administrador – modal editar animal

Após clicar em editar animal na página animais do tutor, o administrador poderá editar animais do usuário selecionado caso haja algum erro.

Figura 60: ADM - Página com Modal de Editar Animal

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Editar –** Após clicar no botão Editar este Pop-up aparecerá.

2. **Campo Nome do Animal –** Campo para o usuário administrador inserir o nome do animal.

3. **Campo Espécie –** Campo para o usuário administrador inserir a espécie do animal.

4. **Campo Sexo–** Campo para o usuário administrador inserir o sexo do animal.

5. **Campo Cor –** Campo para o usuário administrador inserir a cor do animal.

6. **Campo Raça –** Campo para o usuário administrador inserir a raça do animal.

7. **Foto do Animal –** Exibe a foto que o usuário fez upload na hora do cadastro do animal.

8. **Campo Idade –** Campo para o usuário administrador inserir a idade do animal.

9. **Campo Pelagem –** Campo para o usuário administrador escolher a pelagem do animal.

10. **Campo Porte –** Campo para o usuário administrador escolher o porte do animal.

11. **Campo Animal Comunitário –** Campo onde o usuário administrador irá informar se o animal possui apenas um tutor ou é comunitário.

12. **Botão Escolher Arquivo –** Ao clicar neste botão o usuário administrador poderá substituir a imagem por outra.

13. **Botão Editar –** Atualiza as informações do usuário no banco de dados.

14. **Botão Cancelar –** Cancela a edição.

#### Página do administrador - consultar clínicas

Após clicar em consultar clínicas, o administrador poderá realizar consultas em todas as clínicas ou apenas uma clínica específica cadastrada no sistema PetCastra, além de conseguir exportar o banco de dados para CSV ou Excel.

Figura 61: ADM - Página Consultar Clínicas.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de

solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Botão Visibilidade da Coluna –** Exibe as opções de visibilidade.

7. **Botão CSV –** Exporta as informações para um arquivo CSV.

8. **Botão Excel –** Exporta as informações para um arquivo Excel.

9. **Botão Imprimir –** Imprime as informações.

10. **Campo Pesquisar –** Campo para inserir informações para pesquisa.

11. **Campo ID –** Exibe o ID da clínica.

12. **Campo Nome –** Exibe o nome da clínica.

13. **Campo E-mail –** Exibe o e-mail da clínica.

14. **Campo CNPJ –** Exibe o CNPJ da clínica.

15. **Campo Telefone –** Exibe o telefone da clínica.

16. **Campo CEP –** Exibe o CEP da clínica.

17. **Campo Bairro -** Exibe o bairro da clínica.

18. **Campo Rua –** Exibe a rua da clínica.

19. **Campo Número –** Exibe o número da clínica.

20. **Campo Vagas –** Exibe a quantidade de vagas que a clínica possui.

21. **Campo Ativo –** Exibe se a clínica está ativa.

22. **Campo Ações –** Exibe ações que o administrador tem.

23. **Botão Editar –** Ao clicar neste botãoo usuário administrador poderá editar as informações da clínica.

24. **Botão Excluir –** Ao clicar neste botão o usuário administrador pode excluir uma clínica.

25. **Botão Anterior –** Mostra a página de registros anterior.

26. **Botão Próximo –** Mostra a página de registros seguinte.

27. **Botão Voltar –** Direciona o usuário administrador para uma página anterior.

#### Página do administrador – modal editar clínica

Após clicar no botão editar na página consultar clínica, o administrador poderá editar informações da clínica caso haja erro.

Figura 62: ADM - Página com Modal de Editar Clínica

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Editar –** Após clicar no botão Editar este Pop-up aparecerá.

2. **Campo Nome –** Campo para o usuário administrador inserir o nome da clínica.

3. **Campo E-mail –** Campo para o usuário administrador inserir o e-mail da clínica.

4. **Campo CNPJ –** Campo para o usuário administrador inserir o CNPJ do usuário.

5. **Campo Telefone –** Campo para o usuário administrador inserir o telefone do usuário.

6. **Campo Ativo –** Campo que mostra o ativo.

7. **Campo Vagas -** Campo para o usuário administrador inserir as vagas que a clínica possui.

8. **Campo Ativo –** Campo para o usuário administrador confirmar se a clínica está ativa ou não.

9. **Campo CEP –** Campo para o usuário administrador inserir o CEP do usuário.

10. **Campo Bairro –** Campo para o usuário administrador inserir o bairro do usuário.

11. **Campo Rua –** Campo para o usuário administrador inserir a rua do usuário.

12. **Campo Benefício NIS –** Campo para o usuário administrador confirmar se o usuário possui algum benefício.

13. **Campo Crie uma senha -** Campo para o usuário administrador criar a senha de login da clínica.

14. **Campo Confirme a senha –** Campo para o usuário administrador confirmar a senha de login da clínica.

15. **Botão Editar –** Atualiza as informações do usuário no banco de dados.

16. **Botão Cancelar –** Cancela a edição.

#### Página do administrador - consultar castrações

Após clicar em consultar castrações, o administrador poderá realizar consultas em todas as castrações ou apenas uma castração específica cadastrada no sistema PetCastra, além de conseguir exportar o banco de dados para CSV ou Excel.

Figura 63: ADM - Página Consultar Castrações.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Botão Visibilidade da Coluna –** Exibe as opções de visibilidade.

7. **Botão CSV –** Exporta as informações para um arquivo CSV.

8. **Botão Excel –** Exporta as informações para um arquivo Excel.

9. **Botão Imprimir –** Imprime as informações.

10. **Campo Pesquisar –** Campo para inserir informações para pesquisa.

11. **Campo ID –** Exibe o ID do animal.

12. **Foto do Animal –** Exibe a foto que o usuário fez upload na hora do cadastro do animal.

13. **Campo Animal –** Exibe o nome do animal.

14. **Campo Nome do Tutor –** Exibe o nome do responsável pelo animal.

15. **Campo CPF do Responsável –** Exibe o CPF do responsável pelo animal.

16. **Campo Clínica –** Exibe o nome da clínica designada.

17. **Campo Data –** Exibe a data que a castração foi marcada.

18. **Campo Hora –** Exibe a hora que a castração foi marcada.

19. **Campo Status –** Exibe o status da castração.

20. **Campo Observações –** Exibe as observações feitas pelo responsável.

21. **Campo Observações –** Exibe as observações feitas pela clínica.

22. **Campo Ações –** Exibe ações que o administrador tem.

23. **Botão Excluir –** Ao clicar neste botão o usuário administrador poderá excluir uma castração.

#### Página do administrador – consultar raça

Na página consultar raça, o administrador obterá informações sobre todas as raças cadastradas no sistema, além de conseguir editar ou excluir.

Figura 64: ADM - Página Consultar Raça.

Tabela

Descrição gerada automaticamente com confiança baixa

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Campo ID –** Exibe o ID da raça.

7. **Campo Nome –** Exibe o nome da raça.

8. **Campo Espécie –** Exibe a espécie da raça.

9. **Campo Ações –** Exibe ações que o administrador tem.

10. **Botão Editar –** Ao clicar neste botãoo usuário administrador poderá editar as informações da raça.

11. **Botão Excluir –** Ao clicar neste botão o usuário administrador pode excluir uma raça.

12. **Botão Voltar –** Direciona o usuário administrador para uma página anterior.

#### Página do administrador – modal editar raça

Após clicar em editar na página consultar raças, o administrador conseguirá editar a raça cadastrada caso haja erro.

Figura 65: ADM - Página com Modal de Editar Raça.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Editar –** Após clicar no botão Editar este Pop-up aparecerá.

2. **Campo Espécie –** Campo para o usuário administrador inserir a espécie.

3. **Campo Raça –** Campo para o usuário administrador inserir a raça.

4. **Botão Editar –** Atualiza as informações do usuário no banco de dados.

5. **Botão Cancelar –** Cancela a edição

#### Página do administrador - solicitações

Após clicar em solicitações, o administrador conseguirá ter acesso a todas as solicitações de castração feita por usuários, podendo assim escolher a clínica que realizará aquela castração.

Figura 66: ADM - Página de Solicitações.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Campo Exibir –** Exibe quantos resultados aparecerão por página.

7. **Campo Pesquisar –** Campo para inserir informações para pesquisa.

8. **Campo ID –** Exibe o ID da solicitação.

9. **Campo Animal –** Exibe o nome do animal.

10. **Campo Tutor –** Exibe o nome do responsável pelo animal.

11. **Botão Analisar –** Ao clicar neste botão um Pop-up aparecerá, nele será possível analisar a castração.

12. **Botão Anterior –** Mostra a página de registros anterior.

13. **Botão Próximo –** Mostra a página de registros seguinte.

14. **Botão Voltar –** Direciona o usuário administrador para uma página anterior.

#### Página do administrador – solicitação clínica

Na página solicitação, o administrador poderá decidir se o animal vai realizar ou não a castração, além de ter informações sobre o animal.

Figura 67: ADM - Página Solicitação Clínica

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área do Administrador –** Direciona o usuário administrador para a página inicial do administrador.

2. **Botão Cadastrar –** Este botão possui Dropdown. Ao clicar nele duas opções de cadastro irão aparecer: cadastro de clínicas e cadastro de raças.

3. **Botão Consultar –** Este botão possui Dropdown. Ao clicar nele três opções de consulta irão aparecer: consulta de usuários, consulta de clínicas e consulta de castrações.

4. **Botão Solicitações –** Direciona o usuário administrador para a página de solicitações, nela todas as solicitações de castração estarão listadas.

5. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

6. **Campo Número da Solicitação –** Exibe o número da solicitação.

7. **Campo Nome do Animal –** Exibe o nome do animal.

8. **Campo CPF do Responsável –** Exibe o CPF do responsável pelo animal.

9. **Campo Observação do Tutor –** Observações sobre o animal.

10. **Campo Foto –** Exibe a foto do animal.

11. **Campo Selecione uma Clínica–** Campo onde o usuário administrador irá definir a clínica da castração.

12. **Botão Confirmar –** Confirma as informações inseridas.

13. **Botão Recusar –** Recusa a castração.

14. **Botão Voltar –** Direciona o usuário para uma página anterior.

### Clínica

Descrição de como o usuário clínica pode usar o software e quais são as funções disponíveis para ele.

#### Página principal do usuário clínica

Após fazer login como clínica, a clínica conseguirá consultar castrações ou solicitações.

Figura 68: Clínica - Página Principal.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área da Clínica –** Direciona o usuário clínica para a página inicial da clínica.

2. **Botão Consultar Castrações –** Direciona o usuário clínica para a página de castrações onde todas as castrações designadas a clínica estarão listadas.

3. **Botão Lista Solicitações –** Direciona o usuário clínica para a página de solicitações, nela todas as solicitações estarão listadas.

4. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

5. **Campo para texto –** Campo deixado para inserção de texto.

#### Página do usuário clínica - consultar castrações

Após clicar em consultar castrações, a clínica obterá acesso a informações sobre possíveis castrações que ela terá que realizar, com data e horário.

Figura 69: Clínica - Página Consultar Castrações.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área da Clínica –** Direciona o usuário clínica para a página inicial da clínica.

2. **Botão Consultar Castrações –** Direciona o usuário clínica para a página de castrações onde todas as castrações designadas a clínica estarão listadas.

3. **Botão Lista Solicitações –** Direciona o usuário clínica para a página de solicitações, nela todas as solicitações estarão listadas.

4. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

5. **Campo Pesquisar –** Campo para inserir informações para pesquisa.

6. **Campo ID –** Exibe o ID da castração.

7. **Campo Foto –** Exibe a foto do animal.

8. **Campo Animal –** Exibe o nome do animal.

9. **Campo Nome do Tutor –** Exibe o nome do responsável pelo animal.

10. **Campo CPF do Responsável –** Exibe o CPF do responsável pelo animal.

11. **Campo Telefone –** Exibe o telefone do responsável.

12. **Campo Celular –** Exibe o celular do responsável.

13. **Campo Data –** Exibe a data que a castração foi marcada.

14. **Campo Hora –** Exibe a hora que a castração foi marcada.

15. **Campo Status –** Exibe o status da castração.

16. **Campo Observações –** Exibe as observações feitas pelo responsável.

17. **Campo Ações –** Exibe ações que a clínica tem.

18. **Botão Editar –** Ao clicar neste botão o usuário clínica poderá editar o status do animal.

19. **Botão Anterior –** Mostra a página de registros anterior.

20. **Botão Próximo –** Mostra a página de registros seguinte.

#### Página do usuário clínica - modal editar castração

Após clicar no botão atualizar a clínica poderá editar a situação da castração.

Figura 70: Clínica - Página Modal Editar Castrações.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Editar –** Após clicar no botão Editar este Pop-up aparecerá.

2. **Campo** **Status da Castração –** Ao clicar neste campo o usuário clínica poderá escolher o status da castração.

3. **Campo Chip –** Campo para o usuário clínica adicionar o número do chip do animal.

4. **Campo Observação –** Caso o usuário clínica precise fazer alguma observação em relação ao animal.

5. **Botão Editar –** Atualiza as informações do animal no banco de dados.

6. **Botão Cancelar –** Cancela a edição.

#### Página do usuário clínica - solicitações

Após clicar em lista solicitações, a clínica conseguirá acesso a solicitações de castração podendo assim escolher a data e o horário que ela realizará aquela castração.

Figura 71: Clínica - Página de Solicitações.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área da Clínica –** Direciona o usuário clínica para a página inicial da clínica.

2. **Botão Consultar Castrações –** Direciona o usuário clínica para a página de castrações onde todas as castrações designadas a clínica estarão listadas.

3. **Botão Lista Solicitações –** Direciona o usuário clínica para a página de solicitações, nela todas as solicitações estarão listadas.

4. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

5. **Campo Exibir –** Exibe quantos resultados aparecerão por página.

6. **Campo Pesquisar –** Campo para inserir informações para pesquisa.

7. **Campo ID –** Exibe o ID da solicitação.

8. **Campo Animal –** Exibe o nome do animal.

9. **Campo Tutor –** Exibe o nome do responsável pelo animal.

10. **Botão Analisar –** Ao clicar neste botão o usuário clínica será direcionado para outra página onde será possível analisar a castração.

11. **Botão Anterior –** Mostra a página de registros anterior.

12. **Botão Próximo –** Mostra a página de registros seguinte.

13. **Botão Voltar –** Direciona o usuário para uma página anterior.

#### Página do usuário clínica – solicitação data/hora

Na página solicitação, o administrador conseguirá escolher a data e a hora que aquele animal será castrado.

Figura 72: Clínica - Página Solicitação de Data/Hora.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Projeto PetCastra.

1. **Botão Área da Clínica –** Direciona o usuário clínica para a página inicial da clínica.

2. **Botão Consultar Castrações –** Direciona o usuário clínica para a página de castrações onde todas as castrações designadas a clínica estarão listadas.

3. **Botão Lista Solicitações –** Direciona o usuário clínica para a página de solicitações, nela todas as solicitações estarão listadas.

4. **Encerrar Sessão -** Após pressionado encerrará a sessão atual, possibilitando fazer um novo login ou cadastro.

5. **Campo Número da Solicitação –** Exibe o número da solicitação.

6. **Campo Animal –** Exibe o nome do animal.

7. **Campo CPF do Responsável –** Exibe o CPF do responsável pelo animal.

8. **Campo Observação do Tutor –** Observações sobre o animal.

9. **Campo Foto –** Exibe a foto do animal.

10. **Campo Data/Hora –** Campo onde o usuário clínica irá definir a data e horário da castração.

11. **Botão Confirmar –** Confirma as informações inseridas.

12. **Botão Voltar –** Direciona o usuário para uma página anterior.

# CONSIDERAÇÕES FINAIS

O intuito do projeto PetCastra é a facilitação na comunicação entre o programa de castração e os cidadãos por meio de um website. Esse website visa atrair mais visibilidade ao programa e informação aos cidadãos; já para os responsáveis dele, facilita e inova o gerenciamento

Para começarmos o desenvolvimento do projeto, realizamos uma pesquisa com os residentes da região. Com a pesquisa, obtivemos dados que mostravam insatisfação com o programa e com base nisso listamos seus aspectos a serem melhorados.

## Comunicação entre usuário e prefeitura.

Desde o começo do projeto percebemos que este era o grande impasse, pois a forma em que era realizada tomava muito tempo tanto dos usuários quanto dos responsáveis, visto que era feita através de ligações e agendamento presencial.

## Visibilidade.

Até o momento o programa não tinha tanta visibilidade, pois não era divulgado, com website o mesmo pode vir a ter uma maior visibilidade pelo fato de a população estar mais conectada à internet, portanto promovendo confiança aos usuários.

## Informação sobre início e fim do programa.

Pela falta de comunicação entre ambas as partes muitos usuários acabavam perdendo o período de cadastro, com o website todas as informações detalhadas sobre o programa estarão disponíveis a todo momento, principalmente sobre início e fim do agendamento.

## Punições em caso de faltas.

Em caso de ausências sem justificativa no dia da castração o usuário é bloqueado por um tempo determinado, não conseguindo solicitar uma nova data para realizar o procedimento, antes do website não havia formas de punir o usuário, sendo assim ele poderia realizar novamente o cadastro e vir a faltar novamente.

## Facilidade no transporte de informação para Excel.

De acordo com a falta de transporte de informação a equipe PetCastra desenvolveu uma facilidade para os gestores do programa, onde com apenas um clique o administrador consegue transportar as informações do banco de dados para uma tabela Excel.

# CONCLUSÃO

Esse trabalho teve como foco principal o desenvolvimento de uma aplicação web, no qual foi possível aplicar as técnicas e linguagens de programação que aprendemos no curso de desenvolvimento de sistemas.

Com a utilização da ferramenta Bootstrap foi desenvolvido um produto que além de possuir um visual agradável fornece características responsivas, ou seja, a interface se adapta dependendo do tamanho da tela do dispositivo acessado, possibilitando dessa forma a elaboração e produção de uma segunda aplicação para dispositivos móveis com sistema operacional Android.

Para o desenvolvimento deste trabalho foi necessária a criação do website, baseado no programa de castração da Prefeitura de Franco da Rocha, cujo principal objetivo do projeto é agilizar o processo do cadastro de castrações de animais domésticos, facilitar a comunicação entre os tutores com o servidor público, informatizar sobre a verificação de disponibilidade de vagas da campanha e o período em que ela durara. Analisamos o atual método usado e encontramos diversas fraquezas, portanto, chegamos à conclusão de que tal método poderia ser melhorado.

Ao realizarmos algumas pesquisas conhecemos os principais problemas que o programa de castração tem. A partir deste momento, analisamos os obstáculos que poderiam surgir no decorrer do projeto e criamos estratégias para que nossos objetivos pudessem ser alcançados e que para atingir nosso objetivo, buscamos aplicar uma metodologia focada no desenvolvimento de sistemas, e através da metodologia escolhida, pudemos aprofundar nosso conhecimento.

Para tal análise foi aplicado um formulário a população. O formulário aplicado conseguiu mostrar a situação do conhecimento das pessoas diante do programa de castração oferecido pela prefeitura. Com base na análise levantada tivemos a oportunidade de analisar as dificuldades que foram encontradas, sendo ela, o problema de insatisfação com o atual método de cadastro, o que ajudou a comprovar que o nosso projeto (PetCastra) foi realmente necessário, e como solução proposta foi desenvolvido um sistema a fim de facilitar o acesso ao cidadão que planeja fazer parte do serviço de castração, sendo eles administradores, analistas ou quaisquer leitores deste documento.

# APÊNDICE A

-- phpMyAdmin SQL Dump

-- version 4.7.5

-- https://www.phpmyadmin.net/

--

-- Host: localhost

-- Generation Time: 13-Jun-2022 às 17:54

-- Versão do servidor: 5.6.34

-- PHP Version: 7.1.11

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `bdpetcastra`

--

-- --------------------------------------------------------

--

-- Estrutura da tabela `animal`

--

CREATE TABLE `animal` (

`idanimal` int(11) NOT NULL,

`idusuario` int(11) NOT NULL,

`idraca` int(11) NOT NULL,

`aninome` varchar(50) NOT NULL,

`especie` tinyint(4) NOT NULL COMMENT '0 para Canina e 1 para Felina',

`sexo` tinyint(4) NOT NULL COMMENT '0 para Fêmea; 1 para Macho',

`cor` varchar(30) NOT NULL,

`pelagem` tinyint(4) NOT NULL COMMENT '0 pra curta; 1 média; 2 pra alta;',

`porte` tinyint(4) NOT NULL COMMENT '0 pra pequeno; 1 pra médio; 2 pra grande;',

`idade` tinyint(4) NOT NULL,

`comunitario` tinyint(4) NOT NULL COMMENT '0 pra não; 1 pra sim;',

`foto` varchar(50) DEFAULT NULL,

`codchip` char(15) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estrutura da tabela `castracao`

--

CREATE TABLE `castracao` (

`idcastracao` int(11) NOT NULL,

`idanimal` int(11) NOT NULL,

`idclinica` int(11) DEFAULT NULL,

`horario` datetime DEFAULT NULL,

`status` tinyint(4) NOT NULL COMMENT '0 - Solicitação em análise; 1 - Solicitação aprovada; 2 - Animal Castrado; 3 - Solicitação Reprovada; 4 - Tutor não compareceu; 5 - Solicitação Cancelada; 6 - Solicitação Reagendada; 7 - Animal foi a óbito',

`observacao` varchar(100) DEFAULT NULL,

`obsclinica` varchar(100) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estrutura da tabela `clinica`

--

CREATE TABLE `clinica` (

`idclinica` int(11) NOT NULL,

`idlogin` int(11) NOT NULL,

`cnpj` varchar(14) NOT NULL,

`clitelefone` varchar(11) NOT NULL,

`vagas` int(11) NOT NULL,

`clirua` varchar(50) NOT NULL,

`clibairro` varchar(50) NOT NULL,

`clinumero` varchar(5) NOT NULL,

`clicep` varchar(8) NOT NULL,

`ativo` tinyint(4) NOT NULL COMMENT '0 - Clínica não ativa; 1 - Clínica ativa'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estrutura da tabela `login`

--

CREATE TABLE `login` (

`idlogin` int(11) NOT NULL,

`nome` varchar(70) NOT NULL,

`email` varchar(100) NOT NULL,

`senha` varchar(60) NOT NULL,

`nivelacesso` tinyint(4) NOT NULL COMMENT '0 pra Usuário; 1 pra clínica; 2 pra adm;',

`codsenha` char(6) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Extraindo dados da tabela `login`

--

INSERT INTO `login` (`idlogin`, `nome`, `email`, `senha`, `nivelacesso`, `codsenha`) VALUES

(1, 'Administrador', 'adm@adm.com', '$2y$10$h8Sde8BNocqPM4Oo4/MWtuI2Zjmx7H7oAetg8Q5kAnQj3pQAJMl0W', 2, NULL);

-- --------------------------------------------------------

--

-- Estrutura da tabela `raca`

--

CREATE TABLE `raca` (

`idraca` int(11) NOT NULL,

`raca` varchar(30) CHARACTER SET latin1 NOT NULL,

`tipoespecie` tinyint(4) NOT NULL COMMENT '0 pra cachorro; 1 pra gato; 2 para os dois'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Extraindo dados da tabela `raca`

--

INSERT INTO `raca` (`idraca`, `raca`, `tipoespecie`) VALUES

(1, 'Pastor AlemÃ£o', 0),

(2, 'Buldogue FrancÃªs', 0),

(4, 'SiamÃªs', 0),

(5, 'Persa', 1),

(6, 'Sphynx', 1),

(7, 'S.R.D - Sem RaÃ§a Definida', 2),

(8, 'Chow-chow', 0);

-- --------------------------------------------------------

--

-- Estrutura da tabela `usuario`

--

CREATE TABLE `usuario` (

`idusuario` int(11) NOT NULL,

`idlogin` int(11) NOT NULL,

`rg` char(9) NOT NULL,

`cpf` char(11) NOT NULL,

`beneficio` tinyint(4) NOT NULL COMMENT '0 - Sem benefício; 1 - Benefício Soical; 2 - Protetor de Animais; 3 - Em análise',

`telefone` varchar(10) DEFAULT NULL,

`celular` varchar(11) NOT NULL,

`whatsapp` tinyint(4) NOT NULL COMMENT '0 - Celular não é whatsapp; 1 - Celular é whatsapp',

`punicao` tinyint(4) NOT NULL,

`usurua` varchar(50) NOT NULL,

`usubairro` varchar(50) NOT NULL,

`usunumero` varchar(5) NOT NULL,

`usucep` varchar(8) NOT NULL,

`nis` char(11) DEFAULT NULL,

`doccomprovante` varchar(255) DEFAULT NULL,

`quantcastracoes` tinyint(4) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `animal`

--

ALTER TABLE `animal`

ADD PRIMARY KEY (`idanimal`),

ADD KEY `idusuario` (`idusuario`),

ADD KEY `idraca` (`idraca`);

--

-- Indexes for table `castracao`

--

ALTER TABLE `castracao`

ADD PRIMARY KEY (`idcastracao`),

ADD UNIQUE KEY `horario` (`horario`),

ADD KEY `idanimal` (`idanimal`,`idclinica`),

ADD KEY `fkidclinica\_castracao` (`idclinica`);

--

-- Indexes for table `clinica`

--

ALTER TABLE `clinica`

ADD PRIMARY KEY (`idclinica`),

ADD UNIQUE KEY `cnpj` (`cnpj`),

ADD KEY `idlogin` (`idlogin`);

--

-- Indexes for table `login`

--

ALTER TABLE `login`

ADD PRIMARY KEY (`idlogin`),

ADD UNIQUE KEY `email` (`email`);

--

-- Indexes for table `raca`

--

ALTER TABLE `raca`

ADD PRIMARY KEY (`idraca`),

ADD UNIQUE KEY `Raca` (`raca`);

--

-- Indexes for table `usuario`

--

ALTER TABLE `usuario`

ADD PRIMARY KEY (`idusuario`),

ADD UNIQUE KEY `cpf` (`cpf`),

ADD UNIQUE KEY `nis` (`nis`),

ADD KEY `idlogin` (`idlogin`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `animal`

--

ALTER TABLE `animal`

MODIFY `idanimal` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=28;

--

-- AUTO\_INCREMENT for table `castracao`

--

ALTER TABLE `castracao`

MODIFY `idcastracao` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;

--

-- AUTO\_INCREMENT for table `clinica`

--

ALTER TABLE `clinica`

MODIFY `idclinica` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT for table `login`

--

ALTER TABLE `login`

MODIFY `idlogin` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=34;

--

-- AUTO\_INCREMENT for table `raca`

--

ALTER TABLE `raca`

MODIFY `idraca` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;

--

-- AUTO\_INCREMENT for table `usuario`

--

ALTER TABLE `usuario`

MODIFY `idusuario` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=23;

--

-- Constraints for dumped tables

--

--

-- Limitadores para a tabela `animal`

--

ALTER TABLE `animal`

ADD CONSTRAINT `fkidraca` FOREIGN KEY (`idraca`) REFERENCES `raca` (`idraca`),

ADD CONSTRAINT `fkidusuario` FOREIGN KEY (`idusuario`) REFERENCES `usuario` (`idusuario`);

--

-- Limitadores para a tabela `castracao`

--

ALTER TABLE `castracao`

ADD CONSTRAINT `fkidanimal\_castracao` FOREIGN KEY (`idanimal`) REFERENCES `animal` (`idanimal`),

ADD CONSTRAINT `fkidclinica\_castracao` FOREIGN KEY (`idclinica`) REFERENCES `clinica` (`idclinica`);

--

-- Limitadores para a tabela `clinica`

--

ALTER TABLE `clinica`

ADD CONSTRAINT `fkidlogin\_clinica` FOREIGN KEY (`idlogin`) REFERENCES `login` (`idlogin`);

--

-- Limitadores para a tabela `usuario`

--

ALTER TABLE `usuario`

ADD CONSTRAINT `fkidlogin\_usuario` FOREIGN KEY (`idlogin`) REFERENCES `login` (`idlogin`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

# APÊNDICE B

**AnimalController**

<?php

include\_once "model/Animal.php";

include\_once "model/Raca.php";

class AnimalController

{

    function carregarRaca($id){

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        $raca = new Raca();

        $raca->tipoespecie = $id;

        $dadosRacaAnimal = $raca->consultarRaca();

        $options = "";

        foreach ($dadosRacaAnimal as $value){

            $options .= "<option value='$value->idraca'>$value->raca</option>";

        }

        echo $options;

    }

    function cadastrarAnimal()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 0) {

            $animal = new Animal();

            $animal->idusuario =   $\_SESSION["dadosUsuario"]->idusuario;

            $animal->especie =     $\_POST["tipoEspecie"];

            $animal->idraca =      $\_POST["racas"];

            $animal->aninome =     $\_POST["txtNome"];

            $animal->sexo =        $\_POST["slcSexo"];

            $animal->porte =       $\_POST["slcPorte"];

            $animal->cor =         $\_POST["txtCor"];

            $animal->pelagem =     $\_POST["slcPelagem"];

            $animal->idade =       $\_POST["numIdade"];

            $animal->comunitario = $\_POST["slcComunitario"];

            /\* UPLOAD IMAGEM \*/

            if($\_FILES["imgAnimal"]["error"] == 0)

            {

                $nomeArquivo = $\_FILES["imgAnimal"]["name"];    //Nome do arquivo

                $nomeTemp =    $\_FILES["imgAnimal"]["tmp\_name"];   //nome temporário

                //pegar a extensão do arquivo

                $info = new SplFileInfo($nomeArquivo);

                $extensao = $info->getExtension();

                //gerar novo nome

                $novoNome = md5(microtime()) . ".$extensao";

                $pastaDestino = "recursos/img/Animais/$novoNome";   //pasta destino

                move\_uploaded\_file($nomeTemp, $pastaDestino);       //mover o arquivo

                //enviando para o banco de dados

                $animal->foto = $novoNome;

            }

            $animal->cadastrar();

            header("Location:".URL."meus-animais");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function atualizarAnimal()

    {

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 0 || $\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $animal = new Animal();

            $animal->idanimal =    $\_POST["idanimal"];

            $animal->aninome =     $\_POST["txtNome"];

            $animal->especie =     $\_POST["tipoEspecie"];

            $animal->idade =       $\_POST["numIdade"];

            $animal->sexo =        $\_POST["slcSexo"];

            $animal->pelagem =     $\_POST["slcPelagem"];

            $animal->cor =         $\_POST["txtCor"];

            $animal->porte =       $\_POST["slcPorte"];

            $animal->idraca =      $\_POST["racas"];

            $animal->comunitario = $\_POST["slcComunitario"];

            /\* UPLOAD IMAGEM \*/

            $infoanimal = new Animal();

            $infoanimal->idanimal = $\_POST["idanimal"];

            $dadosAnimal = $infoanimal->retornar();

            if($\_FILES["foto"]["error"] == 0)

            {

                if(empty($dadosAnimal->foto))

                {

                    $nomeArquivo = $\_FILES["foto"]["name"];       //Nome do arquivo

                    $nomeTemp =    $\_FILES["foto"]["tmp\_name"];      //nome temporário

                    //pegar a extensão do arquivo

                    $info = new SplFileInfo($nomeArquivo);

                    $extensao = $info->getExtension();

                    //gerar novo nome

                    $novoNome = md5(microtime()) . ".$extensao";

                    $pastaDestino = "recursos/img/Animais/$novoNome";    //pasta destino

                    move\_uploaded\_file($nomeTemp, $pastaDestino);       //mover o arquivo

                    //enviando para o banco

                    $animal->foto = $novoNome;

                }

                else {

                    unlink("recursos/img/Animais/$dadosAnimal->foto"); //excluir o arquivo

                    $nomeArquivo = $\_FILES["foto"]["name"];       //Nome do arquivo

                    $nomeTemp =    $\_FILES["foto"]["tmp\_name"];      //nome temporário

                    //pegar a extensão do arquivo

                    $info = new SplFileInfo($nomeArquivo);

                    $extensao = $info->getExtension();

                    //gerar novo nome

                    $novoNome = md5(microtime()) . ".$extensao";

                    $pastaDestino = "recursos/img/Animais/$novoNome";    //pasta destino

                    move\_uploaded\_file($nomeTemp, $pastaDestino);       //mover o arquivo

                    //enviando para o banco

                    $animal->foto = $novoNome;

                }

            }

            else

            $animal->foto = $dadosAnimal->foto;

            $animal->atualizar();

            if($\_SESSION["dadosLogin"]->nivelacesso == 0){ header("Location:".URL."meus-animais"); }

            else{ header("Location:".URL."consulta-animais/".$\_POST['idusuario']);}

        }

    }

    function excluirAnimal($idanimal,$idusuario,$foto)

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 0 || $\_SESSION["dadosLogin"]->nivelacesso == 2) {

            try{

                $animal = new Animal();

                $animal->idanimal = $idanimal;

                //removendo imagem

                $dadosAnimal = $animal->retornar();

                if(is\_file("recursos/img/$dadosAnimal->foto")) //verificar se o arquivo existe

                {

                    unlink("recursos/img/$dadosAnimal->foto"); //excluir o arquivo

                }

                $animal->excluir();

            }

            catch(Exception $e){

                if($\_SESSION["dadosLogin"]->nivelacesso == 2)

                {

                    echo "<script>alert('Erro ao excluir o animal! Verifique se esse animal não possui um cadastro nas castrações'); window.location='".URL."consulta-animais/$idusuario'; </script>";

                }

                else

                {

                    echo "<script>alert('Erro ao excluir o animal'); window.location='".URL."meus-animais'; </script>";

                }

                return;

            }

            if(is\_file("recursos/img/Animais/$foto"))

            {

                unlink("recursos/img/Animais/$foto");

            }

            if($\_SESSION["dadosLogin"]->nivelacesso == 0){ header("Location:".URL."meus-animais"); }

            else{ header("Location:".URL."consulta-animais/$idusuario");}

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function cadastrarRaca()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            try

            {

                $raca = new Raca();

                $raca->raca = $\_POST["txtRaca"];

                $raca->tipoespecie = $\_POST["tipoEspecie"];

                $raca->cadastrar();

                echo"<script>alert('Raça cadastrada com sucesso'); window.location='".URL."cadastra-raca'; </script>";

            }

            catch(Exception $e)

            {

                echo"<script>alert('Erro: $e'); window.location='".URL."cadastra-raca'; </script>";

            }

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    //Atualizar Raça

    function atualizarRaca()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            try

            {

                $raca = new Raca();

                $raca->idraca = $\_POST["idRaca"];

                $raca->raca = $\_POST["txtRaca"];

                $raca->tipoespecie = $\_POST["tipoEspecie"];

                $raca->atualizar();

                echo"<script>alert('Raça atualizada com sucesso'); window.location='".URL."consulta-raca'; </script>";

            }

            catch(Exception $e)

            {

                echo"<script>alert('Erro: $e'); window.location='".URL."consulta-raca'; </script>";

            }

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    //Excluir Raça

    function excluirRaca($idraca)

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            try

            {

                $raca = new Raca();

                $raca->idraca = $idraca;

                $raca->excluir();

            }

            catch(Exception $e)

            {

                echo "<script>alert('Erro: $e'); window.location='".URL."consulta-raca'; </script>";

            }

            header("Location:".URL."consulta-raca");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

}

?>

**ClinicaController**

<?php

use FFI\Exception;

include\_once "model/Usuario.php";

include\_once "model/Clinica.php";

include\_once "model/Castracao.php";

include\_once "model/email.php";

class ClinicaController

{

    function cadastrarClinica()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $filtros = array(".", "-", "(", ")", "/", " ");

            $cnpj = str\_replace($filtros,'',$\_POST["txtCNPJ"]);

            $cep = str\_replace($filtros,'',$\_POST["txtCEP"]);

            $tel = str\_replace($filtros,'',$\_POST["txtTelefone"]);

            $login = new Login();

            $login->nome =  $\_POST["txtNome"];

            $login->email = $\_POST["txtEmail"];

            $login->senha = password\_hash($\_POST["txtSenha"], PASSWORD\_DEFAULT);

            $login->nivelacesso = 1;

            $clinica = new Clinica();

            $clinica->idlogin = $login->cadastrar();

            $clinica->cnpj = $cnpj;

            $clinica->clitelefone = $tel;

            $clinica->vagas =       $\_POST["txtVagas"];

            $clinica->clirua =      $\_POST["txtRua"];

            $clinica->clibairro =   $\_POST["txtBairro"];

            $clinica->clinumero =   $\_POST["txtNumero"];

            $clinica->clicep = $cep;

            $clinica->ativo = 1;

            $clinica->cadastrar();

            echo "<script>alert('Clínica cadastrada com sucesso!'); window.location='".URL."cadastra-clinica';</script>";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function atualizarClinica()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $login = new Login();

            $login->idlogin = $\_POST["idLogin"];

            $login->nome = $\_POST["txtNome"];

            $login->email = $\_POST["txtEmail"];

            $login->senha = password\_hash($\_POST["txtSenha"], PASSWORD\_DEFAULT);

            $login->nivelacesso = 1;

            $login->atualizar();

            $clinica = new Clinica();

            $clinica->idclinica =   $\_POST["idClinica"];

            $clinica->cnpj =        $\_POST["txtCNPJ"];

            $clinica->clitelefone = $\_POST["txtTelefone"];

            $clinica->vagas =       $\_POST["txtVagas"];

            $clinica->clirua =      $\_POST["txtRua"];

            $clinica->clibairro =   $\_POST["txtBairro"];

            $clinica->clinumero =   $\_POST["txtNumero"];

            $clinica->clicep =      $\_POST["txtCEP"];

            if($\_POST["chkAtivo"] == 1)

            {

                $clinica->ativo = 1;

            }

            else

            $clinica->ativo = 0;

            $clinica->atualizar();

            header("location:".URL."consulta-clinica");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function excluirClinica($idClinica, $idLogin)

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $clinica = new Clinica();

            $clinica->idclinica = $idClinica;

            $clinica->excluir();

            $login = new Login();

            $login->idlogin = $idLogin;

            $login->excluir();

            header("Location:".URL."consulta-clinica");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function agendarDataCastracao()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 1) {

            $castracao = new Castracao();

            $castracao->idcastracao = $\_POST["idcastracao"];

            $castracao->idclinica = $\_SESSION["dadosClinica"]->idclinica;

            $castracao->status = 1;

            $castracao->horario = $\_POST["horario"];

            $castracao->aprovarCastracao();

            //enviar o email

            $email = new Email();

            $email->data = $\_POST["horario"];

            $email->nomeClinica = $\_SESSION["dadosClinica"]->nome;

            $email->ruaClinica = $\_SESSION["dadosClinica"]->clirua;

            $email->bairroClinica = $\_SESSION["dadosClinica"]->clibairro;

            $email->numeroClinica = $\_SESSION["dadosClinica"]->clinumero;

            $email->emailDestinatario = $\_POST["emailDestinatario"];

            $email->nomeDestinatario = $\_POST["nomeDestinatario"];

            $email->nomeAnimal = $\_POST["aninome"];

            $email->clitelefone = $\_POST["dadosClinica"]->clitelefone;

            $email->enviarConfirmacao();

            header("Location:".URL."lista-solicitacao");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

}

?>

**Controller**

<?php

/\* USADO PARA ABRIR TODAS AS PÁGINAS \*/

include\_once "model/Animal.php";

include\_once "model/Castracao.php";

include\_once "model/Clinica.php";

include\_once "model/Login.php";

include\_once "model/Raca.php";

include\_once "model/Usuario.php";

class Controller

{

    // PÁGINA INCIAL

    function abrirInicio(){

        include\_once "view/home.php";

    }

    // PÁGINAS IMPORTANTES

    function abrirSobre(){

        include\_once "view/sobre.php";

    }

    function paginaNaoEncontrada(){

        include\_once "view/paginaNaoEncontrada.php";

    }

    // LOGIN

    function abrirLogin(){

        include\_once "view/login.php";

    }

    function abrirEsqSenha(){

        include\_once "view/esqSenha.php";

    }

    function abrirRecuperacao($email)

    {

        include\_once "view/confirmarCodigoSenha.php";

    }

    // USUÁRIO

    function abrirHomeUsuario(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if ($\_SESSION["dadosLogin"]->nivelacesso == 0) {

            include\_once "view/homeUsuario.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirCadastro(){

        include\_once "view/cadastro.php";

    }

    function abrirPerfil(){

        //Caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if ($\_SESSION["dadosLogin"]->nivelacesso == 0) {

            include\_once "view/infoUsuario.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirAlterarSenha($idlogin){

        //só abrir a tela alterar senha se existir sessão de usuário ou se confirmar login for igual ao login retornado pelo banco

        if(isset($\_SESSION["dadosUsuario"]) || $\_SESSION["idlogin"] == $idlogin){

            include\_once "view/alterarSenha.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirCadAnimal(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if ($\_SESSION["dadosLogin"]->nivelacesso == 0) {

            $animal = new Raca();

            $dadosRacaAnimal = $animal->consultarRaca();

            include\_once "view/cadAnimal.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirMeusAnimais(){

        //Caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 0) {

            $animal = new Animal();

            $animal->idusuario = $\_SESSION["dadosUsuario"]->idusuario;

            $dadosAnimais = $animal->retornarAnimais();

            $usuario = new Usuario();

            $usuario->idusuario = $\_SESSION["dadosUsuario"]->idusuario;

            $quantcastracoes = $usuario->retornarQuantCastracao();

            include\_once "view/meusAnimais.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    /\*function abrirAtualizaAnimal($id){

        $animal = new Animal();

        $animal->idanimal = $id;

        $dadosAnimal = $animal->retornar();

        $raca = new Raca();

        $dadosRacaCanino = $raca->consultarCanino();

        $dadosRacaFelino = $raca->consultarFelino();

        include\_once "view/editaAnimal.php";

    }\*/

    // ADMINISTRADOR

    function abrirHomeAdm(){

        //Caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) { include\_once "view/homeAdm.php"; }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    #CADASTROS

    function abrirCadRaca(){

        //Caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) { include\_once "view/cadRaca.php"; }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirCadClinica(){

        //Caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) { include\_once "view/cadClinica.php"; }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    #CONSULTAS

    function abrirConsultaUsuario(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //caso não tenha privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $usuario = new Usuario();

            $dadosUsuario = $usuario->consultar();

            include\_once "view/consultaUsuario.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirConsultaClinica(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //caso não tenha privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $clinica = new Clinica();

            $dadosClinica = $clinica->consultar();

            include\_once "view/consultaClinica.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirConsultaCastracao(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return;}

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $castracao = new Castracao();

            $dadosCastracao = $castracao->consultar();

            include\_once "view/consultaCastracao.php";

        }

        else if($\_SESSION["dadosLogin"]->nivelacesso == 1){

            $castracao = new Castracao();

            $castracao->idclinica = $\_SESSION["dadosClinica"]->idclinica;

            $dadosCastracaoClinica = $castracao->consultarPraClinica();

            include\_once "view/consultaCastracao.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirConsultaAnimais($idusuario){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //caso não tenha privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2){

            $animal = new Animal();

            $animal->idusuario = $idusuario;

            $dadosAnimal = $animal->retornarAnimais();

            $raca = new Raca();

            $dadosRaca = $raca->consultar();

            include\_once "view/consultaAnimais.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirConsultaRaca(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //caso não tenha privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $raca = new Raca();

            $dadosRaca = $raca->consultar();

            include\_once "view/consultaRaca.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    #AGENDAMENTO

    function abrirListaSolicitacao(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $castracao = new Castracao();

            $dadosSolicitacao = $castracao->consultarSolicitacao();

            include\_once "view/listaSolicitacao.php";

        }

        else if($\_SESSION["dadosLogin"]->nivelacesso == 1){

            $castracao = new Castracao();

            $castracao->idclinica = $\_SESSION["dadosClinica"]->idclinica;

            $dadosSolicitacao = $castracao->clinicaConsultarSolicitacao();

            include\_once "view/listaSolicitacao.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function abrirAgendamento($id){

        //Caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        if ($\_SESSION["dadosLogin"]->nivelacesso > 0) {

            $agendamento = new Castracao();

            $agendamento->idcastracao = $id;

            $dadosCastracao = $agendamento->retornar();

            $clinica = new Clinica();

            $dadosClinicas = $clinica->consultarComVagas();

            include\_once "view/confirmaSolicitacao.php";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    // CLÍNICA

    function abrirHomeClinica(){

        //caso o usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 1) { include\_once "view/homeClinica.php"; }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    // TESTE

    function abrirTeste(){

        include\_once "view/Teste.php";

    }

}

?>

**UsuarioController**

<?php

include\_once "model/Usuario.php";

include\_once "model/Login.php";

include\_once "model/Animal.php";

include\_once "model/Castracao.php";

include\_once "model/Email.php";

include\_once "model/Clinica.php";

class UsuarioController

{

    function cadastrarUsuario(){

        //Cadastro do Login

        //verificando se o comprovante de endereço foi enviado corretamente

        if($\_FILES["btnComprovante"]["error"] == 0)

        {

            //filtrando a mascara dos inputs

            $filtros = array(".", "-", "(", ")", " ");

            $cpf = str\_replace($filtros,'',$\_POST["txtCPF"]);

            $cep = str\_replace($filtros,'',$\_POST["txtCEP"]);

            $rg = str\_replace($filtros,'',$\_POST["txtRG"]);

            $tel = str\_replace($filtros,'',$\_POST["txtTel"]);

            $celular = str\_replace($filtros,'',$\_POST["txtCelular"]);

            $nis = str\_replace($filtros,'',$\_POST["txtNIS"]);

            $consultarEmail = new Login();

            $consultarEmail->email = $\_POST["txtEmail"];

            $consultarCPF = new Usuario();

            $consultarCPF->cpf = $\_POST["txtCPF"];

            //Verificando o NIS

            if($\_POST["chkNIS"] == "sim" && isset($nis))

            {

                if(strlen($nis) == 11)

                {

                    $consultarNIS = new Usuario();

                    $consultarNIS->nis = $nis;

                    $dadosConusultaNIS = $consultarNIS->verificarNis();

                }

                else

                {

                    echo"<script>alert('digite um NIS válido'); window.location='".URL."cadastra-tutor'; </script>";

                    return;

                }

            }

            else

                $dadosConusultaNIS = null;

            //verificando se o email ou cpf ou nis já existem no banco ou não

            if($consultarEmail->logar() == null && $consultarCPF->verificarCPF() == null && $dadosConusultaNIS == null)

            {

                $login = new Login();

                $login->nome =  $\_POST["txtNome"];

                $login->email = $\_POST["txtEmail"];

                $login->senha = password\_hash($\_POST["txtSenha"], PASSWORD\_DEFAULT);

                $login->nivelacesso = 0;

                //Cadastro do Usuário

                $cadastra = new Usuario();

                $cadastra->idlogin = $login->cadastrar();

                $cadastra->rg = strtoupper($rg);

                $cadastra->cpf = $cpf;

                $cadastra->beneficio = 0;

                $cadastra->telefone = $tel;

                $cadastra->celular = $celular;

                //Arrumar o Whatsapp dps

                $cadastra->whatsapp = 0;

                $cadastra->punicao = 0;

                $cadastra->usurua =    $\_POST["txtRua"];

                $cadastra->usubairro = $\_POST["txtBairro"];

                $cadastra->usunumero = $\_POST["txtNumero"];

                $cadastra->usucep = $cep;

                if($\_POST["chkWhats"] == "sim")

                    $cadastra->whatsapp = 1;

                else

                    $cadastra->whatsapp = 0;

                //TRATANDO O COMPROVANTE DE ENDEREÇO

                    //Tratar o envio da imagem

                    $docComprovante = $\_FILES["btnComprovante"]["name"];       //Nome do arquivo

                    $docComprovanteTemp = $\_FILES["btnComprovante"]["tmp\_name"];      //nome temporário

                    //pegar a extensão do arquivo

                    $info = new SplFileInfo($docComprovante);

                    $extensao = $info->getExtension();

                    //gerar novo nome

                    $novoNomeComprovante = md5(microtime()) . ".$extensao";

                    $pastaDestino = "recursos/img/docComprovantes/$novoNomeComprovante";    //pasta destino

                    move\_uploaded\_file($docComprovanteTemp, $pastaDestino);       //mover o arquivo

                    $cadastra->doccomprovante = $novoNomeComprovante;

                    $cadastra->quantcastracoes = 1;

                if(empty($nis))

                {

                    $cadastra->nis = "";

                }

                else

                {

                    $cadastra->nis = $nis;

                    $cadastra->beneficio = 1;

                    $cadastra->quantcastracoes = 2;

                }

                //TRATANDO O DOCUMENTO DO PROTETOR DE ANIMAIS

                if($\_POST["chkProtetor"] == "sim")

                {

                    $cadastra->beneficio = 3;

                }

                $cadastra->cadastrar();

                $this->logar();

                header("Location:".URL);

            }

            else

            {

                echo"<script>alert('Já existe um perfil com esse e-mail, CPF ou NIS cadastrados'); window.location='".URL."cadastra-tutor'; </script>";

                return;

            }

        }

        else

        {

            echo"<script>alert('Houve um erro ao enviar o comprovante de residência'); window.location='".URL."cadastra-tutor'; </script>";

            return;

        }

    }

    function atualizarUsuario()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            //filtrando a mascara dos inputs

            $filtros = array(".", "-", "(", ")", " ");

            $cpf = str\_replace($filtros,'',$\_POST["txtCPF"]);

            $cep = str\_replace($filtros,'',$\_POST["txtCEP"]);

            $rg = str\_replace($filtros,'',$\_POST["txtRG"]);

            $tel = str\_replace($filtros,'',$\_POST["txtTel"]);

            $celular = str\_replace($filtros,'',$\_POST["txtCelular"]);

            $nis = str\_replace($filtros,'',$\_POST["txtNIS"]);

            $login = new Login();

            $login->idlogin = $\_POST["idlogin"];

            $login->nome =    $\_POST["txtNome"];

            $login->email =   $\_POST["txtEmail"];

            $login->atualizarLogin();

            $usu = new Usuario();

            $usu->rg =        strtoupper($rg);

            $usu->cpf =       $cpf;

            // Que tipo de benefício tem

            if (isset($\_POST["chkProtetor"])) {

                $usu->beneficio = $\_POST["chkProtetor"];

            }

            else if(isset($\_POST["chkNIS"])){

                $usu->beneficio = $\_POST["chkNIS"];

            }

            else{ $usu->beneficio = 0; }

            $usu->telefone =  $tel;

            $usu->celular =   $celular;

            $usu->usurua =    $\_POST["txtRua"];

            $usu->usubairro = $\_POST["txtBairro"];

            $usu->usunumero = $\_POST["txtNumero"];

            $usu->usucep =    $cep;

                // NIS

                if(empty($nis)){

                    $usu->nis = "";

                }

                else{ $usu->nis = $nis;}

            $usu->idusuario = $\_POST["idusuario"];

            if(isset($\_POST["chkPunicao"]))

            {

                $usu->punicao = $\_POST["chkPunicao"];

            }else{

                $usu->punicao = 0;

            }

            $usu->atualizar();

            echo "<script>

                    alert('Dados alterados com sucesso!');

                    window.location='".URL."consulta-usuario';

                </script>";

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function alterarSenha()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 0) {

            $alterar = new Login();

            $alterar->senha = password\_hash($\_POST["senha"], PASSWORD\_DEFAULT);

            $alterar->alterarSenha();

            header("Location:".URL."perfil");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function solicitarCastracao()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 0) {

            $castracao = new Castracao();

            $usuario = new Usuario();

            $usuario->idusuario = $\_SESSION["dadosUsuario"]->idusuario;

            $dadosUsuario = $usuario->retornar();

            if($dadosUsuario->quantcastracoes >= 0)

            {

                $castracao->idusuario = $usuario->idusuario;

                $castracao->idanimal = $\_POST["idAnimal"];

                $castracao->observacao = $\_POST["obsCastracao"];

                $castracao->status = 0;

                $\_SESSION["dadosUsuario"]->quantcastracoes--;

                $usuario->quantcastracoes = $\_SESSION["dadosUsuario"]->quantcastracoes;

                $usuario->atualizarQuantCastracoes();

                $castracao->cadastrar();

                echo"<script>alert('Solicitação enviada'); window.location='".URL."meus-animais'; </script>";

            }

            else

            {

                echo"<script>alert('Seu limite de castrações foi atingido'); window.location='".URL."meus-animais'; </script>";

            }

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function agendarClinicaCastracao()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $idcastracao = $\_POST["idcastracao"];

            //Caso o botão recusado não seja apertado

            if(!isset($\_POST["btnRecusa"]))

            {

                if($\_POST["selectClinica"] != 0)

                {

                    $castracao = new Castracao();

                    $castracao->idclinica = $\_POST["selectClinica"];

                    $castracao->status = 1;

                    $castracao->idcastracao = $idcastracao;

                    $castracao->aprovarCastracao();

                    $clinica = new Clinica();

                    $clinica->idclinica = $\_POST["selectClinica"];

                    $dadosClinica = $clinica->retornar();

                    $clinica->vagas = $dadosClinica->vagas - 1;

                    $clinica->subtrairVagas();

                    header("Location:".URL."lista-solicitacao");

                }

                else

                {

                    echo "<script>alert('Selecione a clínica e/ou a data'); window.location='".URL."agendamento/$idcastracao'; </script>";

                }

            }

            else if($\_POST["btnRecusa"] == "Recusar")

            {

                $castracao = new Castracao();

                $castracao->idcastracao = $idcastracao;

                $castracao->status = 3;

                $castracao->recusarCastracao();

                header("Location:".URL."lista-solicitacao");

            }

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function atualizarCastracao()

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 1) {

            $castracao = new Castracao();

            $castracao->idcastracao = $\_POST["idcastracao"];

            switch($\_POST["statusAtualizado"])

            {

                case 2:

                    // Animal castrado

                    $castracao->status = 2;

                    $animal = new Animal();

                    $animal->codchip = $\_POST["codChip"];

                    $animal->atualizarCastrado();

                    $castracao->atualizar();

                    if($\_POST["status"] != 4)

                    {

                        //Liberar a vaga de castração para a clínica

                        $clinica = new Clinica();

                        $clinica->idclinica = $\_SESSION["dadosClinica"]->idclinica;

                        $dadosClinica = $clinica->retornar();

                        $clinica->vagas = $dadosClinica->vagas - 1;

                        $clinica->subtrairVagas();

                    }

                break;

                case 4:

                    // Não compareceu

                    //Adicionar o status de que o animal não compareceu à castração

                    $castracao->status = 4;

                    $castracao->atualizar();

                    //Aplicar uma punição ao tutor que não compareceu à castração

                    $usuario = new Usuario();

                    $usuario->idusuario = $\_POST["idTutor"];

                    $usuario->punicao = 1;

                    $usuario->aplicarPunicao();

                    //Enviar aviso ao usuário dizendo que não compareceu à castração

                    $email = new Email();

                    $email->emailDestinatario = $\_POST["emailTutor"];

                    $email->nomeDestinatario =  $\_POST["nomeTutor"];

                    $email->nomeAnimal =        $\_POST["nomeAnimal"];

                    $email->data =              $\_POST["dataCastracao"];

                    $email->enviarAviso();

                    if($\_POST["status"] == 4)

                    {

                        //Liberar a vaga de castração para a clínica

                        $clinica = new Clinica();

                        $clinica->idclinica = $\_SESSION["dadosClinica"]->idclinica;

                        $dadosClinica = $clinica->retornar();

                        $clinica->vagas = $dadosClinica->vagas + 1;

                        $clinica->adicionarVagas();

                    }

                break;

                case 5:

                    // Castração cancelada

                    $castracao->status = 5;

                    $castracao->excluir();

                    if($\_POST["status"] != 4)

                    {

                        //Liberar a vaga de castração para a clínica

                        $clinica = new Clinica();

                        $clinica->idclinica = $\_SESSION["dadosClinica"]->idclinica;

                        $dadosClinica = $clinica->retornar();

                        $clinica->vagas = $dadosClinica->vagas + 1;

                        $clinica->adicionarVagas();

                    }

                break;

                case 6:

                    // Reagendar castração

                    //Adicionar o status de que o animal retornou para a análise da castração

                    $castracao->status = 6;

                    $castracao->obsclinica = $\_POST["obsclinica"];

                    $castracao->reagendar();

                    if($\_POST["status"] != 4)

                    {

                        //Liberar a vaga de castração para a clínica

                        $clinica = new Clinica();

                        $clinica->idclinica = $\_SESSION["dadosClinica"]->idclinica;

                        $dadosClinica = $clinica->retornar();

                        $clinica->vagas = $dadosClinica->vagas + 1;

                        $clinica->adicionarVagas();

                    }

                break;

                case 7:

                    // Animal veio a óbito

                    $castracao->status = 7;

                    $castracao->atualizar();

                    if($\_POST["status"] != 4)

                    {

                        //Liberar a vaga de castração para a clínica

                        $clinica = new Clinica();

                        $clinica->idclinica = $\_SESSION["dadosClinica"]->idclinica;

                        $dadosClinica = $clinica->retornar();

                        $clinica->vagas = $dadosClinica->vagas + 1;

                        $clinica->adicionarVagas();

                    }

                break;

                case 8:

                    // Animal castrado(mas com alterações)

                    $castracao->status = 8;

                    $castracao->obsclinica = $\_POST["obsClinica"];

                    $castracao->atualizar();

                    if($\_POST["status"] != 4)

                    {

                        //Liberar a vaga de castração para a clínica

                        $clinica = new Clinica();

                        $clinica->idclinica = $\_SESSION["dadosClinica"]->idclinica;

                        $dadosClinica = $clinica->retornar();

                        $clinica->vagas = $dadosClinica->vagas - 1;

                        $clinica->subtrairVagas();

                    }

                break;

                default:

                    echo"<script>alert('Insira um valor válido'); window.location='".URL."consulta-castracao'; </script>"; return;

            }

            header("Location:".URL."consulta-castracao");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function logar()

    {

        $logar = new Login();

        $logar->email = $\_POST["txtEmail"];

        $dadosLogin = $logar->logar();

        if($dadosLogin && password\_verify($\_POST["txtSenha"], $dadosLogin->senha))

        {

            $\_SESSION["dadosLogin"] = $dadosLogin;

            switch($dadosLogin->nivelacesso)

            {

                //caso seja usuário

                case 0:

                    $usuario = new Login();

                    $usuario->idlogin = $dadosLogin->idlogin;

                    $dadosUsuario = $usuario->retornarUsuario();

                    $animal = new Animal();

                    $animal->idusuario = $dadosUsuario->idusuario;

                    $dadosAnimais = $animal->retornarAnimais();

                    $\_SESSION["dadosUsuario"] = $dadosUsuario;

                    $\_SESSION["dadosAnimais"] = $dadosAnimais;

                    echo"<script>alert('Usuário Logado'); window.location='".URL."home-usuario'; </script>";

                break;

                //caso seja clínica

                case 1:

                    //Buscando as informações da clínica

                    $clinica = new Login();

                    $clinica->idlogin = $dadosLogin->idlogin;

                    $dadosClinica = $clinica->retornarClinica();

                    //Colocando em uma Sessão

                    $\_SESSION["dadosClinica"] = $dadosClinica;

                    echo"<script>alert('Usuário Clínica Logado'); window.location='".URL."home-clinica'; </script>";

                break;

                //caso seja adm

                case 2:

                    echo"<script>alert('Usuário Administrador Logado'); window.location='".URL."home-adm'; </script>";

                break;

                default:

                    header("location:".URL."login");

            }

        }

        else

        {

            echo"<script>alert('Email ou senha estão errados'); window.location='".URL."login'; </script>";

        }

    }

    function sair()

    {

        $\_SESSION[] = null;

        session\_destroy();

        header("Location:".URL);

    }

    function excluir($idUsuario, $idLogin)

    {

        //caso não usuário não esteja logado

        if(!isset($\_SESSION["dadosLogin"])) { header("Location:".URL."login"); return; }

        //Controle de privilégio

        if($\_SESSION["dadosLogin"]->nivelacesso == 2) {

            $usuario = new Usuario();

            $usuario->idusuario = $idUsuario;

            $usuario->excluir();

            $login = new Login();

            $login->idlogin = $idLogin;

            $login->excluir();

            header("location:".URL."consulta-usuario");

        }

        else{ include\_once "view/paginaNaoEncontrada.php"; }

    }

    function recuperarSenha()

    {

        $login = new Login();

        $login->email = $\_POST["txtEmail"];

        $dadosRecuperacao = $login->logar();

        if($dadosRecuperacao && $dadosRecuperacao->nivelacesso == 0)

        {

            $codigo = rand(100000,999999);

            $email = new Email();

            $email->emailDestinatario = $\_POST["txtEmail"];

            $email->codsenha = $codigo;

            $email->enviarRecuperacao();

            $login->codsenha = $codigo;

            $login->gerarCodigo();

            header("Location:".URL."codigo-de-recuperacao/$dadosRecuperacao->email");

        }

        else

        {

            setcookie("msg","Parece que esse E-mail não existe no sistema");

            header("Location:".URL."esqueci-a-senha");

        }

    }

    function confirmarCodigo()

    {

        $login = new Login();

        $login->email = $\_POST["txtEmail"];

        $dadosConfirmacao = $login->confirmarCodigo();

        $\_SESSION["idlogin"] = $dadosConfirmacao->idlogin;

        $codigo = $\_POST["txtCod"];

        if(empty($\_COOKIE["tentativas"])){setcookie("tentativas", 1, time() + 3600, "/");}

        //RESETAR O COOKIE QUE LIMITA A QUANTITADE DE TENTATIVAS PARA O CÓDIGO. USAR PARA DEBUGAR

        //setcookie("tentativas", "", time() - 1000, "/");

        if($\_COOKIE["tentativas"] < 5)

        {

            if(isset($codigo) && strlen($codigo) === 6 && $codigo === $dadosConfirmacao->codsenha)

            {

                //cookie para permitir o usuário de abrir a tela alterarsenha

                setcookie("tentativas", "", time() - 1000, "/");

                setcookie("confirmacao",$dadosConfirmacao->idlogin);

                header("Location:".URL."alterar-senha/$dadosConfirmacao->idlogin");

                return;

            }

            else

            {

                setcookie("tentativas", 1 + $\_COOKIE["tentativas"], time() + 3600, "/");

                setcookie("msg","Codigo inválido! Tente novamente");

                header("Location:".URL."codigo-de-recuperacao/".$\_POST["txtEmail"]);

                return;

            }

        }

        else

        {

            echo"<script>alert('Você fez muitas tentativas, tente novamente mais tarde'); window.location='".URL."login'; </script>";

        }

    }

    function redefinirSenha()

    {

        if(isset($\_SESSION["dadosLogin"]) && $\_SESSION["dadosLogin"]->nivelacesso == 0)

        {

            //redefinir senha e limpar o campo codsenha

            $login = new Login();

            $login->idlogin = $\_SESSION["dadosLogin"]->idlogin;

            $login->senha = password\_hash($\_POST["confSenha"], PASSWORD\_DEFAULT);

            $login->redefinirSenha();

            $\_SESSION["idlogin"] = null;

            $this->sair();

            header("Location:".URL."login");

        }

        else if(!isset($\_SESSION["dadosLogin"]) && $\_COOKIE["confirmacao"] == $\_SESSION["idlogin"])

        {

            //redefinir senha e limpar o campo codsenha

            $login = new Login();

            $login->idlogin = $\_SESSION["idlogin"];

            $login->senha = password\_hash($\_POST["confSenha"], PASSWORD\_DEFAULT);

            $login->redefinirSenha();

            setcookie("confirmacao","", time() - 3600);

            $\_SESSION["idlogin"] = null;

            $this->sair();

            header("Location:".URL."login");

        }

        else

        echo"<script>alert('Erro ao alterar a senha'); window.location='".URL."alterar-senha'; </script>";

    }

}

?>

**Animal**

<?php

    class Animal{

        //Atributos

        private $idanimal;

        private $idusuario;

        private $idraca;

        private $aninome;

        private $especie;

        private $sexo;

        private $cor;

        private $pelagem;

        private $porte;

        private $idade;

        private $comunitario;

        private $foto;

        private $codchip;

        //Método get

        function \_\_get($atributo)

        {

            return $this->$atributo;

        }

        //Método set

        function \_\_set($atributo, $valor)

        {

            $this->$atributo = $valor;

        }

        //Método construtor

        function \_\_construct()

        {

            //Importando arquivo de conexão com banco de dados

            include\_once "Conexao.php";

        }

        //Método cadastrar

        function cadastrar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para cadastrar

            $cmd = $con->prepare("INSERT INTO animal (idusuario, idraca, aninome, especie, sexo, cor, pelagem, porte, idade, comunitario, foto)

                                    VALUES (:idusuario, :idraca, :aninome, :especie, :sexo, :cor, :pelagem, :porte, :idade, :comunitario, :foto)");

            //Parâmetros SQL

            $cmd->bindParam(":idusuario",   $this->idusuario);

            $cmd->bindParam(":idraca",      $this->idraca);

            $cmd->bindParam(":aninome",     $this->aninome);

            $cmd->bindParam(":especie",     $this->especie);

            $cmd->bindParam(":sexo",        $this->sexo);

            $cmd->bindParam(":cor",         $this->cor);

            $cmd->bindParam(":porte",       $this->porte);

            $cmd->bindParam(":pelagem",     $this->pelagem);

            $cmd->bindParam(":porte",       $this->porte);

            $cmd->bindParam(":idade",       $this->idade);

            $cmd->bindParam(":comunitario", $this->comunitario);

            $cmd->bindParam(":foto",        $this->foto);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método consultar

        function consultar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            $cmd = $con->prepare("SELECT \* FROM animal JOIN raca ON animal.idraca = raca.idraca");

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        //Método excluir

        function excluir()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para deletar

            $cmd = $con->prepare("DELETE FROM animal WHERE idanimal = :idanimal");

            //Parâmetros SQL

            $cmd->bindParam(":idanimal", $this->idanimal);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Atualizar

        function atualizar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

            $cmd = $con->prepare("UPDATE animal SET idraca = :idraca, aninome = :aninome, especie = :especie, sexo = :sexo, cor = :cor, pelagem = :pelagem,

                                    porte = :porte, idade = :idade, comunitario = :comunitario, foto = :foto WHERE idanimal = :idanimal");

            //Parâmetros SQL

            $cmd->bindParam(":idraca",      $this->idraca);

            $cmd->bindParam(":aninome",     $this->aninome);

            $cmd->bindParam(":especie",     $this->especie);

            $cmd->bindParam(":sexo",        $this->sexo);

            $cmd->bindParam(":cor",         $this->cor);

            $cmd->bindParam(":porte",       $this->porte);

            $cmd->bindParam(":pelagem",     $this->pelagem);

            $cmd->bindParam(":porte",       $this->porte);

            $cmd->bindParam(":idade",       $this->idade);

            $cmd->bindParam(":comunitario", $this->comunitario);

            $cmd->bindParam(":foto",        $this->foto);

            $cmd->bindParam(":idanimal", $this->idanimal);

            //$cmd->bindParam(":foto", $this->foto);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Retornar

        function retornar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT \* FROM animal WHERE idanimal = :idanimal");

            //Parâmetros SQL

            $cmd->bindParam(":idanimal", $this->idanimal);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function retornarAnimais()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT animal.\*, raca.\*, status FROM animal join raca on

                                    animal.idraca = raca.idraca left join castracao on

                                    animal.idanimal = castracao.idanimal WHERE idusuario = :idusuario");

            //Parâmetros SQL

            $cmd->bindParam(":idusuario", $this->idusuario);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        function atualizarCastrado(){

            $con = Conexao::conectar();

            $cmd = $con->prepare("UPDATE animaL SET codchip = :codchip WHERE idanimal = :idanimal");

            $cmd->bindParam(":codchip", $this->codchip);

            $cmd->bindParam(":idanimal", $this->idanimal);

            $cmd->execute();

        }

    }

?>

**Castracao**

<?php

    class Castracao{

        //Atributos

        private $idcastracao;

        private $idanimal;

        private $idclinica;

        private $horario;

        private $status;

        private $observacao;

        private $obsclinica;

        //Método get

        function \_\_get($atributo)

        {

            return $this->$atributo;

        }

        //Método set

        function \_\_set($atributo, $valor)

        {

            $this->$atributo = $valor;

        }

        //Método construtor

        function \_\_construct()

        {

            //Importando arquivo de conexão com banco de dados

            include\_once "Conexao.php";

        }

        //Método cadastrar

        function cadastrar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para cadastrar

            $cmd = $con->prepare("INSERT INTO castracao (idanimal, idclinica, horario, status, observacao)

                                    VALUES (:idanimal, :idclinica, NULLIF(:horario,''), :status, :observacao)");

            //Parâmetros SQL

            $cmd->bindParam(":idanimal",   $this->idanimal);

            $cmd->bindParam(":idclinica",  $this->idclinica);

            $cmd->bindParam(":horario",    $this->horario);

            $cmd->bindParam(":status",     $this->status);

            $cmd->bindParam(":observacao", $this->observacao);

            //Executando o comando SQL

            $cmd->execute();

        }

        function consultarSolicitacao()

        {

            $con = Conexao::conectar();

            $cmd = $con->prepare("SELECT idcastracao, aninome, nome FROM castracao

                                        JOIN animal ON animal.idanimal = castracao.idanimal

                                        JOIN usuario ON usuario.idusuario = animal.idusuario

                                        JOIN login ON login.idlogin = usuario.idlogin

                                    WHERE status = 'null' ");

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        function clinicaConsultarSolicitacao()

        {

            $con = Conexao::conectar();

            $cmd = $con->prepare("SELECT idcastracao, aninome, nome FROM castracao

                                        join animal on animal.idanimal = castracao.idanimal

                                        join usuario on usuario.idusuario = animal.idusuario

                                        join login on login.idlogin = usuario.idlogin

                                    WHERE castracao.idclinica = :idclinica AND horario IS NULL");

            $cmd->bindParam(":idclinica", $this->idclinica);

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        function aprovarCastracao()

        {

            $con = Conexao::conectar();

            $cmd = $con->prepare("UPDATE castracao SET idclinica = :idclinica, horario = nullif(:horario,''), status = :status WHERE idcastracao = :idcastracao");

            $cmd->bindParam(":idclinica",   $this->idclinica);

            $cmd->bindParam(":horario",     $this->horario);

            $cmd->bindParam(":status",      $this->status);

            $cmd->bindParam(":idcastracao", $this->idcastracao);

            $cmd->execute();

        }

        function recusarCastracao()

        {

            $con = Conexao::conectar();

            $cmd = $con->prepare("UPDATE castracao SET status = :status WHERE idcastracao = :idcastracao");

            $cmd->bindParam(":status",      $this->status);

            $cmd->bindParam(":idcastracao", $this->idcastracao);

            $cmd->execute();

        }

        //Método consultar

        function consultar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            $cmd = $con->prepare("SELECT idcastracao, foto, aninome, nome AS 'nometutor', horario, status, observacao, cpf, cnpj, nome AS 'nomeclinica', obsclinica FROM castracao

                                    JOIN animal ON castracao.idanimal = animal.idanimal

                                    JOIN usuario ON animal.idusuario = usuario.idusuario

                                    LEFT JOIN clinica ON castracao.idclinica = clinica.idclinica

                                    JOIN login ON login.idlogin = clinica.idlogin");

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        function consultarPraClinica()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            //$cmd = $con->prepare("SELECT \* FROM castracao");

            $cmd = $con->prepare("SELECT idcastracao, clinica.idclinica, animal.foto, animal.aninome, nome AS 'nometutor', horario, status, observacao, cpf, usuario.idusuario, email, rg, telefone, celular

                                    FROM castracao

                                        JOIN animal ON castracao.idanimal = animal.idanimal

                                        JOIN usuario ON animal.idusuario = usuario.idusuario

                                        JOIN login ON login.idlogin = usuario.idlogin

                                        JOIN clinica ON castracao.idclinica = clinica.idclinica

                                    WHERE clinica.idclinica = :idclinica ORDER BY horario DESC");

            $cmd->bindParam(":idclinica", $this->idclinica);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        //Método excluir

        function excluir()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para deletar

            $cmd = $con->prepare("DELETE FROM castracao WHERE idcastracao = :idcastracao");

            //Parâmetros SQL

            $cmd->bindParam(":idcastracao", $this->idcastracao);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Atualizar

        function atualizar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

            $cmd = $con->prepare("UPDATE castracao SET status = :status, obsclinica = NULLIF(:obsclinica, '') WHERE idcastracao = :idcastracao");

            //Parâmetros SQL

            $cmd->bindParam(":status",      $this->status);

            $cmd->bindParam(":obsclinica",  $this->obsclinica);

            $cmd->bindParam(":idcastracao", $this->idcastracao);

            //Executando o comando SQL

            $cmd->execute();

        }

        function reagendar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

            $cmd = $con->prepare("UPDATE castracao SET idclinica = null, horario = null, status = :status, obsclinica = :obsclinica WHERE idcastracao = :idcastracao");

            //Parâmetros SQL

            $cmd->bindParam(":status",      $this->status);

            $cmd->bindParam(":obsclinica",  $this->obsclinica);

            $cmd->bindParam(":idcastracao", $this->idcastracao);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Retornar

        function retornar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT idcastracao, aninome, cpf, observacao, status, foto, email, nome FROM castracao

                                        JOIN animal ON animal.idanimal = castracao.idanimal

                                        JOIN usuario ON usuario.idusuario = animal.idusuario

                                        JOIN login ON usuario.idlogin = login.idlogin

                                    WHERE castracao.idcastracao = :idcastracao");

            //Parâmetros SQL

            $cmd->bindParam(":idcastracao", $this->idcastracao);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

    }

?>

**Clinica**

<?php

    class Clinica{

        //Atributos

        private $idclinica;

        private $idlogin;

        private $cnpj;

        private $clitelefone;

        private $vagas;

        private $clirua;

        private $clibairro;

        private $clinumero;

        private $clicep;

        private $ativo;

        //Método get

        function \_\_get($atributo)

        {

            return $this->$atributo;

        }

        //Método set

        function \_\_set($atributo, $valor)

        {

            $this->$atributo = $valor;

        }

        //Método construtor

        function \_\_construct()

        {

            //Importando arquivo de conexão com banco de dados

            include\_once "Conexao.php";

        }

        //Método cadastrar

        function cadastrar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para cadastrar

            $cmd = $con->prepare("INSERT INTO clinica (idlogin, cnpj, clitelefone, vagas, clirua, clibairro, clinumero, clicep, ativo)

                                        VALUES (:idlogin, :cnpj, :clitelefone, :vagas, :clirua, :clibairro, :clinumero, :clicep, :ativo)");

            //Parâmetros SQL

            $cmd->bindParam(":idlogin",     $this->idlogin);

            $cmd->bindParam(":cnpj",        $this->cnpj);

            $cmd->bindParam(":clitelefone", $this->clitelefone);

            $cmd->bindParam(":vagas",       $this->vagas);

            $cmd->bindParam(":clirua",      $this->clirua);

            $cmd->bindParam(":clibairro",   $this->clibairro);

            $cmd->bindParam(":clinumero",   $this->clinumero);

            $cmd->bindParam(":clicep",      $this->clicep);

            $cmd->bindParam("ativo",        $this->ativo);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método consultar

        function consultar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            $cmd = $con->prepare("SELECT idclinica, login.idlogin, nome, email, cnpj, clitelefone, vagas, clirua, clibairro, clinumero, clicep, ativo

                                    FROM clinica JOIN login ON clinica.idlogin = login.idlogin");

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        function consultarComVagas()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            $cmd = $con->prepare("SELECT idclinica, nome, email, cnpj, clitelefone, vagas, clirua, clibairro, clinumero, clicep

                                    FROM clinica JOIN login ON clinica.idlogin = login.idlogin WHERE clinica.vagas > 0 and clinica.ativo = 1");

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        //Método excluir

        function excluir()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para deletar

            $cmd = $con->prepare("DELETE FROM clinica WHERE idclinica = :idclinica");

            //Parâmetros SQL

            $cmd->bindParam(":idclinica", $this->idclinica);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Atualizar

        function atualizar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

            $cmd = $con->prepare("UPDATE clinica SET cnpj = :cnpj, clitelefone = :clitelefone, vagas = :vagas, clirua = :clirua,

                                    clibairro = :clibairro, clinumero = :clinumero, clicep = :clicep, ativo = :ativo WHERE idclinica = :idclinica");

            //Parâmetros SQL

            $cmd->bindParam(":cnpj",        $this->cnpj);

            $cmd->bindParam(":clitelefone", $this->clitelefone);

            $cmd->bindParam(":vagas",       $this->vagas);

            $cmd->bindParam(":clirua",      $this->clirua);

            $cmd->bindParam(":clibairro",   $this->clibairro);

            $cmd->bindParam(":clinumero",   $this->clinumero);

            $cmd->bindParam(":clicep",      $this->clicep);

            $cmd->bindParam(":idclinica",   $this->idclinica);

            $cmd->bindParam(":ativo",       $this->ativo);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Retornar

        function retornar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT clinica.\*, nome FROM clinica join login on login.idlogin = clinica.idlogin WHERE idclinica = :idclinica");

            //Parâmetros SQL

            $cmd->bindParam(":idclinica", $this->idclinica);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function subtrairVagas()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("UPDATE clinica SET vagas = :vagas WHERE idclinica = :idclinica");

            //Parâmetros SQL

            $cmd->bindParam(":vagas", $this->vagas);

            $cmd->bindParam(":idclinica", $this->idclinica);

            //Executando o comando SQL

            $cmd->execute();

        }

        function adicionarVagas()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("UPDATE clinica SET vagas = :vagas WHERE idclinica = :idclinica");

            //Parâmetros SQL

            $cmd->bindParam(":vagas",     $this->vagas);

            $cmd->bindParam(":idclinica", $this->idclinica);

            //Executando o comando SQL

            $cmd->execute();

        }

    }

?>

**Conexao**

<?php

class Conexao{

    static function conectar(){

        //Informações para acessar o servidor do banco de dados

        $localhost = "mysql:host=localhost;port=3306;dbname=bdcastrapet";

        $usuario = "root";

        $senha = "usbw";

        $con = new PDO($localhost,$usuario,$senha);

        //Ativando recurso de exibição de erro

        $con->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_EXCEPTION);

        return $con;//Retorna conexão para uso

    }

}

?>

**Email**

<?php

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

require "recursos/PHPMailer/Exception.php";

require "recursos/PHPMailer/PHPMailer.php";

require "recursos/PHPMailer/SMTP.php";

    class Email

    {

        //Atributos

        private $host = "smtp.office365.com";                                                //servidor do protocolo de envio de email

        private $emailRemetente = "castrapet.franco@outlook.com";                           //email do remetente

        private $senhaRemetente = "pet.CastraF";                                            //senha do remetente

        private $nomeRemetente = "PetCastra";               //nome do remetente

        private $porta = 587;                               //porta do servidor TLS/STARTTLS/SSL

        private $emailDestinatario;                         //email a ser enviado

        private $nomeDestinatario;                          //nome do destinatário

        private $nomeAnimal;                                //nome do Animal

        private $data;

        private $nomeClinica;

        private $ruaClinica;

        private $bairroClinica;

        private $numeroClinica;

        private $codsenha;

        private $clitelefone;

        //Método get

        function \_\_get($atributo)

        {

            return $this->$atributo;

        }

        //Método set

        function \_\_set($atributo, $valor)

        {

            $this->$atributo = $valor;

        }

        function enviarConfirmacao()

        {

            $Conteudo =

            "

                <p><b>Olá, $this->nomeDestinatario!</b><br/>

                A castração do(a) $this->nomeAnimal foi agendada! Seguem informações:

                </p>

                <p><b>Data:</b> ".date('d-m-Y',strtotime($this->data))."<br/>

                <b>Hora:</b> ".date('H:i',strtotime($this->data))."</p>

                <p><b>Clínica: </b>$this->nomeClinica <br/>

                <b>Endereço: </b>$this->ruaClinica, $this->numeroClinica - $this->bairroClinica, Franco da Rocha - SP</p>

                <b>Telefone: </b>$this->telefoneClinica</p>

                <p><u>Chegar com 10 minutos de antecedência.</u></p>

                <h3>Orientações sobre a castração:</h3>

                <ul>

                    <li>A clínica veterinária credenciada ficará responsável pelo <u><b>contato e agendamento</b></u> da castração. <br/>Seram realizadas três

                        (03) tentativas de contato, caso não atenda, automaticamente perderá a vaga cadastrada.</li>

                    <li>Comparecer no horário agendado pele clínica.</li>

                    <li>Deixar o animal em <u><b>JEJUM</b></u> (sólido e líquido) <u><b>08 horas antes da castração.</b></u> <br/>(Ex: a castração será ás 08hs da manhã, portanto, a última refeição

                        deverá ser até as 23hs da noite).</li>

                    <li>É recomendado o uso de colar de proteção (elisabetano) ou a roupa cirúgica (você encontrará na clínica veterinária ou em petshops)

                        procure em lugares mais baratos ou até mesmo empréstimo. </li>

                    <li>A clínica veterinária já medica os animais, caso considerem necessário, a clínica poderá prescrever medicamentos

                    com nome genérico (menor custo) que podem ser comprados em farmácia humana.</li>

                    <li>Ter 18 anos ou mais e residir no município.</li>

                    <li>Levar cobertor ou um lençol limpo.</li>

                    <li>O proprietário que não fizer a retirada do animal no horário estipulado pela clínica estará sujeito a aplicação de

                    Auto de Infração, conforme o Art 29 da Resolução SIMA n° 05/2021 e demais legislações federais, estaduais e municipais.</li>

                    <li>Em caso de motivo que impeça o animal à cirurgia de castração (fugiu, doença, castrou no particular, óbito e etc),

                        entrar em contato imediatamente com a clínica veterinária para comunicar o cancelamento da cirurgia.</li>

                    <li><b>Impedem o procedimento cirúrgico:</b> Cio, prenhez, amamentação de filhotes com menos de 45 dias, animais muito idosos

                    (acima de 08 anos) ou filhotes com menos de 03 meses.</li>

                    <li>Todos os animais passarão por uma avaliação clínica pelos médicos veterinários da clínica credenciada e caso detectado

                    alguma alteração o procedimento poderá ser cancelado ou remarcado.</li>

                </ul>

            ";

            $email = new PHPMailer(true);

            try{

                //Configurações

                $email->isSMTP();

                $email->Host = $this->host;

                $email->SMTPAuth = true;

                $email->Username = $this->emailRemetente;

                $email->Password = $this->senhaRemetente;

                $email->SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS;           //criptografia do email

                $email->Port = $this->porta;

                $email->CharSet = 'UTF-8';

                //Informações de quem enviou

                $email->setFrom($this->emailRemetente, $this->nomeRemetente);

                $email->addReplyTo($this->emailRemetente);

                //endereço para qual será enviado o email

                $email->addAddress($this->emailDestinatario);

                //define se é html

                $email->isHTML(true);

                //Corpo do e-mail

                $email->Subject = "✨ Confirmação da castração do(a) $this->nomeAnimal ✨";     //Assunto / Título

                //conteúdo

                $email->Body = $Conteudo;

                $email->AltBody = 'Para conseguir ver esse e-mail corretamente,

                use um visualizador de e-mail com suporte a HTML';

                //envia

                $email->send();

                echo "Mensagem enviada com sucesso";

            }

            catch(Exception $e)

            {

                echo 'Houve um erro: ' . $e;

            }

        }

        function enviarAviso()

        {

            //Conteúdo da mensagem enviada

            $Conteudo =

            "

            <p><b>Olá, $this->nomeDestinatario.</b><br/>

            Notamos que você não compareceu a castração de $this->nomeAnimal agendada para ".date('d-m-Y',strtotime($this->data)).",

            às ".date('H:i',strtotime($this->data))."</p>

            <p>Por este motivo, foi dado a você uma penalidade que o impede de solicitar mais castrações por um determinado período de tempo.<br/>

            Caso haja um motivo plausível para este ocorrido, entre em contato com a clínica o mais breve possível para a punição ser retirada e uma nova data ser definida.

            </p>

            ";

            $email = new PHPMailer(true);

            try{

                //Configurações

                $email->isSMTP();

                $email->Host = $this->host;

                $email->SMTPAuth = true;

                $email->Username = $this->emailRemetente;

                $email->Password = $this->senhaRemetente;

                $email->SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS;           //criptografia do email

                $email->Port = $this->porta;

                $email->CharSet = 'UTF-8';

                //Informações de quem enviou

                $email->setFrom($this->emailRemetente, $this->nomeRemetente);

                $email->addReplyTo($this->emailRemetente);

                //endereço para qual será enviado o email

                $email->addAddress($this->emailDestinatario);

                //define se é html

                $email->isHTML(true);

                //Corpo do e-mail

                $email->Subject = "Ausência na castração.";     //Assunto / Título

                //conteúdo

                $email->Body = $Conteudo;

                $email->AltBody = 'Para conseguir ver esse e-mail corretamente,

                use um visualizador de e-mail com suporte a HTML';

                //envia

                $email->send();

                echo "Mensagem enviada com sucesso";

            }

            catch(Exception $e)

            {

                echo 'Houve um erro: ' . $e;

            }

        }

        function enviarRecuperacao()

        {

            //Conteúdo da mensagem enviada

            $Conteudo =

            "

            <p><b>Recuperação de senha</b><br/>

            Foi solicitada a recuperação da senha na sua conta</p>

            <p>Codigo de Confirmação: <b>$this->codsenha</b></p>

            ";

            $email = new PHPMailer(true);

            try{

                //Configurações

                $email->isSMTP();

                $email->SMTPDebug = 0;

                $email->Host = $this->host;

                $email->SMTPAuth = true;

                $email->Username = $this->emailRemetente;

                $email->Password = $this->senhaRemetente;

                $email->SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS;           //criptografia do email

                $email->Port = $this->porta;

                $email->CharSet = 'UTF-8';

                //Informações de quem enviou

                $email->setFrom($this->emailRemetente, $this->nomeRemetente);

                $email->addReplyTo($this->emailRemetente);

                //endereço para qual será enviado o email

                $email->addAddress($this->emailDestinatario);

                //define se é html

                $email->isHTML(true);

                //Corpo do e-mail

                $email->Subject = "Recuperar senha";     //Assunto / Título

                //conteúdo

                $email->Body = $Conteudo;

                $email->AltBody = "Foi soliciatada a recuperação da senha na sua conta \n\nCódigo de recuperação: $this->codsenha";

                //envia

                $email->send();

                echo "Mensagem enviada com sucesso";

            }

            catch(Exception $e)

            {

                echo 'Houve um erro: ' . $e;

            }

        }

    }

?>

**Login**

<?php

    class Login{

        //Atributos

        private $idlogin;

        private $nome;

        private $email;

        private $senha;

        private $nivelacesso;

        private $codsenha;

        //Método get

        function \_\_get($atributo)

        {

            return $this->$atributo;

        }

        //Método set

        function \_\_set($atributo, $valor)

        {

            $this->$atributo = $valor;

        }

        //Método construtor

        function \_\_construct()

        {

            //Importando arquivo de conexão com banco de dados

            include\_once "Conexao.php";

        }

        //Método cadastrar

        function cadastrar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para cadastrar

            $cmd = $con->prepare("INSERT INTO login (nome, email, senha, nivelacesso)

                                    VALUES (:nome, :email, :senha, :nivelacesso)");

            //Parâmetros SQL

            $cmd->bindParam(":nome",        $this->nome);

            $cmd->bindParam(":email",       $this->email);

            $cmd->bindParam(":senha",       $this->senha);

            $cmd->bindParam(":nivelacesso", $this->nivelacesso);

            //Executando o comando SQL

            $cmd->execute();

            return $con->lastInsertId();

        }

        //Método consultar

        function consultar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            $cmd = $con->prepare("SELECT \* FROM login");

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        //Método excluir

        function excluir()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para deletar

            $cmd = $con->prepare("DELETE FROM login WHERE idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":idlogin", $this->idlogin);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Atualizar

        function atualizar(){

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

            $cmd = $con->prepare("UPDATE login SET nome = :nome, email = :email, senha = :senha WHERE idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":nome",    $this->nome);

            $cmd->bindParam(":email",   $this->email);

            $cmd->bindParam(":senha",   $this->senha);

            $cmd->bindParam(":idlogin", $this->idlogin);

            //Executando o comando SQL

            $cmd->execute();

        }

        function atualizarLogin()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

            $cmd = $con->prepare("UPDATE login SET nome = :nome, email = :email WHERE idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":nome",    $this->nome);

            $cmd->bindParam(":email",   $this->email);

            $cmd->bindParam(":idlogin", $this->idlogin);

            //Executando o comando SQL

            $cmd->execute();

        }

        function alterarSenha()

        {

            $con = Conexao::conectar();

            $cmd = $con->prepare("UPDATE login SET senha = :senha WHERE idlogin = :idlogin");

            $cmd->bindParam(":senha",   $this->senha);

            $cmd->bindParam(":idlogin", $this->idlogin);

            $cmd->execute();

        }

        //Método Retornar

        function retornar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT \* FROM login WHERE idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":idlogin", $this->idlogin);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function logar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT \* FROM login WHERE email = :email");

            //Parâmetros SQL

            $cmd->bindParam(":email", $this->email);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function retornarUsuario()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT \* FROM usuario JOIN login ON usuario.idlogin = login.idlogin WHERE login.idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":idlogin", $this->idlogin);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function retornarClinica()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT \* FROM clinica JOIN login ON clinica.idlogin = login.idlogin WHERE login.idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":idlogin", $this->idlogin);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function gerarCodigo()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("UPDATE login SET codsenha = :codsenha WHERE email = :email OR idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":email", $this->email);

            $cmd->bindParam(":idlogin", $this->idlogin);

            $cmd->bindParam(":codsenha", $this->codsenha);

            //Executando o comando SQL

            $cmd->execute();

        }

        function confirmarCodigo()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT idlogin, codsenha FROM login WHERE email = :email");

            //Parâmetros SQL

            $cmd->bindParam(":email", $this->email);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function redefinirSenha()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("UPDATE login SET senha = :senha, codsenha = null WHERE idlogin = :idlogin");

            //Parâmetros SQL

            $cmd->bindParam(":idlogin", $this->idlogin);

            $cmd->bindParam(":senha", $this->senha);

            //Executando o comando SQL

            $cmd->execute();

        }

    }

?>

**Raca**

<?php

    class Raca

    {

        //Atributos

        private $idraca;

        private $raca;

        private $tipoespecie;

        function \_\_get($atributo)

        {

            return $this->$atributo;

        }

        //Método set

        function \_\_set($atributo, $valor)

        {

            $this->$atributo = $valor;

        }

        //Método construtor

        function \_\_construct()

        {

            //Importando arquivo de conexão com banco de dados

            include\_once "Conexao.php";

        }

        //Método Cadastrar

        function cadastrar()

        {

            $con = Conexao::conectar();

            //Preparar comando SQL para cadastrar

            $cmd = $con->prepare("INSERT INTO raca (raca, tipoespecie) VALUES (:raca, :tipoespecie)");

            //Parâmetros SQL

            $cmd->bindParam(":raca",        $this->raca);

            $cmd->bindParam(":tipoespecie", $this->tipoespecie);

            //executa o comando SQL

            $cmd->execute();

        }

        //Método Consultar

        function consultar()

        {

            //Conectar

            $con = Conexao::conectar();

            //Preparar o comando SQL para cadastrar

            $cmd = $con->prepare("SELECT \* FROM raca");

            //Executando o comando SQL

            $cmd->execute();

            //Retorna os usuários em forma de objeto de dentro de uma matriz

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        function consultarRaca()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            $cmd = $con->prepare("SELECT \* FROM raca WHERE tipoespecie in(:tipoespecie,2) ");

            //Parâmetros SQL

            $cmd->bindParam(":tipoespecie", $this->tipoespecie);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        //Método Excluir

        function excluir()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para deletar

            $cmd = $con->prepare("DELETE FROM raca WHERE idraca = :idraca");

            //Parâmetros SQL

            $cmd->bindParam(":idraca", $this->idraca);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Atualizar

        function atualizar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

            $cmd = $con->prepare("UPDATE raca SET raca = :raca, tipoespecie = :tipoespecie WHERE idraca = :idraca");

            //Parâmetros SQL

            $cmd->bindParam(":idraca",      $this->idraca);

            $cmd->bindParam(":raca",        $this->raca);

            $cmd->bindParam(":tipoespecie", $this->tipoespecie);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Retornar

        function retornar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT \* FROM raca WHERE idraca = :idraca");

            //Parâmetros SQL

            $cmd->bindParam(":idraca", $this->idraca);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

    }

?>

**Usuario**

<?php

    class Usuario{

        //Atributos

        private $idusuario;

        private $idlogin;

        private $rg;

        private $cpf;

        private $beneficio;

        private $telefone;

        private $celular;

        private $punicao;

        private $usurua;

        private $usubairro;

        private $usunumero;

        private $usucep;

        private $nis;

        private $whatsapp;

        private $doccomprovante;

        //private $docprotetor;

        private $quantcastracoes;

        //Método get

        function \_\_get($atributo)

        {

            return $this->$atributo;

        }

        //Método set

        function \_\_set($atributo, $valor)

        {

            $this->$atributo = $valor;

        }

        //Método construtor

        function \_\_construct()

        {

            //Importando arquivo de conexão com banco de dados

            include\_once "Conexao.php";

        }

        //Método cadastrar

        function cadastrar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para cadastrar

            $cmd = $con->prepare("INSERT INTO usuario (idlogin, rg, cpf, beneficio, telefone, celular, punicao, usurua, usubairro, usunumero, usucep, nis, whatsapp, doccomprovante, docprotetor, quantcastracoes)

                                    VALUES (:idlogin, :rg, :cpf, :beneficio, NULLIF(:telefone,''), :celular, :punicao, :usurua, :usubairro, :usunumero, :usucep, NULLIF(:nis,''), :whatsapp, :doccomprovante, /\*NULLIF(:docprotetor,''),\*/ :quantcastracoes)");

            //Parâmetros SQL

            $cmd->bindParam(":idlogin",         $this->idlogin);

            $cmd->bindParam(":rg",              $this->rg);

            $cmd->bindParam(":cpf",             $this->cpf);

            $cmd->bindParam(":beneficio",       $this->beneficio);

            $cmd->bindParam(":telefone",        $this->telefone);

            $cmd->bindParam(":celular",         $this->celular);

            $cmd->bindParam(":punicao",         $this->punicao);

            $cmd->bindParam(":usurua",          $this->usurua);

            $cmd->bindParam(":usubairro",       $this->usubairro);

            $cmd->bindParam(":usunumero",       $this->usunumero);

            $cmd->bindParam(":usucep",          $this->usucep);

            $cmd->bindParam(":nis",             $this->nis);

            $cmd->bindParam(":whatsapp",        $this->whatsapp);

            $cmd->bindParam(":doccomprovante",  $this->doccomprovante);

            //$cmd->bindParam(":docprotetor",     $this->docprotetor);

            $cmd->bindParam(":quantcastracoes", $this->quantcastracoes);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método consultar

        function consultar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para consultar

            $cmd = $con->prepare("SELECT idusuario, login.idlogin, doccomprovante, nome, email, cpf, rg, beneficio, nis, telefone, celular, usucep, usubairro, usurua, usunumero, punicao

                                    FROM usuario JOIN login ON usuario.idlogin = login.idlogin");

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetchAll(PDO::FETCH\_OBJ);

        }

        //Método excluir

        function excluir()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para deletar

            $cmd = $con->prepare("DELETE FROM usuario WHERE idusuario = :idusuario");

            //Parâmetros SQL

            $cmd->bindParam(":idusuario", $this->idusuario);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Atualizar

        function atualizar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar o comando SQL para atualizar

           /\* $cmd = $con->prepare("UPDATE usuario SET

                                                    rg = :rg,

                                                    cpf = :cpf,

                                                    beneficio = :beneficio,

                                                    telefone = NULLIF(:telefone,''),

                                                    celular = :celular,

                                                    punicao = :punicao,

                                                    usurua = :usurua,

                                                    usubairro = :usubairro,

                                                    usunumero = :usunumero,

                                                    usucep = :usucep,

                                                    nis = NULLIF(:nis,''),

                                                    whatsapp = :whatsapp,

                                                    doccomprovante = :doccomprovante,

                                                    docprotetor = NULLIF(:docprotetor,''),

                                                    quantcastracoes = :quantcastracoes

                                      WHERE idusuario = :idusuario");\*/

            $cmd = $con->prepare("UPDATE usuario SET

                                                rg = :rg,

                                                cpf = :cpf,

                                                beneficio = :beneficio,

                                                telefone = NULLIF(:telefone,''),

                                                celular = :celular,

                                                usurua = :usurua,

                                                usubairro = :usubairro,

                                                usunumero = :usunumero,

                                                usucep = :usucep,

                                                nis = NULLIF(:nis,''),

                                                punicao = :punicao

                                            WHERE idusuario = :idusuario");

            //Parâmetros SQL

            $cmd->bindParam(":rg",              $this->rg);

            $cmd->bindParam(":cpf",             $this->cpf);

            $cmd->bindParam(":beneficio",       $this->beneficio);

            $cmd->bindParam(":telefone",        $this->telefone);

            $cmd->bindParam(":celular",         $this->celular);

           // $cmd->bindParam(":punicao",          $this->punicao);

            $cmd->bindParam(":usurua",          $this->usurua);

            $cmd->bindParam(":usubairro",       $this->usubairro);

            $cmd->bindParam(":usunumero",       $this->usunumero);

            $cmd->bindParam(":usucep",          $this->usucep);

            $cmd->bindParam(":nis",             $this->nis);

            /\*$cmd->bindParam(":whatsapp",        $this->whatsapp);

            $cmd->bindParam(":doccomprovante",  $this->doccomprovante);

            $cmd->bindParam(":docprotetor",     $this->docprotetor);

            $cmd->bindParam(":quantcastracoes", $this->quantcastracoes);\*/

            $cmd->bindParam(":punicao",         $this->punicao);

            $cmd->bindParam(":idusuario",       $this->idusuario);

            //Executando o comando SQL

            $cmd->execute();

        }

        //Método Retornar

        function retornar()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT \* FROM usuario JOIN login ON usuario.idlogin = login.idlogin WHERE idusuario = :idusuario");

            //Parâmetros SQL

            $cmd->bindParam(":idusuario", $this->idusuario);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function verificarNis()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT nis FROM usuario WHERE nis = :nis");

            //Parâmetros SQL

            $cmd->bindParam(":nis", $this->nis);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        function verificarCPF()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT cpf FROM usuario WHERE cpf = :cpf");

            //Parâmetros SQL

            $cmd->bindParam(":cpf", $this->cpf);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        //Método para punição

        function aplicarPunicao()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("UPDATE usuario SET punicao = :punicao WHERE idusuario = :idusuario");

            //Parâmetros SQL

            $cmd->bindParam(":punicao",   $this->punicao);

            $cmd->bindParam(":idusuario", $this->idusuario);

            //Executando o comando SQL

            $cmd->execute();

        }

        function retornarQuantCastracao()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para retornar

            $cmd = $con->prepare("SELECT quantcastracoes FROM usuario WHERE idusuario = :idusuario");

            //Parâmetros SQL

            $cmd->bindParam(":idusuario", $this->idusuario);

            //Executando o comando SQL

            $cmd->execute();

            return $cmd->fetch(PDO::FETCH\_OBJ);

        }

        //Atualizar quantidade de castrações

        function atualizarQuantCastracoes()

        {

            //Conectando ao banco de dados

            $con = Conexao::conectar();

            //Preparar comando SQL para atualizar

            $cmd = $con->prepare("UPDATE usuario SET quantcastracoes = :quantcastracoes WHERE idusuario = :idusuario");

            //Parâmetros SQL

            $cmd->bindParam(":quantcastracoes", $this->quantcastracoes);

            $cmd->bindParam(":idusuario", $this->idusuario);

            //Executando o comando SQL

            $cmd->execute();

        }

    }

?>

**Exception**

<?php

/\*\*

 \* PHPMailer Exception class.

 \* PHP Version 5.5.

 \*

 \* @see       https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub project

 \*

 \* @author    Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \* @author    Jim Jagielski (jimjag) <jimjag@gmail.com>

 \* @author    Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>

 \* @author    Brent R. Matzelle (original founder)

 \* @copyright 2012 - 2020 Marcus Bointon

 \* @copyright 2010 - 2012 Jim Jagielski

 \* @copyright 2004 - 2009 Andy Prevost

 \* @license   http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License

 \* @note      This program is distributed in the hope that it will be useful - WITHOUT

 \* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

 \* FITNESS FOR A PARTICULAR PURPOSE.

 \*/

namespace PHPMailer\PHPMailer;

/\*\*

 \* PHPMailer exception handler.

 \*

 \* @author Marcus Bointon <phpmailer@synchromedia.co.uk>

 \*/

class Exception extends \Exception

{

    /\*\*

     \* Prettify error message output.

     \*

     \* @return string

     \*/

    public function errorMessage()

    {

        return '<strong>' . htmlspecialchars($this->getMessage(), ENT\_COMPAT | ENT\_HTML401) . "</strong><br />\n";

    }

}

**OAuth**

<?php

/\*\*

 \* PHPMailer - PHP email creation and transport class.

 \* PHP Version 5.5.

 \*

 \* @see       https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub project

 \*

 \* @author    Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \* @author    Jim Jagielski (jimjag) <jimjag@gmail.com>

 \* @author    Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>

 \* @author    Brent R. Matzelle (original founder)

 \* @copyright 2012 - 2020 Marcus Bointon

 \* @copyright 2010 - 2012 Jim Jagielski

 \* @copyright 2004 - 2009 Andy Prevost

 \* @license   http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License

 \* @note      This program is distributed in the hope that it will be useful - WITHOUT

 \* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

 \* FITNESS FOR A PARTICULAR PURPOSE.

 \*/

namespace PHPMailer\PHPMailer;

use League\OAuth2\Client\Grant\RefreshToken;

use League\OAuth2\Client\Provider\AbstractProvider;

use League\OAuth2\Client\Token\AccessToken;

/\*\*

 \* OAuth - OAuth2 authentication wrapper class.

 \* Uses the oauth2-client package from the League of Extraordinary Packages.

 \*

 \* @see     http://oauth2-client.thephpleague.com

 \*

 \* @author  Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \*/

class OAuth implements OAuthTokenProvider

{

    /\*\*

     \* An instance of the League OAuth Client Provider.

     \*

     \* @var AbstractProvider

     \*/

    protected $provider;

    /\*\*

     \* The current OAuth access token.

     \*

     \* @var AccessToken

     \*/

    protected $oauthToken;

    /\*\*

     \* The user's email address, usually used as the login ID

     \* and also the from address when sending email.

     \*

     \* @var string

     \*/

    protected $oauthUserEmail = '';

    /\*\*

     \* The client secret, generated in the app definition of the service you're connecting to.

     \*

     \* @var string

     \*/

    protected $oauthClientSecret = '';

    /\*\*

     \* The client ID, generated in the app definition of the service you're connecting to.

     \*

     \* @var string

     \*/

    protected $oauthClientId = '';

    /\*\*

     \* The refresh token, used to obtain new AccessTokens.

     \*

     \* @var string

     \*/

    protected $oauthRefreshToken = '';

    /\*\*

     \* OAuth constructor.

     \*

     \* @param array $options Associative array containing

     \*                       `provider`, `userName`, `clientSecret`, `clientId` and `refreshToken` elements

     \*/

    public function \_\_construct($options)

    {

        $this->provider = $options['provider'];

        $this->oauthUserEmail = $options['userName'];

        $this->oauthClientSecret = $options['clientSecret'];

        $this->oauthClientId = $options['clientId'];

        $this->oauthRefreshToken = $options['refreshToken'];

    }

    /\*\*

     \* Get a new RefreshToken.

     \*

     \* @return RefreshToken

     \*/

    protected function getGrant()

    {

        return new RefreshToken();

    }

    /\*\*

     \* Get a new AccessToken.

     \*

     \* @return AccessToken

     \*/

    protected function getToken()

    {

        return $this->provider->getAccessToken(

            $this->getGrant(),

            ['refresh\_token' => $this->oauthRefreshToken]

        );

    }

    /\*\*

     \* Generate a base64-encoded OAuth token.

     \*

     \* @return string

     \*/

    public function getOauth64()

    {

        //Get a new token if it's not available or has expired

        if (null === $this->oauthToken || $this->oauthToken->hasExpired()) {

            $this->oauthToken = $this->getToken();

        }

        return base64\_encode(

            'user=' .

            $this->oauthUserEmail .

            "\001auth=Bearer " .

            $this->oauthToken .

            "\001\001"

        );

    }

}

**OAuthTokenProvider**

<?php

/\*\*

 \* PHPMailer - PHP email creation and transport class.

 \* PHP Version 5.5.

 \*

 \* @see https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub project

 \*

 \* @author    Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \* @author    Jim Jagielski (jimjag) <jimjag@gmail.com>

 \* @author    Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>

 \* @author    Brent R. Matzelle (original founder)

 \* @copyright 2012 - 2020 Marcus Bointon

 \* @copyright 2010 - 2012 Jim Jagielski

 \* @copyright 2004 - 2009 Andy Prevost

 \* @license   http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License

 \* @note      This program is distributed in the hope that it will be useful - WITHOUT

 \* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

 \* FITNESS FOR A PARTICULAR PURPOSE.

 \*/

namespace PHPMailer\PHPMailer;

/\*\*

 \* OAuthTokenProvider - OAuth2 token provider interface.

 \* Provides base64 encoded OAuth2 auth strings for SMTP authentication.

 \*

 \* @see     OAuth

 \* @see     SMTP::authenticate()

 \*

 \* @author  Peter Scopes (pdscopes)

 \* @author  Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \*/

interface OAuthTokenProvider

{

    /\*\*

     \* Generate a base64-encoded OAuth token ensuring that the access token has not expired.

     \* The string to be base 64 encoded should be in the form:

     \* "user=<user\_email\_address>\001auth=Bearer <access\_token>\001\001"

     \*

     \* @return string

     \*/

    public function getOauth64();

}

**PHPMailer**

**<?php**

**/\*\***

**\* PHPMailer - PHP email creation and transport class.**

**\* PHP Version 5.5.**

**\***

**\* @see https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub project**

**\***

**\* @author Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>**

**\* @author Jim Jagielski (jimjag) <jimjag@gmail.com>**

**\* @author Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>**

**\* @author Brent R. Matzelle (original founder)**

**\* @copyright 2012 - 2020 Marcus Bointon**

**\* @copyright 2010 - 2012 Jim Jagielski**

**\* @copyright 2004 - 2009 Andy Prevost**

**\* @license http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License**

**\* @note This program is distributed in the hope that it will be useful - WITHOUT**

**\* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or**

**\* FITNESS FOR A PARTICULAR PURPOSE.**

**\*/**

**namespace PHPMailer\PHPMailer;**

**/\*\***

**\* PHPMailer - PHP email creation and transport class.**

**\***

**\* @author Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>**

**\* @author Jim Jagielski (jimjag) <jimjag@gmail.com>**

**\* @author Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>**

**\* @author Brent R. Matzelle (original founder)**

**\*/**

**class PHPMailer**

**{**

**const CHARSET\_ASCII = 'us-ascii';**

**const CHARSET\_ISO88591 = 'iso-8859-1';**

**const CHARSET\_UTF8 = 'utf-8';**

**const CONTENT\_TYPE\_PLAINTEXT = 'text/plain';**

**const CONTENT\_TYPE\_TEXT\_CALENDAR = 'text/calendar';**

**const CONTENT\_TYPE\_TEXT\_HTML = 'text/html';**

**const CONTENT\_TYPE\_MULTIPART\_ALTERNATIVE = 'multipart/alternative';**

**const CONTENT\_TYPE\_MULTIPART\_MIXED = 'multipart/mixed';**

**const CONTENT\_TYPE\_MULTIPART\_RELATED = 'multipart/related';**

**const ENCODING\_7BIT = '7bit';**

**const ENCODING\_8BIT = '8bit';**

**const ENCODING\_BASE64 = 'base64';**

**const ENCODING\_BINARY = 'binary';**

**const ENCODING\_QUOTED\_PRINTABLE = 'quoted-printable';**

**const ENCRYPTION\_STARTTLS = 'tls';**

**const ENCRYPTION\_SMTPS = 'ssl';**

**const ICAL\_METHOD\_REQUEST = 'REQUEST';**

**const ICAL\_METHOD\_PUBLISH = 'PUBLISH';**

**const ICAL\_METHOD\_REPLY = 'REPLY';**

**const ICAL\_METHOD\_ADD = 'ADD';**

**const ICAL\_METHOD\_CANCEL = 'CANCEL';**

**const ICAL\_METHOD\_REFRESH = 'REFRESH';**

**const ICAL\_METHOD\_COUNTER = 'COUNTER';**

**const ICAL\_METHOD\_DECLINECOUNTER = 'DECLINECOUNTER';**

**/\*\***

**\* Email priority.**

**\* Options: null (default), 1 = High, 3 = Normal, 5 = low.**

**\* When null, the header is not set at all.**

**\***

**\* @var int|null**

**\*/**

**public $Priority;**

**/\*\***

**\* The character set of the message.**

**\***

**\* @var string**

**\*/**

**public $CharSet = self::CHARSET\_ISO88591;**

**/\*\***

**\* The MIME Content-type of the message.**

**\***

**\* @var string**

**\*/**

**public $ContentType = self::CONTENT\_TYPE\_PLAINTEXT;**

**/\*\***

**\* The message encoding.**

**\* Options: "8bit", "7bit", "binary", "base64", and "quoted-printable".**

**\***

**\* @var string**

**\*/**

**public $Encoding = self::ENCODING\_8BIT;**

**/\*\***

**\* Holds the most recent mailer error message.**

**\***

**\* @var string**

**\*/**

**public $ErrorInfo = '';**

**/\*\***

**\* The From email address for the message.**

**\***

**\* @var string**

**\*/**

**public $From = '';**

**/\*\***

**\* The From name of the message.**

**\***

**\* @var string**

**\*/**

**public $FromName = '';**

**/\*\***

**\* The envelope sender of the message.**

**\* This will usually be turned into a Return-Path header by the receiver,**

**\* and is the address that bounces will be sent to.**

**\* If not empty, will be passed via `-f` to sendmail or as the 'MAIL FROM' value over SMTP.**

**\***

**\* @var string**

**\*/**

**public $Sender = '';**

**/\*\***

**\* The Subject of the message.**

**\***

**\* @var string**

**\*/**

**public $Subject = '';**

**/\*\***

**\* An HTML or plain text message body.**

**\* If HTML then call isHTML(true).**

**\***

**\* @var string**

**\*/**

**public $Body = '';**

**/\*\***

**\* The plain-text message body.**

**\* This body can be read by mail clients that do not have HTML email**

**\* capability such as mutt & Eudora.**

**\* Clients that can read HTML will view the normal Body.**

**\***

**\* @var string**

**\*/**

**public $AltBody = '';**

**/\*\***

**\* An iCal message part body.**

**\* Only supported in simple alt or alt\_inline message types**

**\* To generate iCal event structures, use classes like EasyPeasyICS or iCalcreator.**

**\***

**\* @see http://sprain.ch/blog/downloads/php-class-easypeasyics-create-ical-files-with-php/**

**\* @see http://kigkonsult.se/iCalcreator/**

**\***

**\* @var string**

**\*/**

**public $Ical = '';**

**/\*\***

**\* Value-array of "method" in Contenttype header "text/calendar"**

**\***

**\* @var string[]**

**\*/**

**protected static $IcalMethods = [**

**self::ICAL\_METHOD\_REQUEST,**

**self::ICAL\_METHOD\_PUBLISH,**

**self::ICAL\_METHOD\_REPLY,**

**self::ICAL\_METHOD\_ADD,**

**self::ICAL\_METHOD\_CANCEL,**

**self::ICAL\_METHOD\_REFRESH,**

**self::ICAL\_METHOD\_COUNTER,**

**self::ICAL\_METHOD\_DECLINECOUNTER,**

**];**

**/\*\***

**\* The complete compiled MIME message body.**

**\***

**\* @var string**

**\*/**

**protected $MIMEBody = '';**

**/\*\***

**\* The complete compiled MIME message headers.**

**\***

**\* @var string**

**\*/**

**protected $MIMEHeader = '';**

**/\*\***

**\* Extra headers that createHeader() doesn't fold in.**

**\***

**\* @var string**

**\*/**

**protected $mailHeader = '';**

**/\*\***

**\* Word-wrap the message body to this number of chars.**

**\* Set to 0 to not wrap. A useful value here is 78, for RFC2822 section 2.1.1 compliance.**

**\***

**\* @see static::STD\_LINE\_LENGTH**

**\***

**\* @var int**

**\*/**

**public $WordWrap = 0;**

**/\*\***

**\* Which method to use to send mail.**

**\* Options: "mail", "sendmail", or "smtp".**

**\***

**\* @var string**

**\*/**

**public $Mailer = 'mail';**

**/\*\***

**\* The path to the sendmail program.**

**\***

**\* @var string**

**\*/**

**public $Sendmail = '/usr/sbin/sendmail';**

**/\*\***

**\* Whether mail() uses a fully sendmail-compatible MTA.**

**\* One which supports sendmail's "-oi -f" options.**

**\***

**\* @var bool**

**\*/**

**public $UseSendmailOptions = true;**

**/\*\***

**\* The email address that a reading confirmation should be sent to, also known as read receipt.**

**\***

**\* @var string**

**\*/**

**public $ConfirmReadingTo = '';**

**/\*\***

**\* The hostname to use in the Message-ID header and as default HELO string.**

**\* If empty, PHPMailer attempts to find one with, in order,**

**\* $\_SERVER['SERVER\_NAME'], gethostname(), php\_uname('n'), or the value**

**\* 'localhost.localdomain'.**

**\***

**\* @see PHPMailer::$Helo**

**\***

**\* @var string**

**\*/**

**public $Hostname = '';**

**/\*\***

**\* An ID to be used in the Message-ID header.**

**\* If empty, a unique id will be generated.**

**\* You can set your own, but it must be in the format "<id@domain>",**

**\* as defined in RFC5322 section 3.6.4 or it will be ignored.**

**\***

**\* @see https://tools.ietf.org/html/rfc5322#section-3.6.4**

**\***

**\* @var string**

**\*/**

**public $MessageID = '';**

**/\*\***

**\* The message Date to be used in the Date header.**

**\* If empty, the current date will be added.**

**\***

**\* @var string**

**\*/**

**public $MessageDate = '';**

**/\*\***

**\* SMTP hosts.**

**\* Either a single hostname or multiple semicolon-delimited hostnames.**

**\* You can also specify a different port**

**\* for each host by using this format: [hostname:port]**

**\* (e.g. "smtp1.example.com:25;smtp2.example.com").**

**\* You can also specify encryption type, for example:**

**\* (e.g. "tls://smtp1.example.com:587;ssl://smtp2.example.com:465").**

**\* Hosts will be tried in order.**

**\***

**\* @var string**

**\*/**

**public $Host = 'localhost';**

**/\*\***

**\* The default SMTP server port.**

**\***

**\* @var int**

**\*/**

**public $Port = 25;**

**/\*\***

**\* The SMTP HELO/EHLO name used for the SMTP connection.**

**\* Default is $Hostname. If $Hostname is empty, PHPMailer attempts to find**

**\* one with the same method described above for $Hostname.**

**\***

**\* @see PHPMailer::$Hostname**

**\***

**\* @var string**

**\*/**

**public $Helo = '';**

**/\*\***

**\* What kind of encryption to use on the SMTP connection.**

**\* Options: '', static::ENCRYPTION\_STARTTLS, or static::ENCRYPTION\_SMTPS.**

**\***

**\* @var string**

**\*/**

**public $SMTPSecure = '';**

**/\*\***

**\* Whether to enable TLS encryption automatically if a server supports it,**

**\* even if `SMTPSecure` is not set to 'tls'.**

**\* Be aware that in PHP >= 5.6 this requires that the server's certificates are valid.**

**\***

**\* @var bool**

**\*/**

**public $SMTPAutoTLS = true;**

**/\*\***

**\* Whether to use SMTP authentication.**

**\* Uses the Username and Password properties.**

**\***

**\* @see PHPMailer::$Username**

**\* @see PHPMailer::$Password**

**\***

**\* @var bool**

**\*/**

**public $SMTPAuth = false;**

**/\*\***

**\* Options array passed to stream\_context\_create when connecting via SMTP.**

**\***

**\* @var array**

**\*/**

**public $SMTPOptions = [];**

**/\*\***

**\* SMTP username.**

**\***

**\* @var string**

**\*/**

**public $Username = '';**

**/\*\***

**\* SMTP password.**

**\***

**\* @var string**

**\*/**

**public $Password = '';**

**/\*\***

**\* SMTP auth type.**

**\* Options are CRAM-MD5, LOGIN, PLAIN, XOAUTH2, attempted in that order if not specified.**

**\***

**\* @var string**

**\*/**

**public $AuthType = '';**

**/\*\***

**\* An implementation of the PHPMailer OAuthTokenProvider interface.**

**\***

**\* @var OAuthTokenProvider**

**\*/**

**protected $oauth;**

**/\*\***

**\* The SMTP server timeout in seconds.**

**\* Default of 5 minutes (300sec) is from RFC2821 section 4.5.3.2.**

**\***

**\* @var int**

**\*/**

**public $Timeout = 300;**

**/\*\***

**\* Comma separated list of DSN notifications**

**\* 'NEVER' under no circumstances a DSN must be returned to the sender.**

**\* If you use NEVER all other notifications will be ignored.**

**\* 'SUCCESS' will notify you when your mail has arrived at its destination.**

**\* 'FAILURE' will arrive if an error occurred during delivery.**

**\* 'DELAY' will notify you if there is an unusual delay in delivery, but the actual**

**\* delivery's outcome (success or failure) is not yet decided.**

**\***

**\* @see https://tools.ietf.org/html/rfc3461 See section 4.1 for more information about NOTIFY**

**\*/**

**public $dsn = '';**

**/\*\***

**\* SMTP class debug output mode.**

**\* Debug output level.**

**\* Options:**

**\* @see SMTP::DEBUG\_OFF: No output**

**\* @see SMTP::DEBUG\_CLIENT: Client messages**

**\* @see SMTP::DEBUG\_SERVER: Client and server messages**

**\* @see SMTP::DEBUG\_CONNECTION: As SERVER plus connection status**

**\* @see SMTP::DEBUG\_LOWLEVEL: Noisy, low-level data output, rarely needed**

**\***

**\* @see SMTP::$do\_debug**

**\***

**\* @var int**

**\*/**

**public $SMTPDebug = 0;**

**/\*\***

**\* How to handle debug output.**

**\* Options:**

**\* \* `echo` Output plain-text as-is, appropriate for CLI**

**\* \* `html` Output escaped, line breaks converted to `<br>`, appropriate for browser output**

**\* \* `error\_log` Output to error log as configured in php.ini**

**\* By default PHPMailer will use `echo` if run from a `cli` or `cli-server` SAPI, `html` otherwise.**

**\* Alternatively, you can provide a callable expecting two params: a message string and the debug level:**

**\***

**\* ```php**

**\* $mail->Debugoutput = function($str, $level) {echo "debug level $level; message: $str";};**

**\* ```**

**\***

**\* Alternatively, you can pass in an instance of a PSR-3 compatible logger, though only `debug`**

**\* level output is used:**

**\***

**\* ```php**

**\* $mail->Debugoutput = new myPsr3Logger;**

**\* ```**

**\***

**\* @see SMTP::$Debugoutput**

**\***

**\* @var string|callable|\Psr\Log\LoggerInterface**

**\*/**

**public $Debugoutput = 'echo';**

**/\*\***

**\* Whether to keep the SMTP connection open after each message.**

**\* If this is set to true then the connection will remain open after a send,**

**\* and closing the connection will require an explicit call to smtpClose().**

**\* It's a good idea to use this if you are sending multiple messages as it reduces overhead.**

**\* See the mailing list example for how to use it.**

**\***

**\* @var bool**

**\*/**

**public $SMTPKeepAlive = false;**

**/\*\***

**\* Whether to split multiple to addresses into multiple messages**

**\* or send them all in one message.**

**\* Only supported in `mail` and `sendmail` transports, not in SMTP.**

**\***

**\* @var bool**

**\***

**\* @deprecated 6.0.0 PHPMailer isn't a mailing list manager!**

**\*/**

**public $SingleTo = false;**

**/\*\***

**\* Storage for addresses when SingleTo is enabled.**

**\***

**\* @var array**

**\*/**

**protected $SingleToArray = [];**

**/\*\***

**\* Whether to generate VERP addresses on send.**

**\* Only applicable when sending via SMTP.**

**\***

**\* @see https://en.wikipedia.org/wiki/Variable\_envelope\_return\_path**

**\* @see http://www.postfix.org/VERP\_README.html Postfix VERP info**

**\***

**\* @var bool**

**\*/**

**public $do\_verp = false;**

**/\*\***

**\* Whether to allow sending messages with an empty body.**

**\***

**\* @var bool**

**\*/**

**public $AllowEmpty = false;**

**/\*\***

**\* DKIM selector.**

**\***

**\* @var string**

**\*/**

**public $DKIM\_selector = '';**

**/\*\***

**\* DKIM Identity.**

**\* Usually the email address used as the source of the email.**

**\***

**\* @var string**

**\*/**

**public $DKIM\_identity = '';**

**/\*\***

**\* DKIM passphrase.**

**\* Used if your key is encrypted.**

**\***

**\* @var string**

**\*/**

**public $DKIM\_passphrase = '';**

**/\*\***

**\* DKIM signing domain name.**

**\***

**\* @example 'example.com'**

**\***

**\* @var string**

**\*/**

**public $DKIM\_domain = '';**

**/\*\***

**\* DKIM Copy header field values for diagnostic use.**

**\***

**\* @var bool**

**\*/**

**public $DKIM\_copyHeaderFields = true;**

**/\*\***

**\* DKIM Extra signing headers.**

**\***

**\* @example ['List-Unsubscribe', 'List-Help']**

**\***

**\* @var array**

**\*/**

**public $DKIM\_extraHeaders = [];**

**/\*\***

**\* DKIM private key file path.**

**\***

**\* @var string**

**\*/**

**public $DKIM\_private = '';**

**/\*\***

**\* DKIM private key string.**

**\***

**\* If set, takes precedence over `$DKIM\_private`.**

**\***

**\* @var string**

**\*/**

**public $DKIM\_private\_string = '';**

**/\*\***

**\* Callback Action function name.**

**\***

**\* The function that handles the result of the send email action.**

**\* It is called out by send() for each email sent.**

**\***

**\* Value can be any php callable: http://www.php.net/is\_callable**

**\***

**\* Parameters:**

**\* bool $result result of the send action**

**\* array $to email addresses of the recipients**

**\* array $cc cc email addresses**

**\* array $bcc bcc email addresses**

**\* string $subject the subject**

**\* string $body the email body**

**\* string $from email address of sender**

**\* string $extra extra information of possible use**

**\* "smtp\_transaction\_id' => last smtp transaction id**

**\***

**\* @var string**

**\*/**

**public $action\_function = '';**

**/\*\***

**\* What to put in the X-Mailer header.**

**\* Options: An empty string for PHPMailer default, whitespace/null for none, or a string to use.**

**\***

**\* @var string|null**

**\*/**

**public $XMailer = '';**

**/\*\***

**\* Which validator to use by default when validating email addresses.**

**\* May be a callable to inject your own validator, but there are several built-in validators.**

**\* The default validator uses PHP's FILTER\_VALIDATE\_EMAIL filter\_var option.**

**\***

**\* @see PHPMailer::validateAddress()**

**\***

**\* @var string|callable**

**\*/**

**public static $validator = 'php';**

**/\*\***

**\* An instance of the SMTP sender class.**

**\***

**\* @var SMTP**

**\*/**

**protected $smtp;**

**/\*\***

**\* The array of 'to' names and addresses.**

**\***

**\* @var array**

**\*/**

**protected $to = [];**

**/\*\***

**\* The array of 'cc' names and addresses.**

**\***

**\* @var array**

**\*/**

**protected $cc = [];**

**/\*\***

**\* The array of 'bcc' names and addresses.**

**\***

**\* @var array**

**\*/**

**protected $bcc = [];**

**/\*\***

**\* The array of reply-to names and addresses.**

**\***

**\* @var array**

**\*/**

**protected $ReplyTo = [];**

**/\*\***

**\* An array of all kinds of addresses.**

**\* Includes all of $to, $cc, $bcc.**

**\***

**\* @see PHPMailer::$to**

**\* @see PHPMailer::$cc**

**\* @see PHPMailer::$bcc**

**\***

**\* @var array**

**\*/**

**protected $all\_recipients = [];**

**/\*\***

**\* An array of names and addresses queued for validation.**

**\* In send(), valid and non duplicate entries are moved to $all\_recipients**

**\* and one of $to, $cc, or $bcc.**

**\* This array is used only for addresses with IDN.**

**\***

**\* @see PHPMailer::$to**

**\* @see PHPMailer::$cc**

**\* @see PHPMailer::$bcc**

**\* @see PHPMailer::$all\_recipients**

**\***

**\* @var array**

**\*/**

**protected $RecipientsQueue = [];**

**/\*\***

**\* An array of reply-to names and addresses queued for validation.**

**\* In send(), valid and non duplicate entries are moved to $ReplyTo.**

**\* This array is used only for addresses with IDN.**

**\***

**\* @see PHPMailer::$ReplyTo**

**\***

**\* @var array**

**\*/**

**protected $ReplyToQueue = [];**

**/\*\***

**\* The array of attachments.**

**\***

**\* @var array**

**\*/**

**protected $attachment = [];**

**/\*\***

**\* The array of custom headers.**

**\***

**\* @var array**

**\*/**

**protected $CustomHeader = [];**

**/\*\***

**\* The most recent Message-ID (including angular brackets).**

**\***

**\* @var string**

**\*/**

**protected $lastMessageID = '';**

**/\*\***

**\* The message's MIME type.**

**\***

**\* @var string**

**\*/**

**protected $message\_type = '';**

**/\*\***

**\* The array of MIME boundary strings.**

**\***

**\* @var array**

**\*/**

**protected $boundary = [];**

**/\*\***

**\* The array of available text strings for the current language.**

**\***

**\* @var array**

**\*/**

**protected $language = [];**

**/\*\***

**\* The number of errors encountered.**

**\***

**\* @var int**

**\*/**

**protected $error\_count = 0;**

**/\*\***

**\* The S/MIME certificate file path.**

**\***

**\* @var string**

**\*/**

**protected $sign\_cert\_file = '';**

**/\*\***

**\* The S/MIME key file path.**

**\***

**\* @var string**

**\*/**

**protected $sign\_key\_file = '';**

**/\*\***

**\* The optional S/MIME extra certificates ("CA Chain") file path.**

**\***

**\* @var string**

**\*/**

**protected $sign\_extracerts\_file = '';**

**/\*\***

**\* The S/MIME password for the key.**

**\* Used only if the key is encrypted.**

**\***

**\* @var string**

**\*/**

**protected $sign\_key\_pass = '';**

**/\*\***

**\* Whether to throw exceptions for errors.**

**\***

**\* @var bool**

**\*/**

**protected $exceptions = false;**

**/\*\***

**\* Unique ID used for message ID and boundaries.**

**\***

**\* @var string**

**\*/**

**protected $uniqueid = '';**

**/\*\***

**\* The PHPMailer Version number.**

**\***

**\* @var string**

**\*/**

**const VERSION = '6.6.0';**

**/\*\***

**\* Error severity: message only, continue processing.**

**\***

**\* @var int**

**\*/**

**const STOP\_MESSAGE = 0;**

**/\*\***

**\* Error severity: message, likely ok to continue processing.**

**\***

**\* @var int**

**\*/**

**const STOP\_CONTINUE = 1;**

**/\*\***

**\* Error severity: message, plus full stop, critical error reached.**

**\***

**\* @var int**

**\*/**

**const STOP\_CRITICAL = 2;**

**/\*\***

**\* The SMTP standard CRLF line break.**

**\* If you want to change line break format, change static::$LE, not this.**

**\*/**

**const CRLF = "\r\n";**

**/\*\***

**\* "Folding White Space" a white space string used for line folding.**

**\*/**

**const FWS = ' ';**

**/\*\***

**\* SMTP RFC standard line ending; Carriage Return, Line Feed.**

**\***

**\* @var string**

**\*/**

**protected static $LE = self::CRLF;**

**/\*\***

**\* The maximum line length supported by mail().**

**\***

**\* Background: mail() will sometimes corrupt messages**

**\* with headers headers longer than 65 chars, see #818.**

**\***

**\* @var int**

**\*/**

**const MAIL\_MAX\_LINE\_LENGTH = 63;**

**/\*\***

**\* The maximum line length allowed by RFC 2822 section 2.1.1.**

**\***

**\* @var int**

**\*/**

**const MAX\_LINE\_LENGTH = 998;**

**/\*\***

**\* The lower maximum line length allowed by RFC 2822 section 2.1.1.**

**\* This length does NOT include the line break**

**\* 76 means that lines will be 77 or 78 chars depending on whether**

**\* the line break format is LF or CRLF; both are valid.**

**\***

**\* @var int**

**\*/**

**const STD\_LINE\_LENGTH = 76;**

**/\*\***

**\* Constructor.**

**\***

**\* @param bool $exceptions Should we throw external exceptions?**

**\*/**

**public function \_\_construct($exceptions = null)**

**{**

**if (null !== $exceptions) {**

**$this->exceptions = (bool) $exceptions;**

**}**

**//Pick an appropriate debug output format automatically**

**$this->Debugoutput = (strpos(PHP\_SAPI, 'cli') !== false ? 'echo' : 'html');**

**}**

**/\*\***

**\* Destructor.**

**\*/**

**public function \_\_destruct()**

**{**

**//Close any open SMTP connection nicely**

**$this->smtpClose();**

**}**

**/\*\***

**\* Call mail() in a safe\_mode-aware fashion.**

**\* Also, unless sendmail\_path points to sendmail (or something that**

**\* claims to be sendmail), don't pass params (not a perfect fix,**

**\* but it will do).**

**\***

**\* @param string $to To**

**\* @param string $subject Subject**

**\* @param string $body Message Body**

**\* @param string $header Additional Header(s)**

**\* @param string|null $params Params**

**\***

**\* @return bool**

**\*/**

**private function mailPassthru($to, $subject, $body, $header, $params)**

**{**

**//Check overloading of mail function to avoid double-encoding**

**if (ini\_get('mbstring.func\_overload') & 1) {**

**$subject = $this->secureHeader($subject);**

**} else {**

**$subject = $this->encodeHeader($this->secureHeader($subject));**

**}**

**//Calling mail() with null params breaks**

**$this->edebug('Sending with mail()');**

**$this->edebug('Sendmail path: ' . ini\_get('sendmail\_path'));**

**$this->edebug("Envelope sender: {$this->Sender}");**

**$this->edebug("To: {$to}");**

**$this->edebug("Subject: {$subject}");**

**$this->edebug("Headers: {$header}");**

**if (!$this->UseSendmailOptions || null === $params) {**

**$result = @mail($to, $subject, $body, $header);**

**} else {**

**$this->edebug("Additional params: {$params}");**

**$result = @mail($to, $subject, $body, $header, $params);**

**}**

**$this->edebug('Result: ' . ($result ? 'true' : 'false'));**

**return $result;**

**}**

**/\*\***

**\* Output debugging info via a user-defined method.**

**\* Only generates output if debug output is enabled.**

**\***

**\* @see PHPMailer::$Debugoutput**

**\* @see PHPMailer::$SMTPDebug**

**\***

**\* @param string $str**

**\*/**

**protected function edebug($str)**

**{**

**if ($this->SMTPDebug <= 0) {**

**return;**

**}**

**//Is this a PSR-3 logger?**

**if ($this->Debugoutput instanceof \Psr\Log\LoggerInterface) {**

**$this->Debugoutput->debug($str);**

**return;**

**}**

**//Avoid clash with built-in function names**

**if (is\_callable($this->Debugoutput) && !in\_array($this->Debugoutput, ['error\_log', 'html', 'echo'])) {**

**call\_user\_func($this->Debugoutput, $str, $this->SMTPDebug);**

**return;**

**}**

**switch ($this->Debugoutput) {**

**case 'error\_log':**

**//Don't output, just log**

**/\*\* @noinspection ForgottenDebugOutputInspection \*/**

**error\_log($str);**

**break;**

**case 'html':**

**//Cleans up output a bit for a better looking, HTML-safe output**

**echo htmlentities(**

**preg\_replace('/[\r\n]+/', '', $str),**

**ENT\_QUOTES,**

**'UTF-8'**

**), "<br>\n";**

**break;**

**case 'echo':**

**default:**

**//Normalize line breaks**

**$str = preg\_replace('/\r\n|\r/m', "\n", $str);**

**echo gmdate('Y-m-d H:i:s'),**

**"\t",**

**//Trim trailing space**

**trim(**

**//Indent for readability, except for trailing break**

**str\_replace(**

**"\n",**

**"\n \t ",**

**trim($str)**

**)**

**),**

**"\n";**

**}**

**}**

**/\*\***

**\* Sets message type to HTML or plain.**

**\***

**\* @param bool $isHtml True for HTML mode**

**\*/**

**public function isHTML($isHtml = true)**

**{**

**if ($isHtml) {**

**$this->ContentType = static::CONTENT\_TYPE\_TEXT\_HTML;**

**} else {**

**$this->ContentType = static::CONTENT\_TYPE\_PLAINTEXT;**

**}**

**}**

**/\*\***

**\* Send messages using SMTP.**

**\*/**

**public function isSMTP()**

**{**

**$this->Mailer = 'smtp';**

**}**

**/\*\***

**\* Send messages using PHP's mail() function.**

**\*/**

**public function isMail()**

**{**

**$this->Mailer = 'mail';**

**}**

**/\*\***

**\* Send messages using $Sendmail.**

**\*/**

**public function isSendmail()**

**{**

**$ini\_sendmail\_path = ini\_get('sendmail\_path');**

**if (false === stripos($ini\_sendmail\_path, 'sendmail')) {**

**$this->Sendmail = '/usr/sbin/sendmail';**

**} else {**

**$this->Sendmail = $ini\_sendmail\_path;**

**}**

**$this->Mailer = 'sendmail';**

**}**

**/\*\***

**\* Send messages using qmail.**

**\*/**

**public function isQmail()**

**{**

**$ini\_sendmail\_path = ini\_get('sendmail\_path');**

**if (false === stripos($ini\_sendmail\_path, 'qmail')) {**

**$this->Sendmail = '/var/qmail/bin/qmail-inject';**

**} else {**

**$this->Sendmail = $ini\_sendmail\_path;**

**}**

**$this->Mailer = 'qmail';**

**}**

**/\*\***

**\* Add a "To" address.**

**\***

**\* @param string $address The email address to send to**

**\* @param string $name**

**\***

**\* @throws Exception**

**\***

**\* @return bool true on success, false if address already used or invalid in some way**

**\*/**

**public function addAddress($address, $name = '')**

**{**

**return $this->addOrEnqueueAnAddress('to', $address, $name);**

**}**

**/\*\***

**\* Add a "CC" address.**

**\***

**\* @param string $address The email address to send to**

**\* @param string $name**

**\***

**\* @throws Exception**

**\***

**\* @return bool true on success, false if address already used or invalid in some way**

**\*/**

**public function addCC($address, $name = '')**

**{**

**return $this->addOrEnqueueAnAddress('cc', $address, $name);**

**}**

**/\*\***

**\* Add a "BCC" address.**

**\***

**\* @param string $address The email address to send to**

**\* @param string $name**

**\***

**\* @throws Exception**

**\***

**\* @return bool true on success, false if address already used or invalid in some way**

**\*/**

**public function addBCC($address, $name = '')**

**{**

**return $this->addOrEnqueueAnAddress('bcc', $address, $name);**

**}**

**/\*\***

**\* Add a "Reply-To" address.**

**\***

**\* @param string $address The email address to reply to**

**\* @param string $name**

**\***

**\* @throws Exception**

**\***

**\* @return bool true on success, false if address already used or invalid in some way**

**\*/**

**public function addReplyTo($address, $name = '')**

**{**

**return $this->addOrEnqueueAnAddress('Reply-To', $address, $name);**

**}**

**/\*\***

**\* Add an address to one of the recipient arrays or to the ReplyTo array. Because PHPMailer**

**\* can't validate addresses with an IDN without knowing the PHPMailer::$CharSet (that can still**

**\* be modified after calling this function), addition of such addresses is delayed until send().**

**\* Addresses that have been added already return false, but do not throw exceptions.**

**\***

**\* @param string $kind One of 'to', 'cc', 'bcc', or 'ReplyTo'**

**\* @param string $address The email address**

**\* @param string $name An optional username associated with the address**

**\***

**\* @throws Exception**

**\***

**\* @return bool true on success, false if address already used or invalid in some way**

**\*/**

**protected function addOrEnqueueAnAddress($kind, $address, $name)**

**{**

**$pos = false;**

**if ($address !== null) {**

**$address = trim($address);**

**$pos = strrpos($address, '@');**

**}**

**if (false === $pos) {**

**//At-sign is missing.**

**$error\_message = sprintf(**

**'%s (%s): %s',**

**$this->lang('invalid\_address'),**

**$kind,**

**$address**

**);**

**$this->setError($error\_message);**

**$this->edebug($error\_message);**

**if ($this->exceptions) {**

**throw new Exception($error\_message);**

**}**

**return false;**

**}**

**if ($name !== null) {**

**$name = trim(preg\_replace('/[\r\n]+/', '', $name)); //Strip breaks and trim**

**} else {**

**$name = '';**

**}**

**$params = [$kind, $address, $name];**

**//Enqueue addresses with IDN until we know the PHPMailer::$CharSet.**

**//Domain is assumed to be whatever is after the last @ symbol in the address**

**if (static::idnSupported() && $this->has8bitChars(substr($address, ++$pos))) {**

**if ('Reply-To' !== $kind) {**

**if (!array\_key\_exists($address, $this->RecipientsQueue)) {**

**$this->RecipientsQueue[$address] = $params;**

**return true;**

**}**

**} elseif (!array\_key\_exists($address, $this->ReplyToQueue)) {**

**$this->ReplyToQueue[$address] = $params;**

**return true;**

**}**

**return false;**

**}**

**//Immediately add standard addresses without IDN.**

**return call\_user\_func\_array([$this, 'addAnAddress'], $params);**

**}**

**/\*\***

**\* Add an address to one of the recipient arrays or to the ReplyTo array.**

**\* Addresses that have been added already return false, but do not throw exceptions.**

**\***

**\* @param string $kind One of 'to', 'cc', 'bcc', or 'ReplyTo'**

**\* @param string $address The email address to send, resp. to reply to**

**\* @param string $name**

**\***

**\* @throws Exception**

**\***

**\* @return bool true on success, false if address already used or invalid in some way**

**\*/**

**protected function addAnAddress($kind, $address, $name = '')**

**{**

**if (!in\_array($kind, ['to', 'cc', 'bcc', 'Reply-To'])) {**

**$error\_message = sprintf(**

**'%s: %s',**

**$this->lang('Invalid recipient kind'),**

**$kind**

**);**

**$this->setError($error\_message);**

**$this->edebug($error\_message);**

**if ($this->exceptions) {**

**throw new Exception($error\_message);**

**}**

**return false;**

**}**

**if (!static::validateAddress($address)) {**

**$error\_message = sprintf(**

**'%s (%s): %s',**

**$this->lang('invalid\_address'),**

**$kind,**

**$address**

**);**

**$this->setError($error\_message);**

**$this->edebug($error\_message);**

**if ($this->exceptions) {**

**throw new Exception($error\_message);**

**}**

**return false;**

**}**

**if ('Reply-To' !== $kind) {**

**if (!array\_key\_exists(strtolower($address), $this->all\_recipients)) {**

**$this->{$kind}[] = [$address, $name];**

**$this->all\_recipients[strtolower($address)] = true;**

**return true;**

**}**

**} elseif (!array\_key\_exists(strtolower($address), $this->ReplyTo)) {**

**$this->ReplyTo[strtolower($address)] = [$address, $name];**

**return true;**

**}**

**return false;**

**}**

**/\*\***

**\* Parse and validate a string containing one or more RFC822-style comma-separated email addresses**

**\* of the form "display name <address>" into an array of name/address pairs.**

**\* Uses the imap\_rfc822\_parse\_adrlist function if the IMAP extension is available.**

**\* Note that quotes in the name part are removed.**

**\***

**\* @see http://www.andrew.cmu.edu/user/agreen1/testing/mrbs/web/Mail/RFC822.php A more careful implementation**

**\***

**\* @param string $addrstr The address list string**

**\* @param bool $useimap Whether to use the IMAP extension to parse the list**

**\* @param string $charset The charset to use when decoding the address list string.**

**\***

**\* @return array**

**\*/**

**public static function parseAddresses($addrstr, $useimap = true, $charset = self::CHARSET\_ISO88591)**

**{**

**$addresses = [];**

**if ($useimap && function\_exists('imap\_rfc822\_parse\_adrlist')) {**

**//Use this built-in parser if it's available**

**$list = imap\_rfc822\_parse\_adrlist($addrstr, '');**

**// Clear any potential IMAP errors to get rid of notices being thrown at end of script.**

**imap\_errors();**

**foreach ($list as $address) {**

**if (**

**'.SYNTAX-ERROR.' !== $address->host &&**

**static::validateAddress($address->mailbox . '@' . $address->host)**

**) {**

**//Decode the name part if it's present and encoded**

**if (**

**property\_exists($address, 'personal') &&**

**//Check for a Mbstring constant rather than using extension\_loaded, which is sometimes disabled**

**defined('MB\_CASE\_UPPER') &&**

**preg\_match('/^=\?.\*\?=$/s', $address->personal)**

**) {**

**$origCharset = mb\_internal\_encoding();**

**mb\_internal\_encoding($charset);**

**//Undo any RFC2047-encoded spaces-as-underscores**

**$address->personal = str\_replace('\_', '=20', $address->personal);**

**//Decode the name**

**$address->personal = mb\_decode\_mimeheader($address->personal);**

**mb\_internal\_encoding($origCharset);**

**}**

**$addresses[] = [**

**'name' => (property\_exists($address, 'personal') ? $address->personal : ''),**

**'address' => $address->mailbox . '@' . $address->host,**

**];**

**}**

**}**

**} else {**

**//Use this simpler parser**

**$list = explode(',', $addrstr);**

**foreach ($list as $address) {**

**$address = trim($address);**

**//Is there a separate name part?**

**if (strpos($address, '<') === false) {**

**//No separate name, just use the whole thing**

**if (static::validateAddress($address)) {**

**$addresses[] = [**

**'name' => '',**

**'address' => $address,**

**];**

**}**

**} else {**

**list($name, $email) = explode('<', $address);**

**$email = trim(str\_replace('>', '', $email));**

**$name = trim($name);**

**if (static::validateAddress($email)) {**

**//Check for a Mbstring constant rather than using extension\_loaded, which is sometimes disabled**

**//If this name is encoded, decode it**

**if (defined('MB\_CASE\_UPPER') && preg\_match('/^=\?.\*\?=$/s', $name)) {**

**$origCharset = mb\_internal\_encoding();**

**mb\_internal\_encoding($charset);**

**//Undo any RFC2047-encoded spaces-as-underscores**

**$name = str\_replace('\_', '=20', $name);**

**//Decode the name**

**$name = mb\_decode\_mimeheader($name);**

**mb\_internal\_encoding($origCharset);**

**}**

**$addresses[] = [**

**//Remove any surrounding quotes and spaces from the name**

**'name' => trim($name, '\'" '),**

**'address' => $email,**

**];**

**}**

**}**

**}**

**}**

**return $addresses;**

**}**

**/\*\***

**\* Set the From and FromName properties.**

**\***

**\* @param string $address**

**\* @param string $name**

**\* @param bool $auto Whether to also set the Sender address, defaults to true**

**\***

**\* @throws Exception**

**\***

**\* @return bool**

**\*/**

**public function setFrom($address, $name = '', $auto = true)**

**{**

**$address = trim($address);**

**$name = trim(preg\_replace('/[\r\n]+/', '', $name)); //Strip breaks and trim**

**//Don't validate now addresses with IDN. Will be done in send().**

**$pos = strrpos($address, '@');**

**if (**

**(false === $pos)**

**|| ((!$this->has8bitChars(substr($address, ++$pos)) || !static::idnSupported())**

**&& !static::validateAddress($address))**

**) {**

**$error\_message = sprintf(**

**'%s (From): %s',**

**$this->lang('invalid\_address'),**

**$address**

**);**

**$this->setError($error\_message);**

**$this->edebug($error\_message);**

**if ($this->exceptions) {**

**throw new Exception($error\_message);**

**}**

**return false;**

**}**

**$this->From = $address;**

**$this->FromName = $name;**

**if ($auto && empty($this->Sender)) {**

**$this->Sender = $address;**

**}**

**return true;**

**}**

**/\*\***

**\* Return the Message-ID header of the last email.**

**\* Technically this is the value from the last time the headers were created,**

**\* but it's also the message ID of the last sent message except in**

**\* pathological cases.**

**\***

**\* @return string**

**\*/**

**public function getLastMessageID()**

**{**

**return $this->lastMessageID;**

**}**

**/\*\***

**\* Check that a string looks like an email address.**

**\* Validation patterns supported:**

**\* \* `auto` Pick best pattern automatically;**

**\* \* `pcre8` Use the squiloople.com pattern, requires PCRE > 8.0;**

**\* \* `pcre` Use old PCRE implementation;**

**\* \* `php` Use PHP built-in FILTER\_VALIDATE\_EMAIL;**

**\* \* `html5` Use the pattern given by the HTML5 spec for 'email' type form input elements.**

**\* \* `noregex` Don't use a regex: super fast, really dumb.**

**\* Alternatively you may pass in a callable to inject your own validator, for example:**

**\***

**\* ```php**

**\* PHPMailer::validateAddress('user@example.com', function($address) {**

**\* return (strpos($address, '@') !== false);**

**\* });**

**\* ```**

**\***

**\* You can also set the PHPMailer::$validator static to a callable, allowing built-in methods to use your validator.**

**\***

**\* @param string $address The email address to check**

**\* @param string|callable $patternselect Which pattern to use**

**\***

**\* @return bool**

**\*/**

**public static function validateAddress($address, $patternselect = null)**

**{**

**if (null === $patternselect) {**

**$patternselect = static::$validator;**

**}**

**//Don't allow strings as callables, see SECURITY.md and CVE-2021-3603**

**if (is\_callable($patternselect) && !is\_string($patternselect)) {**

**return call\_user\_func($patternselect, $address);**

**}**

**//Reject line breaks in addresses; it's valid RFC5322, but not RFC5321**

**if (strpos($address, "\n") !== false || strpos($address, "\r") !== false) {**

**return false;**

**}**

**switch ($patternselect) {**

**case 'pcre': //Kept for BC**

**case 'pcre8':**

**/\***

**\* A more complex and more permissive version of the RFC5322 regex on which FILTER\_VALIDATE\_EMAIL**

**\* is based.**

**\* In addition to the addresses allowed by filter\_var, also permits:**

**\* \* dotless domains: `a@b`**

**\* \* comments: `1234 @ local(blah) .machine .example`**

**\* \* quoted elements: `'"test blah"@example.org'`**

**\* \* numeric TLDs: `a@b.123`**

**\* \* unbracketed IPv4 literals: `a@192.168.0.1`**

**\* \* IPv6 literals: 'first.last@[IPv6:a1::]'**

**\* Not all of these will necessarily work for sending!**

**\***

**\* @see http://squiloople.com/2009/12/20/email-address-validation/**

**\* @copyright 2009-2010 Michael Rushton**

**\* Feel free to use and redistribute this code. But please keep this copyright notice.**

**\*/**

**return (bool) preg\_match(**

**'/^(?!(?>(?1)"?(?>\\\[ -~]|[^"])"?(?1)){255,})(?!(?>(?1)"?(?>\\\[ -~]|[^"])"?(?1)){65,}@)' .**

**'((?>(?>(?>((?>(?>(?>\x0D\x0A)?[\t ])+|(?>[\t ]\*\x0D\x0A)?[\t ]+)?)(\((?>(?2)' .**

**'(?>[\x01-\x08\x0B\x0C\x0E-\'\*-\[\]-\x7F]|\\\[\x00-\x7F]|(?3)))\*(?2)\)))+(?2))|(?2))?)' .**

**'([!#-\'\*+\/-9=?^-~-]+|"(?>(?2)(?>[\x01-\x08\x0B\x0C\x0E-!#-\[\]-\x7F]|\\\[\x00-\x7F]))\*' .**

**'(?2)")(?>(?1)\.(?1)(?4))\*(?1)@(?!(?1)[a-z0-9-]{64,})(?1)(?>([a-z0-9](?>[a-z0-9-]\*[a-z0-9])?)' .**

**'(?>(?1)\.(?!(?1)[a-z0-9-]{64,})(?1)(?5)){0,126}|\[(?:(?>IPv6:(?>([a-f0-9]{1,4})(?>:(?6)){7}' .**

**'|(?!(?:.\*[a-f0-9][:\]]){8,})((?6)(?>:(?6)){0,6})?::(?7)?))|(?>(?>IPv6:(?>(?6)(?>:(?6)){5}:' .**

**'|(?!(?:.\*[a-f0-9]:){6,})(?8)?::(?>((?6)(?>:(?6)){0,4}):)?))?(25[0-5]|2[0-4][0-9]|1[0-9]{2}' .**

**'|[1-9]?[0-9])(?>\.(?9)){3}))\])(?1)$/isD',**

**$address**

**);**

**case 'html5':**

**/\***

**\* This is the pattern used in the HTML5 spec for validation of 'email' type form input elements.**

**\***

**\* @see https://html.spec.whatwg.org/#e-mail-state-(type=email)**

**\*/**

**return (bool) preg\_match(**

**'/^[a-zA-Z0-9.!#$%&\'\*+\/=?^\_`{|}~-]+@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}' .**

**'[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)\*$/sD',**

**$address**

**);**

**case 'php':**

**default:**

**return filter\_var($address, FILTER\_VALIDATE\_EMAIL) !== false;**

**}**

**}**

**/\*\***

**\* Tells whether IDNs (Internationalized Domain Names) are supported or not. This requires the**

**\* `intl` and `mbstring` PHP extensions.**

**\***

**\* @return bool `true` if required functions for IDN support are present**

**\*/**

**public static function idnSupported()**

**{**

**return function\_exists('idn\_to\_ascii') && function\_exists('mb\_convert\_encoding');**

**}**

**/\*\***

**\* Converts IDN in given email address to its ASCII form, also known as punycode, if possible.**

**\* Important: Address must be passed in same encoding as currently set in PHPMailer::$CharSet.**

**\* This function silently returns unmodified address if:**

**\* - No conversion is necessary (i.e. domain name is not an IDN, or is already in ASCII form)**

**\* - Conversion to punycode is impossible (e.g. required PHP functions are not available)**

**\* or fails for any reason (e.g. domain contains characters not allowed in an IDN).**

**\***

**\* @see PHPMailer::$CharSet**

**\***

**\* @param string $address The email address to convert**

**\***

**\* @return string The encoded address in ASCII form**

**\*/**

**public function punyencodeAddress($address)**

**{**

**//Verify we have required functions, CharSet, and at-sign.**

**$pos = strrpos($address, '@');**

**if (**

**!empty($this->CharSet) &&**

**false !== $pos &&**

**static::idnSupported()**

**) {**

**$domain = substr($address, ++$pos);**

**//Verify CharSet string is a valid one, and domain properly encoded in this CharSet.**

**if ($this->has8bitChars($domain) && @mb\_check\_encoding($domain, $this->CharSet)) {**

**//Convert the domain from whatever charset it's in to UTF-8**

**$domain = mb\_convert\_encoding($domain, self::CHARSET\_UTF8, $this->CharSet);**

**//Ignore IDE complaints about this line - method signature changed in PHP 5.4**

**$errorcode = 0;**

**if (defined('INTL\_IDNA\_VARIANT\_UTS46')) {**

**//Use the current punycode standard (appeared in PHP 7.2)**

**$punycode = idn\_to\_ascii(**

**$domain,**

**\IDNA\_DEFAULT | \IDNA\_USE\_STD3\_RULES | \IDNA\_CHECK\_BIDI |**

**\IDNA\_CHECK\_CONTEXTJ | \IDNA\_NONTRANSITIONAL\_TO\_ASCII,**

**\INTL\_IDNA\_VARIANT\_UTS46**

**);**

**} elseif (defined('INTL\_IDNA\_VARIANT\_2003')) {**

**//Fall back to this old, deprecated/removed encoding**

**$punycode = idn\_to\_ascii($domain, $errorcode, \INTL\_IDNA\_VARIANT\_2003);**

**} else {**

**//Fall back to a default we don't know about**

**$punycode = idn\_to\_ascii($domain, $errorcode);**

**}**

**if (false !== $punycode) {**

**return substr($address, 0, $pos) . $punycode;**

**}**

**}**

**}**

**return $address;**

**}**

**/\*\***

**\* Create a message and send it.**

**\* Uses the sending method specified by $Mailer.**

**\***

**\* @throws Exception**

**\***

**\* @return bool false on error - See the ErrorInfo property for details of the error**

**\*/**

**public function send()**

**{**

**try {**

**if (!$this->preSend()) {**

**return false;**

**}**

**return $this->postSend();**

**} catch (Exception $exc) {**

**$this->mailHeader = '';**

**$this->setError($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**return false;**

**}**

**}**

**/\*\***

**\* Prepare a message for sending.**

**\***

**\* @throws Exception**

**\***

**\* @return bool**

**\*/**

**public function preSend()**

**{**

**if (**

**'smtp' === $this->Mailer**

**|| ('mail' === $this->Mailer && (\PHP\_VERSION\_ID >= 80000 || stripos(PHP\_OS, 'WIN') === 0))**

**) {**

**//SMTP mandates RFC-compliant line endings**

**//and it's also used with mail() on Windows**

**static::setLE(self::CRLF);**

**} else {**

**//Maintain backward compatibility with legacy Linux command line mailers**

**static::setLE(PHP\_EOL);**

**}**

**//Check for buggy PHP versions that add a header with an incorrect line break**

**if (**

**'mail' === $this->Mailer**

**&& ((\PHP\_VERSION\_ID >= 70000 && \PHP\_VERSION\_ID < 70017)**

**|| (\PHP\_VERSION\_ID >= 70100 && \PHP\_VERSION\_ID < 70103))**

**&& ini\_get('mail.add\_x\_header') === '1'**

**&& stripos(PHP\_OS, 'WIN') === 0**

**) {**

**trigger\_error($this->lang('buggy\_php'), E\_USER\_WARNING);**

**}**

**try {**

**$this->error\_count = 0; //Reset errors**

**$this->mailHeader = '';**

**//Dequeue recipient and Reply-To addresses with IDN**

**foreach (array\_merge($this->RecipientsQueue, $this->ReplyToQueue) as $params) {**

**$params[1] = $this->punyencodeAddress($params[1]);**

**call\_user\_func\_array([$this, 'addAnAddress'], $params);**

**}**

**if (count($this->to) + count($this->cc) + count($this->bcc) < 1) {**

**throw new Exception($this->lang('provide\_address'), self::STOP\_CRITICAL);**

**}**

**//Validate From, Sender, and ConfirmReadingTo addresses**

**foreach (['From', 'Sender', 'ConfirmReadingTo'] as $address\_kind) {**

**$this->$address\_kind = trim($this->$address\_kind);**

**if (empty($this->$address\_kind)) {**

**continue;**

**}**

**$this->$address\_kind = $this->punyencodeAddress($this->$address\_kind);**

**if (!static::validateAddress($this->$address\_kind)) {**

**$error\_message = sprintf(**

**'%s (%s): %s',**

**$this->lang('invalid\_address'),**

**$address\_kind,**

**$this->$address\_kind**

**);**

**$this->setError($error\_message);**

**$this->edebug($error\_message);**

**if ($this->exceptions) {**

**throw new Exception($error\_message);**

**}**

**return false;**

**}**

**}**

**//Set whether the message is multipart/alternative**

**if ($this->alternativeExists()) {**

**$this->ContentType = static::CONTENT\_TYPE\_MULTIPART\_ALTERNATIVE;**

**}**

**$this->setMessageType();**

**//Refuse to send an empty message unless we are specifically allowing it**

**if (!$this->AllowEmpty && empty($this->Body)) {**

**throw new Exception($this->lang('empty\_message'), self::STOP\_CRITICAL);**

**}**

**//Trim subject consistently**

**$this->Subject = trim($this->Subject);**

**//Create body before headers in case body makes changes to headers (e.g. altering transfer encoding)**

**$this->MIMEHeader = '';**

**$this->MIMEBody = $this->createBody();**

**//createBody may have added some headers, so retain them**

**$tempheaders = $this->MIMEHeader;**

**$this->MIMEHeader = $this->createHeader();**

**$this->MIMEHeader .= $tempheaders;**

**//To capture the complete message when using mail(), create**

**//an extra header list which createHeader() doesn't fold in**

**if ('mail' === $this->Mailer) {**

**if (count($this->to) > 0) {**

**$this->mailHeader .= $this->addrAppend('To', $this->to);**

**} else {**

**$this->mailHeader .= $this->headerLine('To', 'undisclosed-recipients:;');**

**}**

**$this->mailHeader .= $this->headerLine(**

**'Subject',**

**$this->encodeHeader($this->secureHeader($this->Subject))**

**);**

**}**

**//Sign with DKIM if enabled**

**if (**

**!empty($this->DKIM\_domain)**

**&& !empty($this->DKIM\_selector)**

**&& (!empty($this->DKIM\_private\_string)**

**|| (!empty($this->DKIM\_private)**

**&& static::isPermittedPath($this->DKIM\_private)**

**&& file\_exists($this->DKIM\_private)**

**)**

**)**

**) {**

**$header\_dkim = $this->DKIM\_Add(**

**$this->MIMEHeader . $this->mailHeader,**

**$this->encodeHeader($this->secureHeader($this->Subject)),**

**$this->MIMEBody**

**);**

**$this->MIMEHeader = static::stripTrailingWSP($this->MIMEHeader) . static::$LE .**

**static::normalizeBreaks($header\_dkim) . static::$LE;**

**}**

**return true;**

**} catch (Exception $exc) {**

**$this->setError($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**return false;**

**}**

**}**

**/\*\***

**\* Actually send a message via the selected mechanism.**

**\***

**\* @throws Exception**

**\***

**\* @return bool**

**\*/**

**public function postSend()**

**{**

**try {**

**//Choose the mailer and send through it**

**switch ($this->Mailer) {**

**case 'sendmail':**

**case 'qmail':**

**return $this->sendmailSend($this->MIMEHeader, $this->MIMEBody);**

**case 'smtp':**

**return $this->smtpSend($this->MIMEHeader, $this->MIMEBody);**

**case 'mail':**

**return $this->mailSend($this->MIMEHeader, $this->MIMEBody);**

**default:**

**$sendMethod = $this->Mailer . 'Send';**

**if (method\_exists($this, $sendMethod)) {**

**return $this->$sendMethod($this->MIMEHeader, $this->MIMEBody);**

**}**

**return $this->mailSend($this->MIMEHeader, $this->MIMEBody);**

**}**

**} catch (Exception $exc) {**

**if ($this->Mailer === 'smtp' && $this->SMTPKeepAlive == true) {**

**$this->smtp->reset();**

**}**

**$this->setError($exc->getMessage());**

**$this->edebug($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**}**

**return false;**

**}**

**/\*\***

**\* Send mail using the $Sendmail program.**

**\***

**\* @see PHPMailer::$Sendmail**

**\***

**\* @param string $header The message headers**

**\* @param string $body The message body**

**\***

**\* @throws Exception**

**\***

**\* @return bool**

**\*/**

**protected function sendmailSend($header, $body)**

**{**

**if ($this->Mailer === 'qmail') {**

**$this->edebug('Sending with qmail');**

**} else {**

**$this->edebug('Sending with sendmail');**

**}**

**$header = static::stripTrailingWSP($header) . static::$LE . static::$LE;**

**//This sets the SMTP envelope sender which gets turned into a return-path header by the receiver**

**//A space after `-f` is optional, but there is a long history of its presence**

**//causing problems, so we don't use one**

**//Exim docs: http://www.exim.org/exim-html-current/doc/html/spec\_html/ch-the\_exim\_command\_line.html**

**//Sendmail docs: http://www.sendmail.org/~ca/email/man/sendmail.html**

**//Qmail docs: http://www.qmail.org/man/man8/qmail-inject.html**

**//Example problem: https://www.drupal.org/node/1057954**

**//PHP 5.6 workaround**

**$sendmail\_from\_value = ini\_get('sendmail\_from');**

**if (empty($this->Sender) && !empty($sendmail\_from\_value)) {**

**//PHP config has a sender address we can use**

**$this->Sender = ini\_get('sendmail\_from');**

**}**

**//CVE-2016-10033, CVE-2016-10045: Don't pass -f if characters will be escaped.**

**if (!empty($this->Sender) && static::validateAddress($this->Sender) && self::isShellSafe($this->Sender)) {**

**if ($this->Mailer === 'qmail') {**

**$sendmailFmt = '%s -f%s';**

**} else {**

**$sendmailFmt = '%s -oi -f%s -t';**

**}**

**} else {**

**//allow sendmail to choose a default envelope sender. It may**

**//seem preferable to force it to use the From header as with**

**//SMTP, but that introduces new problems (see**

**//<https://github.com/PHPMailer/PHPMailer/issues/2298>), and**

**//it has historically worked this way.**

**$sendmailFmt = '%s -oi -t';**

**}**

**$sendmail = sprintf($sendmailFmt, escapeshellcmd($this->Sendmail), $this->Sender);**

**$this->edebug('Sendmail path: ' . $this->Sendmail);**

**$this->edebug('Sendmail command: ' . $sendmail);**

**$this->edebug('Envelope sender: ' . $this->Sender);**

**$this->edebug("Headers: {$header}");**

**if ($this->SingleTo) {**

**foreach ($this->SingleToArray as $toAddr) {**

**$mail = @popen($sendmail, 'w');**

**if (!$mail) {**

**throw new Exception($this->lang('execute') . $this->Sendmail, self::STOP\_CRITICAL);**

**}**

**$this->edebug("To: {$toAddr}");**

**fwrite($mail, 'To: ' . $toAddr . "\n");**

**fwrite($mail, $header);**

**fwrite($mail, $body);**

**$result = pclose($mail);**

**$addrinfo = static::parseAddresses($toAddr, true, $this->CharSet);**

**$this->doCallback(**

**($result === 0),**

**[[$addrinfo['address'], $addrinfo['name']]],**

**$this->cc,**

**$this->bcc,**

**$this->Subject,**

**$body,**

**$this->From,**

**[]**

**);**

**$this->edebug("Result: " . ($result === 0 ? 'true' : 'false'));**

**if (0 !== $result) {**

**throw new Exception($this->lang('execute') . $this->Sendmail, self::STOP\_CRITICAL);**

**}**

**}**

**} else {**

**$mail = @popen($sendmail, 'w');**

**if (!$mail) {**

**throw new Exception($this->lang('execute') . $this->Sendmail, self::STOP\_CRITICAL);**

**}**

**fwrite($mail, $header);**

**fwrite($mail, $body);**

**$result = pclose($mail);**

**$this->doCallback(**

**($result === 0),**

**$this->to,**

**$this->cc,**

**$this->bcc,**

**$this->Subject,**

**$body,**

**$this->From,**

**[]**

**);**

**$this->edebug("Result: " . ($result === 0 ? 'true' : 'false'));**

**if (0 !== $result) {**

**throw new Exception($this->lang('execute') . $this->Sendmail, self::STOP\_CRITICAL);**

**}**

**}**

**return true;**

**}**

**/\*\***

**\* Fix CVE-2016-10033 and CVE-2016-10045 by disallowing potentially unsafe shell characters.**

**\* Note that escapeshellarg and escapeshellcmd are inadequate for our purposes, especially on Windows.**

**\***

**\* @see https://github.com/PHPMailer/PHPMailer/issues/924 CVE-2016-10045 bug report**

**\***

**\* @param string $string The string to be validated**

**\***

**\* @return bool**

**\*/**

**protected static function isShellSafe($string)**

**{**

**//It's not possible to use shell commands safely (which includes the mail() function) without escapeshellarg,**

**//but some hosting providers disable it, creating a security problem that we don't want to have to deal with,**

**//so we don't.**

**if (!function\_exists('escapeshellarg') || !function\_exists('escapeshellcmd')) {**

**return false;**

**}**

**if (**

**escapeshellcmd($string) !== $string**

**|| !in\_array(escapeshellarg($string), ["'$string'", "\"$string\""])**

**) {**

**return false;**

**}**

**$length = strlen($string);**

**for ($i = 0; $i < $length; ++$i) {**

**$c = $string[$i];**

**//All other characters have a special meaning in at least one common shell, including = and +.**

**//Full stop (.) has a special meaning in cmd.exe, but its impact should be negligible here.**

**//Note that this does permit non-Latin alphanumeric characters based on the current locale.**

**if (!ctype\_alnum($c) && strpos('@\_-.', $c) === false) {**

**return false;**

**}**

**}**

**return true;**

**}**

**/\*\***

**\* Check whether a file path is of a permitted type.**

**\* Used to reject URLs and phar files from functions that access local file paths,**

**\* such as addAttachment.**

**\***

**\* @param string $path A relative or absolute path to a file**

**\***

**\* @return bool**

**\*/**

**protected static function isPermittedPath($path)**

**{**

**//Matches scheme definition from https://tools.ietf.org/html/rfc3986#section-3.1**

**return !preg\_match('#^[a-z][a-z\d+.-]\*://#i', $path);**

**}**

**/\*\***

**\* Check whether a file path is safe, accessible, and readable.**

**\***

**\* @param string $path A relative or absolute path to a file**

**\***

**\* @return bool**

**\*/**

**protected static function fileIsAccessible($path)**

**{**

**if (!static::isPermittedPath($path)) {**

**return false;**

**}**

**$readable = file\_exists($path);**

**//If not a UNC path (expected to start with \\), check read permission, see #2069**

**if (strpos($path, '\\\\') !== 0) {**

**$readable = $readable && is\_readable($path);**

**}**

**return $readable;**

**}**

**/\*\***

**\* Send mail using the PHP mail() function.**

**\***

**\* @see http://www.php.net/manual/en/book.mail.php**

**\***

**\* @param string $header The message headers**

**\* @param string $body The message body**

**\***

**\* @throws Exception**

**\***

**\* @return bool**

**\*/**

**protected function mailSend($header, $body)**

**{**

**$header = static::stripTrailingWSP($header) . static::$LE . static::$LE;**

**$toArr = [];**

**foreach ($this->to as $toaddr) {**

**$toArr[] = $this->addrFormat($toaddr);**

**}**

**$to = implode(', ', $toArr);**

**$params = null;**

**//This sets the SMTP envelope sender which gets turned into a return-path header by the receiver**

**//A space after `-f` is optional, but there is a long history of its presence**

**//causing problems, so we don't use one**

**//Exim docs: http://www.exim.org/exim-html-current/doc/html/spec\_html/ch-the\_exim\_command\_line.html**

**//Sendmail docs: http://www.sendmail.org/~ca/email/man/sendmail.html**

**//Qmail docs: http://www.qmail.org/man/man8/qmail-inject.html**

**//Example problem: https://www.drupal.org/node/1057954**

**//CVE-2016-10033, CVE-2016-10045: Don't pass -f if characters will be escaped.**

**//PHP 5.6 workaround**

**$sendmail\_from\_value = ini\_get('sendmail\_from');**

**if (empty($this->Sender) && !empty($sendmail\_from\_value)) {**

**//PHP config has a sender address we can use**

**$this->Sender = ini\_get('sendmail\_from');**

**}**

**if (!empty($this->Sender) && static::validateAddress($this->Sender)) {**

**if (self::isShellSafe($this->Sender)) {**

**$params = sprintf('-f%s', $this->Sender);**

**}**

**$old\_from = ini\_get('sendmail\_from');**

**ini\_set('sendmail\_from', $this->Sender);**

**}**

**$result = false;**

**if ($this->SingleTo && count($toArr) > 1) {**

**foreach ($toArr as $toAddr) {**

**$result = $this->mailPassthru($toAddr, $this->Subject, $body, $header, $params);**

**$addrinfo = static::parseAddresses($toAddr, true, $this->CharSet);**

**$this->doCallback(**

**$result,**

**[[$addrinfo['address'], $addrinfo['name']]],**

**$this->cc,**

**$this->bcc,**

**$this->Subject,**

**$body,**

**$this->From,**

**[]**

**);**

**}**

**} else {**

**$result = $this->mailPassthru($to, $this->Subject, $body, $header, $params);**

**$this->doCallback($result, $this->to, $this->cc, $this->bcc, $this->Subject, $body, $this->From, []);**

**}**

**if (isset($old\_from)) {**

**ini\_set('sendmail\_from', $old\_from);**

**}**

**if (!$result) {**

**throw new Exception($this->lang('instantiate'), self::STOP\_CRITICAL);**

**}**

**return true;**

**}**

**/\*\***

**\* Get an instance to use for SMTP operations.**

**\* Override this function to load your own SMTP implementation,**

**\* or set one with setSMTPInstance.**

**\***

**\* @return SMTP**

**\*/**

**public function getSMTPInstance()**

**{**

**if (!is\_object($this->smtp)) {**

**$this->smtp = new SMTP();**

**}**

**return $this->smtp;**

**}**

**/\*\***

**\* Provide an instance to use for SMTP operations.**

**\***

**\* @return SMTP**

**\*/**

**public function setSMTPInstance(SMTP $smtp)**

**{**

**$this->smtp = $smtp;**

**return $this->smtp;**

**}**

**/\*\***

**\* Send mail via SMTP.**

**\* Returns false if there is a bad MAIL FROM, RCPT, or DATA input.**

**\***

**\* @see PHPMailer::setSMTPInstance() to use a different class.**

**\***

**\* @uses \PHPMailer\PHPMailer\SMTP**

**\***

**\* @param string $header The message headers**

**\* @param string $body The message body**

**\***

**\* @throws Exception**

**\***

**\* @return bool**

**\*/**

**protected function smtpSend($header, $body)**

**{**

**$header = static::stripTrailingWSP($header) . static::$LE . static::$LE;**

**$bad\_rcpt = [];**

**if (!$this->smtpConnect($this->SMTPOptions)) {**

**throw new Exception($this->lang('smtp\_connect\_failed'), self::STOP\_CRITICAL);**

**}**

**//Sender already validated in preSend()**

**if ('' === $this->Sender) {**

**$smtp\_from = $this->From;**

**} else {**

**$smtp\_from = $this->Sender;**

**}**

**if (!$this->smtp->mail($smtp\_from)) {**

**$this->setError($this->lang('from\_failed') . $smtp\_from . ' : ' . implode(',', $this->smtp->getError()));**

**throw new Exception($this->ErrorInfo, self::STOP\_CRITICAL);**

**}**

**$callbacks = [];**

**//Attempt to send to all recipients**

**foreach ([$this->to, $this->cc, $this->bcc] as $togroup) {**

**foreach ($togroup as $to) {**

**if (!$this->smtp->recipient($to[0], $this->dsn)) {**

**$error = $this->smtp->getError();**

**$bad\_rcpt[] = ['to' => $to[0], 'error' => $error['detail']];**

**$isSent = false;**

**} else {**

**$isSent = true;**

**}**

**$callbacks[] = ['issent' => $isSent, 'to' => $to[0], 'name' => $to[1]];**

**}**

**}**

**//Only send the DATA command if we have viable recipients**

**if ((count($this->all\_recipients) > count($bad\_rcpt)) && !$this->smtp->data($header . $body)) {**

**throw new Exception($this->lang('data\_not\_accepted'), self::STOP\_CRITICAL);**

**}**

**$smtp\_transaction\_id = $this->smtp->getLastTransactionID();**

**if ($this->SMTPKeepAlive) {**

**$this->smtp->reset();**

**} else {**

**$this->smtp->quit();**

**$this->smtp->close();**

**}**

**foreach ($callbacks as $cb) {**

**$this->doCallback(**

**$cb['issent'],**

**[[$cb['to'], $cb['name']]],**

**[],**

**[],**

**$this->Subject,**

**$body,**

**$this->From,**

**['smtp\_transaction\_id' => $smtp\_transaction\_id]**

**);**

**}**

**//Create error message for any bad addresses**

**if (count($bad\_rcpt) > 0) {**

**$errstr = '';**

**foreach ($bad\_rcpt as $bad) {**

**$errstr .= $bad['to'] . ': ' . $bad['error'];**

**}**

**throw new Exception($this->lang('recipients\_failed') . $errstr, self::STOP\_CONTINUE);**

**}**

**return true;**

**}**

**/\*\***

**\* Initiate a connection to an SMTP server.**

**\* Returns false if the operation failed.**

**\***

**\* @param array $options An array of options compatible with stream\_context\_create()**

**\***

**\* @throws Exception**

**\***

**\* @uses \PHPMailer\PHPMailer\SMTP**

**\***

**\* @return bool**

**\*/**

**public function smtpConnect($options = null)**

**{**

**if (null === $this->smtp) {**

**$this->smtp = $this->getSMTPInstance();**

**}**

**//If no options are provided, use whatever is set in the instance**

**if (null === $options) {**

**$options = $this->SMTPOptions;**

**}**

**//Already connected?**

**if ($this->smtp->connected()) {**

**return true;**

**}**

**$this->smtp->setTimeout($this->Timeout);**

**$this->smtp->setDebugLevel($this->SMTPDebug);**

**$this->smtp->setDebugOutput($this->Debugoutput);**

**$this->smtp->setVerp($this->do\_verp);**

**$hosts = explode(';', $this->Host);**

**$lastexception = null;**

**foreach ($hosts as $hostentry) {**

**$hostinfo = [];**

**if (**

**!preg\_match(**

**'/^(?:(ssl|tls):\/\/)?(.+?)(?::(\d+))?$/',**

**trim($hostentry),**

**$hostinfo**

**)**

**) {**

**$this->edebug($this->lang('invalid\_hostentry') . ' ' . trim($hostentry));**

**//Not a valid host entry**

**continue;**

**}**

**//$hostinfo[1]: optional ssl or tls prefix**

**//$hostinfo[2]: the hostname**

**//$hostinfo[3]: optional port number**

**//The host string prefix can temporarily override the current setting for SMTPSecure**

**//If it's not specified, the default value is used**

**//Check the host name is a valid name or IP address before trying to use it**

**if (!static::isValidHost($hostinfo[2])) {**

**$this->edebug($this->lang('invalid\_host') . ' ' . $hostinfo[2]);**

**continue;**

**}**

**$prefix = '';**

**$secure = $this->SMTPSecure;**

**$tls = (static::ENCRYPTION\_STARTTLS === $this->SMTPSecure);**

**if ('ssl' === $hostinfo[1] || ('' === $hostinfo[1] && static::ENCRYPTION\_SMTPS === $this->SMTPSecure)) {**

**$prefix = 'ssl://';**

**$tls = false; //Can't have SSL and TLS at the same time**

**$secure = static::ENCRYPTION\_SMTPS;**

**} elseif ('tls' === $hostinfo[1]) {**

**$tls = true;**

**//TLS doesn't use a prefix**

**$secure = static::ENCRYPTION\_STARTTLS;**

**}**

**//Do we need the OpenSSL extension?**

**$sslext = defined('OPENSSL\_ALGO\_SHA256');**

**if (static::ENCRYPTION\_STARTTLS === $secure || static::ENCRYPTION\_SMTPS === $secure) {**

**//Check for an OpenSSL constant rather than using extension\_loaded, which is sometimes disabled**

**if (!$sslext) {**

**throw new Exception($this->lang('extension\_missing') . 'openssl', self::STOP\_CRITICAL);**

**}**

**}**

**$host = $hostinfo[2];**

**$port = $this->Port;**

**if (**

**array\_key\_exists(3, $hostinfo) &&**

**is\_numeric($hostinfo[3]) &&**

**$hostinfo[3] > 0 &&**

**$hostinfo[3] < 65536**

**) {**

**$port = (int) $hostinfo[3];**

**}**

**if ($this->smtp->connect($prefix . $host, $port, $this->Timeout, $options)) {**

**try {**

**if ($this->Helo) {**

**$hello = $this->Helo;**

**} else {**

**$hello = $this->serverHostname();**

**}**

**$this->smtp->hello($hello);**

**//Automatically enable TLS encryption if:**

**//\* it's not disabled**

**//\* we have openssl extension**

**//\* we are not already using SSL**

**//\* the server offers STARTTLS**

**if ($this->SMTPAutoTLS && $sslext && 'ssl' !== $secure && $this->smtp->getServerExt('STARTTLS')) {**

**$tls = true;**

**}**

**if ($tls) {**

**if (!$this->smtp->startTLS()) {**

**$message = $this->getSmtpErrorMessage('connect\_host');**

**throw new Exception($message);**

**}**

**//We must resend EHLO after TLS negotiation**

**$this->smtp->hello($hello);**

**}**

**if (**

**$this->SMTPAuth && !$this->smtp->authenticate(**

**$this->Username,**

**$this->Password,**

**$this->AuthType,**

**$this->oauth**

**)**

**) {**

**throw new Exception($this->lang('authenticate'));**

**}**

**return true;**

**} catch (Exception $exc) {**

**$lastexception = $exc;**

**$this->edebug($exc->getMessage());**

**//We must have connected, but then failed TLS or Auth, so close connection nicely**

**$this->smtp->quit();**

**}**

**}**

**}**

**//If we get here, all connection attempts have failed, so close connection hard**

**$this->smtp->close();**

**//As we've caught all exceptions, just report whatever the last one was**

**if ($this->exceptions && null !== $lastexception) {**

**throw $lastexception;**

**} elseif ($this->exceptions) {**

**// no exception was thrown, likely $this->smtp->connect() failed**

**$message = $this->getSmtpErrorMessage('connect\_host');**

**throw new Exception($message);**

**}**

**return false;**

**}**

**/\*\***

**\* Close the active SMTP session if one exists.**

**\*/**

**public function smtpClose()**

**{**

**if ((null !== $this->smtp) && $this->smtp->connected()) {**

**$this->smtp->quit();**

**$this->smtp->close();**

**}**

**}**

**/\*\***

**\* Set the language for error messages.**

**\* The default language is English.**

**\***

**\* @param string $langcode ISO 639-1 2-character language code (e.g. French is "fr")**

**\* Optionally, the language code can be enhanced with a 4-character**

**\* script annotation and/or a 2-character country annotation.**

**\* @param string $lang\_path Path to the language file directory, with trailing separator (slash)**

**\* Do not set this from user input!**

**\***

**\* @return bool Returns true if the requested language was loaded, false otherwise.**

**\*/**

**public function setLanguage($langcode = 'en', $lang\_path = '')**

**{**

**//Backwards compatibility for renamed language codes**

**$renamed\_langcodes = [**

**'br' => 'pt\_br',**

**'cz' => 'cs',**

**'dk' => 'da',**

**'no' => 'nb',**

**'se' => 'sv',**

**'rs' => 'sr',**

**'tg' => 'tl',**

**'am' => 'hy',**

**];**

**if (array\_key\_exists($langcode, $renamed\_langcodes)) {**

**$langcode = $renamed\_langcodes[$langcode];**

**}**

**//Define full set of translatable strings in English**

**$PHPMAILER\_LANG = [**

**'authenticate' => 'SMTP Error: Could not authenticate.',**

**'buggy\_php' => 'Your version of PHP is affected by a bug that may result in corrupted messages.' .**

**' To fix it, switch to sending using SMTP, disable the mail.add\_x\_header option in' .**

**' your php.ini, switch to MacOS or Linux, or upgrade your PHP to version 7.0.17+ or 7.1.3+.',**

**'connect\_host' => 'SMTP Error: Could not connect to SMTP host.',**

**'data\_not\_accepted' => 'SMTP Error: data not accepted.',**

**'empty\_message' => 'Message body empty',**

**'encoding' => 'Unknown encoding: ',**

**'execute' => 'Could not execute: ',**

**'extension\_missing' => 'Extension missing: ',**

**'file\_access' => 'Could not access file: ',**

**'file\_open' => 'File Error: Could not open file: ',**

**'from\_failed' => 'The following From address failed: ',**

**'instantiate' => 'Could not instantiate mail function.',**

**'invalid\_address' => 'Invalid address: ',**

**'invalid\_header' => 'Invalid header name or value',**

**'invalid\_hostentry' => 'Invalid hostentry: ',**

**'invalid\_host' => 'Invalid host: ',**

**'mailer\_not\_supported' => ' mailer is not supported.',**

**'provide\_address' => 'You must provide at least one recipient email address.',**

**'recipients\_failed' => 'SMTP Error: The following recipients failed: ',**

**'signing' => 'Signing Error: ',**

**'smtp\_code' => 'SMTP code: ',**

**'smtp\_code\_ex' => 'Additional SMTP info: ',**

**'smtp\_connect\_failed' => 'SMTP connect() failed.',**

**'smtp\_detail' => 'Detail: ',**

**'smtp\_error' => 'SMTP server error: ',**

**'variable\_set' => 'Cannot set or reset variable: ',**

**];**

**if (empty($lang\_path)) {**

**//Calculate an absolute path so it can work if CWD is not here**

**$lang\_path = dirname(\_\_DIR\_\_) . DIRECTORY\_SEPARATOR . 'language' . DIRECTORY\_SEPARATOR;**

**}**

**//Validate $langcode**

**$foundlang = true;**

**$langcode = strtolower($langcode);**

**if (**

**!preg\_match('/^(?P<lang>[a-z]{2})(?P<script>\_[a-z]{4})?(?P<country>\_[a-z]{2})?$/', $langcode, $matches)**

**&& $langcode !== 'en'**

**) {**

**$foundlang = false;**

**$langcode = 'en';**

**}**

**//There is no English translation file**

**if ('en' !== $langcode) {**

**$langcodes = [];**

**if (!empty($matches['script']) && !empty($matches['country'])) {**

**$langcodes[] = $matches['lang'] . $matches['script'] . $matches['country'];**

**}**

**if (!empty($matches['country'])) {**

**$langcodes[] = $matches['lang'] . $matches['country'];**

**}**

**if (!empty($matches['script'])) {**

**$langcodes[] = $matches['lang'] . $matches['script'];**

**}**

**$langcodes[] = $matches['lang'];**

**//Try and find a readable language file for the requested language.**

**$foundFile = false;**

**foreach ($langcodes as $code) {**

**$lang\_file = $lang\_path . 'phpmailer.lang-' . $code . '.php';**

**if (static::fileIsAccessible($lang\_file)) {**

**$foundFile = true;**

**break;**

**}**

**}**

**if ($foundFile === false) {**

**$foundlang = false;**

**} else {**

**$lines = file($lang\_file);**

**foreach ($lines as $line) {**

**//Translation file lines look like this:**

**//$PHPMAILER\_LANG['authenticate'] = 'SMTP-Fehler: Authentifizierung fehlgeschlagen.';**

**//These files are parsed as text and not PHP so as to avoid the possibility of code injection**

**//See https://blog.stevenlevithan.com/archives/match-quoted-string**

**$matches = [];**

**if (**

**preg\_match(**

**'/^\$PHPMAILER\_LANG\[\'([a-z\d\_]+)\'\]\s\*=\s\*(["\'])(.+)\*?\2;/',**

**$line,**

**$matches**

**) &&**

**//Ignore unknown translation keys**

**array\_key\_exists($matches[1], $PHPMAILER\_LANG)**

**) {**

**//Overwrite language-specific strings so we'll never have missing translation keys.**

**$PHPMAILER\_LANG[$matches[1]] = (string)$matches[3];**

**}**

**}**

**}**

**}**

**$this->language = $PHPMAILER\_LANG;**

**return $foundlang; //Returns false if language not found**

**}**

**/\*\***

**\* Get the array of strings for the current language.**

**\***

**\* @return array**

**\*/**

**public function getTranslations()**

**{**

**if (empty($this->language)) {**

**$this->setLanguage(); // Set the default language.**

**}**

**return $this->language;**

**}**

**/\*\***

**\* Create recipient headers.**

**\***

**\* @param string $type**

**\* @param array $addr An array of recipients,**

**\* where each recipient is a 2-element indexed array with element 0 containing an address**

**\* and element 1 containing a name, like:**

**\* [['joe@example.com', 'Joe User'], ['zoe@example.com', 'Zoe User']]**

**\***

**\* @return string**

**\*/**

**public function addrAppend($type, $addr)**

**{**

**$addresses = [];**

**foreach ($addr as $address) {**

**$addresses[] = $this->addrFormat($address);**

**}**

**return $type . ': ' . implode(', ', $addresses) . static::$LE;**

**}**

**/\*\***

**\* Format an address for use in a message header.**

**\***

**\* @param array $addr A 2-element indexed array, element 0 containing an address, element 1 containing a name like**

**\* ['joe@example.com', 'Joe User']**

**\***

**\* @return string**

**\*/**

**public function addrFormat($addr)**

**{**

**if (empty($addr[1])) { //No name provided**

**return $this->secureHeader($addr[0]);**

**}**

**return $this->encodeHeader($this->secureHeader($addr[1]), 'phrase') .**

**' <' . $this->secureHeader($addr[0]) . '>';**

**}**

**/\*\***

**\* Word-wrap message.**

**\* For use with mailers that do not automatically perform wrapping**

**\* and for quoted-printable encoded messages.**

**\* Original written by philippe.**

**\***

**\* @param string $message The message to wrap**

**\* @param int $length The line length to wrap to**

**\* @param bool $qp\_mode Whether to run in Quoted-Printable mode**

**\***

**\* @return string**

**\*/**

**public function wrapText($message, $length, $qp\_mode = false)**

**{**

**if ($qp\_mode) {**

**$soft\_break = sprintf(' =%s', static::$LE);**

**} else {**

**$soft\_break = static::$LE;**

**}**

**//If utf-8 encoding is used, we will need to make sure we don't**

**//split multibyte characters when we wrap**

**$is\_utf8 = static::CHARSET\_UTF8 === strtolower($this->CharSet);**

**$lelen = strlen(static::$LE);**

**$crlflen = strlen(static::$LE);**

**$message = static::normalizeBreaks($message);**

**//Remove a trailing line break**

**if (substr($message, -$lelen) === static::$LE) {**

**$message = substr($message, 0, -$lelen);**

**}**

**//Split message into lines**

**$lines = explode(static::$LE, $message);**

**//Message will be rebuilt in here**

**$message = '';**

**foreach ($lines as $line) {**

**$words = explode(' ', $line);**

**$buf = '';**

**$firstword = true;**

**foreach ($words as $word) {**

**if ($qp\_mode && (strlen($word) > $length)) {**

**$space\_left = $length - strlen($buf) - $crlflen;**

**if (!$firstword) {**

**if ($space\_left > 20) {**

**$len = $space\_left;**

**if ($is\_utf8) {**

**$len = $this->utf8CharBoundary($word, $len);**

**} elseif ('=' === substr($word, $len - 1, 1)) {**

**--$len;**

**} elseif ('=' === substr($word, $len - 2, 1)) {**

**$len -= 2;**

**}**

**$part = substr($word, 0, $len);**

**$word = substr($word, $len);**

**$buf .= ' ' . $part;**

**$message .= $buf . sprintf('=%s', static::$LE);**

**} else {**

**$message .= $buf . $soft\_break;**

**}**

**$buf = '';**

**}**

**while ($word !== '') {**

**if ($length <= 0) {**

**break;**

**}**

**$len = $length;**

**if ($is\_utf8) {**

**$len = $this->utf8CharBoundary($word, $len);**

**} elseif ('=' === substr($word, $len - 1, 1)) {**

**--$len;**

**} elseif ('=' === substr($word, $len - 2, 1)) {**

**$len -= 2;**

**}**

**$part = substr($word, 0, $len);**

**$word = (string) substr($word, $len);**

**if ($word !== '') {**

**$message .= $part . sprintf('=%s', static::$LE);**

**} else {**

**$buf = $part;**

**}**

**}**

**} else {**

**$buf\_o = $buf;**

**if (!$firstword) {**

**$buf .= ' ';**

**}**

**$buf .= $word;**

**if ('' !== $buf\_o && strlen($buf) > $length) {**

**$message .= $buf\_o . $soft\_break;**

**$buf = $word;**

**}**

**}**

**$firstword = false;**

**}**

**$message .= $buf . static::$LE;**

**}**

**return $message;**

**}**

**/\*\***

**\* Find the last character boundary prior to $maxLength in a utf-8**

**\* quoted-printable encoded string.**

**\* Original written by Colin Brown.**

**\***

**\* @param string $encodedText utf-8 QP text**

**\* @param int $maxLength Find the last character boundary prior to this length**

**\***

**\* @return int**

**\*/**

**public function utf8CharBoundary($encodedText, $maxLength)**

**{**

**$foundSplitPos = false;**

**$lookBack = 3;**

**while (!$foundSplitPos) {**

**$lastChunk = substr($encodedText, $maxLength - $lookBack, $lookBack);**

**$encodedCharPos = strpos($lastChunk, '=');**

**if (false !== $encodedCharPos) {**

**//Found start of encoded character byte within $lookBack block.**

**//Check the encoded byte value (the 2 chars after the '=')**

**$hex = substr($encodedText, $maxLength - $lookBack + $encodedCharPos + 1, 2);**

**$dec = hexdec($hex);**

**if ($dec < 128) {**

**//Single byte character.**

**//If the encoded char was found at pos 0, it will fit**

**//otherwise reduce maxLength to start of the encoded char**

**if ($encodedCharPos > 0) {**

**$maxLength -= $lookBack - $encodedCharPos;**

**}**

**$foundSplitPos = true;**

**} elseif ($dec >= 192) {**

**//First byte of a multi byte character**

**//Reduce maxLength to split at start of character**

**$maxLength -= $lookBack - $encodedCharPos;**

**$foundSplitPos = true;**

**} elseif ($dec < 192) {**

**//Middle byte of a multi byte character, look further back**

**$lookBack += 3;**

**}**

**} else {**

**//No encoded character found**

**$foundSplitPos = true;**

**}**

**}**

**return $maxLength;**

**}**

**/\*\***

**\* Apply word wrapping to the message body.**

**\* Wraps the message body to the number of chars set in the WordWrap property.**

**\* You should only do this to plain-text bodies as wrapping HTML tags may break them.**

**\* This is called automatically by createBody(), so you don't need to call it yourself.**

**\*/**

**public function setWordWrap()**

**{**

**if ($this->WordWrap < 1) {**

**return;**

**}**

**switch ($this->message\_type) {**

**case 'alt':**

**case 'alt\_inline':**

**case 'alt\_attach':**

**case 'alt\_inline\_attach':**

**$this->AltBody = $this->wrapText($this->AltBody, $this->WordWrap);**

**break;**

**default:**

**$this->Body = $this->wrapText($this->Body, $this->WordWrap);**

**break;**

**}**

**}**

**/\*\***

**\* Assemble message headers.**

**\***

**\* @return string The assembled headers**

**\*/**

**public function createHeader()**

**{**

**$result = '';**

**$result .= $this->headerLine('Date', '' === $this->MessageDate ? self::rfcDate() : $this->MessageDate);**

**//The To header is created automatically by mail(), so needs to be omitted here**

**if ('mail' !== $this->Mailer) {**

**if ($this->SingleTo) {**

**foreach ($this->to as $toaddr) {**

**$this->SingleToArray[] = $this->addrFormat($toaddr);**

**}**

**} elseif (count($this->to) > 0) {**

**$result .= $this->addrAppend('To', $this->to);**

**} elseif (count($this->cc) === 0) {**

**$result .= $this->headerLine('To', 'undisclosed-recipients:;');**

**}**

**}**

**$result .= $this->addrAppend('From', [[trim($this->From), $this->FromName]]);**

**//sendmail and mail() extract Cc from the header before sending**

**if (count($this->cc) > 0) {**

**$result .= $this->addrAppend('Cc', $this->cc);**

**}**

**//sendmail and mail() extract Bcc from the header before sending**

**if (**

**(**

**'sendmail' === $this->Mailer || 'qmail' === $this->Mailer || 'mail' === $this->Mailer**

**)**

**&& count($this->bcc) > 0**

**) {**

**$result .= $this->addrAppend('Bcc', $this->bcc);**

**}**

**if (count($this->ReplyTo) > 0) {**

**$result .= $this->addrAppend('Reply-To', $this->ReplyTo);**

**}**

**//mail() sets the subject itself**

**if ('mail' !== $this->Mailer) {**

**$result .= $this->headerLine('Subject', $this->encodeHeader($this->secureHeader($this->Subject)));**

**}**

**//Only allow a custom message ID if it conforms to RFC 5322 section 3.6.4**

**//https://tools.ietf.org/html/rfc5322#section-3.6.4**

**if (**

**'' !== $this->MessageID &&**

**preg\_match(**

**'/^<((([a-z\d!#$%&\'\*+\/=?^\_`{|}~-]+(\.[a-z\d!#$%&\'\*+\/=?^\_`{|}~-]+)\*)' .**

**'|("(([\x01-\x08\x0B\x0C\x0E-\x1F\x7F]|[\x21\x23-\x5B\x5D-\x7E])' .**

**'|(\\[\x01-\x09\x0B\x0C\x0E-\x7F]))\*"))@(([a-z\d!#$%&\'\*+\/=?^\_`{|}~-]+' .**

**'(\.[a-z\d!#$%&\'\*+\/=?^\_`{|}~-]+)\*)|(\[(([\x01-\x08\x0B\x0C\x0E-\x1F\x7F]' .**

**'|[\x21-\x5A\x5E-\x7E])|(\\[\x01-\x09\x0B\x0C\x0E-\x7F]))\*\])))>$/Di',**

**$this->MessageID**

**)**

**) {**

**$this->lastMessageID = $this->MessageID;**

**} else {**

**$this->lastMessageID = sprintf('<%s@%s>', $this->uniqueid, $this->serverHostname());**

**}**

**$result .= $this->headerLine('Message-ID', $this->lastMessageID);**

**if (null !== $this->Priority) {**

**$result .= $this->headerLine('X-Priority', $this->Priority);**

**}**

**if ('' === $this->XMailer) {**

**//Empty string for default X-Mailer header**

**$result .= $this->headerLine(**

**'X-Mailer',**

**'PHPMailer ' . self::VERSION . ' (https://github.com/PHPMailer/PHPMailer)'**

**);**

**} elseif (is\_string($this->XMailer) && trim($this->XMailer) !== '') {**

**//Some string**

**$result .= $this->headerLine('X-Mailer', trim($this->XMailer));**

**} //Other values result in no X-Mailer header**

**if ('' !== $this->ConfirmReadingTo) {**

**$result .= $this->headerLine('Disposition-Notification-To', '<' . $this->ConfirmReadingTo . '>');**

**}**

**//Add custom headers**

**foreach ($this->CustomHeader as $header) {**

**$result .= $this->headerLine(**

**trim($header[0]),**

**$this->encodeHeader(trim($header[1]))**

**);**

**}**

**if (!$this->sign\_key\_file) {**

**$result .= $this->headerLine('MIME-Version', '1.0');**

**$result .= $this->getMailMIME();**

**}**

**return $result;**

**}**

**/\*\***

**\* Get the message MIME type headers.**

**\***

**\* @return string**

**\*/**

**public function getMailMIME()**

**{**

**$result = '';**

**$ismultipart = true;**

**switch ($this->message\_type) {**

**case 'inline':**

**$result .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_RELATED . ';');**

**$result .= $this->textLine(' boundary="' . $this->boundary[1] . '"');**

**break;**

**case 'attach':**

**case 'inline\_attach':**

**case 'alt\_attach':**

**case 'alt\_inline\_attach':**

**$result .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_MIXED . ';');**

**$result .= $this->textLine(' boundary="' . $this->boundary[1] . '"');**

**break;**

**case 'alt':**

**case 'alt\_inline':**

**$result .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_ALTERNATIVE . ';');**

**$result .= $this->textLine(' boundary="' . $this->boundary[1] . '"');**

**break;**

**default:**

**//Catches case 'plain': and case '':**

**$result .= $this->textLine('Content-Type: ' . $this->ContentType . '; charset=' . $this->CharSet);**

**$ismultipart = false;**

**break;**

**}**

**//RFC1341 part 5 says 7bit is assumed if not specified**

**if (static::ENCODING\_7BIT !== $this->Encoding) {**

**//RFC 2045 section 6.4 says multipart MIME parts may only use 7bit, 8bit or binary CTE**

**if ($ismultipart) {**

**if (static::ENCODING\_8BIT === $this->Encoding) {**

**$result .= $this->headerLine('Content-Transfer-Encoding', static::ENCODING\_8BIT);**

**}**

**//The only remaining alternatives are quoted-printable and base64, which are both 7bit compatible**

**} else {**

**$result .= $this->headerLine('Content-Transfer-Encoding', $this->Encoding);**

**}**

**}**

**return $result;**

**}**

**/\*\***

**\* Returns the whole MIME message.**

**\* Includes complete headers and body.**

**\* Only valid post preSend().**

**\***

**\* @see PHPMailer::preSend()**

**\***

**\* @return string**

**\*/**

**public function getSentMIMEMessage()**

**{**

**return static::stripTrailingWSP($this->MIMEHeader . $this->mailHeader) .**

**static::$LE . static::$LE . $this->MIMEBody;**

**}**

**/\*\***

**\* Create a unique ID to use for boundaries.**

**\***

**\* @return string**

**\*/**

**protected function generateId()**

**{**

**$len = 32; //32 bytes = 256 bits**

**$bytes = '';**

**if (function\_exists('random\_bytes')) {**

**try {**

**$bytes = random\_bytes($len);**

**} catch (\Exception $e) {**

**//Do nothing**

**}**

**} elseif (function\_exists('openssl\_random\_pseudo\_bytes')) {**

**/\*\* @noinspection CryptographicallySecureRandomnessInspection \*/**

**$bytes = openssl\_random\_pseudo\_bytes($len);**

**}**

**if ($bytes === '') {**

**//We failed to produce a proper random string, so make do.**

**//Use a hash to force the length to the same as the other methods**

**$bytes = hash('sha256', uniqid((string) mt\_rand(), true), true);**

**}**

**//We don't care about messing up base64 format here, just want a random string**

**return str\_replace(['=', '+', '/'], '', base64\_encode(hash('sha256', $bytes, true)));**

**}**

**/\*\***

**\* Assemble the message body.**

**\* Returns an empty string on failure.**

**\***

**\* @throws Exception**

**\***

**\* @return string The assembled message body**

**\*/**

**public function createBody()**

**{**

**$body = '';**

**//Create unique IDs and preset boundaries**

**$this->uniqueid = $this->generateId();**

**$this->boundary[1] = 'b1\_' . $this->uniqueid;**

**$this->boundary[2] = 'b2\_' . $this->uniqueid;**

**$this->boundary[3] = 'b3\_' . $this->uniqueid;**

**if ($this->sign\_key\_file) {**

**$body .= $this->getMailMIME() . static::$LE;**

**}**

**$this->setWordWrap();**

**$bodyEncoding = $this->Encoding;**

**$bodyCharSet = $this->CharSet;**

**//Can we do a 7-bit downgrade?**

**if (static::ENCODING\_8BIT === $bodyEncoding && !$this->has8bitChars($this->Body)) {**

**$bodyEncoding = static::ENCODING\_7BIT;**

**//All ISO 8859, Windows codepage and UTF-8 charsets are ascii compatible up to 7-bit**

**$bodyCharSet = static::CHARSET\_ASCII;**

**}**

**//If lines are too long, and we're not already using an encoding that will shorten them,**

**//change to quoted-printable transfer encoding for the body part only**

**if (static::ENCODING\_BASE64 !== $this->Encoding && static::hasLineLongerThanMax($this->Body)) {**

**$bodyEncoding = static::ENCODING\_QUOTED\_PRINTABLE;**

**}**

**$altBodyEncoding = $this->Encoding;**

**$altBodyCharSet = $this->CharSet;**

**//Can we do a 7-bit downgrade?**

**if (static::ENCODING\_8BIT === $altBodyEncoding && !$this->has8bitChars($this->AltBody)) {**

**$altBodyEncoding = static::ENCODING\_7BIT;**

**//All ISO 8859, Windows codepage and UTF-8 charsets are ascii compatible up to 7-bit**

**$altBodyCharSet = static::CHARSET\_ASCII;**

**}**

**//If lines are too long, and we're not already using an encoding that will shorten them,**

**//change to quoted-printable transfer encoding for the alt body part only**

**if (static::ENCODING\_BASE64 !== $altBodyEncoding && static::hasLineLongerThanMax($this->AltBody)) {**

**$altBodyEncoding = static::ENCODING\_QUOTED\_PRINTABLE;**

**}**

**//Use this as a preamble in all multipart message types**

**$mimepre = 'This is a multi-part message in MIME format.' . static::$LE . static::$LE;**

**switch ($this->message\_type) {**

**case 'inline':**

**$body .= $mimepre;**

**$body .= $this->getBoundary($this->boundary[1], $bodyCharSet, '', $bodyEncoding);**

**$body .= $this->encodeString($this->Body, $bodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->attachAll('inline', $this->boundary[1]);**

**break;**

**case 'attach':**

**$body .= $mimepre;**

**$body .= $this->getBoundary($this->boundary[1], $bodyCharSet, '', $bodyEncoding);**

**$body .= $this->encodeString($this->Body, $bodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->attachAll('attachment', $this->boundary[1]);**

**break;**

**case 'inline\_attach':**

**$body .= $mimepre;**

**$body .= $this->textLine('--' . $this->boundary[1]);**

**$body .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_RELATED . ';');**

**$body .= $this->textLine(' boundary="' . $this->boundary[2] . '";');**

**$body .= $this->textLine(' type="' . static::CONTENT\_TYPE\_TEXT\_HTML . '"');**

**$body .= static::$LE;**

**$body .= $this->getBoundary($this->boundary[2], $bodyCharSet, '', $bodyEncoding);**

**$body .= $this->encodeString($this->Body, $bodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->attachAll('inline', $this->boundary[2]);**

**$body .= static::$LE;**

**$body .= $this->attachAll('attachment', $this->boundary[1]);**

**break;**

**case 'alt':**

**$body .= $mimepre;**

**$body .= $this->getBoundary(**

**$this->boundary[1],**

**$altBodyCharSet,**

**static::CONTENT\_TYPE\_PLAINTEXT,**

**$altBodyEncoding**

**);**

**$body .= $this->encodeString($this->AltBody, $altBodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->getBoundary(**

**$this->boundary[1],**

**$bodyCharSet,**

**static::CONTENT\_TYPE\_TEXT\_HTML,**

**$bodyEncoding**

**);**

**$body .= $this->encodeString($this->Body, $bodyEncoding);**

**$body .= static::$LE;**

**if (!empty($this->Ical)) {**

**$method = static::ICAL\_METHOD\_REQUEST;**

**foreach (static::$IcalMethods as $imethod) {**

**if (stripos($this->Ical, 'METHOD:' . $imethod) !== false) {**

**$method = $imethod;**

**break;**

**}**

**}**

**$body .= $this->getBoundary(**

**$this->boundary[1],**

**'',**

**static::CONTENT\_TYPE\_TEXT\_CALENDAR . '; method=' . $method,**

**''**

**);**

**$body .= $this->encodeString($this->Ical, $this->Encoding);**

**$body .= static::$LE;**

**}**

**$body .= $this->endBoundary($this->boundary[1]);**

**break;**

**case 'alt\_inline':**

**$body .= $mimepre;**

**$body .= $this->getBoundary(**

**$this->boundary[1],**

**$altBodyCharSet,**

**static::CONTENT\_TYPE\_PLAINTEXT,**

**$altBodyEncoding**

**);**

**$body .= $this->encodeString($this->AltBody, $altBodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->textLine('--' . $this->boundary[1]);**

**$body .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_RELATED . ';');**

**$body .= $this->textLine(' boundary="' . $this->boundary[2] . '";');**

**$body .= $this->textLine(' type="' . static::CONTENT\_TYPE\_TEXT\_HTML . '"');**

**$body .= static::$LE;**

**$body .= $this->getBoundary(**

**$this->boundary[2],**

**$bodyCharSet,**

**static::CONTENT\_TYPE\_TEXT\_HTML,**

**$bodyEncoding**

**);**

**$body .= $this->encodeString($this->Body, $bodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->attachAll('inline', $this->boundary[2]);**

**$body .= static::$LE;**

**$body .= $this->endBoundary($this->boundary[1]);**

**break;**

**case 'alt\_attach':**

**$body .= $mimepre;**

**$body .= $this->textLine('--' . $this->boundary[1]);**

**$body .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_ALTERNATIVE . ';');**

**$body .= $this->textLine(' boundary="' . $this->boundary[2] . '"');**

**$body .= static::$LE;**

**$body .= $this->getBoundary(**

**$this->boundary[2],**

**$altBodyCharSet,**

**static::CONTENT\_TYPE\_PLAINTEXT,**

**$altBodyEncoding**

**);**

**$body .= $this->encodeString($this->AltBody, $altBodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->getBoundary(**

**$this->boundary[2],**

**$bodyCharSet,**

**static::CONTENT\_TYPE\_TEXT\_HTML,**

**$bodyEncoding**

**);**

**$body .= $this->encodeString($this->Body, $bodyEncoding);**

**$body .= static::$LE;**

**if (!empty($this->Ical)) {**

**$method = static::ICAL\_METHOD\_REQUEST;**

**foreach (static::$IcalMethods as $imethod) {**

**if (stripos($this->Ical, 'METHOD:' . $imethod) !== false) {**

**$method = $imethod;**

**break;**

**}**

**}**

**$body .= $this->getBoundary(**

**$this->boundary[2],**

**'',**

**static::CONTENT\_TYPE\_TEXT\_CALENDAR . '; method=' . $method,**

**''**

**);**

**$body .= $this->encodeString($this->Ical, $this->Encoding);**

**}**

**$body .= $this->endBoundary($this->boundary[2]);**

**$body .= static::$LE;**

**$body .= $this->attachAll('attachment', $this->boundary[1]);**

**break;**

**case 'alt\_inline\_attach':**

**$body .= $mimepre;**

**$body .= $this->textLine('--' . $this->boundary[1]);**

**$body .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_ALTERNATIVE . ';');**

**$body .= $this->textLine(' boundary="' . $this->boundary[2] . '"');**

**$body .= static::$LE;**

**$body .= $this->getBoundary(**

**$this->boundary[2],**

**$altBodyCharSet,**

**static::CONTENT\_TYPE\_PLAINTEXT,**

**$altBodyEncoding**

**);**

**$body .= $this->encodeString($this->AltBody, $altBodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->textLine('--' . $this->boundary[2]);**

**$body .= $this->headerLine('Content-Type', static::CONTENT\_TYPE\_MULTIPART\_RELATED . ';');**

**$body .= $this->textLine(' boundary="' . $this->boundary[3] . '";');**

**$body .= $this->textLine(' type="' . static::CONTENT\_TYPE\_TEXT\_HTML . '"');**

**$body .= static::$LE;**

**$body .= $this->getBoundary(**

**$this->boundary[3],**

**$bodyCharSet,**

**static::CONTENT\_TYPE\_TEXT\_HTML,**

**$bodyEncoding**

**);**

**$body .= $this->encodeString($this->Body, $bodyEncoding);**

**$body .= static::$LE;**

**$body .= $this->attachAll('inline', $this->boundary[3]);**

**$body .= static::$LE;**

**$body .= $this->endBoundary($this->boundary[2]);**

**$body .= static::$LE;**

**$body .= $this->attachAll('attachment', $this->boundary[1]);**

**break;**

**default:**

**//Catch case 'plain' and case '', applies to simple `text/plain` and `text/html` body content types**

**//Reset the `Encoding` property in case we changed it for line length reasons**

**$this->Encoding = $bodyEncoding;**

**$body .= $this->encodeString($this->Body, $this->Encoding);**

**break;**

**}**

**if ($this->isError()) {**

**$body = '';**

**if ($this->exceptions) {**

**throw new Exception($this->lang('empty\_message'), self::STOP\_CRITICAL);**

**}**

**} elseif ($this->sign\_key\_file) {**

**try {**

**if (!defined('PKCS7\_TEXT')) {**

**throw new Exception($this->lang('extension\_missing') . 'openssl');**

**}**

**$file = tempnam(sys\_get\_temp\_dir(), 'srcsign');**

**$signed = tempnam(sys\_get\_temp\_dir(), 'mailsign');**

**file\_put\_contents($file, $body);**

**//Workaround for PHP bug https://bugs.php.net/bug.php?id=69197**

**if (empty($this->sign\_extracerts\_file)) {**

**$sign = @openssl\_pkcs7\_sign(**

**$file,**

**$signed,**

**'file://' . realpath($this->sign\_cert\_file),**

**['file://' . realpath($this->sign\_key\_file), $this->sign\_key\_pass],**

**[]**

**);**

**} else {**

**$sign = @openssl\_pkcs7\_sign(**

**$file,**

**$signed,**

**'file://' . realpath($this->sign\_cert\_file),**

**['file://' . realpath($this->sign\_key\_file), $this->sign\_key\_pass],**

**[],**

**PKCS7\_DETACHED,**

**$this->sign\_extracerts\_file**

**);**

**}**

**@unlink($file);**

**if ($sign) {**

**$body = file\_get\_contents($signed);**

**@unlink($signed);**

**//The message returned by openssl contains both headers and body, so need to split them up**

**$parts = explode("\n\n", $body, 2);**

**$this->MIMEHeader .= $parts[0] . static::$LE . static::$LE;**

**$body = $parts[1];**

**} else {**

**@unlink($signed);**

**throw new Exception($this->lang('signing') . openssl\_error\_string());**

**}**

**} catch (Exception $exc) {**

**$body = '';**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**}**

**}**

**return $body;**

**}**

**/\*\***

**\* Return the start of a message boundary.**

**\***

**\* @param string $boundary**

**\* @param string $charSet**

**\* @param string $contentType**

**\* @param string $encoding**

**\***

**\* @return string**

**\*/**

**protected function getBoundary($boundary, $charSet, $contentType, $encoding)**

**{**

**$result = '';**

**if ('' === $charSet) {**

**$charSet = $this->CharSet;**

**}**

**if ('' === $contentType) {**

**$contentType = $this->ContentType;**

**}**

**if ('' === $encoding) {**

**$encoding = $this->Encoding;**

**}**

**$result .= $this->textLine('--' . $boundary);**

**$result .= sprintf('Content-Type: %s; charset=%s', $contentType, $charSet);**

**$result .= static::$LE;**

**//RFC1341 part 5 says 7bit is assumed if not specified**

**if (static::ENCODING\_7BIT !== $encoding) {**

**$result .= $this->headerLine('Content-Transfer-Encoding', $encoding);**

**}**

**$result .= static::$LE;**

**return $result;**

**}**

**/\*\***

**\* Return the end of a message boundary.**

**\***

**\* @param string $boundary**

**\***

**\* @return string**

**\*/**

**protected function endBoundary($boundary)**

**{**

**return static::$LE . '--' . $boundary . '--' . static::$LE;**

**}**

**/\*\***

**\* Set the message type.**

**\* PHPMailer only supports some preset message types, not arbitrary MIME structures.**

**\*/**

**protected function setMessageType()**

**{**

**$type = [];**

**if ($this->alternativeExists()) {**

**$type[] = 'alt';**

**}**

**if ($this->inlineImageExists()) {**

**$type[] = 'inline';**

**}**

**if ($this->attachmentExists()) {**

**$type[] = 'attach';**

**}**

**$this->message\_type = implode('\_', $type);**

**if ('' === $this->message\_type) {**

**//The 'plain' message\_type refers to the message having a single body element, not that it is plain-text**

**$this->message\_type = 'plain';**

**}**

**}**

**/\*\***

**\* Format a header line.**

**\***

**\* @param string $name**

**\* @param string|int $value**

**\***

**\* @return string**

**\*/**

**public function headerLine($name, $value)**

**{**

**return $name . ': ' . $value . static::$LE;**

**}**

**/\*\***

**\* Return a formatted mail line.**

**\***

**\* @param string $value**

**\***

**\* @return string**

**\*/**

**public function textLine($value)**

**{**

**return $value . static::$LE;**

**}**

**/\*\***

**\* Add an attachment from a path on the filesystem.**

**\* Never use a user-supplied path to a file!**

**\* Returns false if the file could not be found or read.**

**\* Explicitly \*does not\* support passing URLs; PHPMailer is not an HTTP client.**

**\* If you need to do that, fetch the resource yourself and pass it in via a local file or string.**

**\***

**\* @param string $path Path to the attachment**

**\* @param string $name Overrides the attachment name**

**\* @param string $encoding File encoding (see $Encoding)**

**\* @param string $type MIME type, e.g. `image/jpeg`; determined automatically from $path if not specified**

**\* @param string $disposition Disposition to use**

**\***

**\* @throws Exception**

**\***

**\* @return bool**

**\*/**

**public function addAttachment(**

**$path,**

**$name = '',**

**$encoding = self::ENCODING\_BASE64,**

**$type = '',**

**$disposition = 'attachment'**

**) {**

**try {**

**if (!static::fileIsAccessible($path)) {**

**throw new Exception($this->lang('file\_access') . $path, self::STOP\_CONTINUE);**

**}**

**//If a MIME type is not specified, try to work it out from the file name**

**if ('' === $type) {**

**$type = static::filenameToType($path);**

**}**

**$filename = (string) static::mb\_pathinfo($path, PATHINFO\_BASENAME);**

**if ('' === $name) {**

**$name = $filename;**

**}**

**if (!$this->validateEncoding($encoding)) {**

**throw new Exception($this->lang('encoding') . $encoding);**

**}**

**$this->attachment[] = [**

**0 => $path,**

**1 => $filename,**

**2 => $name,**

**3 => $encoding,**

**4 => $type,**

**5 => false, //isStringAttachment**

**6 => $disposition,**

**7 => $name,**

**];**

**} catch (Exception $exc) {**

**$this->setError($exc->getMessage());**

**$this->edebug($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**return false;**

**}**

**return true;**

**}**

**/\*\***

**\* Return the array of attachments.**

**\***

**\* @return array**

**\*/**

**public function getAttachments()**

**{**

**return $this->attachment;**

**}**

**/\*\***

**\* Attach all file, string, and binary attachments to the message.**

**\* Returns an empty string on failure.**

**\***

**\* @param string $disposition\_type**

**\* @param string $boundary**

**\***

**\* @throws Exception**

**\***

**\* @return string**

**\*/**

**protected function attachAll($disposition\_type, $boundary)**

**{**

**//Return text of body**

**$mime = [];**

**$cidUniq = [];**

**$incl = [];**

**//Add all attachments**

**foreach ($this->attachment as $attachment) {**

**//Check if it is a valid disposition\_filter**

**if ($attachment[6] === $disposition\_type) {**

**//Check for string attachment**

**$string = '';**

**$path = '';**

**$bString = $attachment[5];**

**if ($bString) {**

**$string = $attachment[0];**

**} else {**

**$path = $attachment[0];**

**}**

**$inclhash = hash('sha256', serialize($attachment));**

**if (in\_array($inclhash, $incl, true)) {**

**continue;**

**}**

**$incl[] = $inclhash;**

**$name = $attachment[2];**

**$encoding = $attachment[3];**

**$type = $attachment[4];**

**$disposition = $attachment[6];**

**$cid = $attachment[7];**

**if ('inline' === $disposition && array\_key\_exists($cid, $cidUniq)) {**

**continue;**

**}**

**$cidUniq[$cid] = true;**

**$mime[] = sprintf('--%s%s', $boundary, static::$LE);**

**//Only include a filename property if we have one**

**if (!empty($name)) {**

**$mime[] = sprintf(**

**'Content-Type: %s; name=%s%s',**

**$type,**

**static::quotedString($this->encodeHeader($this->secureHeader($name))),**

**static::$LE**

**);**

**} else {**

**$mime[] = sprintf(**

**'Content-Type: %s%s',**

**$type,**

**static::$LE**

**);**

**}**

**//RFC1341 part 5 says 7bit is assumed if not specified**

**if (static::ENCODING\_7BIT !== $encoding) {**

**$mime[] = sprintf('Content-Transfer-Encoding: %s%s', $encoding, static::$LE);**

**}**

**//Only set Content-IDs on inline attachments**

**if ((string) $cid !== '' && $disposition === 'inline') {**

**$mime[] = 'Content-ID: <' . $this->encodeHeader($this->secureHeader($cid)) . '>' . static::$LE;**

**}**

**//Allow for bypassing the Content-Disposition header**

**if (!empty($disposition)) {**

**$encoded\_name = $this->encodeHeader($this->secureHeader($name));**

**if (!empty($encoded\_name)) {**

**$mime[] = sprintf(**

**'Content-Disposition: %s; filename=%s%s',**

**$disposition,**

**static::quotedString($encoded\_name),**

**static::$LE . static::$LE**

**);**

**} else {**

**$mime[] = sprintf(**

**'Content-Disposition: %s%s',**

**$disposition,**

**static::$LE . static::$LE**

**);**

**}**

**} else {**

**$mime[] = static::$LE;**

**}**

**//Encode as string attachment**

**if ($bString) {**

**$mime[] = $this->encodeString($string, $encoding);**

**} else {**

**$mime[] = $this->encodeFile($path, $encoding);**

**}**

**if ($this->isError()) {**

**return '';**

**}**

**$mime[] = static::$LE;**

**}**

**}**

**$mime[] = sprintf('--%s--%s', $boundary, static::$LE);**

**return implode('', $mime);**

**}**

**/\*\***

**\* Encode a file attachment in requested format.**

**\* Returns an empty string on failure.**

**\***

**\* @param string $path The full path to the file**

**\* @param string $encoding The encoding to use; one of 'base64', '7bit', '8bit', 'binary', 'quoted-printable'**

**\***

**\* @return string**

**\*/**

**protected function encodeFile($path, $encoding = self::ENCODING\_BASE64)**

**{**

**try {**

**if (!static::fileIsAccessible($path)) {**

**throw new Exception($this->lang('file\_open') . $path, self::STOP\_CONTINUE);**

**}**

**$file\_buffer = file\_get\_contents($path);**

**if (false === $file\_buffer) {**

**throw new Exception($this->lang('file\_open') . $path, self::STOP\_CONTINUE);**

**}**

**$file\_buffer = $this->encodeString($file\_buffer, $encoding);**

**return $file\_buffer;**

**} catch (Exception $exc) {**

**$this->setError($exc->getMessage());**

**$this->edebug($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**return '';**

**}**

**}**

**/\*\***

**\* Encode a string in requested format.**

**\* Returns an empty string on failure.**

**\***

**\* @param string $str The text to encode**

**\* @param string $encoding The encoding to use; one of 'base64', '7bit', '8bit', 'binary', 'quoted-printable'**

**\***

**\* @throws Exception**

**\***

**\* @return string**

**\*/**

**public function encodeString($str, $encoding = self::ENCODING\_BASE64)**

**{**

**$encoded = '';**

**switch (strtolower($encoding)) {**

**case static::ENCODING\_BASE64:**

**$encoded = chunk\_split(**

**base64\_encode($str),**

**static::STD\_LINE\_LENGTH,**

**static::$LE**

**);**

**break;**

**case static::ENCODING\_7BIT:**

**case static::ENCODING\_8BIT:**

**$encoded = static::normalizeBreaks($str);**

**//Make sure it ends with a line break**

**if (substr($encoded, -(strlen(static::$LE))) !== static::$LE) {**

**$encoded .= static::$LE;**

**}**

**break;**

**case static::ENCODING\_BINARY:**

**$encoded = $str;**

**break;**

**case static::ENCODING\_QUOTED\_PRINTABLE:**

**$encoded = $this->encodeQP($str);**

**break;**

**default:**

**$this->setError($this->lang('encoding') . $encoding);**

**if ($this->exceptions) {**

**throw new Exception($this->lang('encoding') . $encoding);**

**}**

**break;**

**}**

**return $encoded;**

**}**

**/\*\***

**\* Encode a header value (not including its label) optimally.**

**\* Picks shortest of Q, B, or none. Result includes folding if needed.**

**\* See RFC822 definitions for phrase, comment and text positions.**

**\***

**\* @param string $str The header value to encode**

**\* @param string $position What context the string will be used in**

**\***

**\* @return string**

**\*/**

**public function encodeHeader($str, $position = 'text')**

**{**

**$matchcount = 0;**

**switch (strtolower($position)) {**

**case 'phrase':**

**if (!preg\_match('/[\200-\377]/', $str)) {**

**//Can't use addslashes as we don't know the value of magic\_quotes\_sybase**

**$encoded = addcslashes($str, "\0..\37\177\\\"");**

**if (($str === $encoded) && !preg\_match('/[^A-Za-z0-9!#$%&\'\*+\/=?^\_`{|}~ -]/', $str)) {**

**return $encoded;**

**}**

**return "\"$encoded\"";**

**}**

**$matchcount = preg\_match\_all('/[^\040\041\043-\133\135-\176]/', $str, $matches);**

**break;**

**/\* @noinspection PhpMissingBreakStatementInspection \*/**

**case 'comment':**

**$matchcount = preg\_match\_all('/[()"]/', $str, $matches);**

**//fallthrough**

**case 'text':**

**default:**

**$matchcount += preg\_match\_all('/[\000-\010\013\014\016-\037\177-\377]/', $str, $matches);**

**break;**

**}**

**if ($this->has8bitChars($str)) {**

**$charset = $this->CharSet;**

**} else {**

**$charset = static::CHARSET\_ASCII;**

**}**

**//Q/B encoding adds 8 chars and the charset ("` =?<charset>?[QB]?<content>?=`").**

**$overhead = 8 + strlen($charset);**

**if ('mail' === $this->Mailer) {**

**$maxlen = static::MAIL\_MAX\_LINE\_LENGTH - $overhead;**

**} else {**

**$maxlen = static::MAX\_LINE\_LENGTH - $overhead;**

**}**

**//Select the encoding that produces the shortest output and/or prevents corruption.**

**if ($matchcount > strlen($str) / 3) {**

**//More than 1/3 of the content needs encoding, use B-encode.**

**$encoding = 'B';**

**} elseif ($matchcount > 0) {**

**//Less than 1/3 of the content needs encoding, use Q-encode.**

**$encoding = 'Q';**

**} elseif (strlen($str) > $maxlen) {**

**//No encoding needed, but value exceeds max line length, use Q-encode to prevent corruption.**

**$encoding = 'Q';**

**} else {**

**//No reformatting needed**

**$encoding = false;**

**}**

**switch ($encoding) {**

**case 'B':**

**if ($this->hasMultiBytes($str)) {**

**//Use a custom function which correctly encodes and wraps long**

**//multibyte strings without breaking lines within a character**

**$encoded = $this->base64EncodeWrapMB($str, "\n");**

**} else {**

**$encoded = base64\_encode($str);**

**$maxlen -= $maxlen % 4;**

**$encoded = trim(chunk\_split($encoded, $maxlen, "\n"));**

**}**

**$encoded = preg\_replace('/^(.\*)$/m', ' =?' . $charset . "?$encoding?\\1?=", $encoded);**

**break;**

**case 'Q':**

**$encoded = $this->encodeQ($str, $position);**

**$encoded = $this->wrapText($encoded, $maxlen, true);**

**$encoded = str\_replace('=' . static::$LE, "\n", trim($encoded));**

**$encoded = preg\_replace('/^(.\*)$/m', ' =?' . $charset . "?$encoding?\\1?=", $encoded);**

**break;**

**default:**

**return $str;**

**}**

**return trim(static::normalizeBreaks($encoded));**

**}**

**/\*\***

**\* Check if a string contains multi-byte characters.**

**\***

**\* @param string $str multi-byte text to wrap encode**

**\***

**\* @return bool**

**\*/**

**public function hasMultiBytes($str)**

**{**

**if (function\_exists('mb\_strlen')) {**

**return strlen($str) > mb\_strlen($str, $this->CharSet);**

**}**

**//Assume no multibytes (we can't handle without mbstring functions anyway)**

**return false;**

**}**

**/\*\***

**\* Does a string contain any 8-bit chars (in any charset)?**

**\***

**\* @param string $text**

**\***

**\* @return bool**

**\*/**

**public function has8bitChars($text)**

**{**

**return (bool) preg\_match('/[\x80-\xFF]/', $text);**

**}**

**/\*\***

**\* Encode and wrap long multibyte strings for mail headers**

**\* without breaking lines within a character.**

**\* Adapted from a function by paravoid.**

**\***

**\* @see http://www.php.net/manual/en/function.mb-encode-mimeheader.php#60283**

**\***

**\* @param string $str multi-byte text to wrap encode**

**\* @param string $linebreak string to use as linefeed/end-of-line**

**\***

**\* @return string**

**\*/**

**public function base64EncodeWrapMB($str, $linebreak = null)**

**{**

**$start = '=?' . $this->CharSet . '?B?';**

**$end = '?=';**

**$encoded = '';**

**if (null === $linebreak) {**

**$linebreak = static::$LE;**

**}**

**$mb\_length = mb\_strlen($str, $this->CharSet);**

**//Each line must have length <= 75, including $start and $end**

**$length = 75 - strlen($start) - strlen($end);**

**//Average multi-byte ratio**

**$ratio = $mb\_length / strlen($str);**

**//Base64 has a 4:3 ratio**

**$avgLength = floor($length \* $ratio \* .75);**

**$offset = 0;**

**for ($i = 0; $i < $mb\_length; $i += $offset) {**

**$lookBack = 0;**

**do {**

**$offset = $avgLength - $lookBack;**

**$chunk = mb\_substr($str, $i, $offset, $this->CharSet);**

**$chunk = base64\_encode($chunk);**

**++$lookBack;**

**} while (strlen($chunk) > $length);**

**$encoded .= $chunk . $linebreak;**

**}**

**//Chomp the last linefeed**

**return substr($encoded, 0, -strlen($linebreak));**

**}**

**/\*\***

**\* Encode a string in quoted-printable format.**

**\* According to RFC2045 section 6.7.**

**\***

**\* @param string $string The text to encode**

**\***

**\* @return string**

**\*/**

**public function encodeQP($string)**

**{**

**return static::normalizeBreaks(quoted\_printable\_encode($string));**

**}**

**/\*\***

**\* Encode a string using Q encoding.**

**\***

**\* @see http://tools.ietf.org/html/rfc2047#section-4.2**

**\***

**\* @param string $str the text to encode**

**\* @param string $position Where the text is going to be used, see the RFC for what that means**

**\***

**\* @return string**

**\*/**

**public function encodeQ($str, $position = 'text')**

**{**

**//There should not be any EOL in the string**

**$pattern = '';**

**$encoded = str\_replace(["\r", "\n"], '', $str);**

**switch (strtolower($position)) {**

**case 'phrase':**

**//RFC 2047 section 5.3**

**$pattern = '^A-Za-z0-9!\*+\/ -';**

**break;**

**/\***

**\* RFC 2047 section 5.2.**

**\* Build $pattern without including delimiters and []**

**\*/**

**/\* @noinspection PhpMissingBreakStatementInspection \*/**

**case 'comment':**

**$pattern = '\(\)"';**

**/\* Intentional fall through \*/**

**case 'text':**

**default:**

**//RFC 2047 section 5.1**

**//Replace every high ascii, control, =, ? and \_ characters**

**$pattern = '\000-\011\013\014\016-\037\075\077\137\177-\377' . $pattern;**

**break;**

**}**

**$matches = [];**

**if (preg\_match\_all("/[{$pattern}]/", $encoded, $matches)) {**

**//If the string contains an '=', make sure it's the first thing we replace**

**//so as to avoid double-encoding**

**$eqkey = array\_search('=', $matches[0], true);**

**if (false !== $eqkey) {**

**unset($matches[0][$eqkey]);**

**array\_unshift($matches[0], '=');**

**}**

**foreach (array\_unique($matches[0]) as $char) {**

**$encoded = str\_replace($char, '=' . sprintf('%02X', ord($char)), $encoded);**

**}**

**}**

**//Replace spaces with \_ (more readable than =20)**

**//RFC 2047 section 4.2(2)**

**return str\_replace(' ', '\_', $encoded);**

**}**

**/\*\***

**\* Add a string or binary attachment (non-filesystem).**

**\* This method can be used to attach ascii or binary data,**

**\* such as a BLOB record from a database.**

**\***

**\* @param string $string String attachment data**

**\* @param string $filename Name of the attachment**

**\* @param string $encoding File encoding (see $Encoding)**

**\* @param string $type File extension (MIME) type**

**\* @param string $disposition Disposition to use**

**\***

**\* @throws Exception**

**\***

**\* @return bool True on successfully adding an attachment**

**\*/**

**public function addStringAttachment(**

**$string,**

**$filename,**

**$encoding = self::ENCODING\_BASE64,**

**$type = '',**

**$disposition = 'attachment'**

**) {**

**try {**

**//If a MIME type is not specified, try to work it out from the file name**

**if ('' === $type) {**

**$type = static::filenameToType($filename);**

**}**

**if (!$this->validateEncoding($encoding)) {**

**throw new Exception($this->lang('encoding') . $encoding);**

**}**

**//Append to $attachment array**

**$this->attachment[] = [**

**0 => $string,**

**1 => $filename,**

**2 => static::mb\_pathinfo($filename, PATHINFO\_BASENAME),**

**3 => $encoding,**

**4 => $type,**

**5 => true, //isStringAttachment**

**6 => $disposition,**

**7 => 0,**

**];**

**} catch (Exception $exc) {**

**$this->setError($exc->getMessage());**

**$this->edebug($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**return false;**

**}**

**return true;**

**}**

**/\*\***

**\* Add an embedded (inline) attachment from a file.**

**\* This can include images, sounds, and just about any other document type.**

**\* These differ from 'regular' attachments in that they are intended to be**

**\* displayed inline with the message, not just attached for download.**

**\* This is used in HTML messages that embed the images**

**\* the HTML refers to using the $cid value.**

**\* Never use a user-supplied path to a file!**

**\***

**\* @param string $path Path to the attachment**

**\* @param string $cid Content ID of the attachment; Use this to reference**

**\* the content when using an embedded image in HTML**

**\* @param string $name Overrides the attachment name**

**\* @param string $encoding File encoding (see $Encoding)**

**\* @param string $type File MIME type**

**\* @param string $disposition Disposition to use**

**\***

**\* @throws Exception**

**\***

**\* @return bool True on successfully adding an attachment**

**\*/**

**public function addEmbeddedImage(**

**$path,**

**$cid,**

**$name = '',**

**$encoding = self::ENCODING\_BASE64,**

**$type = '',**

**$disposition = 'inline'**

**) {**

**try {**

**if (!static::fileIsAccessible($path)) {**

**throw new Exception($this->lang('file\_access') . $path, self::STOP\_CONTINUE);**

**}**

**//If a MIME type is not specified, try to work it out from the file name**

**if ('' === $type) {**

**$type = static::filenameToType($path);**

**}**

**if (!$this->validateEncoding($encoding)) {**

**throw new Exception($this->lang('encoding') . $encoding);**

**}**

**$filename = (string) static::mb\_pathinfo($path, PATHINFO\_BASENAME);**

**if ('' === $name) {**

**$name = $filename;**

**}**

**//Append to $attachment array**

**$this->attachment[] = [**

**0 => $path,**

**1 => $filename,**

**2 => $name,**

**3 => $encoding,**

**4 => $type,**

**5 => false, //isStringAttachment**

**6 => $disposition,**

**7 => $cid,**

**];**

**} catch (Exception $exc) {**

**$this->setError($exc->getMessage());**

**$this->edebug($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**return false;**

**}**

**return true;**

**}**

**/\*\***

**\* Add an embedded stringified attachment.**

**\* This can include images, sounds, and just about any other document type.**

**\* If your filename doesn't contain an extension, be sure to set the $type to an appropriate MIME type.**

**\***

**\* @param string $string The attachment binary data**

**\* @param string $cid Content ID of the attachment; Use this to reference**

**\* the content when using an embedded image in HTML**

**\* @param string $name A filename for the attachment. If this contains an extension,**

**\* PHPMailer will attempt to set a MIME type for the attachment.**

**\* For example 'file.jpg' would get an 'image/jpeg' MIME type.**

**\* @param string $encoding File encoding (see $Encoding), defaults to 'base64'**

**\* @param string $type MIME type - will be used in preference to any automatically derived type**

**\* @param string $disposition Disposition to use**

**\***

**\* @throws Exception**

**\***

**\* @return bool True on successfully adding an attachment**

**\*/**

**public function addStringEmbeddedImage(**

**$string,**

**$cid,**

**$name = '',**

**$encoding = self::ENCODING\_BASE64,**

**$type = '',**

**$disposition = 'inline'**

**) {**

**try {**

**//If a MIME type is not specified, try to work it out from the name**

**if ('' === $type && !empty($name)) {**

**$type = static::filenameToType($name);**

**}**

**if (!$this->validateEncoding($encoding)) {**

**throw new Exception($this->lang('encoding') . $encoding);**

**}**

**//Append to $attachment array**

**$this->attachment[] = [**

**0 => $string,**

**1 => $name,**

**2 => $name,**

**3 => $encoding,**

**4 => $type,**

**5 => true, //isStringAttachment**

**6 => $disposition,**

**7 => $cid,**

**];**

**} catch (Exception $exc) {**

**$this->setError($exc->getMessage());**

**$this->edebug($exc->getMessage());**

**if ($this->exceptions) {**

**throw $exc;**

**}**

**return false;**

**}**

**return true;**

**}**

**/\*\***

**\* Validate encodings.**

**\***

**\* @param string $encoding**

**\***

**\* @return bool**

**\*/**

**protected function validateEncoding($encoding)**

**{**

**return in\_array(**

**$encoding,**

**[**

**self::ENCODING\_7BIT,**

**self::ENCODING\_QUOTED\_PRINTABLE,**

**self::ENCODING\_BASE64,**

**self::ENCODING\_8BIT,**

**self::ENCODING\_BINARY,**

**],**

**true**

**);**

**}**

**/\*\***

**\* Check if an embedded attachment is present with this cid.**

**\***

**\* @param string $cid**

**\***

**\* @return bool**

**\*/**

**protected function cidExists($cid)**

**{**

**foreach ($this->attachment as $attachment) {**

**if ('inline' === $attachment[6] && $cid === $attachment[7]) {**

**return true;**

**}**

**}**

**return false;**

**}**

**/\*\***

**\* Check if an inline attachment is present.**

**\***

**\* @return bool**

**\*/**

**public function inlineImageExists()**

**{**

**foreach ($this->attachment as $attachment) {**

**if ('inline' === $attachment[6]) {**

**return true;**

**}**

**}**

**return false;**

**}**

**/\*\***

**\* Check if an attachment (non-inline) is present.**

**\***

**\* @return bool**

**\*/**

**public function attachmentExists()**

**{**

**foreach ($this->attachment as $attachment) {**

**if ('attachment' === $attachment[6]) {**

**return true;**

**}**

**}**

**return false;**

**}**

**/\*\***

**\* Check if this message has an alternative body set.**

**\***

**\* @return bool**

**\*/**

**public function alternativeExists()**

**{**

**return !empty($this->AltBody);**

**}**

**/\*\***

**\* Clear queued addresses of given kind.**

**\***

**\* @param string $kind 'to', 'cc', or 'bcc'**

**\*/**

**public function clearQueuedAddresses($kind)**

**{**

**$this->RecipientsQueue = array\_filter(**

**$this->RecipientsQueue,**

**static function ($params) use ($kind) {**

**return $params[0] !== $kind;**

**}**

**);**

**}**

**/\*\***

**\* Clear all To recipients.**

**\*/**

**public function clearAddresses()**

**{**

**foreach ($this->to as $to) {**

**unset($this->all\_recipients[strtolower($to[0])]);**

**}**

**$this->to = [];**

**$this->clearQueuedAddresses('to');**

**}**

**/\*\***

**\* Clear all CC recipients.**

**\*/**

**public function clearCCs()**

**{**

**foreach ($this->cc as $cc) {**

**unset($this->all\_recipients[strtolower($cc[0])]);**

**}**

**$this->cc = [];**

**$this->clearQueuedAddresses('cc');**

**}**

**/\*\***

**\* Clear all BCC recipients.**

**\*/**

**public function clearBCCs()**

**{**

**foreach ($this->bcc as $bcc) {**

**unset($this->all\_recipients[strtolower($bcc[0])]);**

**}**

**$this->bcc = [];**

**$this->clearQueuedAddresses('bcc');**

**}**

**/\*\***

**\* Clear all ReplyTo recipients.**

**\*/**

**public function clearReplyTos()**

**{**

**$this->ReplyTo = [];**

**$this->ReplyToQueue = [];**

**}**

**/\*\***

**\* Clear all recipient types.**

**\*/**

**public function clearAllRecipients()**

**{**

**$this->to = [];**

**$this->cc = [];**

**$this->bcc = [];**

**$this->all\_recipients = [];**

**$this->RecipientsQueue = [];**

**}**

**/\*\***

**\* Clear all filesystem, string, and binary attachments.**

**\*/**

**public function clearAttachments()**

**{**

**$this->attachment = [];**

**}**

**/\*\***

**\* Clear all custom headers.**

**\*/**

**public function clearCustomHeaders()**

**{**

**$this->CustomHeader = [];**

**}**

**/\*\***

**\* Add an error message to the error container.**

**\***

**\* @param string $msg**

**\*/**

**protected function setError($msg)**

**{**

**++$this->error\_count;**

**if ('smtp' === $this->Mailer && null !== $this->smtp) {**

**$lasterror = $this->smtp->getError();**

**if (!empty($lasterror['error'])) {**

**$msg .= $this->lang('smtp\_error') . $lasterror['error'];**

**if (!empty($lasterror['detail'])) {**

**$msg .= ' ' . $this->lang('smtp\_detail') . $lasterror['detail'];**

**}**

**if (!empty($lasterror['smtp\_code'])) {**

**$msg .= ' ' . $this->lang('smtp\_code') . $lasterror['smtp\_code'];**

**}**

**if (!empty($lasterror['smtp\_code\_ex'])) {**

**$msg .= ' ' . $this->lang('smtp\_code\_ex') . $lasterror['smtp\_code\_ex'];**

**}**

**}**

**}**

**$this->ErrorInfo = $msg;**

**}**

**/\*\***

**\* Return an RFC 822 formatted date.**

**\***

**\* @return string**

**\*/**

**public static function rfcDate()**

**{**

**//Set the time zone to whatever the default is to avoid 500 errors**

**//Will default to UTC if it's not set properly in php.ini**

**date\_default\_timezone\_set(@date\_default\_timezone\_get());**

**return date('D, j M Y H:i:s O');**

**}**

**/\*\***

**\* Get the server hostname.**

**\* Returns 'localhost.localdomain' if unknown.**

**\***

**\* @return string**

**\*/**

**protected function serverHostname()**

**{**

**$result = '';**

**if (!empty($this->Hostname)) {**

**$result = $this->Hostname;**

**} elseif (isset($\_SERVER) && array\_key\_exists('SERVER\_NAME', $\_SERVER)) {**

**$result = $\_SERVER['SERVER\_NAME'];**

**} elseif (function\_exists('gethostname') && gethostname() !== false) {**

**$result = gethostname();**

**} elseif (php\_uname('n') !== false) {**

**$result = php\_uname('n');**

**}**

**if (!static::isValidHost($result)) {**

**return 'localhost.localdomain';**

**}**

**return $result;**

**}**

**/\*\***

**\* Validate whether a string contains a valid value to use as a hostname or IP address.**

**\* IPv6 addresses must include [], e.g. `[::1]`, not just `::1`.**

**\***

**\* @param string $host The host name or IP address to check**

**\***

**\* @return bool**

**\*/**

**public static function isValidHost($host)**

**{**

**//Simple syntax limits**

**if (**

**empty($host)**

**|| !is\_string($host)**

**|| strlen($host) > 256**

**|| !preg\_match('/^([a-zA-Z\d.-]\*|\[[a-fA-F\d:]+\])$/', $host)**

**) {**

**return false;**

**}**

**//Looks like a bracketed IPv6 address**

**if (strlen($host) > 2 && substr($host, 0, 1) === '[' && substr($host, -1, 1) === ']') {**

**return filter\_var(substr($host, 1, -1), FILTER\_VALIDATE\_IP, FILTER\_FLAG\_IPV6) !== false;**

**}**

**//If removing all the dots results in a numeric string, it must be an IPv4 address.**

**//Need to check this first because otherwise things like `999.0.0.0` are considered valid host names**

**if (is\_numeric(str\_replace('.', '', $host))) {**

**//Is it a valid IPv4 address?**

**return filter\_var($host, FILTER\_VALIDATE\_IP, FILTER\_FLAG\_IPV4) !== false;**

**}**

**if (filter\_var('http://' . $host, FILTER\_VALIDATE\_URL) !== false) {**

**//Is it a syntactically valid hostname?**

**return true;**

**}**

**return false;**

**}**

**/\*\***

**\* Get an error message in the current language.**

**\***

**\* @param string $key**

**\***

**\* @return string**

**\*/**

**protected function lang($key)**

**{**

**if (count($this->language) < 1) {**

**$this->setLanguage(); //Set the default language**

**}**

**if (array\_key\_exists($key, $this->language)) {**

**if ('smtp\_connect\_failed' === $key) {**

**//Include a link to troubleshooting docs on SMTP connection failure.**

**//This is by far the biggest cause of support questions**

**//but it's usually not PHPMailer's fault.**

**return $this->language[$key] . ' https://github.com/PHPMailer/PHPMailer/wiki/Troubleshooting';**

**}**

**return $this->language[$key];**

**}**

**//Return the key as a fallback**

**return $key;**

**}**

**/\*\***

**\* Build an error message starting with a generic one and adding details if possible.**

**\***

**\* @param string $base\_key**

**\* @return string**

**\*/**

**private function getSmtpErrorMessage($base\_key)**

**{**

**$message = $this->lang($base\_key);**

**$error = $this->smtp->getError();**

**if (!empty($error['error'])) {**

**$message .= ' ' . $error['error'];**

**if (!empty($error['detail'])) {**

**$message .= ' ' . $error['detail'];**

**}**

**}**

**return $message;**

**}**

**/\*\***

**\* Check if an error occurred.**

**\***

**\* @return bool True if an error did occur**

**\*/**

**public function isError()**

**{**

**return $this->error\_count > 0;**

**}**

**/\*\***

**\* Add a custom header.**

**\* $name value can be overloaded to contain**

**\* both header name and value (name:value).**

**\***

**\* @param string $name Custom header name**

**\* @param string|null $value Header value**

**\***

**\* @throws Exception**

**\*/**

**public function addCustomHeader($name, $value = null)**

**{**

**if (null === $value && strpos($name, ':') !== false) {**

**//Value passed in as name:value**

**list($name, $value) = explode(':', $name, 2);**

**}**

**$name = trim($name);**

**$value = (null === $value) ? '' : trim($value);**

**//Ensure name is not empty, and that neither name nor value contain line breaks**

**if (empty($name) || strpbrk($name . $value, "\r\n") !== false) {**

**if ($this->exceptions) {**

**throw new Exception($this->lang('invalid\_header'));**

**}**

**return false;**

**}**

**$this->CustomHeader[] = [$name, $value];**

**return true;**

**}**

**/\*\***

**\* Returns all custom headers.**

**\***

**\* @return array**

**\*/**

**public function getCustomHeaders()**

**{**

**return $this->CustomHeader;**

**}**

**/\*\***

**\* Create a message body from an HTML string.**

**\* Automatically inlines images and creates a plain-text version by converting the HTML,**

**\* overwriting any existing values in Body and AltBody.**

**\* Do not source $message content from user input!**

**\* $basedir is prepended when handling relative URLs, e.g. <img src="/images/a.png"> and must not be empty**

**\* will look for an image file in $basedir/images/a.png and convert it to inline.**

**\* If you don't provide a $basedir, relative paths will be left untouched (and thus probably break in email)**

**\* Converts data-uri images into embedded attachments.**

**\* If you don't want to apply these transformations to your HTML, just set Body and AltBody directly.**

**\***

**\* @param string $message HTML message string**

**\* @param string $basedir Absolute path to a base directory to prepend to relative paths to images**

**\* @param bool|callable $advanced Whether to use the internal HTML to text converter**

**\* or your own custom converter**

**\* @return string The transformed message body**

**\***

**\* @throws Exception**

**\***

**\* @see PHPMailer::html2text()**

**\*/**

**public function msgHTML($message, $basedir = '', $advanced = false)**

**{**

**preg\_match\_all('/(?<!-)(src|background)=["\'](.\*)["\']/Ui', $message, $images);**

**if (array\_key\_exists(2, $images)) {**

**if (strlen($basedir) > 1 && '/' !== substr($basedir, -1)) {**

**//Ensure $basedir has a trailing /**

**$basedir .= '/';**

**}**

**foreach ($images[2] as $imgindex => $url) {**

**//Convert data URIs into embedded images**

**//e.g. "data:image/gif;base64,R0lGODlhAQABAAAAACH5BAEKAAEALAAAAAABAAEAAAICTAEAOw=="**

**$match = [];**

**if (preg\_match('#^data:(image/(?:jpe?g|gif|png));?(base64)?,(.+)#', $url, $match)) {**

**if (count($match) === 4 && static::ENCODING\_BASE64 === $match[2]) {**

**$data = base64\_decode($match[3]);**

**} elseif ('' === $match[2]) {**

**$data = rawurldecode($match[3]);**

**} else {**

**//Not recognised so leave it alone**

**continue;**

**}**

**//Hash the decoded data, not the URL, so that the same data-URI image used in multiple places**

**//will only be embedded once, even if it used a different encoding**

**$cid = substr(hash('sha256', $data), 0, 32) . '@phpmailer.0'; //RFC2392 S 2**

**if (!$this->cidExists($cid)) {**

**$this->addStringEmbeddedImage(**

**$data,**

**$cid,**

**'embed' . $imgindex,**

**static::ENCODING\_BASE64,**

**$match[1]**

**);**

**}**

**$message = str\_replace(**

**$images[0][$imgindex],**

**$images[1][$imgindex] . '="cid:' . $cid . '"',**

**$message**

**);**

**continue;**

**}**

**if (**

**//Only process relative URLs if a basedir is provided (i.e. no absolute local paths)**

**!empty($basedir)**

**//Ignore URLs containing parent dir traversal (..)**

**&& (strpos($url, '..') === false)**

**//Do not change urls that are already inline images**

**&& 0 !== strpos($url, 'cid:')**

**//Do not change absolute URLs, including anonymous protocol**

**&& !preg\_match('#^[a-z][a-z0-9+.-]\*:?//#i', $url)**

**) {**

**$filename = static::mb\_pathinfo($url, PATHINFO\_BASENAME);**

**$directory = dirname($url);**

**if ('.' === $directory) {**

**$directory = '';**

**}**

**//RFC2392 S 2**

**$cid = substr(hash('sha256', $url), 0, 32) . '@phpmailer.0';**

**if (strlen($basedir) > 1 && '/' !== substr($basedir, -1)) {**

**$basedir .= '/';**

**}**

**if (strlen($directory) > 1 && '/' !== substr($directory, -1)) {**

**$directory .= '/';**

**}**

**if (**

**$this->addEmbeddedImage(**

**$basedir . $directory . $filename,**

**$cid,**

**$filename,**

**static::ENCODING\_BASE64,**

**static::\_mime\_types((string) static::mb\_pathinfo($filename, PATHINFO\_EXTENSION))**

**)**

**) {**

**$message = preg\_replace(**

**'/' . $images[1][$imgindex] . '=["\']' . preg\_quote($url, '/') . '["\']/Ui',**

**$images[1][$imgindex] . '="cid:' . $cid . '"',**

**$message**

**);**

**}**

**}**

**}**

**}**

**$this->isHTML();**

**//Convert all message body line breaks to LE, makes quoted-printable encoding work much better**

**$this->Body = static::normalizeBreaks($message);**

**$this->AltBody = static::normalizeBreaks($this->html2text($message, $advanced));**

**if (!$this->alternativeExists()) {**

**$this->AltBody = 'This is an HTML-only message. To view it, activate HTML in your email application.'**

**. static::$LE;**

**}**

**return $this->Body;**

**}**

**/\*\***

**\* Convert an HTML string into plain text.**

**\* This is used by msgHTML().**

**\* Note - older versions of this function used a bundled advanced converter**

**\* which was removed for license reasons in #232.**

**\* Example usage:**

**\***

**\* ```php**

**\* //Use default conversion**

**\* $plain = $mail->html2text($html);**

**\* //Use your own custom converter**

**\* $plain = $mail->html2text($html, function($html) {**

**\* $converter = new MyHtml2text($html);**

**\* return $converter->get\_text();**

**\* });**

**\* ```**

**\***

**\* @param string $html The HTML text to convert**

**\* @param bool|callable $advanced Any boolean value to use the internal converter,**

**\* or provide your own callable for custom conversion.**

**\* \*Never\* pass user-supplied data into this parameter**

**\***

**\* @return string**

**\*/**

**public function html2text($html, $advanced = false)**

**{**

**if (is\_callable($advanced)) {**

**return call\_user\_func($advanced, $html);**

**}**

**return html\_entity\_decode(**

**trim(strip\_tags(preg\_replace('/<(head|title|style|script)[^>]\*>.\*?<\/\\1>/si', '', $html))),**

**ENT\_QUOTES,**

**$this->CharSet**

**);**

**}**

**/\*\***

**\* Get the MIME type for a file extension.**

**\***

**\* @param string $ext File extension**

**\***

**\* @return string MIME type of file**

**\*/**

**public static function \_mime\_types($ext = '')**

**{**

**$mimes = [**

**'xl' => 'application/excel',**

**'js' => 'application/javascript',**

**'hqx' => 'application/mac-binhex40',**

**'cpt' => 'application/mac-compactpro',**

**'bin' => 'application/macbinary',**

**'doc' => 'application/msword',**

**'word' => 'application/msword',**

**'xlsx' => 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',**

**'xltx' => 'application/vnd.openxmlformats-officedocument.spreadsheetml.template',**

**'potx' => 'application/vnd.openxmlformats-officedocument.presentationml.template',**

**'ppsx' => 'application/vnd.openxmlformats-officedocument.presentationml.slideshow',**

**'pptx' => 'application/vnd.openxmlformats-officedocument.presentationml.presentation',**

**'sldx' => 'application/vnd.openxmlformats-officedocument.presentationml.slide',**

**'docx' => 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',**

**'dotx' => 'application/vnd.openxmlformats-officedocument.wordprocessingml.template',**

**'xlam' => 'application/vnd.ms-excel.addin.macroEnabled.12',**

**'xlsb' => 'application/vnd.ms-excel.sheet.binary.macroEnabled.12',**

**'class' => 'application/octet-stream',**

**'dll' => 'application/octet-stream',**

**'dms' => 'application/octet-stream',**

**'exe' => 'application/octet-stream',**

**'lha' => 'application/octet-stream',**

**'lzh' => 'application/octet-stream',**

**'psd' => 'application/octet-stream',**

**'sea' => 'application/octet-stream',**

**'so' => 'application/octet-stream',**

**'oda' => 'application/oda',**

**'pdf' => 'application/pdf',**

**'ai' => 'application/postscript',**

**'eps' => 'application/postscript',**

**'ps' => 'application/postscript',**

**'smi' => 'application/smil',**

**'smil' => 'application/smil',**

**'mif' => 'application/vnd.mif',**

**'xls' => 'application/vnd.ms-excel',**

**'ppt' => 'application/vnd.ms-powerpoint',**

**'wbxml' => 'application/vnd.wap.wbxml',**

**'wmlc' => 'application/vnd.wap.wmlc',**

**'dcr' => 'application/x-director',**

**'dir' => 'application/x-director',**

**'dxr' => 'application/x-director',**

**'dvi' => 'application/x-dvi',**

**'gtar' => 'application/x-gtar',**

**'php3' => 'application/x-httpd-php',**

**'php4' => 'application/x-httpd-php',**

**'php' => 'application/x-httpd-php',**

**'phtml' => 'application/x-httpd-php',**

**'phps' => 'application/x-httpd-php-source',**

**'swf' => 'application/x-shockwave-flash',**

**'sit' => 'application/x-stuffit',**

**'tar' => 'application/x-tar',**

**'tgz' => 'application/x-tar',**

**'xht' => 'application/xhtml+xml',**

**'xhtml' => 'application/xhtml+xml',**

**'zip' => 'application/zip',**

**'mid' => 'audio/midi',**

**'midi' => 'audio/midi',**

**'mp2' => 'audio/mpeg',**

**'mp3' => 'audio/mpeg',**

**'m4a' => 'audio/mp4',**

**'mpga' => 'audio/mpeg',**

**'aif' => 'audio/x-aiff',**

**'aifc' => 'audio/x-aiff',**

**'aiff' => 'audio/x-aiff',**

**'ram' => 'audio/x-pn-realaudio',**

**'rm' => 'audio/x-pn-realaudio',**

**'rpm' => 'audio/x-pn-realaudio-plugin',**

**'ra' => 'audio/x-realaudio',**

**'wav' => 'audio/x-wav',**

**'mka' => 'audio/x-matroska',**

**'bmp' => 'image/bmp',**

**'gif' => 'image/gif',**

**'jpeg' => 'image/jpeg',**

**'jpe' => 'image/jpeg',**

**'jpg' => 'image/jpeg',**

**'png' => 'image/png',**

**'tiff' => 'image/tiff',**

**'tif' => 'image/tiff',**

**'webp' => 'image/webp',**

**'avif' => 'image/avif',**

**'heif' => 'image/heif',**

**'heifs' => 'image/heif-sequence',**

**'heic' => 'image/heic',**

**'heics' => 'image/heic-sequence',**

**'eml' => 'message/rfc822',**

**'css' => 'text/css',**

**'html' => 'text/html',**

**'htm' => 'text/html',**

**'shtml' => 'text/html',**

**'log' => 'text/plain',**

**'text' => 'text/plain',**

**'txt' => 'text/plain',**

**'rtx' => 'text/richtext',**

**'rtf' => 'text/rtf',**

**'vcf' => 'text/vcard',**

**'vcard' => 'text/vcard',**

**'ics' => 'text/calendar',**

**'xml' => 'text/xml',**

**'xsl' => 'text/xml',**

**'wmv' => 'video/x-ms-wmv',**

**'mpeg' => 'video/mpeg',**

**'mpe' => 'video/mpeg',**

**'mpg' => 'video/mpeg',**

**'mp4' => 'video/mp4',**

**'m4v' => 'video/mp4',**

**'mov' => 'video/quicktime',**

**'qt' => 'video/quicktime',**

**'rv' => 'video/vnd.rn-realvideo',**

**'avi' => 'video/x-msvideo',**

**'movie' => 'video/x-sgi-movie',**

**'webm' => 'video/webm',**

**'mkv' => 'video/x-matroska',**

**];**

**$ext = strtolower($ext);**

**if (array\_key\_exists($ext, $mimes)) {**

**return $mimes[$ext];**

**}**

**return 'application/octet-stream';**

**}**

**/\*\***

**\* Map a file name to a MIME type.**

**\* Defaults to 'application/octet-stream', i.e.. arbitrary binary data.**

**\***

**\* @param string $filename A file name or full path, does not need to exist as a file**

**\***

**\* @return string**

**\*/**

**public static function filenameToType($filename)**

**{**

**//In case the path is a URL, strip any query string before getting extension**

**$qpos = strpos($filename, '?');**

**if (false !== $qpos) {**

**$filename = substr($filename, 0, $qpos);**

**}**

**$ext = static::mb\_pathinfo($filename, PATHINFO\_EXTENSION);**

**return static::\_mime\_types($ext);**

**}**

**/\*\***

**\* Multi-byte-safe pathinfo replacement.**

**\* Drop-in replacement for pathinfo(), but multibyte- and cross-platform-safe.**

**\***

**\* @see http://www.php.net/manual/en/function.pathinfo.php#107461**

**\***

**\* @param string $path A filename or path, does not need to exist as a file**

**\* @param int|string $options Either a PATHINFO\_\* constant,**

**\* or a string name to return only the specified piece**

**\***

**\* @return string|array**

**\*/**

**public static function mb\_pathinfo($path, $options = null)**

**{**

**$ret = ['dirname' => '', 'basename' => '', 'extension' => '', 'filename' => ''];**

**$pathinfo = [];**

**if (preg\_match('#^(.\*?)[\\\\/]\*(([^/\\\\]\*?)(\.([^.\\\\/]+?)|))[\\\\/.]\*$#m', $path, $pathinfo)) {**

**if (array\_key\_exists(1, $pathinfo)) {**

**$ret['dirname'] = $pathinfo[1];**

**}**

**if (array\_key\_exists(2, $pathinfo)) {**

**$ret['basename'] = $pathinfo[2];**

**}**

**if (array\_key\_exists(5, $pathinfo)) {**

**$ret['extension'] = $pathinfo[5];**

**}**

**if (array\_key\_exists(3, $pathinfo)) {**

**$ret['filename'] = $pathinfo[3];**

**}**

**}**

**switch ($options) {**

**case PATHINFO\_DIRNAME:**

**case 'dirname':**

**return $ret['dirname'];**

**case PATHINFO\_BASENAME:**

**case 'basename':**

**return $ret['basename'];**

**case PATHINFO\_EXTENSION:**

**case 'extension':**

**return $ret['extension'];**

**case PATHINFO\_FILENAME:**

**case 'filename':**

**return $ret['filename'];**

**default:**

**return $ret;**

**}**

**}**

**/\*\***

**\* Set or reset instance properties.**

**\* You should avoid this function - it's more verbose, less efficient, more error-prone and**

**\* harder to debug than setting properties directly.**

**\* Usage Example:**

**\* `$mail->set('SMTPSecure', static::ENCRYPTION\_STARTTLS);`**

**\* is the same as:**

**\* `$mail->SMTPSecure = static::ENCRYPTION\_STARTTLS;`.**

**\***

**\* @param string $name The property name to set**

**\* @param mixed $value The value to set the property to**

**\***

**\* @return bool**

**\*/**

**public function set($name, $value = '')**

**{**

**if (property\_exists($this, $name)) {**

**$this->$name = $value;**

**return true;**

**}**

**$this->setError($this->lang('variable\_set') . $name);**

**return false;**

**}**

**/\*\***

**\* Strip newlines to prevent header injection.**

**\***

**\* @param string $str**

**\***

**\* @return string**

**\*/**

**public function secureHeader($str)**

**{**

**return trim(str\_replace(["\r", "\n"], '', $str));**

**}**

**/\*\***

**\* Normalize line breaks in a string.**

**\* Converts UNIX LF, Mac CR and Windows CRLF line breaks into a single line break format.**

**\* Defaults to CRLF (for message bodies) and preserves consecutive breaks.**

**\***

**\* @param string $text**

**\* @param string $breaktype What kind of line break to use; defaults to static::$LE**

**\***

**\* @return string**

**\*/**

**public static function normalizeBreaks($text, $breaktype = null)**

**{**

**if (null === $breaktype) {**

**$breaktype = static::$LE;**

**}**

**//Normalise to \n**

**$text = str\_replace([self::CRLF, "\r"], "\n", $text);**

**//Now convert LE as needed**

**if ("\n" !== $breaktype) {**

**$text = str\_replace("\n", $breaktype, $text);**

**}**

**return $text;**

**}**

**/\*\***

**\* Remove trailing breaks from a string.**

**\***

**\* @param string $text**

**\***

**\* @return string The text to remove breaks from**

**\*/**

**public static function stripTrailingWSP($text)**

**{**

**return rtrim($text, " \r\n\t");**

**}**

**/\*\***

**\* Return the current line break format string.**

**\***

**\* @return string**

**\*/**

**public static function getLE()**

**{**

**return static::$LE;**

**}**

**/\*\***

**\* Set the line break format string, e.g. "\r\n".**

**\***

**\* @param string $le**

**\*/**

**protected static function setLE($le)**

**{**

**static::$LE = $le;**

**}**

**/\*\***

**\* Set the public and private key files and password for S/MIME signing.**

**\***

**\* @param string $cert\_filename**

**\* @param string $key\_filename**

**\* @param string $key\_pass Password for private key**

**\* @param string $extracerts\_filename Optional path to chain certificate**

**\*/**

**public function sign($cert\_filename, $key\_filename, $key\_pass, $extracerts\_filename = '')**

**{**

**$this->sign\_cert\_file = $cert\_filename;**

**$this->sign\_key\_file = $key\_filename;**

**$this->sign\_key\_pass = $key\_pass;**

**$this->sign\_extracerts\_file = $extracerts\_filename;**

**}**

**/\*\***

**\* Quoted-Printable-encode a DKIM header.**

**\***

**\* @param string $txt**

**\***

**\* @return string**

**\*/**

**public function DKIM\_QP($txt)**

**{**

**$line = '';**

**$len = strlen($txt);**

**for ($i = 0; $i < $len; ++$i) {**

**$ord = ord($txt[$i]);**

**if (((0x21 <= $ord) && ($ord <= 0x3A)) || $ord === 0x3C || ((0x3E <= $ord) && ($ord <= 0x7E))) {**

**$line .= $txt[$i];**

**} else {**

**$line .= '=' . sprintf('%02X', $ord);**

**}**

**}**

**return $line;**

**}**

**/\*\***

**\* Generate a DKIM signature.**

**\***

**\* @param string $signHeader**

**\***

**\* @throws Exception**

**\***

**\* @return string The DKIM signature value**

**\*/**

**public function DKIM\_Sign($signHeader)**

**{**

**if (!defined('PKCS7\_TEXT')) {**

**if ($this->exceptions) {**

**throw new Exception($this->lang('extension\_missing') . 'openssl');**

**}**

**return '';**

**}**

**$privKeyStr = !empty($this->DKIM\_private\_string) ?**

**$this->DKIM\_private\_string :**

**file\_get\_contents($this->DKIM\_private);**

**if ('' !== $this->DKIM\_passphrase) {**

**$privKey = openssl\_pkey\_get\_private($privKeyStr, $this->DKIM\_passphrase);**

**} else {**

**$privKey = openssl\_pkey\_get\_private($privKeyStr);**

**}**

**if (openssl\_sign($signHeader, $signature, $privKey, 'sha256WithRSAEncryption')) {**

**if (\PHP\_MAJOR\_VERSION < 8) {**

**openssl\_pkey\_free($privKey);**

**}**

**return base64\_encode($signature);**

**}**

**if (\PHP\_MAJOR\_VERSION < 8) {**

**openssl\_pkey\_free($privKey);**

**}**

**return '';**

**}**

**/\*\***

**\* Generate a DKIM canonicalization header.**

**\* Uses the 'relaxed' algorithm from RFC6376 section 3.4.2.**

**\* Canonicalized headers should \*always\* use CRLF, regardless of mailer setting.**

**\***

**\* @see https://tools.ietf.org/html/rfc6376#section-3.4.2**

**\***

**\* @param string $signHeader Header**

**\***

**\* @return string**

**\*/**

**public function DKIM\_HeaderC($signHeader)**

**{**

**//Normalize breaks to CRLF (regardless of the mailer)**

**$signHeader = static::normalizeBreaks($signHeader, self::CRLF);**

**//Unfold header lines**

**//Note PCRE \s is too broad a definition of whitespace; RFC5322 defines it as `[ \t]`**

**//@see https://tools.ietf.org/html/rfc5322#section-2.2**

**//That means this may break if you do something daft like put vertical tabs in your headers.**

**$signHeader = preg\_replace('/\r\n[ \t]+/', ' ', $signHeader);**

**//Break headers out into an array**

**$lines = explode(self::CRLF, $signHeader);**

**foreach ($lines as $key => $line) {**

**//If the header is missing a :, skip it as it's invalid**

**//This is likely to happen because the explode() above will also split**

**//on the trailing LE, leaving an empty line**

**if (strpos($line, ':') === false) {**

**continue;**

**}**

**list($heading, $value) = explode(':', $line, 2);**

**//Lower-case header name**

**$heading = strtolower($heading);**

**//Collapse white space within the value, also convert WSP to space**

**$value = preg\_replace('/[ \t]+/', ' ', $value);**

**//RFC6376 is slightly unclear here - it says to delete space at the \*end\* of each value**

**//But then says to delete space before and after the colon.**

**//Net result is the same as trimming both ends of the value.**

**//By elimination, the same applies to the field name**

**$lines[$key] = trim($heading, " \t") . ':' . trim($value, " \t");**

**}**

**return implode(self::CRLF, $lines);**

**}**

**/\*\***

**\* Generate a DKIM canonicalization body.**

**\* Uses the 'simple' algorithm from RFC6376 section 3.4.3.**

**\* Canonicalized bodies should \*always\* use CRLF, regardless of mailer setting.**

**\***

**\* @see https://tools.ietf.org/html/rfc6376#section-3.4.3**

**\***

**\* @param string $body Message Body**

**\***

**\* @return string**

**\*/**

**public function DKIM\_BodyC($body)**

**{**

**if (empty($body)) {**

**return self::CRLF;**

**}**

**//Normalize line endings to CRLF**

**$body = static::normalizeBreaks($body, self::CRLF);**

**//Reduce multiple trailing line breaks to a single one**

**return static::stripTrailingWSP($body) . self::CRLF;**

**}**

**/\*\***

**\* Create the DKIM header and body in a new message header.**

**\***

**\* @param string $headers\_line Header lines**

**\* @param string $subject Subject**

**\* @param string $body Body**

**\***

**\* @throws Exception**

**\***

**\* @return string**

**\*/**

**public function DKIM\_Add($headers\_line, $subject, $body)**

**{**

**$DKIMsignatureType = 'rsa-sha256'; //Signature & hash algorithms**

**$DKIMcanonicalization = 'relaxed/simple'; //Canonicalization methods of header & body**

**$DKIMquery = 'dns/txt'; //Query method**

**$DKIMtime = time();**

**//Always sign these headers without being asked**

**//Recommended list from https://tools.ietf.org/html/rfc6376#section-5.4.1**

**$autoSignHeaders = [**

**'from',**

**'to',**

**'cc',**

**'date',**

**'subject',**

**'reply-to',**

**'message-id',**

**'content-type',**

**'mime-version',**

**'x-mailer',**

**];**

**if (stripos($headers\_line, 'Subject') === false) {**

**$headers\_line .= 'Subject: ' . $subject . static::$LE;**

**}**

**$headerLines = explode(static::$LE, $headers\_line);**

**$currentHeaderLabel = '';**

**$currentHeaderValue = '';**

**$parsedHeaders = [];**

**$headerLineIndex = 0;**

**$headerLineCount = count($headerLines);**

**foreach ($headerLines as $headerLine) {**

**$matches = [];**

**if (preg\_match('/^([^ \t]\*?)(?::[ \t]\*)(.\*)$/', $headerLine, $matches)) {**

**if ($currentHeaderLabel !== '') {**

**//We were previously in another header; This is the start of a new header, so save the previous one**

**$parsedHeaders[] = ['label' => $currentHeaderLabel, 'value' => $currentHeaderValue];**

**}**

**$currentHeaderLabel = $matches[1];**

**$currentHeaderValue = $matches[2];**

**} elseif (preg\_match('/^[ \t]+(.\*)$/', $headerLine, $matches)) {**

**//This is a folded continuation of the current header, so unfold it**

**$currentHeaderValue .= ' ' . $matches[1];**

**}**

**++$headerLineIndex;**

**if ($headerLineIndex >= $headerLineCount) {**

**//This was the last line, so finish off this header**

**$parsedHeaders[] = ['label' => $currentHeaderLabel, 'value' => $currentHeaderValue];**

**}**

**}**

**$copiedHeaders = [];**

**$headersToSignKeys = [];**

**$headersToSign = [];**

**foreach ($parsedHeaders as $header) {**

**//Is this header one that must be included in the DKIM signature?**

**if (in\_array(strtolower($header['label']), $autoSignHeaders, true)) {**

**$headersToSignKeys[] = $header['label'];**

**$headersToSign[] = $header['label'] . ': ' . $header['value'];**

**if ($this->DKIM\_copyHeaderFields) {**

**$copiedHeaders[] = $header['label'] . ':' . //Note no space after this, as per RFC**

**str\_replace('|', '=7C', $this->DKIM\_QP($header['value']));**

**}**

**continue;**

**}**

**//Is this an extra custom header we've been asked to sign?**

**if (in\_array($header['label'], $this->DKIM\_extraHeaders, true)) {**

**//Find its value in custom headers**

**foreach ($this->CustomHeader as $customHeader) {**

**if ($customHeader[0] === $header['label']) {**

**$headersToSignKeys[] = $header['label'];**

**$headersToSign[] = $header['label'] . ': ' . $header['value'];**

**if ($this->DKIM\_copyHeaderFields) {**

**$copiedHeaders[] = $header['label'] . ':' . //Note no space after this, as per RFC**

**str\_replace('|', '=7C', $this->DKIM\_QP($header['value']));**

**}**

**//Skip straight to the next header**

**continue 2;**

**}**

**}**

**}**

**}**

**$copiedHeaderFields = '';**

**if ($this->DKIM\_copyHeaderFields && count($copiedHeaders) > 0) {**

**//Assemble a DKIM 'z' tag**

**$copiedHeaderFields = ' z=';**

**$first = true;**

**foreach ($copiedHeaders as $copiedHeader) {**

**if (!$first) {**

**$copiedHeaderFields .= static::$LE . ' |';**

**}**

**//Fold long values**

**if (strlen($copiedHeader) > self::STD\_LINE\_LENGTH - 3) {**

**$copiedHeaderFields .= substr(**

**chunk\_split($copiedHeader, self::STD\_LINE\_LENGTH - 3, static::$LE . self::FWS),**

**0,**

**-strlen(static::$LE . self::FWS)**

**);**

**} else {**

**$copiedHeaderFields .= $copiedHeader;**

**}**

**$first = false;**

**}**

**$copiedHeaderFields .= ';' . static::$LE;**

**}**

**$headerKeys = ' h=' . implode(':', $headersToSignKeys) . ';' . static::$LE;**

**$headerValues = implode(static::$LE, $headersToSign);**

**$body = $this->DKIM\_BodyC($body);**

**//Base64 of packed binary SHA-256 hash of body**

**$DKIMb64 = base64\_encode(pack('H\*', hash('sha256', $body)));**

**$ident = '';**

**if ('' !== $this->DKIM\_identity) {**

**$ident = ' i=' . $this->DKIM\_identity . ';' . static::$LE;**

**}**

**//The DKIM-Signature header is included in the signature \*except for\* the value of the `b` tag**

**//which is appended after calculating the signature**

**//https://tools.ietf.org/html/rfc6376#section-3.5**

**$dkimSignatureHeader = 'DKIM-Signature: v=1;' .**

**' d=' . $this->DKIM\_domain . ';' .**

**' s=' . $this->DKIM\_selector . ';' . static::$LE .**

**' a=' . $DKIMsignatureType . ';' .**

**' q=' . $DKIMquery . ';' .**

**' t=' . $DKIMtime . ';' .**

**' c=' . $DKIMcanonicalization . ';' . static::$LE .**

**$headerKeys .**

**$ident .**

**$copiedHeaderFields .**

**' bh=' . $DKIMb64 . ';' . static::$LE .**

**' b=';**

**//Canonicalize the set of headers**

**$canonicalizedHeaders = $this->DKIM\_HeaderC(**

**$headerValues . static::$LE . $dkimSignatureHeader**

**);**

**$signature = $this->DKIM\_Sign($canonicalizedHeaders);**

**$signature = trim(chunk\_split($signature, self::STD\_LINE\_LENGTH - 3, static::$LE . self::FWS));**

**return static::normalizeBreaks($dkimSignatureHeader . $signature);**

**}**

**/\*\***

**\* Detect if a string contains a line longer than the maximum line length**

**\* allowed by RFC 2822 section 2.1.1.**

**\***

**\* @param string $str**

**\***

**\* @return bool**

**\*/**

**public static function hasLineLongerThanMax($str)**

**{**

**return (bool) preg\_match('/^(.{' . (self::MAX\_LINE\_LENGTH + strlen(static::$LE)) . ',})/m', $str);**

**}**

**/\*\***

**\* If a string contains any "special" characters, double-quote the name,**

**\* and escape any double quotes with a backslash.**

**\***

**\* @param string $str**

**\***

**\* @return string**

**\***

**\* @see RFC822 3.4.1**

**\*/**

**public static function quotedString($str)**

**{**

**if (preg\_match('/[ ()<>@,;:"\/\[\]?=]/', $str)) {**

**//If the string contains any of these chars, it must be double-quoted**

**//and any double quotes must be escaped with a backslash**

**return '"' . str\_replace('"', '\\"', $str) . '"';**

**}**

**//Return the string untouched, it doesn't need quoting**

**return $str;**

**}**

**/\*\***

**\* Allows for public read access to 'to' property.**

**\* Before the send() call, queued addresses (i.e. with IDN) are not yet included.**

**\***

**\* @return array**

**\*/**

**public function getToAddresses()**

**{**

**return $this->to;**

**}**

**/\*\***

**\* Allows for public read access to 'cc' property.**

**\* Before the send() call, queued addresses (i.e. with IDN) are not yet included.**

**\***

**\* @return array**

**\*/**

**public function getCcAddresses()**

**{**

**return $this->cc;**

**}**

**/\*\***

**\* Allows for public read access to 'bcc' property.**

**\* Before the send() call, queued addresses (i.e. with IDN) are not yet included.**

**\***

**\* @return array**

**\*/**

**public function getBccAddresses()**

**{**

**return $this->bcc;**

**}**

**/\*\***

**\* Allows for public read access to 'ReplyTo' property.**

**\* Before the send() call, queued addresses (i.e. with IDN) are not yet included.**

**\***

**\* @return array**

**\*/**

**public function getReplyToAddresses()**

**{**

**return $this->ReplyTo;**

**}**

**/\*\***

**\* Allows for public read access to 'all\_recipients' property.**

**\* Before the send() call, queued addresses (i.e. with IDN) are not yet included.**

**\***

**\* @return array**

**\*/**

**public function getAllRecipientAddresses()**

**{**

**return $this->all\_recipients;**

**}**

**/\*\***

**\* Perform a callback.**

**\***

**\* @param bool $isSent**

**\* @param array $to**

**\* @param array $cc**

**\* @param array $bcc**

**\* @param string $subject**

**\* @param string $body**

**\* @param string $from**

**\* @param array $extra**

**\*/**

**protected function doCallback($isSent, $to, $cc, $bcc, $subject, $body, $from, $extra)**

**{**

**if (!empty($this->action\_function) && is\_callable($this->action\_function)) {**

**call\_user\_func($this->action\_function, $isSent, $to, $cc, $bcc, $subject, $body, $from, $extra);**

**}**

**}**

**/\*\***

**\* Get the OAuthTokenProvider instance.**

**\***

**\* @return OAuthTokenProvider**

**\*/**

**public function getOAuth()**

**{**

**return $this->oauth;**

**}**

**/\*\***

**\* Set an OAuthTokenProvider instance.**

**\*/**

**public function setOAuth(OAuthTokenProvider $oauth)**

**{**

**$this->oauth = $oauth;**

**}**

**}**

**POP3**

<?php

/\*\*

 \* PHPMailer POP-Before-SMTP Authentication Class.

 \* PHP Version 5.5.

 \*

 \* @see https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub project

 \*

 \* @author    Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \* @author    Jim Jagielski (jimjag) <jimjag@gmail.com>

 \* @author    Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>

 \* @author    Brent R. Matzelle (original founder)

 \* @copyright 2012 - 2020 Marcus Bointon

 \* @copyright 2010 - 2012 Jim Jagielski

 \* @copyright 2004 - 2009 Andy Prevost

 \* @license   http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License

 \* @note      This program is distributed in the hope that it will be useful - WITHOUT

 \* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

 \* FITNESS FOR A PARTICULAR PURPOSE.

 \*/

namespace PHPMailer\PHPMailer;

/\*\*

 \* PHPMailer POP-Before-SMTP Authentication Class.

 \* Specifically for PHPMailer to use for RFC1939 POP-before-SMTP authentication.

 \* 1) This class does not support APOP authentication.

 \* 2) Opening and closing lots of POP3 connections can be quite slow. If you need

 \*   to send a batch of emails then just perform the authentication once at the start,

 \*   and then loop through your mail sending script. Providing this process doesn't

 \*   take longer than the verification period lasts on your POP3 server, you should be fine.

 \* 3) This is really ancient technology; you should only need to use it to talk to very old systems.

 \* 4) This POP3 class is deliberately lightweight and incomplete, implementing just

 \*   enough to do authentication.

 \*   If you want a more complete class there are other POP3 classes for PHP available.

 \*

 \* @author Richard Davey (original author) <rich@corephp.co.uk>

 \* @author Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \* @author Jim Jagielski (jimjag) <jimjag@gmail.com>

 \* @author Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>

 \*/

class POP3

{

    /\*\*

     \* The POP3 PHPMailer Version number.

     \*

     \* @var string

     \*/

    const VERSION = '6.6.0';

    /\*\*

     \* Default POP3 port number.

     \*

     \* @var int

     \*/

    const DEFAULT\_PORT = 110;

    /\*\*

     \* Default timeout in seconds.

     \*

     \* @var int

     \*/

    const DEFAULT\_TIMEOUT = 30;

    /\*\*

     \* POP3 class debug output mode.

     \* Debug output level.

     \* Options:

     \* @see POP3::DEBUG\_OFF: No output

     \* @see POP3::DEBUG\_SERVER: Server messages, connection/server errors

     \* @see POP3::DEBUG\_CLIENT: Client and Server messages, connection/server errors

     \*

     \* @var int

     \*/

    public $do\_debug = self::DEBUG\_OFF;

    /\*\*

     \* POP3 mail server hostname.

     \*

     \* @var string

     \*/

    public $host;

    /\*\*

     \* POP3 port number.

     \*

     \* @var int

     \*/

    public $port;

    /\*\*

     \* POP3 Timeout Value in seconds.

     \*

     \* @var int

     \*/

    public $tval;

    /\*\*

     \* POP3 username.

     \*

     \* @var string

     \*/

    public $username;

    /\*\*

     \* POP3 password.

     \*

     \* @var string

     \*/

    public $password;

    /\*\*

     \* Resource handle for the POP3 connection socket.

     \*

     \* @var resource

     \*/

    protected $pop\_conn;

    /\*\*

     \* Are we connected?

     \*

     \* @var bool

     \*/

    protected $connected = false;

    /\*\*

     \* Error container.

     \*

     \* @var array

     \*/

    protected $errors = [];

    /\*\*

     \* Line break constant.

     \*/

    const LE = "\r\n";

    /\*\*

     \* Debug level for no output.

     \*

     \* @var int

     \*/

    const DEBUG\_OFF = 0;

    /\*\*

     \* Debug level to show server -> client messages

     \* also shows clients connection errors or errors from server

     \*

     \* @var int

     \*/

    const DEBUG\_SERVER = 1;

    /\*\*

     \* Debug level to show client -> server and server -> client messages.

     \*

     \* @var int

     \*/

    const DEBUG\_CLIENT = 2;

    /\*\*

     \* Simple static wrapper for all-in-one POP before SMTP.

     \*

     \* @param string   $host        The hostname to connect to

     \* @param int|bool $port        The port number to connect to

     \* @param int|bool $timeout     The timeout value

     \* @param string   $username

     \* @param string   $password

     \* @param int      $debug\_level

     \*

     \* @return bool

     \*/

    public static function popBeforeSmtp(

        $host,

        $port = false,

        $timeout = false,

        $username = '',

        $password = '',

        $debug\_level = 0

    ) {

        $pop = new self();

        return $pop->authorise($host, $port, $timeout, $username, $password, $debug\_level);

    }

    /\*\*

     \* Authenticate with a POP3 server.

     \* A connect, login, disconnect sequence

     \* appropriate for POP-before SMTP authorisation.

     \*

     \* @param string   $host        The hostname to connect to

     \* @param int|bool $port        The port number to connect to

     \* @param int|bool $timeout     The timeout value

     \* @param string   $username

     \* @param string   $password

     \* @param int      $debug\_level

     \*

     \* @return bool

     \*/

    public function authorise($host, $port = false, $timeout = false, $username = '', $password = '', $debug\_level = 0)

    {

        $this->host = $host;

        //If no port value provided, use default

        if (false === $port) {

            $this->port = static::DEFAULT\_PORT;

        } else {

            $this->port = (int) $port;

        }

        //If no timeout value provided, use default

        if (false === $timeout) {

            $this->tval = static::DEFAULT\_TIMEOUT;

        } else {

            $this->tval = (int) $timeout;

        }

        $this->do\_debug = $debug\_level;

        $this->username = $username;

        $this->password = $password;

        //Reset the error log

        $this->errors = [];

        //Connect

        $result = $this->connect($this->host, $this->port, $this->tval);

        if ($result) {

            $login\_result = $this->login($this->username, $this->password);

            if ($login\_result) {

                $this->disconnect();

                return true;

            }

        }

        //We need to disconnect regardless of whether the login succeeded

        $this->disconnect();

        return false;

    }

    /\*\*

     \* Connect to a POP3 server.

     \*

     \* @param string   $host

     \* @param int|bool $port

     \* @param int      $tval

     \*

     \* @return bool

     \*/

    public function connect($host, $port = false, $tval = 30)

    {

        //Are we already connected?

        if ($this->connected) {

            return true;

        }

        //On Windows this will raise a PHP Warning error if the hostname doesn't exist.

        //Rather than suppress it with @fsockopen, capture it cleanly instead

        set\_error\_handler([$this, 'catchWarning']);

        if (false === $port) {

            $port = static::DEFAULT\_PORT;

        }

        //Connect to the POP3 server

        $errno = 0;

        $errstr = '';

        $this->pop\_conn = fsockopen(

            $host, //POP3 Host

            $port, //Port #

            $errno, //Error Number

            $errstr, //Error Message

            $tval

        ); //Timeout (seconds)

        //Restore the error handler

        restore\_error\_handler();

        //Did we connect?

        if (false === $this->pop\_conn) {

            //It would appear not...

            $this->setError(

                "Failed to connect to server $host on port $port. errno: $errno; errstr: $errstr"

            );

            return false;

        }

        //Increase the stream time-out

        stream\_set\_timeout($this->pop\_conn, $tval, 0);

        //Get the POP3 server response

        $pop3\_response = $this->getResponse();

        //Check for the +OK

        if ($this->checkResponse($pop3\_response)) {

            //The connection is established and the POP3 server is talking

            $this->connected = true;

            return true;

        }

        return false;

    }

    /\*\*

     \* Log in to the POP3 server.

     \* Does not support APOP (RFC 2828, 4949).

     \*

     \* @param string $username

     \* @param string $password

     \*

     \* @return bool

     \*/

    public function login($username = '', $password = '')

    {

        if (!$this->connected) {

            $this->setError('Not connected to POP3 server');

            return false;

        }

        if (empty($username)) {

            $username = $this->username;

        }

        if (empty($password)) {

            $password = $this->password;

        }

        //Send the Username

        $this->sendString("USER $username" . static::LE);

        $pop3\_response = $this->getResponse();

        if ($this->checkResponse($pop3\_response)) {

            //Send the Password

            $this->sendString("PASS $password" . static::LE);

            $pop3\_response = $this->getResponse();

            if ($this->checkResponse($pop3\_response)) {

                return true;

            }

        }

        return false;

    }

    /\*\*

     \* Disconnect from the POP3 server.

     \*/

    public function disconnect()

    {

        $this->sendString('QUIT');

        // RFC 1939 shows POP3 server sending a +OK response to the QUIT command.

        // Try to get it.  Ignore any failures here.

        try {

            $this->getResponse();

        } catch (Exception $e) {

            //Do nothing

        }

        //The QUIT command may cause the daemon to exit, which will kill our connection

        //So ignore errors here

        try {

            @fclose($this->pop\_conn);

        } catch (Exception $e) {

            //Do nothing

        }

        // Clean up attributes.

        $this->connected = false;

        $this->pop\_conn  = false;

    }

    /\*\*

     \* Get a response from the POP3 server.

     \*

     \* @param int $size The maximum number of bytes to retrieve

     \*

     \* @return string

     \*/

    protected function getResponse($size = 128)

    {

        $response = fgets($this->pop\_conn, $size);

        if ($this->do\_debug >= self::DEBUG\_SERVER) {

            echo 'Server -> Client: ', $response;

        }

        return $response;

    }

    /\*\*

     \* Send raw data to the POP3 server.

     \*

     \* @param string $string

     \*

     \* @return int

     \*/

    protected function sendString($string)

    {

        if ($this->pop\_conn) {

            if ($this->do\_debug >= self::DEBUG\_CLIENT) { //Show client messages when debug >= 2

                echo 'Client -> Server: ', $string;

            }

            return fwrite($this->pop\_conn, $string, strlen($string));

        }

        return 0;

    }

    /\*\*

     \* Checks the POP3 server response.

     \* Looks for for +OK or -ERR.

     \*

     \* @param string $string

     \*

     \* @return bool

     \*/

    protected function checkResponse($string)

    {

        if (strpos($string, '+OK') !== 0) {

            $this->setError("Server reported an error: $string");

            return false;

        }

        return true;

    }

    /\*\*

     \* Add an error to the internal error store.

     \* Also display debug output if it's enabled.

     \*

     \* @param string $error

     \*/

    protected function setError($error)

    {

        $this->errors[] = $error;

        if ($this->do\_debug >= self::DEBUG\_SERVER) {

            echo '<pre>';

            foreach ($this->errors as $e) {

                print\_r($e);

            }

            echo '</pre>';

        }

    }

    /\*\*

     \* Get an array of error messages, if any.

     \*

     \* @return array

     \*/

    public function getErrors()

    {

        return $this->errors;

    }

    /\*\*

     \* POP3 connection error handler.

     \*

     \* @param int    $errno

     \* @param string $errstr

     \* @param string $errfile

     \* @param int    $errline

     \*/

    protected function catchWarning($errno, $errstr, $errfile, $errline)

    {

        $this->setError(

            'Connecting to the POP3 server raised a PHP warning:' .

            "errno: $errno errstr: $errstr; errfile: $errfile; errline: $errline"

        );

    }

}

**SMTP**

<?php

/\*\*

 \* PHPMailer RFC821 SMTP email transport class.

 \* PHP Version 5.5.

 \*

 \* @see       https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub project

 \*

 \* @author    Marcus Bointon (Synchro/coolbru) <phpmailer@synchromedia.co.uk>

 \* @author    Jim Jagielski (jimjag) <jimjag@gmail.com>

 \* @author    Andy Prevost (codeworxtech) <codeworxtech@users.sourceforge.net>

 \* @author    Brent R. Matzelle (original founder)

 \* @copyright 2012 - 2020 Marcus Bointon

 \* @copyright 2010 - 2012 Jim Jagielski

 \* @copyright 2004 - 2009 Andy Prevost

 \* @license   http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License

 \* @note      This program is distributed in the hope that it will be useful - WITHOUT

 \* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

 \* FITNESS FOR A PARTICULAR PURPOSE.

 \*/

namespace PHPMailer\PHPMailer;

/\*\*

 \* PHPMailer RFC821 SMTP email transport class.

 \* Implements RFC 821 SMTP commands and provides some utility methods for sending mail to an SMTP server.

 \*

 \* @author Chris Ryan

 \* @author Marcus Bointon <phpmailer@synchromedia.co.uk>

 \*/

class SMTP

{

    /\*\*

     \* The PHPMailer SMTP version number.

     \*

     \* @var string

     \*/

    const VERSION = '6.6.0';

    /\*\*

     \* SMTP line break constant.

     \*

     \* @var string

     \*/

    const LE = "\r\n";

    /\*\*

     \* The SMTP port to use if one is not specified.

     \*

     \* @var int

     \*/

    const DEFAULT\_PORT = 25;

    /\*\*

     \* The maximum line length allowed by RFC 5321 section 4.5.3.1.6,

     \* \*excluding\* a trailing CRLF break.

     \*

     \* @see https://tools.ietf.org/html/rfc5321#section-4.5.3.1.6

     \*

     \* @var int

     \*/

    const MAX\_LINE\_LENGTH = 998;

    /\*\*

     \* The maximum line length allowed for replies in RFC 5321 section 4.5.3.1.5,

     \* \*including\* a trailing CRLF line break.

     \*

     \* @see https://tools.ietf.org/html/rfc5321#section-4.5.3.1.5

     \*

     \* @var int

     \*/

    const MAX\_REPLY\_LENGTH = 512;

    /\*\*

     \* Debug level for no output.

     \*

     \* @var int

     \*/

    const DEBUG\_OFF = 0;

    /\*\*

     \* Debug level to show client -> server messages.

     \*

     \* @var int

     \*/

    const DEBUG\_CLIENT = 1;

    /\*\*

     \* Debug level to show client -> server and server -> client messages.

     \*

     \* @var int

     \*/

    const DEBUG\_SERVER = 2;

    /\*\*

     \* Debug level to show connection status, client -> server and server -> client messages.

     \*

     \* @var int

     \*/

    const DEBUG\_CONNECTION = 3;

    /\*\*

     \* Debug level to show all messages.

     \*

     \* @var int

     \*/

    const DEBUG\_LOWLEVEL = 4;

    /\*\*

     \* Debug output level.

     \* Options:

     \* \* self::DEBUG\_OFF (`0`) No debug output, default

     \* \* self::DEBUG\_CLIENT (`1`) Client commands

     \* \* self::DEBUG\_SERVER (`2`) Client commands and server responses

     \* \* self::DEBUG\_CONNECTION (`3`) As DEBUG\_SERVER plus connection status

     \* \* self::DEBUG\_LOWLEVEL (`4`) Low-level data output, all messages.

     \*

     \* @var int

     \*/

    public $do\_debug = self::DEBUG\_OFF;

    /\*\*

     \* How to handle debug output.

     \* Options:

     \* \* `echo` Output plain-text as-is, appropriate for CLI

     \* \* `html` Output escaped, line breaks converted to `<br>`, appropriate for browser output

     \* \* `error\_log` Output to error log as configured in php.ini

     \* Alternatively, you can provide a callable expecting two params: a message string and the debug level:

     \*

     \* ```php

     \* $smtp->Debugoutput = function($str, $level) {echo "debug level $level; message: $str";};

     \* ```

     \*

     \* Alternatively, you can pass in an instance of a PSR-3 compatible logger, though only `debug`

     \* level output is used:

     \*

     \* ```php

     \* $mail->Debugoutput = new myPsr3Logger;

     \* ```

     \*

     \* @var string|callable|\Psr\Log\LoggerInterface

     \*/

    public $Debugoutput = 'echo';

    /\*\*

     \* Whether to use VERP.

     \*

     \* @see http://en.wikipedia.org/wiki/Variable\_envelope\_return\_path

     \* @see http://www.postfix.org/VERP\_README.html Info on VERP

     \*

     \* @var bool

     \*/

    public $do\_verp = false;

    /\*\*

     \* The timeout value for connection, in seconds.

     \* Default of 5 minutes (300sec) is from RFC2821 section 4.5.3.2.

     \* This needs to be quite high to function correctly with hosts using greetdelay as an anti-spam measure.

     \*

     \* @see http://tools.ietf.org/html/rfc2821#section-4.5.3.2

     \*

     \* @var int

     \*/

    public $Timeout = 300;

    /\*\*

     \* How long to wait for commands to complete, in seconds.

     \* Default of 5 minutes (300sec) is from RFC2821 section 4.5.3.2.

     \*

     \* @var int

     \*/

    public $Timelimit = 300;

    /\*\*

     \* Patterns to extract an SMTP transaction id from reply to a DATA command.

     \* The first capture group in each regex will be used as the ID.

     \* MS ESMTP returns the message ID, which may not be correct for internal tracking.

     \*

     \* @var string[]

     \*/

    protected $smtp\_transaction\_id\_patterns = [

        'exim' => '/[\d]{3} OK id=(.\*)/',

        'sendmail' => '/[\d]{3} 2.0.0 (.\*) Message/',

        'postfix' => '/[\d]{3} 2.0.0 Ok: queued as (.\*)/',

        'Microsoft\_ESMTP' => '/[0-9]{3} 2.[\d].0 (.\*)@(?:.\*) Queued mail for delivery/',

        'Amazon\_SES' => '/[\d]{3} Ok (.\*)/',

        'SendGrid' => '/[\d]{3} Ok: queued as (.\*)/',

        'CampaignMonitor' => '/[\d]{3} 2.0.0 OK:([a-zA-Z\d]{48})/',

        'Haraka' => '/[\d]{3} Message Queued **\(**(.\*)**\)**/',

        'Mailjet' => '/[\d]{3} OK queued as (.\*)/',

    ];

    /\*\*

     \* The last transaction ID issued in response to a DATA command,

     \* if one was detected.

     \*

     \* @var string|bool|null

     \*/

    protected $last\_smtp\_transaction\_id;

    /\*\*

     \* The socket for the server connection.

     \*

     \* @var ?resource

     \*/

    protected $smtp\_conn;

    /\*\*

     \* Error information, if any, for the last SMTP command.

     \*

     \* @var array

     \*/

    protected $error = [

        'error' => '',

        'detail' => '',

        'smtp\_code' => '',

        'smtp\_code\_ex' => '',

    ];

    /\*\*

     \* The reply the server sent to us for HELO.

     \* If null, no HELO string has yet been received.

     \*

     \* @var string|null

     \*/

    protected $helo\_rply;

    /\*\*

     \* The set of SMTP extensions sent in reply to EHLO command.

     \* Indexes of the array are extension names.

     \* Value at index 'HELO' or 'EHLO' (according to command that was sent)

     \* represents the server name. In case of HELO it is the only element of the array.

     \* Other values can be boolean TRUE or an array containing extension options.

     \* If null, no HELO/EHLO string has yet been received.

     \*

     \* @var array|null

     \*/

    protected $server\_caps;

    /\*\*

     \* The most recent reply received from the server.

     \*

     \* @var string

     \*/

    protected $last\_reply = '';

    /\*\*

     \* Output debugging info via a user-selected method.

     \*

     \* @param string $str   Debug string to output

     \* @param int    $level The debug level of this message; see DEBUG\_\* constants

     \*

     \* @see SMTP::$Debugoutput

     \* @see SMTP::$do\_debug

     \*/

    protected function edebug($str, $level = 0)

    {

        if ($level > $this->do\_debug) {

            return;

        }

        //Is this a PSR-3 logger?

        if ($this->Debugoutput instanceof \Psr\Log\LoggerInterface) {

            $this->Debugoutput->debug($str);

            return;

        }

        //Avoid clash with built-in function names

        if (is\_callable($this->Debugoutput) && !in\_array($this->Debugoutput, ['error\_log', 'html', 'echo'])) {

            call\_user\_func($this->Debugoutput, $str, $level);

            return;

        }

        switch ($this->Debugoutput) {

            case 'error\_log':

                //Don't output, just log

                error\_log($str);

                break;

            case 'html':

                //Cleans up output a bit for a better looking, HTML-safe output

                echo gmdate('Y-m-d H:i:s'), ' ', htmlentities(

                    preg\_replace('/[\r\n]+/', '', $str),

                    ENT\_QUOTES,

                    'UTF-8'

                ), "<br>\n";

                break;

            case 'echo':

            default:

                //Normalize line breaks

                $str = preg\_replace('/**\r\n**|**\r**/m', "\n", $str);

                echo gmdate('Y-m-d H:i:s'),

                "\t",

                    //Trim trailing space

                trim(

                    //Indent for readability, except for trailing break

                    str\_replace(

                        "\n",

                        "\n                   \t                  ",

                        trim($str)

                    )

                ),

                "\n";

        }

    }

    /\*\*

     \* Connect to an SMTP server.

     \*

     \* @param string $host    SMTP server IP or host name

     \* @param int    $port    The port number to connect to

     \* @param int    $timeout How long to wait for the connection to open

     \* @param array  $options An array of options for stream\_context\_create()

     \*

     \* @return bool

     \*/

    public function connect($host, $port = null, $timeout = 30, $options = [])

    {

        //Clear errors to avoid confusion

        $this->setError('');

        //Make sure we are \_\_not\_\_ connected

        if ($this->connected()) {

            //Already connected, generate error

            $this->setError('Already connected to a server');

            return false;

        }

        if (empty($port)) {

            $port = self::DEFAULT\_PORT;

        }

        //Connect to the SMTP server

        $this->edebug(

            "Connection: opening to $host:$port, timeout=$timeout, options=" .

            (count($options) > 0 ? var\_export($options, true) : 'array()'),

            self::DEBUG\_CONNECTION

        );

        $this->smtp\_conn = $this->getSMTPConnection($host, $port, $timeout, $options);

        if ($this->smtp\_conn === false) {

            //Error info already set inside `getSMTPConnection()`

            return false;

        }

        $this->edebug('Connection: opened', self::DEBUG\_CONNECTION);

        //Get any announcement

        $this->last\_reply = $this->get\_lines();

        $this->edebug('SERVER -> CLIENT: ' . $this->last\_reply, self::DEBUG\_SERVER);

        $responseCode = (int)substr($this->last\_reply, 0, 3);

        if ($responseCode === 220) {

            return true;

        }

        //Anything other than a 220 response means something went wrong

        //RFC 5321 says the server will wait for us to send a QUIT in response to a 554 error

        //https://tools.ietf.org/html/rfc5321#section-3.1

        if ($responseCode === 554) {

            $this->quit();

        }

        //This will handle 421 responses which may not wait for a QUIT (e.g. if the server is being shut down)

        $this->edebug('Connection: closing due to error', self::DEBUG\_CONNECTION);

        $this->close();

        return false;

    }

    /\*\*

     \* Create connection to the SMTP server.

     \*

     \* @param string $host    SMTP server IP or host name

     \* @param int    $port    The port number to connect to

     \* @param int    $timeout How long to wait for the connection to open

     \* @param array  $options An array of options for stream\_context\_create()

     \*

     \* @return false|resource

     \*/

    protected function getSMTPConnection($host, $port = null, $timeout = 30, $options = [])

    {

        static $streamok;

        //This is enabled by default since 5.0.0 but some providers disable it

        //Check this once and cache the result

        if (null === $streamok) {

            $streamok = function\_exists('stream\_socket\_client');

        }

        $errno = 0;

        $errstr = '';

        if ($streamok) {

            $socket\_context = stream\_context\_create($options);

            set\_error\_handler([$this, 'errorHandler']);

            $connection = stream\_socket\_client(

                $host . ':' . $port,

                $errno,

                $errstr,

                $timeout,

                STREAM\_CLIENT\_CONNECT,

                $socket\_context

            );

        } else {

            //Fall back to fsockopen which should work in more places, but is missing some features

            $this->edebug(

                'Connection: stream\_socket\_client not available, falling back to fsockopen',

                self::DEBUG\_CONNECTION

            );

            set\_error\_handler([$this, 'errorHandler']);

            $connection = fsockopen(

                $host,

                $port,

                $errno,

                $errstr,

                $timeout

            );

        }

        restore\_error\_handler();

        //Verify we connected properly

        if (!is\_resource($connection)) {

            $this->setError(

                'Failed to connect to server',

                '',

                (string) $errno,

                $errstr

            );

            $this->edebug(

                'SMTP ERROR: ' . $this->error['error']

                . ": $errstr ($errno)",

                self::DEBUG\_CLIENT

            );

            return false;

        }

        //SMTP server can take longer to respond, give longer timeout for first read

        //Windows does not have support for this timeout function

        if (strpos(PHP\_OS, 'WIN') !== 0) {

            $max = (int)ini\_get('max\_execution\_time');

            //Don't bother if unlimited, or if set\_time\_limit is disabled

            if (0 !== $max && $timeout > $max && strpos(ini\_get('disable\_functions'), 'set\_time\_limit') === false) {

                @set\_time\_limit($timeout);

            }

            stream\_set\_timeout($connection, $timeout, 0);

        }

        return $connection;

    }

    /\*\*

     \* Initiate a TLS (encrypted) session.

     \*

     \* @return bool

     \*/

    public function startTLS()

    {

        if (!$this->sendCommand('STARTTLS', 'STARTTLS', 220)) {

            return false;

        }

        //Allow the best TLS version(s) we can

        $crypto\_method = STREAM\_CRYPTO\_METHOD\_TLS\_CLIENT;

        //PHP 5.6.7 dropped inclusion of TLS 1.1 and 1.2 in STREAM\_CRYPTO\_METHOD\_TLS\_CLIENT

        //so add them back in manually if we can

        if (defined('STREAM\_CRYPTO\_METHOD\_TLSv1\_2\_CLIENT')) {

            $crypto\_method |= STREAM\_CRYPTO\_METHOD\_TLSv1\_2\_CLIENT;

            $crypto\_method |= STREAM\_CRYPTO\_METHOD\_TLSv1\_1\_CLIENT;

        }

        //Begin encrypted connection

        set\_error\_handler([$this, 'errorHandler']);

        $crypto\_ok = stream\_socket\_enable\_crypto(

            $this->smtp\_conn,

            true,

            $crypto\_method

        );

        restore\_error\_handler();

        return (bool) $crypto\_ok;

    }

    /\*\*

     \* Perform SMTP authentication.

     \* Must be run after hello().

     \*

     \* @see    hello()

     \*

     \* @param string $username The user name

     \* @param string $password The password

     \* @param string $authtype The auth type (CRAM-MD5, PLAIN, LOGIN, XOAUTH2)

     \* @param OAuthTokenProvider $OAuth An optional OAuthTokenProvider instance for XOAUTH2 authentication

     \*

     \* @return bool True if successfully authenticated

     \*/

    public function authenticate(

        $username,

        $password,

        $authtype = null,

        $OAuth = null

    ) {

        if (!$this->server\_caps) {

            $this->setError('Authentication is not allowed before HELO/EHLO');

            return false;

        }

        if (array\_key\_exists('EHLO', $this->server\_caps)) {

            //SMTP extensions are available; try to find a proper authentication method

            if (!array\_key\_exists('AUTH', $this->server\_caps)) {

                $this->setError('Authentication is not allowed at this stage');

                //'at this stage' means that auth may be allowed after the stage changes

                //e.g. after STARTTLS

                return false;

            }

            $this->edebug('Auth method requested: ' . ($authtype ?: 'UNSPECIFIED'), self::DEBUG\_LOWLEVEL);

            $this->edebug(

                'Auth methods available on the server: ' . implode(',', $this->server\_caps['AUTH']),

                self::DEBUG\_LOWLEVEL

            );

            //If we have requested a specific auth type, check the server supports it before trying others

            if (null !== $authtype && !in\_array($authtype, $this->server\_caps['AUTH'], true)) {

                $this->edebug('Requested auth method not available: ' . $authtype, self::DEBUG\_LOWLEVEL);

                $authtype = null;

            }

            if (empty($authtype)) {

                //If no auth mechanism is specified, attempt to use these, in this order

                //Try CRAM-MD5 first as it's more secure than the others

                foreach (['CRAM-MD5', 'LOGIN', 'PLAIN', 'XOAUTH2'] as $method) {

                    if (in\_array($method, $this->server\_caps['AUTH'], true)) {

                        $authtype = $method;

                        break;

                    }

                }

                if (empty($authtype)) {

                    $this->setError('No supported authentication methods found');

                    return false;

                }

                $this->edebug('Auth method selected: ' . $authtype, self::DEBUG\_LOWLEVEL);

            }

            if (!in\_array($authtype, $this->server\_caps['AUTH'], true)) {

                $this->setError("The requested authentication method \"$authtype\" is not supported by the server");

                return false;

            }

        } elseif (empty($authtype)) {

            $authtype = 'LOGIN';

        }

        switch ($authtype) {

            case 'PLAIN':

                //Start authentication

                if (!$this->sendCommand('AUTH', 'AUTH PLAIN', 334)) {

                    return false;

                }

                //Send encoded username and password

                if (

                    //Format from https://tools.ietf.org/html/rfc4616#section-2

                    //We skip the first field (it's forgery), so the string starts with a null byte

                    !$this->sendCommand(

                        'User & Password',

                        base64\_encode("\0" . $username . "\0" . $password),

                        235

                    )

                ) {

                    return false;

                }

                break;

            case 'LOGIN':

                //Start authentication

                if (!$this->sendCommand('AUTH', 'AUTH LOGIN', 334)) {

                    return false;

                }

                if (!$this->sendCommand('Username', base64\_encode($username), 334)) {

                    return false;

                }

                if (!$this->sendCommand('Password', base64\_encode($password), 235)) {

                    return false;

                }

                break;

            case 'CRAM-MD5':

                //Start authentication

                if (!$this->sendCommand('AUTH CRAM-MD5', 'AUTH CRAM-MD5', 334)) {

                    return false;

                }

                //Get the challenge

                $challenge = base64\_decode(substr($this->last\_reply, 4));

                //Build the response

                $response = $username . ' ' . $this->hmac($challenge, $password);

                //send encoded credentials

                return $this->sendCommand('Username', base64\_encode($response), 235);

            case 'XOAUTH2':

                //The OAuth instance must be set up prior to requesting auth.

                if (null === $OAuth) {

                    return false;

                }

                $oauth = $OAuth->getOauth64();

                //Start authentication

                if (!$this->sendCommand('AUTH', 'AUTH XOAUTH2 ' . $oauth, 235)) {

                    return false;

                }

                break;

            default:

                $this->setError("Authentication method \"$authtype\" is not supported");

                return false;

        }

        return true;

    }

    /\*\*

     \* Calculate an MD5 HMAC hash.

     \* Works like hash\_hmac('md5', $data, $key)

     \* in case that function is not available.

     \*

     \* @param string $data The data to hash

     \* @param string $key  The key to hash with

     \*

     \* @return string

     \*/

    protected function hmac($data, $key)

    {

        if (function\_exists('hash\_hmac')) {

            return hash\_hmac('md5', $data, $key);

        }

        //The following borrowed from

        //http://php.net/manual/en/function.mhash.php#27225

        //RFC 2104 HMAC implementation for php.

        //Creates an md5 HMAC.

        //Eliminates the need to install mhash to compute a HMAC

        //by Lance Rushing

        $bytelen = 64; //byte length for md5

        if (strlen($key) > $bytelen) {

            $key = pack('H\*', md5($key));

        }

        $key = str\_pad($key, $bytelen, chr(0x00));

        $ipad = str\_pad('', $bytelen, chr(0x36));

        $opad = str\_pad('', $bytelen, chr(0x5c));

        $k\_ipad = $key ^ $ipad;

        $k\_opad = $key ^ $opad;

        return md5($k\_opad . pack('H\*', md5($k\_ipad . $data)));

    }

    /\*\*

     \* Check connection state.

     \*

     \* @return bool True if connected

     \*/

    public function connected()

    {

        if (is\_resource($this->smtp\_conn)) {

            $sock\_status = stream\_get\_meta\_data($this->smtp\_conn);

            if ($sock\_status['eof']) {

                //The socket is valid but we are not connected

                $this->edebug(

                    'SMTP NOTICE: EOF caught while checking if connected',

                    self::DEBUG\_CLIENT

                );

                $this->close();

                return false;

            }

            return true; //everything looks good

        }

        return false;

    }

    /\*\*

     \* Close the socket and clean up the state of the class.

     \* Don't use this function without first trying to use QUIT.

     \*

     \* @see quit()

     \*/

    public function close()

    {

        $this->setError('');

        $this->server\_caps = null;

        $this->helo\_rply = null;

        if (is\_resource($this->smtp\_conn)) {

            //Close the connection and cleanup

            fclose($this->smtp\_conn);

            $this->smtp\_conn = null; //Makes for cleaner serialization

            $this->edebug('Connection: closed', self::DEBUG\_CONNECTION);

        }

    }

    /\*\*

     \* Send an SMTP DATA command.

     \* Issues a data command and sends the msg\_data to the server,

     \* finalizing the mail transaction. $msg\_data is the message

     \* that is to be send with the headers. Each header needs to be

     \* on a single line followed by a <CRLF> with the message headers

     \* and the message body being separated by an additional <CRLF>.

     \* Implements RFC 821: DATA <CRLF>.

     \*

     \* @param string $msg\_data Message data to send

     \*

     \* @return bool

     \*/

    public function data($msg\_data)

    {

        //This will use the standard timelimit

        if (!$this->sendCommand('DATA', 'DATA', 354)) {

            return false;

        }

        /\* The server is ready to accept data!

         \* According to rfc821 we should not send more than 1000 characters on a single line (including the LE)

         \* so we will break the data up into lines by \r and/or \n then if needed we will break each of those into

         \* smaller lines to fit within the limit.

         \* We will also look for lines that start with a '.' and prepend an additional '.'.

         \* NOTE: this does not count towards line-length limit.

         \*/

        //Normalize line breaks before exploding

        $lines = explode("\n", str\_replace(["\r\n", "\r"], "\n", $msg\_data));

        /\* To distinguish between a complete RFC822 message and a plain message body, we check if the first field

         \* of the first line (':' separated) does not contain a space then it \_should\_ be a header and we will

         \* process all lines before a blank line as headers.

         \*/

        $field = substr($lines[0], 0, strpos($lines[0], ':'));

        $in\_headers = false;

        if (!empty($field) && strpos($field, ' ') === false) {

            $in\_headers = true;

        }

        foreach ($lines as $line) {

            $lines\_out = [];

            if ($in\_headers && $line === '') {

                $in\_headers = false;

            }

            //Break this line up into several smaller lines if it's too long

            //Micro-optimisation: isset($str[$len]) is faster than (strlen($str) > $len),

            while (isset($line[self::MAX\_LINE\_LENGTH])) {

                //Working backwards, try to find a space within the last MAX\_LINE\_LENGTH chars of the line to break on

                //so as to avoid breaking in the middle of a word

                $pos = strrpos(substr($line, 0, self::MAX\_LINE\_LENGTH), ' ');

                //Deliberately matches both false and 0

                if (!$pos) {

                    //No nice break found, add a hard break

                    $pos = self::MAX\_LINE\_LENGTH - 1;

                    $lines\_out[] = substr($line, 0, $pos);

                    $line = substr($line, $pos);

                } else {

                    //Break at the found point

                    $lines\_out[] = substr($line, 0, $pos);

                    //Move along by the amount we dealt with

                    $line = substr($line, $pos + 1);

                }

                //If processing headers add a LWSP-char to the front of new line RFC822 section 3.1.1

                if ($in\_headers) {

                    $line = "\t" . $line;

                }

            }

            $lines\_out[] = $line;

            //Send the lines to the server

            foreach ($lines\_out as $line\_out) {

                //Dot-stuffing as per RFC5321 section 4.5.2

                //https://tools.ietf.org/html/rfc5321#section-4.5.2

                if (!empty($line\_out) && $line\_out[0] === '.') {

                    $line\_out = '.' . $line\_out;

                }

                $this->client\_send($line\_out . static::LE, 'DATA');

            }

        }

        //Message data has been sent, complete the command

        //Increase timelimit for end of DATA command

        $savetimelimit = $this->Timelimit;

        $this->Timelimit \*= 2;

        $result = $this->sendCommand('DATA END', '.', 250);

        $this->recordLastTransactionID();

        //Restore timelimit

        $this->Timelimit = $savetimelimit;

        return $result;

    }

    /\*\*

     \* Send an SMTP HELO or EHLO command.

     \* Used to identify the sending server to the receiving server.

     \* This makes sure that client and server are in a known state.

     \* Implements RFC 821: HELO <SP> <domain> <CRLF>

     \* and RFC 2821 EHLO.

     \*

     \* @param string $host The host name or IP to connect to

     \*

     \* @return bool

     \*/

    public function hello($host = '')

    {

        //Try extended hello first (RFC 2821)

        if ($this->sendHello('EHLO', $host)) {

            return true;

        }

        //Some servers shut down the SMTP service here (RFC 5321)

        if (substr($this->helo\_rply, 0, 3) == '421') {

            return false;

        }

        return $this->sendHello('HELO', $host);

    }

    /\*\*

     \* Send an SMTP HELO or EHLO command.

     \* Low-level implementation used by hello().

     \*

     \* @param string $hello The HELO string

     \* @param string $host  The hostname to say we are

     \*

     \* @return bool

     \*

     \* @see hello()

     \*/

    protected function sendHello($hello, $host)

    {

        $noerror = $this->sendCommand($hello, $hello . ' ' . $host, 250);

        $this->helo\_rply = $this->last\_reply;

        if ($noerror) {

            $this->parseHelloFields($hello);

        } else {

            $this->server\_caps = null;

        }

        return $noerror;

    }

    /\*\*

     \* Parse a reply to HELO/EHLO command to discover server extensions.

     \* In case of HELO, the only parameter that can be discovered is a server name.

     \*

     \* @param string $type `HELO` or `EHLO`

     \*/

    protected function parseHelloFields($type)

    {

        $this->server\_caps = [];

        $lines = explode("\n", $this->helo\_rply);

        foreach ($lines as $n => $s) {

            //First 4 chars contain response code followed by - or space

            $s = trim(substr($s, 4));

            if (empty($s)) {

                continue;

            }

            $fields = explode(' ', $s);

            if (!empty($fields)) {

                if (!$n) {

                    $name = $type;

                    $fields = $fields[0];

                } else {

                    $name = array\_shift($fields);

                    switch ($name) {

                        case 'SIZE':

                            $fields = ($fields ? $fields[0] : 0);

                            break;

                        case 'AUTH':

                            if (!is\_array($fields)) {

                                $fields = [];

                            }

                            break;

                        default:

                            $fields = true;

                    }

                }

                $this->server\_caps[$name] = $fields;

            }

        }

    }

    /\*\*

     \* Send an SMTP MAIL command.

     \* Starts a mail transaction from the email address specified in

     \* $from. Returns true if successful or false otherwise. If True

     \* the mail transaction is started and then one or more recipient

     \* commands may be called followed by a data command.

     \* Implements RFC 821: MAIL <SP> FROM:<reverse-path> <CRLF>.

     \*

     \* @param string $from Source address of this message

     \*

     \* @return bool

     \*/

    public function mail($from)

    {

        $useVerp = ($this->do\_verp ? ' XVERP' : '');

        return $this->sendCommand(

            'MAIL FROM',

            'MAIL FROM:<' . $from . '>' . $useVerp,

            250

        );

    }

    /\*\*

     \* Send an SMTP QUIT command.

     \* Closes the socket if there is no error or the $close\_on\_error argument is true.

     \* Implements from RFC 821: QUIT <CRLF>.

     \*

     \* @param bool $close\_on\_error Should the connection close if an error occurs?

     \*

     \* @return bool

     \*/

    public function quit($close\_on\_error = true)

    {

        $noerror = $this->sendCommand('QUIT', 'QUIT', 221);

        $err = $this->error; //Save any error

        if ($noerror || $close\_on\_error) {

            $this->close();

            $this->error = $err; //Restore any error from the quit command

        }

        return $noerror;

    }

    /\*\*

     \* Send an SMTP RCPT command.

     \* Sets the TO argument to $toaddr.

     \* Returns true if the recipient was accepted false if it was rejected.

     \* Implements from RFC 821: RCPT <SP> TO:<forward-path> <CRLF>.

     \*

     \* @param string $address The address the message is being sent to

     \* @param string $dsn     Comma separated list of DSN notifications. NEVER, SUCCESS, FAILURE

     \*                        or DELAY. If you specify NEVER all other notifications are ignored.

     \*

     \* @return bool

     \*/

    public function recipient($address, $dsn = '')

    {

        if (empty($dsn)) {

            $rcpt = 'RCPT TO:<' . $address . '>';

        } else {

            $dsn = strtoupper($dsn);

            $notify = [];

            if (strpos($dsn, 'NEVER') !== false) {

                $notify[] = 'NEVER';

            } else {

                foreach (['SUCCESS', 'FAILURE', 'DELAY'] as $value) {

                    if (strpos($dsn, $value) !== false) {

                        $notify[] = $value;

                    }

                }

            }

            $rcpt = 'RCPT TO:<' . $address . '> NOTIFY=' . implode(',', $notify);

        }

        return $this->sendCommand(

            'RCPT TO',

            $rcpt,

            [250, 251]

        );

    }

    /\*\*

     \* Send an SMTP RSET command.

     \* Abort any transaction that is currently in progress.

     \* Implements RFC 821: RSET <CRLF>.

     \*

     \* @return bool True on success

     \*/

    public function reset()

    {

        return $this->sendCommand('RSET', 'RSET', 250);

    }

    /\*\*

     \* Send a command to an SMTP server and check its return code.

     \*

     \* @param string    $command       The command name - not sent to the server

     \* @param string    $commandstring The actual command to send

     \* @param int|array $expect        One or more expected integer success codes

     \*

     \* @return bool True on success

     \*/

    protected function sendCommand($command, $commandstring, $expect)

    {

        if (!$this->connected()) {

            $this->setError("Called $command without being connected");

            return false;

        }

        //Reject line breaks in all commands

        if ((strpos($commandstring, "\n") !== false) || (strpos($commandstring, "\r") !== false)) {

            $this->setError("Command '$command' contained line breaks");

            return false;

        }

        $this->client\_send($commandstring . static::LE, $command);

        $this->last\_reply = $this->get\_lines();

        //Fetch SMTP code and possible error code explanation

        $matches = [];

        if (preg\_match('/^([\d]{3})[ -](?:([\d]**\\.**[\d]**\\.**[\d]{1,2}) )?/', $this->last\_reply, $matches)) {

            $code = (int) $matches[1];

            $code\_ex = (count($matches) > 2 ? $matches[2] : null);

            //Cut off error code from each response line

            $detail = preg\_replace(

                "/{$code}[ -]" .

                ($code\_ex ? str\_replace('.', '\\.', $code\_ex) . ' ' : '') . '/m',

                '',

                $this->last\_reply

            );

        } else {

            //Fall back to simple parsing if regex fails

            $code = (int) substr($this->last\_reply, 0, 3);

            $code\_ex = null;

            $detail = substr($this->last\_reply, 4);

        }

        $this->edebug('SERVER -> CLIENT: ' . $this->last\_reply, self::DEBUG\_SERVER);

        if (!in\_array($code, (array) $expect, true)) {

            $this->setError(

                "$command command failed",

                $detail,

                $code,

                $code\_ex

            );

            $this->edebug(

                'SMTP ERROR: ' . $this->error['error'] . ': ' . $this->last\_reply,

                self::DEBUG\_CLIENT

            );

            return false;

        }

        $this->setError('');

        return true;

    }

    /\*\*

     \* Send an SMTP SAML command.

     \* Starts a mail transaction from the email address specified in $from.

     \* Returns true if successful or false otherwise. If True

     \* the mail transaction is started and then one or more recipient

     \* commands may be called followed by a data command. This command

     \* will send the message to the users terminal if they are logged

     \* in and send them an email.

     \* Implements RFC 821: SAML <SP> FROM:<reverse-path> <CRLF>.

     \*

     \* @param string $from The address the message is from

     \*

     \* @return bool

     \*/

    public function sendAndMail($from)

    {

        return $this->sendCommand('SAML', "SAML FROM:$from", 250);

    }

    /\*\*

     \* Send an SMTP VRFY command.

     \*

     \* @param string $name The name to verify

     \*

     \* @return bool

     \*/

    public function verify($name)

    {

        return $this->sendCommand('VRFY', "VRFY $name", [250, 251]);

    }

    /\*\*

     \* Send an SMTP NOOP command.

     \* Used to keep keep-alives alive, doesn't actually do anything.

     \*

     \* @return bool

     \*/

    public function noop()

    {

        return $this->sendCommand('NOOP', 'NOOP', 250);

    }

    /\*\*

     \* Send an SMTP TURN command.

     \* This is an optional command for SMTP that this class does not support.

     \* This method is here to make the RFC821 Definition complete for this class

     \* and \_may\_ be implemented in future.

     \* Implements from RFC 821: TURN <CRLF>.

     \*

     \* @return bool

     \*/

    public function turn()

    {

        $this->setError('The SMTP TURN command is not implemented');

        $this->edebug('SMTP NOTICE: ' . $this->error['error'], self::DEBUG\_CLIENT);

        return false;

    }

    /\*\*

     \* Send raw data to the server.

     \*

     \* @param string $data    The data to send

     \* @param string $command Optionally, the command this is part of, used only for controlling debug output

     \*

     \* @return int|bool The number of bytes sent to the server or false on error

     \*/

    public function client\_send($data, $command = '')

    {

        //If SMTP transcripts are left enabled, or debug output is posted online

        //it can leak credentials, so hide credentials in all but lowest level

        if (

            self::DEBUG\_LOWLEVEL > $this->do\_debug &&

            in\_array($command, ['User & Password', 'Username', 'Password'], true)

        ) {

            $this->edebug('CLIENT -> SERVER: [credentials hidden]', self::DEBUG\_CLIENT);

        } else {

            $this->edebug('CLIENT -> SERVER: ' . $data, self::DEBUG\_CLIENT);

        }

        set\_error\_handler([$this, 'errorHandler']);

        $result = fwrite($this->smtp\_conn, $data);

        restore\_error\_handler();

        return $result;

    }

    /\*\*

     \* Get the latest error.

     \*

     \* @return array

     \*/

    public function getError()

    {

        return $this->error;

    }

    /\*\*

     \* Get SMTP extensions available on the server.

     \*

     \* @return array|null

     \*/

    public function getServerExtList()

    {

        return $this->server\_caps;

    }

    /\*\*

     \* Get metadata about the SMTP server from its HELO/EHLO response.

     \* The method works in three ways, dependent on argument value and current state:

     \*   1. HELO/EHLO has not been sent - returns null and populates $this->error.

     \*   2. HELO has been sent -

     \*     $name == 'HELO': returns server name

     \*     $name == 'EHLO': returns boolean false

     \*     $name == any other string: returns null and populates $this->error

     \*   3. EHLO has been sent -

     \*     $name == 'HELO'|'EHLO': returns the server name

     \*     $name == any other string: if extension $name exists, returns True

     \*       or its options (e.g. AUTH mechanisms supported). Otherwise returns False.

     \*

     \* @param string $name Name of SMTP extension or 'HELO'|'EHLO'

     \*

     \* @return string|bool|null

     \*/

    public function getServerExt($name)

    {

        if (!$this->server\_caps) {

            $this->setError('No HELO/EHLO was sent');

            return null;

        }

        if (!array\_key\_exists($name, $this->server\_caps)) {

            if ('HELO' === $name) {

                return $this->server\_caps['EHLO'];

            }

            if ('EHLO' === $name || array\_key\_exists('EHLO', $this->server\_caps)) {

                return false;

            }

            $this->setError('HELO handshake was used; No information about server extensions available');

            return null;

        }

        return $this->server\_caps[$name];

    }

    /\*\*

     \* Get the last reply from the server.

     \*

     \* @return string

     \*/

    public function getLastReply()

    {

        return $this->last\_reply;

    }

    /\*\*

     \* Read the SMTP server's response.

     \* Either before eof or socket timeout occurs on the operation.

     \* With SMTP we can tell if we have more lines to read if the

     \* 4th character is '-' symbol. If it is a space then we don't

     \* need to read anything else.

     \*

     \* @return string

     \*/

    protected function get\_lines()

    {

        //If the connection is bad, give up straight away

        if (!is\_resource($this->smtp\_conn)) {

            return '';

        }

        $data = '';

        $endtime = 0;

        stream\_set\_timeout($this->smtp\_conn, $this->Timeout);

        if ($this->Timelimit > 0) {

            $endtime = time() + $this->Timelimit;

        }

        $selR = [$this->smtp\_conn];

        $selW = null;

        while (is\_resource($this->smtp\_conn) && !feof($this->smtp\_conn)) {

            //Must pass vars in here as params are by reference

            //solution for signals inspired by https://github.com/symfony/symfony/pull/6540

            set\_error\_handler([$this, 'errorHandler']);

            $n = stream\_select($selR, $selW, $selW, $this->Timelimit);

            restore\_error\_handler();

            if ($n === false) {

                $message = $this->getError()['detail'];

                $this->edebug(

                    'SMTP -> get\_lines(): select failed (' . $message . ')',

                    self::DEBUG\_LOWLEVEL

                );

                //stream\_select returns false when the `select` system call is interrupted

                //by an incoming signal, try the select again

                if (stripos($message, 'interrupted system call') !== false) {

                    $this->edebug(

                        'SMTP -> get\_lines(): retrying stream\_select',

                        self::DEBUG\_LOWLEVEL

                    );

                    $this->setError('');

                    continue;

                }

                break;

            }

            if (!$n) {

                $this->edebug(

                    'SMTP -> get\_lines(): select timed-out in (' . $this->Timelimit . ' sec)',

                    self::DEBUG\_LOWLEVEL

                );

                break;

            }

            //Deliberate noise suppression - errors are handled afterwards

            $str = @fgets($this->smtp\_conn, self::MAX\_REPLY\_LENGTH);

            $this->edebug('SMTP INBOUND: "' . trim($str) . '"', self::DEBUG\_LOWLEVEL);

            $data .= $str;

            //If response is only 3 chars (not valid, but RFC5321 S4.2 says it must be handled),

            //or 4th character is a space or a line break char, we are done reading, break the loop.

            //String array access is a significant micro-optimisation over strlen

            if (!isset($str[3]) || $str[3] === ' ' || $str[3] === "\r" || $str[3] === "\n") {

                break;

            }

            //Timed-out? Log and break

            $info = stream\_get\_meta\_data($this->smtp\_conn);

            if ($info['timed\_out']) {

                $this->edebug(

                    'SMTP -> get\_lines(): stream timed-out (' . $this->Timeout . ' sec)',

                    self::DEBUG\_LOWLEVEL

                );

                break;

            }

            //Now check if reads took too long

            if ($endtime && time() > $endtime) {

                $this->edebug(

                    'SMTP -> get\_lines(): timelimit reached (' .

                    $this->Timelimit . ' sec)',

                    self::DEBUG\_LOWLEVEL

                );

                break;

            }

        }

        return $data;

    }

    /\*\*

     \* Enable or disable VERP address generation.

     \*

     \* @param bool $enabled

     \*/

    public function setVerp($enabled = false)

    {

        $this->do\_verp = $enabled;

    }

    /\*\*

     \* Get VERP address generation mode.

     \*

     \* @return bool

     \*/

    public function getVerp()

    {

        return $this->do\_verp;

    }

    /\*\*

     \* Set error messages and codes.

     \*

     \* @param string $message      The error message

     \* @param string $detail       Further detail on the error

     \* @param string $smtp\_code    An associated SMTP error code

     \* @param string $smtp\_code\_ex Extended SMTP code

     \*/

    protected function setError($message, $detail = '', $smtp\_code = '', $smtp\_code\_ex = '')

    {

        $this->error = [

            'error' => $message,

            'detail' => $detail,

            'smtp\_code' => $smtp\_code,

            'smtp\_code\_ex' => $smtp\_code\_ex,

        ];

    }

    /\*\*

     \* Set debug output method.

     \*

     \* @param string|callable $method The name of the mechanism to use for debugging output, or a callable to handle it

     \*/

    public function setDebugOutput($method = 'echo')

    {

        $this->Debugoutput = $method;

    }

    /\*\*

     \* Get debug output method.

     \*

     \* @return string

     \*/

    public function getDebugOutput()

    {

        return $this->Debugoutput;

    }

    /\*\*

     \* Set debug output level.

     \*

     \* @param int $level

     \*/

    public function setDebugLevel($level = 0)

    {

        $this->do\_debug = $level;

    }

    /\*\*

     \* Get debug output level.

     \*

     \* @return int

     \*/

    public function getDebugLevel()

    {

        return $this->do\_debug;

    }

    /\*\*

     \* Set SMTP timeout.

     \*

     \* @param int $timeout The timeout duration in seconds

     \*/

    public function setTimeout($timeout = 0)

    {

        $this->Timeout = $timeout;

    }

    /\*\*

     \* Get SMTP timeout.

     \*

     \* @return int

     \*/

    public function getTimeout()

    {

        return $this->Timeout;

    }

    /\*\*

     \* Reports an error number and string.

     \*

     \* @param int    $errno   The error number returned by PHP

     \* @param string $errmsg  The error message returned by PHP

     \* @param string $errfile The file the error occurred in

     \* @param int    $errline The line number the error occurred on

     \*/

    protected function errorHandler($errno, $errmsg, $errfile = '', $errline = 0)

    {

        $notice = 'Connection failed.';

        $this->setError(

            $notice,

            $errmsg,

            (string) $errno

        );

        $this->edebug(

            "$notice Error #$errno: $errmsg [$errfile line $errline]",

            self::DEBUG\_CONNECTION

        );

    }

    /\*\*

     \* Extract and return the ID of the last SMTP transaction based on

     \* a list of patterns provided in SMTP::$smtp\_transaction\_id\_patterns.

     \* Relies on the host providing the ID in response to a DATA command.

     \* If no reply has been received yet, it will return null.

     \* If no pattern was matched, it will return false.

     \*

     \* @return bool|string|null

     \*/

    protected function recordLastTransactionID()

    {

        $reply = $this->getLastReply();

        if (empty($reply)) {

            $this->last\_smtp\_transaction\_id = null;

        } else {

            $this->last\_smtp\_transaction\_id = false;

            foreach ($this->smtp\_transaction\_id\_patterns as $smtp\_transaction\_id\_pattern) {

                $matches = [];

                if (preg\_match($smtp\_transaction\_id\_pattern, $reply, $matches)) {

                    $this->last\_smtp\_transaction\_id = trim($matches[1]);

                    break;

                }

            }

        }

        return $this->last\_smtp\_transaction\_id;

    }

    /\*\*

     \* Get the queue/transaction ID of the last SMTP transaction

     \* If no reply has been received yet, it will return null.

     \* If no pattern was matched, it will return false.

     \*

     \* @return bool|string|null

     \*

     \* @see recordLastTransactionID()

     \*/

    public function getLastTransactionID()

    {

        return $this->last\_smtp\_transaction\_id;

    }

}

**alterarSenha**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Alterar minha senha</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white">

                        <form action="<?php echo URL.'redefinir-senha'; ?>" class="p-sm-3 p-md-3 p-lg-4 p-3 px-0 m-0" method="post">

                            <div class="row align-items-center justify-content-center m-0">

                                <input type="hidden" name="idlogin" value="<?php echo $idlogin;?>">

                                <div class="form-group">

                                    <label class="form-label" for="novaSenha">Insira uma nova senha:</label>

                                    <input type="password" name="novaSenha" id="novaSenha" required  class="form-control">

                                </div>

                                <div class="form-group">

                                    <label class="form-label" for="novaSenha">Confirme sua senha:</label>

                                    <input type="password" name="confSenha" id="confSenha" required  class="form-control">

                                </div>

                                <div class="text-danger" id="avisoIgualdade" style="display:none;">as Senhas devem ser iguais\*</div>

                                <div class="text-danger" id="avisoComprimento" style="display:none;">A senha deve conter mais que 5 dígitos\*</div>

                                <div>

                                    <input type="submit" value="Alterar" class="btn btn-success col-auto mt-3 float-end" name="alterarSenha">

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-usuario';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado -->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <script type="text/javascript">

        //confirme a senha

        $("form").submit(function(){

            if($("#novaSenha").val() != $("#confSenha").val())

            {

                event.preventDefault();

                $("#avisoIgualdade").show();

            }

            else{$("#avisoIgualdade").hide();}

            if(($("#novaSenha").val().length <= 5) && ($("#confSenha").val().length <= 5))

            {

                event.preventDefault();

                $("#avisoComprimento").show();

            }

            else {$("#avisoComprimento").hide();}

        });

    </script>

</body>

</html>

**cadAnimal**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

    <style type="text/css">

        input::-webkit-outer-spin-button,

        input::-webkit-inner-spin-button {

            display: none;

        }

        #inputImgAnimal + label

        {

            position: absolute;

            height: 250px;

            width: 375px;

            border: solid;

            border-color: #198754;

            cursor: pointer;

        }

        #imgAnimal

        {

            height: 250px;

            width: 399px;

        }

        @media screen and (max-width: 1199px)

        {

            #inputImgAnimal + label

            {

                width: 365px;

            }

        }

        @media screen and (max-width: 992px)

        {

            #inputImgAnimal + label

            {

                height: 200px;

                width: 272px;

            }

            #imgAnimal

            {

                height: 200px;

            }

        }

        @media screen and (max-width: 767px)

        {

            #imgAnimal

            {

                width: 290px;

            }

        }

        @media screen and (max-width: 336px)

        {

            #inputImgAnimal + label

            {

                height: 175px;

                width: 200px;

            }

            #imgAnimal

            {

                height: 175px;

                width: 220px;

            }

        }

    </style>

    <!-- FontAwesome -->

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css" integrity="sha512-KfkfwYDsLkIlwQp6LFnl8zNdLGxu9YAA1QvwINks4PhcElQSvqcyVLLD9aMhXd13uQjoXtEKNosOWaZqXgel0g==" crossorigin="anonymous" referrerpolicy="no-referrer" />

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid vh-100 d-grid corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="container-fluid container-principal">

            <div class="bg-primary h-100 row align-items-center">

                <div class="container mx-auto p-3" style="grid-area:corpo;">

                    <div class="container shadow-lg text-light p-3" style="border: 0; border-radius:0; overflow: hidden; background-color: var(--preto);">

                        <h1 class="h5 m-1">CADASTRAR ANIMAL</h1>

                    </div>

                    <div class="container shadow-lg bg-white p-4">

                        <form method="post" action="<?php echo URL.'cadastrar-animal';?>" enctype="multipart/form-data">

                            <div class="row">

                                <div class="col-md-7">

                                    <div class="row">

                                        <div class="col-12 mb-2">

                                            <label for="txtNome" class="form-label">Nome do Animal:</label>

                                            <input type="text" class="form-control" id="txtNome" name="txtNome" maxlength="50" required>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="tipoEspecie" class="form-label">Espécie:</label>

                                            <select id="tipoEspecie" name="tipoEspecie" class="form-select"  onchange="carregarRaca(this)" required>

                                                <option value="">... SELECIONE A ESPÉCIE ...</option>

                                                <option value="0">Canina</option>

                                                <option value="1">Felina</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="numIdade" class="form-label">Idade:</label>

                                            <input type="number" class="form-control" id="numIdade" name="numIdade" min="0" max="100" required>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="slcSexo" class="form-label">Sexo:</label>

                                            <select id="slcSexo" name="slcSexo" class="form-select" required>

                                                <option value="0">Fêmea</option>

                                                <option value="1">Macho</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPelagem" class="form-label">Pelagem:</label>

                                            <select id="slcPelagem" name="slcPelagem" class="form-select" required>

                                                <option value="0">Curta</option>

                                                <option value="1">Média</option>

                                                <option value="2">Alta</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="txtCor" class="form-label">Cor:</label>

                                            <input type="text" class="form-control" id="txtCor" name="txtCor" maxlength="30" required>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPorte" class="form-label">Porte:</label>

                                            <select id="slcPorte" name="slcPorte" class="form-select" required>

                                                <option value="0">Pequeno</option>

                                                <option value="1">Médio</option>

                                                <option value="2">Grande</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="listRaca" class="form-label">Raça:</label>

                                            <select name="racas" id="racas" class="form-select">

                                                <option value="">... SELECIONE A RAÇA ...</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcComunitario" class="form-label">Animal Comunitário:</label>

                                            <select id="slcComunitario" name="slcComunitario" class="form-select" required>

                                                <option value="0">Não</option>

                                                <option value="1">Sim</option>

                                            </select>

                                        </div>

                                    </div>

                                </div>

                                <div class="col-md-5 mt-2">

                                    <div class="row text-center mb-3">

                                        <label>Foto do animal:</label>

                                    </div>

                                    <div class="row justify-content-center mb-3">

                                        <input type="file" name="imgAnimal" id="inputImgAnimal" accept="image/\*" hidden>

                                        <label id="labelImgAnimal" for="inputImgAnimal" style="background-color: 0;"></label>

                                        <img src="recursos/img/imagem\_exemplo.jpg" alt="Foto do Animal" id="imgAnimal" for="inputImgAnimal">

                                    </div>

                                    <div class="row">

                                        <div class="col-12 text-center mt-3 mb-2">

                                            <input type="submit" value="Cadastrar" class="btn btn-lg btn-success shadow" style="border-radius: 0; border: 0; padding: 12px 30px 12px 30px;">

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area:footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'meus-animais'; ?>" class="btn btn-success my-2 my-sm-0">Voltar</a>

                </div>

            </div>

        </div>

        <!-- /CORPO -->

    </div>

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/popper.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <!-- EXTENSÃO JQUERY PARA O AJAX -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js" integrity="sha512-894YE6QWD5I59HgZOGReFYm4dnWc1Qt5NtvYSaNcOP+u1T9qYdvdihz0PPSiiqn/+/3e7Jo4EaG7TubfWGUrMQ==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

    <!-- SCRIPT PARA MOSTRAR A FOTO ENVIADA EM TEMPO REAL-->

    <script>

        inputImgAnimal.onchange = evt => {

            const [file] = inputImgAnimal.files

            if (file) {

                var url = URL.createObjectURL(file);

                imgAnimal.src = url;

            }

        }

    </script>

    <!-- SCRIPT PARA POPULAR SELECT racas -->

    <script>

        function carregarRaca(id)

        {

            //limpar todos antes de carregar

            $("#racas").empty();

            $.ajax({

                url: '<?php echo URL;?>carregar-raca/'+ id.value,

                success: function(data) {

                    $("#racas").append(data);

                }

            });

        }

    </script>

</body>

</html>

**Cadastro**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

        <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">CADASTRAR</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <form method="post" class="p-sm-3 p-md-3 p-lg-4 p-3 px-0 row m-0" action="cadastrar-tutor" enctype="multipart/form-data">

                            <div class="col-md-6 mb-lg-0 mb-5 p-0 pe-md-3">

                                <div class="form-group row-6 mb-3">

                                    <label for="txtNome" class="form-label">Nome:<*font* color="red">\*</*font*></label>

                                    <input  class="form-control" type="text" name="txtNome" id="txtNome" maxlength="70" required>

                                </div>

                                <div class="form-group row-6 mb-3">

                                    <label for="txtEmail" class="form-label">Email:<*font* color="red">\*</*font*></label>

                                    <input class="form-control" type="email" name="txtEmail" id="txtEmail" maxlength="100" required>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group col-sm-6">

                                        <label for="txtCPF" class="form-label">CPF:<*font* color="red">\*</*font*></label>

                                        <input class="form-control" type="text" name="txtCPF" id="txtCPF" maxlength="14" minlength="11" placeholder="000.000.000-00" required>

                                    </div>

                                    <div class="form-group col-sm-6">

                                        <label for="txtTel" class="form-label">Telefone:</label>

                                        <input class="form-control" type="text" name="txtTel" id="txtTel" placeholder="(00) 0000-0000" maxlength="10">

                                    </div>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group col-sm-6">

                                        <label for="txtRG" class="form-label">RG:<*font* color="red">\*</*font*></label>

                                        <input class="form-control" type="text" name="txtRG" id="txtRG" maxlength="12" placeholder="00.000.000-X" required>

                                    </div>

                                    <div class="form-group col-sm-6">

                                        <label for="txtCelular" class="form-label">Celular:<*font* color="red">\*</*font*></label>

                                        <input class="form-control" type="text" name="txtCelular" id="txtCelular" placeholder="(00) 00000-0000" maxlength="15">

                                        <div class="form-group" style="margin-bottom: -10px;">

                                            <input type="checkbox" name="chkWhats" id="chkWhats" value="sim">

                                            <label for="chkWhats" class="form-label">O número é Whatsapp?</label>

                                        </div>

                                    </div>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group col-sm-6">

                                        <label for="btnComprovante" class="form-label">Comprovante de residência:<*font* color="red">\*</*font*></label>

                                    </div>

                                    <div class="form-group col-sm-6">

                                        <input class="btn btn-success" type="file" value="Fazer upload do documento" accept="image/\*" name="btnComprovante" id="btnComprovante" hidden>

                                        <label id="labelComprovante" for="btnComprovante" class="btn btn-success" style="background-color: 0;">Enviar comprovante de residência</label>

                                    </div>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group col-sm-6">

                                        <input type="checkbox" name="chkNIS" id="chkNIS" value="sim">

                                        <label for="chkNIS" class="form-label">Tenho o benefício do NIS</label>

                                    </div>

                                    <div class="form-group col-sm-6">

                                        <input class="form-control" type="text" name="txtNIS" id="txtNIS" placeholder="Número do NIS" maxlength="11" disabled>

                                    </div>

                                </div>

                                <!--

                                <div class="row mb-3">

                                    <div class="form-group col-sm-6">

                                        <input type="checkbox" name="chkProtetor" id="chkProtetor" value="sim">

                                        <label for="chkProtetor" class="form-label">Sou protetor de animais</label>

                                    </div>

                                </div>

                                -->

                            </div>

                            <div class="col-md-6 p-0 ps-md-3">

                                <div class="row mb-3">

                                    <div class="form-group col-6">

                                        <label for="txtCEP" class="form-label">CEP:<*font* color="red">\*</*font*></label>

                                        <input class="form-control" type="text" name="txtCEP" id="txtCEP" maxlength="9" placeholder="00000-000" required>

                                    </div>

                                    <div class="form-group col-6">

                                        <label for="txtNumero" class="form-label">Número:<*font* color="red">\*</*font*></label>

                                        <input class="form-control" type="text" name="txtNumero" id="txtNumero" maxlength="5" required>

                                    </div>

                                </div>

                                <div class="form-group mb-3">

                                    <label for="txtBairro" class="form-label">Bairro:<*font* color="red">\*</*font*></label>

                                    <input class="form-control" type="text" name="txtBairro" id="txtBairro" maxlength="50" required>

                                </div>

                                <div class="form-group mb-3">

                                    <label for="txtRua" class="form-label">Rua:<*font* color="red">\*</*font*></label>

                                    <input class="form-control" type="text" name="txtRua" id="txtRua" maxlength="50" required>

                                </div>

                                <div class="form-group mb-3">

                                    <label for="txtSenha" class="form-label">Crie uma senha:<*font* color="red">\*</*font*></label>

                                    <input class="form-control" type="password" name="txtSenha" id="txtSenha" maxlength="40" required>

                                </div>

                                <div class="form-group mb-3">

                                    <label for="txtConfirmaSenha" class="form-label">Confirme sua senha:<*font* color="red">\*</*font*></label>

                                    <input class="form-control" type="password" name="txtConfirmaSenha" id="txtConfirmaSenha" maxlength="40" required>

                                    <div class="text-danger" id="avisoIgualdade" style="display:none;">as Senhas devem ser iguais\*</div>

                                    <div class="text-danger" id="avisoComprimento" style="display:none;">A senha deve conter mais que 5 dígitos\*</div>

                                </div>

                                <div class="row">

                                    <div class="form-group">

                                        <input type="submit" class="btn btn-success float-end" value="Cadastrar">

                                    </div>

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'login';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- EXTENSÃO JQUERY DAS MASCARAS -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.16/jquery.mask.min.js"></script>

    <script>

        //Adicionar Máscaras

        $(document).ready(function (){

            $("#txtCPF").mask('000.000.000-00');

            $("#txtCEP").mask('00000-000');

            $("#txtRG").mask('00.000.000-X', {'translation': {X: {pattern: /[0-9xX]/}}});

            $("#txtTel").mask('(00) 0000-0000');

            $("#txtCelular").mask('(00) 00000-0000');

            $("#txtNIS").mask('000.00000.00-0');

        });

    </script>

    <script>

        //Consultar CEP

            const $campoCEP = document.getElementById("txtCEP");

            const $campoBairro = document.getElementById("txtBairro");

            const $campoRua = document.getElementById("txtRua");

            $campoCEP.addEventListener("blur", infosDoEvento => {

                const cep = infosDoEvento.target.value;

                if($campoCEP.value != '')

                {

                    fetch('https://viacep.com.br/ws/'+cep+'/json/')

                    .then((RetornoDoServidor) => {

                        return RetornoDoServidor.json();

                    })

                    .then((objJS) => {

                        if(objJS.erro == 'true')

                        {

                            $campoBairro.value = '';

                            $campoRua.value = '';

                        }else

                        {

                            //07868150

                        console.log(objJS);

                        $campoBairro.value = objJS.bairro;

                        $campoRua.value = objJS.logradouro;

                        }

                    })

                }

                else

                {

                    $campoBairro.value = '';

                    $campoRua.value = '';

                }

            });

    </script>

    <script>

        //NIS

        $('#chkNIS').change(function() {

            if(this.checked)

            {

                $('#txtNIS').prop('disabled',false);

            }

            else

            {

                $('#txtNIS').prop('disabled',true);

                $('#txtNIS').val(null);

            }

        });

    </script>

    <!--

    <script>

        //Protetor de Animais

        $('#chkProtetor').change(function()

        {

            if(this.checked)

            {

                $('#btnProtetorUpload').prop('disabled', false);

                $('#labelProtetor').removeClass("disabled");

            }

            else

            {

                $('#btnProtetorUpload').prop('disabled', true);

                $('#labelProtetor').addClass("disabled");

            }

        });

    </script>

    -->

    <script type="text/javascript">

        //confirme a senha

        $("form").submit(function(){

            if($("#txtSenha").val() != $("#txtConfirmaSenha").val())

            {

                event.preventDefault();

                $("#avisoIgualdade").show();

            }

            else{$("#avisoIgualdade").hide();}

            if(($("#txtSenha").val().length <= 5) && ($("#txtConfirmaSenha").val().length <= 5))

            {

                event.preventDefault();

                $("#avisoComprimento").show();

            }

            else {$("#avisoComprimento").hide();}

        });

    </script>

</body>

</html>

**cadClinica**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-danger container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Cadastrar Clínica</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <form action="cadastrar-clinica" class="p-sm-3 p-md-3 p-lg-4 p-3 px-0 row m-0" method="POST">

                            <div class="col-md-6 mb-lg-0 mb-5 p-0 pe-md-3">

                                <div class="form-group row-3">

                                    <label for="txtNome">Nome:</label>

                                    <input type="text" id="txtNome" name="txtNome" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group row-3">

                                    <label for="txtEmail">E-mail:</label>

                                    <input type="email" id="txtEmail" name="txtEmail" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group row-3">

                                    <label for="txtCNPJ">CNPJ:</label>

                                    <input type="text" id="txtCNPJ" name="txtCNPJ" class="form-control" placeholder="" required>

                                </div>

                                <div class="row">

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtTelefone">Telefone:</label>

                                            <input type="text" id="txtTelefone" name="txtTelefone" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtVagas">Vagas:</label>

                                            <input type="number" id="txtVagas" name="txtVagas" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                </div>

                            </div>

                            <div class="col-md-6 p-0 ps-md-3">

                                <div class="row">

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtCEP">CEP:</label>

                                            <input type="text" id="txtCEP" name="txtCEP" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtNumero">Número:</label>

                                            <input type="text" id="txtNumero" name="txtNumero" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                </div>

                                <div class="form-group">

                                    <label for="txtBairro">Bairro:</label>

                                    <input type="text" id="txtBairro" name="txtBairro" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group">

                                    <label for="txtRua">Rua:</label>

                                    <input type="text" id="txtRua" name="txtRua" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group">

                                    <label for="txtSenha">Crie uma Senha:</label>

                                    <input type="password" id="txtSenha" name="txtSenha" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group">

                                    <label for="txtConfSenha">Confirme a Senha</label>

                                    <input type="password" id="txtConfSenha" name="txtConfSenha" class="form-control" placeholder="" required>

                                    <div class="text-danger" id="avisoIgualdade" style="display:none;">as Senhas devem ser iguais\*</div>

                                    <div class="text-danger" id="avisoComprimento" style="display:none;">A senha deve conter mais que 5 dígitos\*</div>

                                </div>

                                <div class="row">

                                    <div class="form-group">

                                        <input type="submit" class="btn btn-success float-end mt-4" value="Cadastrar">

                                    </div>

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-adm';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/popper.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <!-- EXTENSÃO JQUERY DAS MASCARAS -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.16/jquery.mask.min.js"></script>

    <script>

        //Adicionar Máscaras

        $(document).ready(function (){

            $("#txtCNPJ").mask('00.000.000/0000-00');

            $("#txtCEP").mask('00000-000');

            $("#txtTelefone").mask('(00) 0000 0000');

        });

    </script>

    <script>

        //Consultar CEP

            const $campoCEP = document.getElementById("txtCEP");

            const $campoBairro = document.getElementById("txtBairro");

            const $campoRua = document.getElementById("txtRua");

            $campoCEP.addEventListener("blur", infosDoEvento => {

                const cep = infosDoEvento.target.value;

                if($campoCEP.value != '')

                {

                    fetch('https://viacep.com.br/ws/'+cep+'/json/')

                    .then((RetornoDoServidor) => {

                        return RetornoDoServidor.json();

                    })

                    .then((objJS) => {

                        if(objJS.erro == 'true')

                        {

                            $campoBairro.value = '';

                            $campoRua.value = '';

                        }else

                        {

                            //07868150

                        console.log(objJS);

                        $campoBairro.value = objJS.bairro;

                        $campoRua.value = objJS.logradouro;

                        }

                    })

                }

                else

                {

                    $campoBairro.value = '';

                    $campoRua.value = '';

                }

            });

    </script>

    <script type="text/javascript">

        //confirme a senha

        $("form").submit(function(){

            if($("#txtSenha").val() != $("#txtConfSenha").val())

            {

                event.preventDefault();

                $("#avisoIgualdade").show();

            }

            else{$("#avisoIgualdade").hide();}

            if(($("#txtSenha").val().length <= 5) && ($("#txtConfSenha").val().length <= 5))

            {

                event.preventDefault();

                $("#avisoComprimento").show();

            }

            else {$("#avisoComprimento").hide();}

        });

    </script>

</body>

</html>

**cadRaca**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-danger container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Cadastrar Raça</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <form action="cadastrar-raca" class="p-sm-3 p-md-3 p-lg-4 p-3 px-0" method="POST">

                            <label for="Especie">Espécie:</label>

                            <select name="tipoEspecie" id="Especie" class="form-select">

                                <option value="1">Felina</option>

                                <option value="0">Canina</option>

                            </select>

                            <small class="form-text text-muted">Selecione a Espécie dessa Raça</small>

                            <div class="form-group">

                                <label for="raca">Raça:</label>

                                <input type="text" id="raca" name="txtRaca" class="form-control" placeholder="ex: Siamês, Persa, Sphynx " required>

                                <small class="form-text text-muted">Informe o nome da raça que deseja cadastrar</small>

                            </div>

                            <div class="text-center">

                                <input type="submit" class="btn btn-success align-middle" value="Cadastrar">

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-adm';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**confirmarCodigoSenha**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Codigo de recuperação</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white">

                        <form action="<?php echo URL.'confirmar-recuperacao';?>" class="p-sm-3 p-md-3 p-lg-4 p-3 px-0 row m-0" method="post">

                            <?php

                                //exibindo mensagem de erro

                                if(isset($\_COOKIE["msg"]))

                                {

                                    echo "<div class='alert alert-danger' role='alert'>

                                        ".$\_COOKIE['msg']."

                                    </div>";

                                }

                                //excluindo cookie de erro

                                setcookie("msg","",time() - 3600);

                            ?>

                            <input type="hidden" name="txtEmail" value="<?php echo $email;?>">

                            <div class="form-group mb-3">

                                <label for="txtCod" class="form-label">Codigo de confirmação:</label>

                                <input type="text" name="txtCod" id="txtCod" class="form-control">

                            </div>

                            <div class="form-group">

                                <input type="submit" class="btn btn-success float-end" value="Confirmar">

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'#';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**confirmaSolicitacao**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <?php

            if($\_SESSION["dadosLogin"]->nivelacesso == 2){echo"<div class='bg-danger container-fluid' style='grid-area: corpo;''>";}

            else{echo"<div class='bg-warning container-fluid' style='grid-area: corpo;''>";}

        ?>

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Solicitação</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white">

                        <?php

                        if($\_SESSION["dadosLogin"]->nivelacesso == 2){echo"<form action='".URL."agendar-clinica' class='p-sm-3 p-md-3 p-lg-4 p-3 px-0 row m-0' method='post'>";}

                        else{echo"<form action='".URL."agendar' class='p-sm-3 p-md-3 p-lg-4 p-3 px-0 row m-0' method='post'>";}?>

                            <input type="hidden" name="emailDestinatario" id="emailDestinatario" value="<?php echo $dadosCastracao->email;?>">

                            <input type="hidden" name="nomeDestinatario" id="nomeDestinatairio" value="<?php echo $dadosCastracao->nome;?>">

                            <div class="col-sm-6 px-3">

                                <div class="row container-fluid p-0 m-0 form-group mb-2">

                                    <label for="" class="form-label">Número da solicitação:</label>

                                    <input type="text" name="idcastracao" readonly class="form-control" value="<?php echo $dadosCastracao->idcastracao;?>">

                                </div>

                                <div class="row container-fluid p-0 m-0 form-group mb-2">

                                    <label for="" class="form-label">Nome do animal: </label>

                                    <input type="text" name="aninome" readonly class="form-control" value="<?php echo $dadosCastracao->aninome;?>">

                                </div>

                                <div class="row container-fluid p-0 m-0 form-group mb-2">

                                    <label for="" class="form-label">CPF do responsável:</label>

                                    <input type="text" name="cpf" readonly class="form-control" value="<?php echo $dadosCastracao->cpf;?>">

                                </div>

                                <div class="row container-fluid p-0 m-0 form-group mb-2 ">

                                    <label for="" class="form-label">Observações do tutor: </label>

                                    <textarea class="form-control" name="txtarea" readonly><?php echo $dadosCastracao->observacao;?></textarea>

                                </div>

                            </div>

                            <div  class="col-sm-6 px-3">

                                <div class="row w-100 justify-content-center m-0">

                                    <img src="<?php echo URL."recursos/img/Animais/$dadosCastracao->foto";?>" class="mb-3" style="max-height:200px; width:auto;">

                                    <div class="form-group p-0">

                                        <?php

                                        if($\_SESSION["dadosLogin"]->nivelacesso == 2)

                                        {

                                            echo"

                                                <label for='selectClinica' class='form-label'>Selecione a Clínica</label>

                                                <select name='selectClinica' id='selectClinica' class='form-select'>

                                                    <option value='0' selected>-- SELECIONE UMA CLÍNICA --</option>

                                                        ";

                                                            foreach($dadosClinicas as $values)

                                                            {

                                                                echo "<option value='$values->idclinica'>$values->nome - Vagas Disponíveis: $values->vagas</option>";

                                                            }

                                            echo"</select>";

                                        }

                                        else

                                        {

                                            echo"

                                                <label for='horario' class='form-label'>Selecione a data e hora da castracao</label>

                                                <input type='datetime-local' name='horario' id='horario' class='form-control' required>

                                            ";

                                        }

                                        ?>

                                   </div>

                                </div>

                                <div class="row float-end my-3">

                                    <div class="">

                                        <?php

                                            if($\_SESSION["dadosLogin"]->nivelacesso == 2)

                                            {

                                                echo"<!-- Botão recusar solicitação -->

                                                <input type='submit' id='btnrecusa' value='Recusar' name='btnRecusa' class='btn btn-danger float-end me-3'>";

                                                echo"

                                                <!-- Botão confirmar solicitação -->

                                                <input type='submit' value='Confirmar' class='btn btn-success float-end me-1'>";

                                            }

                                            else

                                            {

                                                echo"

                                                <!-- Botão confirmar solicitação -->

                                                <input type='submit' value='Confirmar' class='btn btn-success float-end me-1'>";

                                            }

                                        ?>

                                    </div>

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'lista-solicitacao';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/popper.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

</body>

</html>

**consultaAnimais**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-danger container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">

                            Animais do tutor

                        </h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white">

                    <?php

                        foreach ($dadosAnimal as $value)

                        {

                            //Reescrevendo a espécie

                            $valorEspecie = $value->especie;

                            $value->especie = str\_replace("0","Canina", $value->especie);

                            $value->especie = str\_replace("1","Felina", $value->especie);

                            //Reescrevendo o sexo

                            $valorSexo = $value->sexo;

                            $value->sexo = str\_replace("0","Fêmea", $value->sexo);

                            $value->sexo = str\_replace("1","Macho", $value->sexo);

                            //Reescrevendo a pelagem

                            $valorPelagem = $value->pelagem;

                            $value->pelagem = str\_replace("0","Curta", $value->pelagem);

                            $value->pelagem = str\_replace("1","Média", $value->pelagem);

                            $value->pelagem = str\_replace("2","Alta", $value->pelagem);

                            //Reescrevendo o porte

                            $valorPorte = $value->porte;

                            $value->porte = str\_replace("0","Pequeno", $value->porte);

                            $value->porte = str\_replace("1","Médio", $value->porte);

                            $value->porte = str\_replace("2","Grande", $value->porte);

                            //Reescrevendo o Comunitário

                            $valorComunitario = $value->comunitario;

                            $value->comunitario = str\_replace("0","Não", $value->comunitario);

                            $value->comunitario = str\_replace("1","Sim", $value->comunitario);

                            echo

                            "<!-- Começo de um animal -->

                                <div class='row mt-3'>

                                    <div class='col-md-3 d-flex align-items-center'>

                                        <img src='".URL."recursos/img/Animais/$value->foto' alt='Imagem' class='mw-100' style='height: 200px; width: 300px;'>

                                    </div>

                                    <div class='col-md-7'>

                                        <div class='row'>

                                            <div class='col-md-9'>

                                                <div class='row'>

                                                    <p>

                                                        Nome:

                                                        ".$value->aninome."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Espécie:

                                                        ".$value->especie."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Sexo:

                                                        ".$value->sexo."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Pelagem:

                                                        ".$value->pelagem."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Porte:

                                                        ".$value->porte."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p class='mb-md-0'>

                                                        Animal Comunitário:

                                                        ".$value->comunitario."

                                                    </p>

                                                </div>

                                            </div>

                                            <div class='col-md-3'>

                                                <div class='row'>

                                                    <p>

                                                        Idade:

                                                        ".$value->idade."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Cor:

                                                        ".$value->cor."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p class='mb-0'>

                                                        Raça:

                                                        ".$value->raca."

                                                    </p>

                                                </div>

                                            </div>

                                        </div>

                                        <div class='col'></div>

                                    </div>

                                    <div class='col-md-2 mt-2 mt-md-0'>

                                    <button class='btn btn-warning w-100 mb-2' id='btnEditar' type='button' data-bs-target='#modalEditar' data-bs-toggle='modal'

                                            data-idanimal='$value->idanimal' data-idusuario='$value->idusuario' data-nome='$value->aninome' data-especie='$valorEspecie'

                                            data-sexo='$valorSexo' data-cor='$value->cor' data-raca='$value->idraca' data-idade='$value->idade' data-pelagem='$valorPelagem'

                                            data-porte='$valorPorte' data-comunitario='$valorComunitario' data-foto='$value->foto'>

                                            Editar animal

                                    </button>

                                        <a class='btn btn-danger float-end w-100' onclick='confirmar($value->idanimal,$value->idusuario,$value->foto)'><i class='fa fa-trash'></i> Excluir</a>

                                    </div>

                                </div>

                                <hr>

                            <!-- Fim de um animal -->

                            ";

                        }

                    ?>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'consulta-usuario';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

      <!-- MODAL: editar animal-->

        <div class="modal fade" id="modalEditar" tabindex="-1" data-bs-backdrop="static" data-bs-keyboard="false" aria-labelledby="staticBackdropLabel" aria-hidden="true" >

            <div class="modal-dialog modal-lg modal-dialog-centered">

                <div class="modal-content">

                    <form method="post" action="<?php echo URL.'atualizar-animal';?>" enctype="multipart/form-data">

                        <div class="modal-header">

                            <h5 class="modal-title">Editar</h5>

                            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close" ></button>

                        </div>

                        <div class="modal-body">

                            <input type="hidden" name="idanimal" id="idanimal">

                            <input type="hidden" name="idusuario" id="idusuario">

                            <div class="row">

                                <div class="col-md">

                                    <div class="row">

                                        <div class="col-12 mb-2">

                                            <label for="txtNome" class="form-label">Nome do Animal:</label>

                                            <input type="text" class="form-control" id="txtNome" name="txtNome" maxlength="50" required>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="tipoEspecie" class="form-label">Espécie:</label>

                                            <select id="tipoEspecie" name="tipoEspecie" class="form-select" onchange="carregarRaca(this)" required>

                                                <option>... SELECIONE A ESPÉCIE ...</option>

                                                <option value="0">Canina</option>

                                                <option value="1">Felina</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="numIdade" class="form-label">Idade:</label>

                                            <input type="number" class="form-control" id="numIdade" name="numIdade" min="0" max="100" required>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="slcSexo" class="form-label">Sexo:</label>

                                            <select id="slcSexo" name="slcSexo" class="form-select" required>

                                                <option value="0">Fêmea</option>

                                                <option value="1">Macho</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPelagem" class="form-label">Pelagem:</label>

                                            <select id="slcPelagem" name="slcPelagem" class="form-select" required>

                                                <option value="0">Curta</option>

                                                <option value="1">Média</option>

                                                <option value="2">Alta</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="txtCor" class="form-label">Cor:</label>

                                            <input type="text" class="form-control" id="txtCor" name="txtCor" maxlength="30" required>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPorte" class="form-label">Porte:</label>

                                            <select id="slcPorte" name="slcPorte" class="form-select" required>

                                                <option value="0">Pequeno</option>

                                                <option value="1">Médio</option>

                                                <option value="2">Grande</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row mb-2">

                                        <div class="col-md-6 mb-2">

                                            <label for="listRaca" class="form-label">Raça:</label>

                                            <select name="racas" id="racas" class="form-select" required>

                                                <option value="">... SELECIONE A RAÇA ...</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcComunitario" class="form-label">Animal Comunitário:</label>

                                            <select id="slcComunitario" name="slcComunitario" class="form-select" required>

                                                <option value="0">Não</option>

                                                <option value="1">Sim</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row text-center mb-3 aling-content-center">

                                        <div class="col-lg-6 justify-content-center aling-content-center">

                                            <label>Foto do animal:</label>

                                            <label id="labelImgAnimal" for="inputImgAnimal" class="form-label"></label>

                                            <img src="recursos/img/" alt="Foto do Animal" id="imgAnimal" for="inputImgAnimal" style="width:100%;">

                                        </div>

                                        <div class="col-lg-6 justify-content-center aling-content-center">

                                            <label id="labelImgAnimal" for="inputImgAnimal" class="form-label">Trocar foto:</label>

                                            <input type="file" name="foto" id="inputImgAnimal" class="form-control" accept="image/\*">

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <div class="modal-footer">

                            <button type="submit" class="btn btn-success">Editar</button>

                            <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Cancelar</button>

                        </div>

                    </form>

                </div>

            </div>

        </div>

        <!--/MODAL -->

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/popper.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- JS SweetAlert 2-->

    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script src="sweetalert2.all.min.js"></script>

    <!-- EXTENSÃO JQUERY PARA O AJAX -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js" integrity="sha512-894YE6QWD5I59HgZOGReFYm4dnWc1Qt5NtvYSaNcOP+u1T9qYdvdihz0PPSiiqn/+/3e7Jo4EaG7TubfWGUrMQ==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

    <!-- ABRIR MODAL EDITAR -->

    <script>

        //Definindo os valores nos inputs da modal

        var modal = document.getElementById('modalEditar')

        modal.addEventListener('show.bs.modal', function (event) {

            // Button that triggered the modal

            var button = event.relatedTarget

            // Extract info from data-bs-\* attributes

            var idusuario = button.getAttribute('data-idusuario')

            var idanimal = button.getAttribute('data-idanimal')

            var nome = button.getAttribute('data-nome')

            var especie = parseInt(button.getAttribute('data-especie'))

            var sexo = parseInt(button.getAttribute('data-sexo'))

            var cor = button.getAttribute('data-cor')

            var idade = button.getAttribute('data-idade')

            var pelagem = parseInt(button.getAttribute('data-pelagem'))

            var foto = button.getAttribute('data-foto')

            if (foto == "")

            {

                foto = "imagem\_exemplo.jpg";

            }

            $(document).ready(function()

            {

                $("#racas").empty(); //limpar todos antes de carregar

                $.ajax({

                    url: '<?php echo URL;?>carregar-raca/'+ especie,

                    success: function(data) {

                        $("#racas").append(data);

                        try{

                            $("#racas option").filter("[value="+raca+"]").attr("selected",true)

                        }

                        catch{}

                    }

                });

            })

            var raca = button.getAttribute('data-raca')

            var porte = button.getAttribute('data-porte')

            var comunitario = button.getAttribute('data-comunitario')

            $("#idusuario").val(idusuario)

            $("#idanimal").val(idanimal)

            $("#txtNome").val(nome)

            $("#tipoEspecie option").filter("[value="+especie+"]").attr("selected",true)

            $("#slcSexo option").filter("option[value="+sexo+"]").attr("selected",true)

            $("#slcPelagem option").filter("option[value="+pelagem+"]").attr("selected",true)

            $("#slcPorte option").filter("option[value="+porte+"]").attr("selected",true)

            $("#slcComunitario option").filter("option[value="+comunitario+"]").attr("selected",true)

            $("#txtCor").val(cor)

            $("#numIdade").val(idade)

            $("#imgAnimal").prop("src","<?php echo URL.'recursos/img/Animais/';?>"+foto);

        })

    </script>

    <script>

        //Carregar raças

        function carregarRaca(id)

        {

            //limpar todos antes de carregar

            $("#racas").empty();

            $.ajax({

                url: '<?php echo URL;?>carregar-raca/'+ id.value,

                success: function(data) {

                    $("#racas").append(data);

                }

            });

        }

    </script>

    <!-- SCRIPT CONFIRMAÇÃO PARA EXCLUIR O ANIMAL -->

    <script>

        function confirmar(idani,idusu,foto)

        {

            Swal.fire({

                title: 'Você tem certeza que deseja excluir?',

                text: "Você não será capaz de desfazer esta ação!",

                icon: 'warning',

                showCancelButton: true,

                confirmButtonColor: '#3085d6',

                cancelButtonColor: '#d33',

                confirmButtonText: 'Excluir',

                cancelButtonText: 'Cancelar'

                }).then((result) => {

                if (result.isConfirmed) {

                    Swal.fire(

                        'Excluído!', //title:

                        'Animal apagado com sucesso.', //text:

                        'success', //icon:

                    ).then(()=> {

                        window.location='<?php echo URL;?>excluir-animal/'+idani+'/'+idusu+'/'+foto;

                        }

                    )}

            })

        }

    </script>

</body>

</html>

**consultaCastracao**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <!-- DataTables -->

    <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.1.3/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/buttons/2.2.3/css/buttons.dataTables.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.12.1/css/dataTables.bootstrap5.min.css">

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <?php

            if($\_SESSION["dadosLogin"]->nivelacesso == 1)

            {

                echo "<div class='bg-warning container-fluid' style='grid-area: corpo;'>";

            }

            else

            {

                echo "<div class='bg-danger container-fluid' style='grid-area: corpo;'>";

            }

        ?>

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Consultar Castrações</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white table-responsive">

                        <table id="tbCastracao" class="table table-hover">

                            <?php

                                if($\_SESSION["dadosLogin"]->nivelacesso == 2)

                                {

                                    echo"

                                        <thead>

                                            <tr>

                                                <th>#</th>

                                                <th>Foto</th>

                                                <th>Animal</th>

                                                <th>Nome do tutor</th>

                                                <th>CPF do tutor</th>

                                                <th>Clínica</th>

                                                <th>Data</th>

                                                <th>Hora</th>

                                                <th>Status</th>

                                                <th>Observação</th>

                                                <th>Observação CLinica</th>

                                                <th>Ações</th>

                                            </tr>

                                        </thead>

                                        <tbody>

                                        ";

                                    foreach($dadosCastracao as $value)

                                    {

                                        $value->status = str\_replace("0", "Solicitação em análise", $value->status);

                                        $value->status = str\_replace("1", "Solicitação aprovada", $value->status);

                                        $value->status = str\_replace("2", "Animal castrado", $value->status);

                                        $value->status = str\_replace("3", "Solicitação reprovada", $value->status);

                                        $value->status = str\_replace("4", "Tutor não compareceu", $value->status);

                                        $value->status = str\_replace("5", "Castração cancelada", $value->status);

                                        $value->status = str\_replace("6", "Reagendar castração", $value->status);

                                        $value->status = str\_replace("7", "Animal foi a óbito", $value->status);

                                        $value->status = str\_replace("8", "Animal Castrado (Porém suas informações devem ser editadas)", $value->status);

                                        $data = is\_null($value->horario) ? '-' : date('d/m/Y', strtotime($value->horario));

                                        $hora = is\_null($value->horario) ? '-' : date('H:i', strtotime($value->horario));

                                        $value->observacao = preg\_replace("/^$/", "-", $value->observacao);

                                        $value->obsclinica = preg\_replace("/^$/", "-", $value->obsclinica);

                                        echo

                                        "

                                        <tr>

                                            <td>$value->idcastracao</td>

                                            <td><img width='300px' class='img-thumbnail' src='".URL."recursos/img/Animais/$value->foto'></td>

                                            <td>$value->aninome</td>

                                            <td>$value->nometutor</td>

                                            <td><a href=". URL . "consulta-usuario/$value->cpf" .">$value->cpf</a></td>

                                            <td><a href=". URL . "consulta-clinica/$value->cnpj" . ">$value->nomeclinica</a></td>

                                            <td>$data</td>

                                            <td>$hora</td>

                                            <td>$value->status</td>

                                            <td>$value->observacao</td>

                                            <td>$value->obsclinica</td>

                                            <td>

                                            <!--<button type='button' class='btn btn-warning' data-bs-toggle='modal' data-bs-target='#modalAtualizar' data-idcastracao='$value->idcastracao'>

                                                Editar

                                            </button>-->

                                            <a class='btn btn-danger' onClick='confirmar($value->idcastracao)'>Excluir</a>

                                            </td>

                                        </tr>

                                        ";

                                    }

                                }

                                else if ($\_SESSION["dadosLogin"]->nivelacesso == 1)

                                {

                                    echo"

                                        <thead>

                                            <tr>

                                                <th>#</th>

                                                <th>Foto</th>

                                                <th>Animal</th>

                                                <th>Nome do tutor</th>

                                                <th>CPF do tutor</th>

                                                <th>Telefone</th>

                                                <th>Celular</th>

                                                <th>Data</th>

                                                <th>Hora</th>

                                                <th>Status</th>

                                                <th>Observação</th>

                                                <th>Ações</th>

                                            </th>

                                        </thead>

                                        <tbody>

                                        ";

                                    foreach($dadosCastracaoClinica as $value)

                                    {

                                        $value->status = str\_replace("0", "Solicitação em análise", $value->status);

                                        $value->status = str\_replace("1", "Solicitação aprovada", $value->status);

                                        $value->status = str\_replace("2", "Animal castrado", $value->status);

                                        $value->status = str\_replace("3", "Solicitação reprovada", $value->status);

                                        $value->status = str\_replace("4", "Tutor não compareceu", $value->status);

                                        $value->status = str\_replace("5", "Castração cancelada", $value->status);

                                        $value->status = str\_replace("6", "Reagendar castração", $value->status);

                                        $value->status = str\_replace("7", "Animal foi a óbito", $value->status);

                                        $value->status = str\_replace("8", "Animal Castrado (Suas informações devem ser editadas)", $value->status);

                                        $value->observacao = preg\_replace("/^$/", "-", $value->observacao);

                                        echo

                                        "

                                        <tr>

                                            <td>$value->idcastracao</td>

                                            <td><img width='300px' class='img-thumbnail' src='".URL."recursos/img/Animais/$value->foto'></td>

                                            <td>$value->aninome</td>

                                            <td>$value->nometutor</td>

                                            <td>$value->cpf</td>

                                            <td>$value->telefone</td>

                                            <td>$value->celular</td>

                                            <td>". date('d/m/Y',strtotime($value->horario)) ."</td>

                                            <td>". date('H:i',strtotime($value->horario)) ."</td>

                                            <td>$value->status</td>

                                            <td>$value->observacao</td>

                                            <td>

                                                <button type='button' class='btn btn-warning' data-bs-toggle='modal' data-bs-target='#modalAtualizar'

                                                        data-idcastracao='$value->idcastracao' data-emailTutor='$value->email' data-idTutor='$value->idusuario'

                                                        data-nomeTutor='$value->nometutor' data-foto='$value->foto' data-nomeAnimal='$value->aninome'

                                                        data-status='$value->status' data-dataCastracao='$value->horario'>

                                                    Editar

                                                </button>

                                            </td>

                                        </tr>

                                        ";

                                    }

                                }

                            ?>

                            </tbody>

                        </table>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <?php if($\_SESSION["dadosLogin"]->nivelacesso == 1){echo "<a href='".URL."home-clinica' class='btn btn-success'>Voltar</a>";}?>

                    <?php if($\_SESSION["dadosLogin"]->nivelacesso == 2){echo "<a href='".URL."home-adm' class='btn btn-success'>Voltar</a>";}?>

                </div>

            </div>

        </div>

        <?php

        //MODAL DA CLÍNICA

            if($\_SESSION["dadosLogin"]->nivelacesso == 1){

                echo"

                <div class='modal fade' id='modalAtualizar' data-bs-keyboard='true' tabindex='-1' aria-labelledby='staticBackdropLabel' aria-hidden='true'>

                    <div class='modal-dialog modal-dialog-centered'>

                        <div class='modal-content'>

                            <form action='".URL."atualizar-castracao' method='post'>

                                <input type='hidden' id='idCastracao' name='idCastracao'>

                                <input type='hidden' id='status' name='status'>

                                <input type='hidden' id='idTutor' name='idTutor'>

                                <input type='hidden' id='emailTutor' name='emailTutor'>

                                <input type='hidden' id='nomeAnimal' name='nomeAnimal'>

                                <input type='hidden' id='nomeTutor' name='nomeTutor'>

                                <input type='hidden' id='dataCastracao' name='dataCastracao'>

                                <div class='modal-header'>

                                    <h5 class='modal-title' id='staticBackdropLabel'>Editar castração</h5>

                                    <button type='button' class='btn-close' data-bs-dismiss='modal' aria-label='Close'></button>

                                </div>

                                <div class='modal-body'>

                                    <div class='form-group mb-4'>

                                        <label for='statusAtualizado' class='label-form'>Status da castração</label>

                                        <select name='statusAtualizado' id='statusAtualizado' class='form-select'>

                                            <option value=''>-- Atualize o status --</option>

                                            <option value='2'>Animal Castrado</option>

                                            <option value='4'>Tutor não compareceu</option>

                                            <option value='5'>Castração cancelada</option>

                                            <option value='6'>Reagendar castração</option>

                                            <option value='7'>Animal foi a óbito</option>

                                            <option value='8'>Animal Castrado (Suas informações devem ser editadas)</option>

                                        </select>

                                        <small class='form-text text-muted'>Coloque em \"Animal Castrado (Suas informações devem ser editadas)\",

                                                somente se alguma informação estiver incorreta!</small>

                                    </div>

                                    <div class='form-group mb-4'>

                                        <label class='form-label' for='codChip'>Número do chip:</label>

                                        <input type='text' name='codChip' id='codChip' class='form-control' maxlength='15'>

                                    <div>

                                    <div class='form-group'>

                                        <label class='form-label' for='obsClinica'>Observação:</label>

                                        <textarea name='obsClinica' id='obsClinica' rows='5' class='form-control'></textarea>

                                    </div>

                                </div>

                                <div class='modal-footer'>

                                    <button type='submit' class='btn btn-primary'>Editar</button>

                                    <button type='button' class='btn btn-secondary' data-bs-dismiss='modal'>Cancelar</button>

                                </div>

                            </form>

                        </div>

                    </div>

                </div>";

            }

        //MODAL DO ADM

            /\*else if($\_SESSION["dadosLogin"]->nivelacesso == 2){

                echo"

                <div class='modal fade' id='modalAtualizar' data-bs-keyboard='true' tabindex='-1' aria-labelledby='staticBackdropLabel' aria-hidden='true'>

                    <div class='modal-dialog modal-dialog-centered'>

                        <div class='modal-content'>

                            <form action='".URL."atualizar-castracao' method='post'>

                                <input type='hidden' id='idCastracao' name='idCastracao'>

                                <div class='modal-header'>

                                    <h5 class='modal-title' id='staticBackdropLabel'>Editar castração</h5>

                                    <button type='button' class='btn-close' data-bs-dismiss='modal' aria-label='Close'></button>

                                </div>

                                <div class='modal-body'>

                                    <label class='form-label' for='obsCastracao'>Observação: (opcional)</label>

                                    <textarea name='obsCastracao' id='obsCastracao' rows='5' class='form-control' disabled></textarea>

                                </div>

                                <div class='modal-footer'>

                                    <button type='submit' class='btn btn-primary'>Editar</button>

                                    <button type='button' class='btn btn-secondary' data-bs-dismiss='modal'>Cancelar</button>

                                </div>

                            </form>

                        </div>

                    </div>

                </div>";

            }\*/

        ?>

        <!-- /MODAL -->

        <!-- /CORPO -->

    </div>

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- DataTables -->

    <script type='text/javascript' charset='utf8' src='https://code.jquery.com/jquery-3.5.1.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/1.11.5/js/jquery.dataTables.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/1.12.1/js/dataTables.bootstrap5.min.js'></script>

    <!-- JS SweetAlert 2-->

    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script src="sweetalert2.all.min.js"></script>

    <?php

        if($\_SESSION["dadosLogin"]->nivelacesso == 2)

        {

            echo"

            <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.2/js/dataTables.buttons.min.js'></script>

            <script type='text/javascript' charset='utf8' src='https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js'></script>

            <script type='text/javascript' charset='utf8' src='https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js'></script>

            <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js'></script>

            <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.2/js/buttons.print.min.js'></script>

            <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.3/js/buttons.colVis.min.js'></script>

            <script>

                $(document).ready(function() {

                    $('#tbCastracao').DataTable( {

                        dom: 'Bfrtip',

                        buttons: [

                            'colvis',

                            {

                                extend:'csv',

                                exportOptions:{

                                    columns: ':visible'

                                }

                            },

                            {

                                extend: 'excel',

                                exportOptions: {

                                    columns: ':visible'

                                }

                            },

                            {

                                extend: 'print',

                                exportOptions:{

                                    columns: ':visible'

                                }

                            }

                        ],

                        'language': {

                            'url': '//cdn.datatables.net/plug-ins/1.11.5/i18n/pt-BR.json'

                        },

                        columnDefs: [

                            {

                                targets: 1,

                                className: 'noVis'

                            }

                        ]

                    } );

                } );

            </script>

            ";

        }

        else

        {

            echo "

                <script>

                    $(document).ready(function() {

                        $('#tbCastracao').DataTable( {

                            dom: 'Bfrtip',

                            buttons: [

                                'csv', 'excel', 'print'

                            ],

                            'language': {

                                'url': '//cdn.datatables.net/plug-ins/1.11.5/i18n/pt-BR.json'

                            }

                        } );

                    } );

                </script>

            ";

        }

    ?>

    <script>

        //Definindo valores aos inputs da modal

        var modal = document.getElementById('modalAtualizar')

        modal.addEventListener('show.bs.modal', function (event) {

        // Button that triggered the modal

        var button = event.relatedTarget

        // Extract info from data-bs-\* attributes

        var idCastracao = button.getAttribute('data-idcastracao')

        var emailTutor = button.getAttribute('data-emailTutor')

        var idTutor = button.getAttribute('data-idTutor')

        var nomeAnimal = button.getAttribute('data-nomeAnimal')

        var nomeTutor = button.getAttribute('data-nomeTutor')

        var status = button.getAttribute('data-status')

        var dataCastracao = button.getAttribute('data-dataCastracao')

        $("#idCastracao").val(idCastracao)

        $("#emailTutor").val(emailTutor)

        $("#idTutor").val(idTutor)

        $("#nomeAnimal").val(nomeAnimal)

        $("#nomeTutor").val(nomeTutor)

        $("#status").val(status)

        $("#dataCastracao").val(dataCastracao)

        })

    </script>

    <?php

        if($\_SESSION["dadosLogin"]->nivelacesso == 2)

        {

            echo

            "

            <!-- SCRIPT CONFIRMAÇÃO PARA EXCLUIR A CASTRAÇÃO -->

            <script>

                function confirmar(id)

                {

                    Swal.fire({

                        title: 'Você tem certeza que deseja excluir?',

                        text: 'Você não será capaz de desfazer esta ação!',

                        icon: 'warning',

                        showCancelButton: true,

                        confirmButtonColor: '#3085d6',

                        cancelButtonColor: '#d33',

                        confirmButtonText: 'Excluir',

                        cancelButtonText: 'Cancelar'

                        }).then((result) => {

                        if (result.isConfirmed) {

                            Swal.fire(

                                'Excluído!', //title:

                                'Castração apagada com sucesso.', //text:

                                'success', //icon:

                            ).then(()=> {

                                window.location='<?php echo URL;?>excluir-raca/'+id;

                                }

                            )}

                    })

                }

            </script>

            ";

        }

    ?>

</body>

</html>

**consultaClinica**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <!-- DataTables -->

    <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.1.3/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/buttons/2.2.3/css/buttons.dataTables.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.12.1/css/dataTables.bootstrap5.min.css">

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

        <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-danger container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container-fluid bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Consultar Clínica</h5>

                    </div>

                    <div class="container-fluid p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white table-responsive">

                        <table id="tbClinica" class="table table-hover">

                            <thead>

                                <th>#</th>

                                <th>Nome</th>

                                <th>E-mail</th>

                                <th>CNPJ</th>

                                <th>Telefone</th>

                                <th>CEP</th>

                                <th>Bairro</th>

                                <th>Rua</th>

                                <th>Número</th>

                                <th>Vagas</th>

                                <th>Ativo</th>

                                <th>Ações</th>

                            </thead>

                            <tbody>

                                <?php

                                    foreach($dadosClinica as $value)

                                    {

                                        $value->clitelefone = preg\_replace("/^$/", "-", $value->clitelefone);

                                        $valorAtivo = $value->ativo;

                                        $value->ativo = str\_replace("1", "Ativo", $value->ativo);

                                        $value->ativo = str\_replace("0", "Inativo", $value->ativo);

                                        echo

                                        "

                                        <tr>

                                            <td>$value->idclinica</td>

                                            <td>$value->nome</td>

                                            <td>$value->email</td>

                                            <td>$value->cnpj</td>

                                            <td>$value->clitelefone</td>

                                            <td>$value->clicep</td>

                                            <td>$value->clibairro</td>

                                            <td>$value->clirua</td>

                                            <td>$value->clinumero</td>

                                            <td>$value->vagas</td>

                                            <td>$value->ativo</td>

                                            <td>

                                            <button type='button' class='btn btn-warning' data-bs-toggle='modal' data-bs-target='#modalAtualizar' data-idclinica='$value->idclinica'

                                                    data-idlogin='$value->idlogin' data-nome='$value->nome' data-email='$value->email' data-cnpj='$value->cnpj' data-telefone='$value->clitelefone'

                                                    data-vagas='$value->vagas' data-cep='$value->clicep' data-numero='$value->clinumero' data-bairro='$value->clibairro' data-rua='$value->clirua' data-ativo='$valorAtivo'>

                                                Editar

                                            </button>

                                            <a onclick='confirmar($value->idclinica, $value->idlogin)' class='btn btn-danger'>Excluir</a></td>

                                            </tr>

                                        ";

                                    }

                                ?>

                            </tbody>

                        </table>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-adm';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- Modal Atualizar Clinica -->

    <div class="modal fade" id="modalAtualizar" tabindex="-1" data-bs-keyboard="false" aria-labelledby="staticBackdropLabel" aria-hidden="true" >

        <div class="modal-dialog modal-lg modal-dialog-centered">

            <div class="modal-content">

                <form action="<?php echo URL . 'atualizar-clinica'; ?>" method="post">

                    <div class="modal-header">

                        <h5 class="modal-title">Editar clínica</h5>

                        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close" ></button>

                    </div>

                    <input type="hidden" name="idClinica" id="idClinica">

                    <input type="hidden" name="idLogin" id="idLogin">

                    <div class="modal-body">

                        <div class="row">

                            <div class="col-sm-6 mb-4">

                                <div class="form-group row-3">

                                    <label for="txtNome">Nome:</label>

                                    <input type="text" id="txtNome" name="txtNome" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group row-3">

                                    <label for="txtEmail">E-mail:</label>

                                    <input type="email" id="txtEmail" name="txtEmail" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group row-3">

                                    <label for="txtCNPJ">CNPJ:</label>

                                    <input type="text" id="txtCNPJ" name="txtCNPJ" class="form-control" placeholder="" required>

                                </div>

                                <div class="row">

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtTelefone">Telefone:</label>

                                            <input type="text" id="txtTelefone" name="txtTelefone" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtVagas">Vagas:</label>

                                            <input type="number" id="txtVagas" name="txtVagas" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                </div>

                                <div class="row">

                                    <label for="chkAtivo" class="form-label">Ativo:</label>

                                    <input type="checkbox" name="chkAtivo" id="chkAtivo" value="1" class="form-check">

                                </div>

                            </div>

                            <div class="col-md-6">

                                <div class="row">

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtCEP">CEP:</label>

                                            <input type="text" id="txtCEP" name="txtCEP" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                    <div class="col-6">

                                        <div class="form-group">

                                            <label for="txtNumero">Número:</label>

                                            <input type="text" id="txtNumero" name="txtNumero" class="form-control" placeholder="" required>

                                        </div>

                                    </div>

                                </div>

                                <div class="form-group">

                                    <label for="txtBairro">Bairro:</label>

                                    <input type="text" id="txtBairro" name="txtBairro" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group">

                                    <label for="txtRua">Rua:</label>

                                    <input type="text" id="txtRua" name="txtRua" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group">

                                    <label for="txtSenha">Crie uma Senha:</label>

                                    <input type="password" id="txtSenha" name="txtSenha" class="form-control" placeholder="" required>

                                </div>

                                <div class="form-group">

                                    <label for="txtConfSenha">Confirme a Senha</label>

                                    <input type="password" id="txtConfSenha" name="txtConfSenha" class="form-control" placeholder="" required>

                                    <div class="text-danger" id="avisoIgualdade" style="display:none;">as Senhas devem ser iguais\*</div>

                                    <div class="text-danger" id="avisoComprimento" style="display:none;">A senha deve conter mais que 5 dígitos\*</div>

                                </div>

                            </div>

                        </div>

                    </div>

                    <div class='modal-footer'>

                        <button type='submit' class='btn btn-primary'>Editar</button>

                        <button type='button' class='btn btn-secondary' data-bs-dismiss='modal'>Cancelar</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

    <!-- /Modal -->

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- DataTables -->

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.js"></script>

    <script type="text/javascript" charset="utf8" src="https://code.jquery.com/jquery-3.5.1.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/buttons/2.2.2/js/dataTables.buttons.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.print.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.12.1/js/dataTables.bootstrap5.min.js"></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.3/js/buttons.colVis.min.js'></script>

    <!-- JS SweetAlert 2-->

    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script src="sweetalert2.all.min.js"></script>

    <script>

        $(document).ready(function() {

            $('#tbClinica').DataTable( {

                dom: 'Bfrtip',

                buttons: [

                    'colvis',

                    {

                        extend:'csv',

                        exportOptions:{

                            columns: ':visible'

                        }

                    },

                    {

                        extend: 'excel',

                        exportOptions: {

                            columns: ':visible'

                        }

                    },

                    {

                        extend: 'print',

                        exportOptions:{

                            columns: ':visible'

                        }

                    }

                ],

                "language": {

                    "url": "//cdn.datatables.net/plug-ins/1.11.5/i18n/pt-BR.json"

                },

                "search": {

                    "search": "<?php if(!isset($cnpj)){$cnpj = '';} echo $cnpj;?>"

                },

            } );

        } );

    </script>

    <script>

        //Definindo os valores nos inputs da modal

        var exampleModal = document.getElementById('modalAtualizar')

        exampleModal.addEventListener('show.bs.modal', function (event) {

        // Button that triggered the modal

        var button = event.relatedTarget

        // Extract info from data-bs-\* attributes

        var idclinica = button.getAttribute('data-idclinica')

        var idLogin = button.getAttribute('data-idLogin')

        var email = button.getAttribute('data-email')

        var cnpj = button.getAttribute('data-cnpj')

        var telefone = button.getAttribute('data-telefone')

        var nome = button.getAttribute('data-nome')

        var vagas = button.getAttribute('data-vagas')

        var cep = button.getAttribute('data-cep')

        var numero = button.getAttribute('data-numero')

        var bairro = button.getAttribute('data-bairro')

        var rua = button.getAttribute('data-rua')

        var ativo = button.getAttribute('data-ativo')

        $("#idClinica").val(idclinica)

        $("#idLogin").val(idLogin)

        $("#txtNome").val(nome)

        $("#txtEmail").val(email)

        $("#txtCNPJ").val(cnpj)

        $("#txtTelefone").val(telefone)

        $("#txtVagas").val(vagas)

        $("#txtCEP").val(cep)

        $("#txtNumero").val(numero)

        $("#txtBairro").val(bairro)

        $("#txtRua").val(rua)

        if(ativo == 1)

        {

            $("#chkAtivo").prop("checked", true);

        }

        else

            $("#chkAtivo").prop("checked", false);

        })

    </script>

    <script type="text/javascript">

        //confirme a senha

        $("form").submit(function(){

            if($("#txtSenha").val() != $("#txtConfSenha").val())

            {

                event.preventDefault();

                $("#avisoIgualdade").show();

            }

            else{$("#avisoIgualdade").hide();}

            if(($("#txtSenha").val().length <= 5) && ($("#txtConfSenha").val().length <= 5))

            {

                event.preventDefault();

                $("#avisoComprimento").show();

            }

            else {$("#avisoComprimento").hide();}

        });

    </script>

    <!-- SCRIPT CONFIRMAÇÃO PARA EXCLUIR A CLÍNICA -->

    <script>

        function confirmar(idcli, idlog)

        {

            Swal.fire({

                title: 'Você tem certeza que deseja excluir?',

                text: "Você não será capaz de desfazer esta ação!",

                icon: 'warning',

                showCancelButton: true,

                confirmButtonColor: '#3085d6',

                cancelButtonColor: '#d33',

                confirmButtonText: 'Excluir',

                cancelButtonText: 'Cancelar'

                }).then((result) => {

                if (result.isConfirmed) {

                    Swal.fire(

                        'Excluído!', //title:

                        'Clínica apagado com sucesso.', //text:

                        'success', //icon:

                    ).then(()=> {

                        window.location='<?php echo URL;?>excluir-clinica/'+idcli+'/'+idlog;

                        }

                    )}

            })

        }

    </script>

</body>

</html>

**consultaRaca**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <!-- DataTables -->

    <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.1.3/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/buttons/2.2.3/css/buttons.dataTables.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.12.1/css/dataTables.bootstrap5.min.css">

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

        <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-danger container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container-fluid bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Consultar Raças</h5>

                    </div>

                    <div class="container-fluid p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white table-responsive">

                        <table id="tbRaca" class="table table-hover">

                            <thead>

                                <th>#</th>

                                <th>Nome</th>

                                <th>Espécie</th>

                                <th>Ações</th>

                            </thead>

                            <tbody>

                                <?php

                                    foreach($dadosRaca as $value)

                                    {

                                        $valorEspecie = $value->tipoespecie;

                                        $value->tipoespecie = str\_replace("0", "Canina", $value->tipoespecie);

                                        $value->tipoespecie = str\_replace("1", "Felina", $value->tipoespecie);

                                        $value->tipoespecie = str\_replace("2", "Felina/Canina", $value->tipoespecie);

                                        echo

                                        "

                                        <tr>

                                            <td>$value->idraca</td>

                                            <td>$value->raca</td>

                                            <td>$value->tipoespecie</td>

                                            <td>

                                            <button type='button' class='btn btn-warning' data-bs-toggle='modal' data-bs-target='#modalAtualizar' data-idraca='$value->idraca' data-raca='$value->raca' data-tipoespecie='$valorEspecie'>

                                                Editar

                                            </button>

                                            <a class='btn btn-danger' onclick='confirmar($value->idraca)'>Excluir</a></td>

                                            </tr>

                                        ";

                                    }

                                ?>

                            </tbody>

                        </table>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-adm';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- Modal Atualizar Raça -->

    <div class="modal fade" id="modalAtualizar" tabindex="-1" data-bs-keyboard="false" aria-labelledby="staticBackdropLabel" aria-hidden="true" >

        <div class="modal-dialog modal-lg modal-dialog-centered">

            <div class="modal-content">

                <form action="<?php echo URL . 'atualiza-raca'; ?>" method="post">

                    <div class="modal-header">

                        <h5 class="modal-title">Editar Raça</h5>

                        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close" ></button>

                    </div>

                    <input type="hidden" name="idRaca" id="idRaca">

                    <div class="modal-body">

                        <div class="row">

                            <div class="form-group">

                                <label for="Especie">Espécie:</label>

                                <select name="tipoEspecie" id="Especie" class="form-select">

                                    <option value="1">Felina</option>

                                    <option value="0">Canina</option>

                                </select>

                            </div>

                            <small class="form-text text-muted">Selecione a Espécie dessa Raça</small>

                            <div class="form-group">

                                <label for="raca">Raça:</label>

                                <input type="text" id="raca" name="txtRaca" class="form-control" placeholder="ex: Siamês, Persa, Sphynx " required>

                                <small class="form-text text-muted">Informe o novo nome da raça</small>

                            </div>

                        </div>

                    </div>

                    <div class='modal-footer'>

                        <button type='submit' class='btn btn-primary'>Editar</button>

                        <button type='button' class='btn btn-secondary' data-bs-dismiss='modal'>Cancelar</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

    <!-- /Modal -->

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- DataTables -->

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.js"></script>

    <script type="text/javascript" charset="utf8" src="https://code.jquery.com/jquery-3.5.1.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/buttons/2.2.2/js/dataTables.buttons.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.print.min.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.12.1/js/dataTables.bootstrap5.min.js"></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.3/js/buttons.colVis.min.js'></script>

    <!-- JS SweetAlert 2-->

    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script src="sweetalert2.all.min.js"></script>

    <script>

        $(document).ready(function() {

            $('#tbRaca').DataTable( {

                dom: 'Bfrtip',

                buttons: [

                    'colvis',

                    {

                        extend:'csv',

                        exportOptions:{

                            columns: ':visible'

                        }

                    },

                    {

                        extend: 'excel',

                        exportOptions: {

                            columns: ':visible'

                        }

                    },

                    {

                        extend: 'print',

                        exportOptions:{

                            columns: ':visible'

                        }

                    }

                ],

                "language": {

                    "url": "//cdn.datatables.net/plug-ins/1.11.5/i18n/pt-BR.json"

                },

                "search": {

                    "search": "<?php if(!isset($cnpj)){$cnpj = '';} echo $cnpj;?>"

                },

            } );

        } );

    </script>

    <script>

        //Definindo os valores nos inputs da modal

        var exampleModal = document.getElementById('modalAtualizar')

        exampleModal.addEventListener('show.bs.modal', function (event) {

        // Button that triggered the modal

        var button = event.relatedTarget

        // Extract info from data-bs-\* attributes

        var idraca = button.getAttribute('data-idraca')

        var raca = button.getAttribute('data-raca')

        var tipoespecie = parseInt(button.getAttribute('data-tipoespecie'))

        $("#idRaca").val(idraca)

        $("#raca").val(raca)

        $("#Especie option").filter("[value="+tipoespecie+"]").attr("selected",true)

        })

    </script>

    <!-- SCRIPT CONFIRMAÇÃO PARA EXCLUIR A RAÇA -->

    <script>

        function confirmar(id)

        {

            Swal.fire({

                title: 'Você tem certeza que deseja excluir?',

                text: "Você não será capaz de desfazer esta ação!",

                icon: 'warning',

                showCancelButton: true,

                confirmButtonColor: '#3085d6',

                cancelButtonColor: '#d33',

                confirmButtonText: 'Excluir',

                cancelButtonText: 'Cancelar'

                }).then((result) => {

                if (result.isConfirmed) {

                    Swal.fire(

                        'Excluído!', //title:

                        'Raça apagada com sucesso.', //text:

                        'success', //icon:

                    ).then(()=> {

                        window.location='<?php echo URL;?>excluir-raca/'+id;

                        }

                    )}

            })

        }

    </script>

</body>

</html>

**consultaUsuario**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <!-- CSS DataTables-->

    <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.1.3/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/buttons/2.2.3/css/buttons.dataTables.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.12.1/css/dataTables.bootstrap5.min.css">

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-danger container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container-fluid bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Consultar Usuários</h5>

                    </div>

                    <div class="container-fluid p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white table-responsive">

                        <div class="table-responsive">

                            <table id="tbUsuario" class="table table-hover">

                                <thead>

                                    <th>#</th>

                                    <th>Comp. Residência</th>

                                    <th>Nome</th>

                                    <th>CPF</th>

                                    <th>RG</th>

                                    <th>Benefício</th>

                                    <th>NIS</th>

                                    <th>E-mail</th>

                                    <th>Telefone</th>

                                    <th>Celular</th>

                                    <th>Punição</th>

                                    <th>Animais</th>

                                    <th>Ações</th>

                                </thead>

                                <tbody>

                                    <?php

                                        foreach($dadosUsuario as $value)

                                        {

                                            $valorBeneficio = $value->beneficio;

                                            $value->beneficio = str\_replace("0", "-", $value->beneficio);

                                            $value->beneficio = str\_replace("1", "Benefício Social", $value->beneficio);

                                            $value->beneficio = str\_replace("2", "Protetor de Animais", $value->beneficio);

                                            $value->beneficio = str\_replace("3", "Em análise", $value->beneficio);

                                            $valorTelefone = $value->telefone;

                                            $value->telefone = preg\_replace("/^$/", "-", $value->telefone);

                                            $valorNis = $value->nis;

                                            $value->nis = preg\_replace("/^$/", "-", $value->nis);

                                            $valorPunicao = $value->punicao;

                                            $value->punicao = str\_replace("0", "-", $value->punicao);

                                            $value->punicao = str\_replace("1", "<span class='badge bg-danger'>Punido</span>", $value->punicao);

                                            //Testar depois:

                                            //$number="(".substr($number,0,2).") ".substr($number,2,-4)." - ".substr($number,-4);

                                            echo

                                            "

                                            <tr>

                                                <td>$value->idusuario</td>

                                                <td>

                                                    <button id='btnImg' type='button' data-bs-target='#modalImg' data-bs-toggle='modal' data-img='$value->doccomprovante'>

                                                        <img width='150px' class='img-thumbnail' src='".URL."recursos/img/docComprovantes/$value->doccomprovante'>

                                                    </button>

                                                </td>

                                                <td>$value->nome</td>

                                                <td>$value->cpf</td>

                                                <td>$value->rg</td>

                                                <td>$value->beneficio</td>

                                                <td>$value->nis</td>

                                                <td>$value->email</td>

                                                <td>$value->telefone</td>

                                                <td>$value->celular</td>

                                                <td>$value->punicao</td>

                                                <td>

                                                    <a href='". URL. "consulta-animais/$value->idusuario' class='btn btn-success col-auto'>

                                                        <img src='". URL ."recursos/img/Logo-Castra-Pet.svg' alt='Animais cadastrados' width='30px' class='aling-itens-center white justify-content-center'>

                                                    </a>

                                                </td>

                                                <td>

                                                    <button class='btn btn-warning btn-sm-sm text-light' id='btnEditar' type='button' data-bs-target='#modalEditar' data-bs-toggle='modal'

                                                            data-idusuario='$value->idusuario' data-nome='$value->nome' data-cpf='$value->cpf' data-beneficio='$valorBeneficio' data-nis='$valorNis'

                                                            data-email='$value->email' data-telefone='$valorTelefone' data-celular='$value->celular' data-punicao='$valorPunicao' data-rg='$value->rg'

                                                            data-cep='$value->usucep' data-numero='$value->usunumero' data-bairro='$value->usubairro' data-rua='$value->usurua' data-idlogin='$value->idlogin'>

                                                        Editar

                                                    </button>

                                                    <a class='btn btn-danger btn-sm-sm' onclick='confirmar($value->idusuario, $value->idlogin)'>Excluir</a>

                                                </td>

                                            </tr>

                                            ";

                                        }

                                    ?>

                                </tbody>

                            </table>

                        </div>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-adm';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- MODAL: editar usuário -->

    <div class="modal fade" id="modalEditar" tabindex="-1" data-bs-keyboard="false" aria-labelledby="staticBackdropLabel" aria-hidden="true" >

        <div class="modal-dialog modal-lg modal-dialog-centered">

            <div class="modal-content">

                <form action="<?php echo URL.'atualiza-tutor';?>" method="post" enctype="multipart/form-data">

                    <div class="modal-header">

                        <h5 class="modal-title">Editar</h5>

                        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close" ></button>

                    </div>

                    <div class="modal-body">

                        <input type="hidden" name="idusuario" id="idusuario">

                        <input type="hidden" name="idlogin" id="idlogin">

                        <div class="row">

                            <div class="col-sm-6 mb-4">

                                <div class="form-group row-6 mb-3">

                                    <label for="txtNome" class="form-label">Nome:</label>

                                    <input  class="form-control" type="text" name="txtNome" id="txtNome"  maxlength="70" required>

                                </div>

                                <div class="form-group row-6 mb-3">

                                    <label for="txtEmail" class="form-label">Email:</label>

                                    <input class="form-control" type="email" name="txtEmail" id="txtEmail" maxlength="100" required>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group col-6">

                                        <label for="txtCPF" class="form-label">CPF:</label>

                                        <input class="form-control" type="text" name="txtCPF" id="txtCPF"  maxlength="14" required>

                                    </div>

                                    <div class="form-group col-6">

                                        <label for="txtTel" class="form-label">Telefone:</label>

                                        <input class="form-control" type="text" name="txtTel" id="txtTel"  maxlength="15">

                                    </div>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group col-6">

                                        <label for="txtRG" class="form-label">RG:</label>

                                        <input class="form-control" type="text" name="txtRG" id="txtRG"  maxlength="12" required>

                                    </div>

                                    <div class="form-group col-6">

                                        <label for="txtCelular" class="form-label">Celular:</label>

                                        <input class="form-control" type="text" name="txtCelular" id="txtCelular"  maxlength="15">

                                    </div>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group">

                                        <label for="chkPunicao" class="form-label">Punição</label>

                                        <input type="checkbox" name="chkPunicao" id="chkPunicao" value="1" class="form-check">

                                    </div>

                                </div>

                            </div>

                            <div class="col-sm-6 mb-4">

                                <div class="row mb-3">

                                    <div class="form-group col-6">

                                        <label for="txtCEP" class="form-label">CEP:</label>

                                        <input class="form-control" type="text" name="txtCEP" id="txtCEP" maxlength="9" required>

                                    </div>

                                    <div class="form-group col-6">

                                        <label for="txtNumero" class="form-label">Número:</label>

                                        <input class="form-control" type="text" name="txtNumero" id="txtNumero"  maxlength="5" required>

                                    </div>

                                </div>

                                <div class="form-group mb-3">

                                    <label for="txtBairro" class="form-label">Bairro:</label>

                                    <input class="form-control" type="text" name="txtBairro" id="txtBairro" maxlength="50" required>

                                </div>

                                <div class="form-group mb-3">

                                    <label for="txtRua" class="form-label">Rua:</label>

                                    <input class="form-control" type="text" name="txtRua" id="txtRua" maxlength="50" required>

                                </div>

                                <div class="row mb-3">

                                    <div class="form-group col-6">

                                        <input type="checkbox" name="chkNIS" id="chkNIS" value="1">

                                        <label for="chkNIS">Tenho o benefício do NIS</label>

                                    </div>

                                    <div class="form-group col-sm-6">

                                        <input class="form-control" type="text" name="txtNIS" id="txtNIS" maxlength="11">

                                    </div>

                                </div>

                                <div class="form-group mb-4">

                                    <input type="checkbox" name="chkProtetor" id="chkProtetor" value="2">

                                    <label for="chkProtetor">Sou protetor de animais</label>

                                </div>

                            </div>

                        </div>

                    </div>

                    <div class="modal-footer">

                        <button type="submit" class="btn btn-success">Editar</button>

                        <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Cancelar</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

    <!-- MODAL -->

    <!-- MODAL: img docComprovante-->

    <div class="modal fade" id="modalImg" tabindex="-1" data-bs-keyboard="false" aria-labelledby="staticBackdropLabel" aria-hidden="true" >

        <div class="modal-dialog modal-lg modal-dialog-centered">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title">Comprovante de Residência</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close" ></button>

                </div>

                <div class="modal-body row justify-content-center">

                    <img id="modalImagem" src="recursos/img/doccomprovantes/">

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Fechar</button>

                </div>

            </div>

        </div>

    </div>

    <!-- MODAL -->

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- DataTables -->

    <script type='text/javascript' charset='utf8' src='https://code.jquery.com/jquery-3.5.1.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/1.11.5/js/jquery.dataTables.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/1.12.1/js/dataTables.bootstrap5.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.2/js/dataTables.buttons.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.2/js/buttons.print.min.js'></script>

    <script type='text/javascript' charset='utf8' src='https://cdn.datatables.net/buttons/2.2.3/js/buttons.colVis.min.js'></script>

    <!-- EXTENSÃO JQUERY DAS MASCARAS -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.16/jquery.mask.min.js"></script>

    <!-- JS SweetAlert 2-->

    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script src="sweetalert2.all.min.js"></script>

    <script>

        $(document).ready(function() {

            $('#tbUsuario').DataTable( {

                dom: 'Bfrtip',

                buttons: [

                    'colvis',

                    {

                        extend:'csv',

                        exportOptions:{

                            columns: ':visible'

                        }

                    },

                    {

                        extend: 'excel',

                        exportOptions: {

                            columns: ':visible'

                        }

                    },

                    {

                        extend: 'print',

                        exportOptions:{

                            columns: ':visible'

                        }

                    }

                ],

                columnDefs: [

                    {

                        targets: 1,

                        className: 'noVis'

                    }

                ],

                "language": {

                    "url": "//cdn.datatables.net/plug-ins/1.11.5/i18n/pt-BR.json"

                },

                "search": {

                    "search": "<?php if(!isset($cpf)){$cpf = '';}echo $cpf;?>"

                }

            } );

        } );

    </script>

    <!-- ABRIR MODAL editar usuário -->

    <script>

        var modalEditar = document.getElementById('modalEditar')

        modalEditar.addEventListener('show.bs.modal', function (event) {

            var button = event.relatedTarget

            var idusuario = button.getAttribute('data-idusuario')

            var idlogin = button.getAttribute('data-idlogin')

            var nome = button.getAttribute('data-nome')

            var cpf = button.getAttribute('data-cpf')

            var beneficio = button.getAttribute('data-beneficio')

            var nis = button.getAttribute('data-nis')

            var email = button.getAttribute('data-email')

            var telefone = button.getAttribute('data-telefone')

            var celular = button.getAttribute('data-celular')

            var rg = button.getAttribute('data-rg')

            var cep = button.getAttribute('data-cep')

            var numero = button.getAttribute('data-numero')

            var bairro = button.getAttribute('data-bairro')

            var rua = button.getAttribute('data-rua')

            var punicao = button.getAttribute('data-punicao')

            var modalId = modalEditar.querySelector('.modal-title')

            modalId.textContent = 'Editar: ' + nome;

            $("#idusuario").val(idusuario);

            $("#idlogin").val(idlogin);

            $("#txtNome").val(nome);

            $("#txtCPF").val(cpf).mask('000.000.000-00');

            if(beneficio == 2)

            {

                $("#chkProtetor").prop("checked", true);

            }

            else{

                $("#chkProtetor").prop("checked", false);

            }

            if(nis.length == 11)

            {

                $("#chkNIS").prop("checked", true);

            }

            else{

                $("#chkNIS").prop("checked", false);

            }

            $("#txtNIS").val(nis);

            $("#txtEmail").val(email);

            $("#txtTel").val(telefone);

            $("#txtCelular").val(celular).mask('(00) 00000-0000');

            $("#txtRG").val(rg).mask('00.000.000-X', {'translation': {X: {pattern: /[0-9xX]/}}});

            $("#txtCEP").val(cep).mask('00000-000');

            $("#txtNumero").val(numero);

            $("#txtBairro").val(bairro);

            $("#txtRua").val(rua);

            if(punicao == 1)

            {

                $("#chkPunicao").prop("checked",true);

            }

            else{

                $("#chkPunicao").prop("checked",false);

            }

        });

    </script>

    <!-- ABRIR MODAL editar usuário -->

    <script>

        var modalEditar = document.getElementById('modalImg')

        modalEditar.addEventListener('show.bs.modal', function (event) {

            var button = event.relatedTarget

            var img = button.getAttribute('data-img')

            $("#modalImagem").prop("src", "<?php echo URL.'recursos/img/doccomprovantes/'?>"+img);

        });

    </script>

    <!-- SCRIPT CONFIRMAÇÃO PARA EXCLUIR O USUÁRIO -->

    <script>

        function confirmar(idusu, idlogin)

        {

            Swal.fire({

                title: 'Você tem certeza que deseja excluir?',

                text: "Você não será capaz de desfazer esta ação!",

                icon: 'warning',

                showCancelButton: true,

                confirmButtonColor: '#3085d6',

                cancelButtonColor: '#d33',

                confirmButtonText: 'Excluir',

                cancelButtonText: 'Cancelar'

                }).then((result) => {

                if (result.isConfirmed) {

                    Swal.fire(

                        'Excluído!', //title:

                        'Usuário apagado com sucesso.', //text:

                        'success', //icon:

                    ).then(()=> {

                        window.location='<?php echo URL;?>excluir-tutor/'+idusu+'/'+idlogin;

                        }

                    )}

            })

        }

    </script>

</body>

</html>

**editaAnimal**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

    <style type="text/css">

        input::-webkit-outer-spin-button,

        input::-webkit-inner-spin-button {

            display: none;

        }

       #inputImgAnimal + label

        {

            position: absolute;

            height: 250px;

            width: 375px;

            border: solid;

            border-color: #198754;

            cursor: pointer;

        }

        #imgAnimal

        {

            height: 250px;

            width: 399px;

        }

        @media screen and (max-width: 1199px)

        {

            #inputImgAnimal + label

            {

                width: 365px;

            }

        }

        @media screen and (max-width: 992px)

        {

            #inputImgAnimal + label

            {

                height: 200px;

                width: 272px;

            }

            #imgAnimal

            {

                height: 200px;

            }

        }

        @media screen and (max-width: 767px)

        {

            #imgAnimal

            {

                width: 290px;

            }

        }

        @media screen and (max-width: 336px)

        {

            #inputImgAnimal + label

            {

                height: 175px;

                width: 200px;

            }

            #imgAnimal

            {

                height: 175px;

                width: 220px;

            }

        }

    </style>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid vh-100 d-grid corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="container-fluid container-principal">

            <div class="bg-primary h-100 row align-items-center">

                <div class="container mx-auto p-3" style="grid-area:corpo;">

                    <div class="container shadow-lg text-light p-3" style="border: 0; border-radius:0; overflow: hidden; background-color: var(--preto);">

                        <h1 class="h5 m-1">EDITAR ANIMAL</h1>

                    </div>

                    <div class="container shadow-lg bg-white p-4">

                        <form method="post" action="<?php echo URL.'editar-animal';?>" enctype="multipart/form-data">

                        <input type="hidden" name="" value="<?php echo $dadosAnimal->idanimal;?>">

                            <div class="row">

                                <div class="col-md-7">

                                    <div class="row">

                                        <div class="col-12 mb-2">

                                            <label for="txtNome" class="form-label">Nome do Animal:</label>

                                            <input type="text" class="form-control" id="txtNome" name="txtNome" maxlength="50" required value="<?php echo $dadosAnimal->aninome;?>">

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="slcEspecie" class="form-label">Espécie:</label>

                                            <select id="slcEspecie" name="slcEspecie" class="form-select" required value="<?php echo $dadosAnimal->especie;?>">

                                                <option value="0">Canina</option>

                                                <option value="1">Felina</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="numIdade" class="form-label">Idade:</label>

                                            <input type="number" class="form-control" id="numIdade" name="numIdade" min="0" max="100" required value="<?php echo $dadosAnimal->idade;?>">

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="slcSexo" class="form-label">Sexo:</label>

                                            <select id="slcSexo" name="slcSexo" class="form-select" required value="<?php echo $dadosAnimal->sexo;?>">

                                                <option value="0">Fêmea</option>

                                                <option value="1">Macho</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPelagem" class="form-label">Pelagem:</label>

                                            <select id="slcPelagem" name="slcPelagem" class="form-select" required value="<?php echo $dadosAnimal->pelagem;?>">

                                                <option value="0">Curta</option>

                                                <option value="1">Média</option>

                                                <option value="2">Alta</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="txtCor" class="form-label">Cor:</label>

                                            <input type="text" class="form-control" id="txtCor" name="txtCor" maxlength="30" required value="<?php echo $dadosAnimal->cor;?>">

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPorte" class="form-label">Porte:</label>

                                            <select id="slcPorte" name="slcPorte" class="form-select" required value="<?php echo $dadosAnimal->porte;?>">

                                                <option value="0">Pequeno</option>

                                                <option value="1">Médio</option>

                                                <option value="2">Grande</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="listRaca" class="form-label">Raça:</label>

                                            <input list="racas" name="listRaca" id="listRaca" class="form-control" maxlength="30" required>

                                            <datalist id="racas">

                                                <?php

                                                    foreach($dadosRaca as $value)

                                                    {

                                                        echo "<option value='$value->raca'></option>";

                                                    }

                                                ?>

                                            </datalist>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcComunitario" class="form-label">Animal Comunitário:</label>

                                            <select id="slcComunitario" name="slcComunitario" class="form-select" required>

                                                <option value="0">Não</option>

                                                <option value="1">Sim</option>

                                            </select>

                                        </div>

                                    </div>

                                </div>

                                <div class="col-md-5 mt-2">

                                    <div class="row text-center mb-3">

                                        <label>Foto do animal:</label>

                                    </div>

                                    <div class="row justify-content-center mb-3">

                                        <input type="file" name="imgAnimal" id="inputImgAnimal" accept="image/\*" hidden>

                                        <label id="labelImgAnimal" for="inputImgAnimal" style="background-color: 0;"></label>

                                        <img src="<?php echo URL.'recursos/img/Animais/'.$dadosAnimal->foto;?>" alt="Foto do Animal" id="imgAnimal" for="inputImgAnimal">

                                    </div>

                                    <div class="row">

                                        <div class="col-12 text-center mt-3 mb-2">

                                            <input type="submit" value="Atualizar" class="btn btn-lg btn-success shadow" style="border-radius: 0; border: 0; padding: 12px 30px 12px 30px;">

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area:footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'meus-animais'; ?>" class="btn btn-success my-2 my-sm-0">Voltar</a>

                </div>

            </div>

        </div>

        <!-- /CORPO -->

    </div>

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

    <script>

        inputImgAnimal.onchange = evt => {

            const [file] = inputImgAnimal.files

            if (file) {

                var url = URL.createObjectURL(file);

                imgAnimal.src = url;

            }

        }

    </script>

</body>

</html>

**esqSenha**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Recuperar de senha</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <form method="post" class="container p-sm-3 p-md-3 p-lg-4 p-3" action="recuperar-senha">

                            <?php

                                //exibindo mensagem de erro

                                if(isset($\_COOKIE["msg"]))

                                {

                                    echo "<div class='alert alert-danger' role='alert'>

                                        ".$\_COOKIE['msg']."

                                    </div>";

                                }

                                //excluindo cookie de erro

                                setcookie("msg","",time() - 3600);

                            ?>

                            <div class="form-group mb-4">

                                <label for="txtEmail" class="form-label">E-mail:</label>

                                <input class="form-control" type="email" name="txtEmail" id="txtEmail" maxlength="100" required>

                            </div>

                            <div class="text-center">

                                <input type="submit" class="btn btn-success mx-auto" value="Enviar">

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'login';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**favicon**

<link rel="icon" href="<?php echo URL.'recursos/img/LogoCP (2).png'?> ">

**head**

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Favicon -->

    <?php include\_once "favicon.php"?>

    <!-- /Favicon-->

    <title>PetCastra</title>

    <!-- EXTENSÃO CSS -->

    <link rel="stylesheet" href="<?php echo URL;?>recursos/css/bootstrap.min.css">

    <link rel="stylesheet" href="<?php echo URL;?>recursos/css/root.css">

    <!-- /EXTENSÃO CSS -->

    <!-- Grid da tela -->

    <style rel="stylesheet" type="text/css">

        .corpo{

            grid-template-areas: 'header''corpo''footer';

            grid-template-rows: max-content auto 100px;

        }

        </style>

    <!-- /Grid da tela -->

**home**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container">

                        <div class="row ">

                            <div class="col-md-6">

                                <div class="p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                                    <p>

                                        Seja bem-vindo(a) ao projeto PetCastra, um site em que você pode se cadastrar para agendar uma castração gratuita para seu gato ou cachorro

                                        através do programa gratuito da Prefeitura de Franco da Rocha. Você sabe sobre as vantagens de castrar seu animalzinho? Algumas delas são:

                                        <ul>

                                            <li>Redução dos passeios externos e problemas associados a esta prática, como o desenvolvimento de doenças infecciosas contraídas nas ruas, traumas e atropelamentos, intoxicações e maus tratos;</li>

                                            <li>Redução de doenças zoonóticas (transmitidas de pets para humanos), como a <span class="fw-bold text-decoration-underline">Raiva</span> e a <span class="fw-bold text-decoration-underline">Leishmaniose</span>;</li>

                                            <li>Redução no <span class="fw-bold text-decoration-underline">comportamento de cio</span> e prevenção de doenças reprodutivas em fêmeas;</li>

                                            <li>Redução do risco de enfermidades mamárias, uterinas e ovarianas como <span class="fw-bold text-decoration-underline">neoplasias</span> e piometra (infecção no útero);</li>

                                            <li>Prevenção de doenças como pseudociese (<span class="fw-bold text-decoration-underline">gravidez psicológica</span>), hipertrofia mamária felina e estro persistente (cio);</li>

                                            <li>Prevenção de <span class="fw-bold text-decoration-underline">distúrbios testiculares</span> em machos (neoplasias, orquites (inflamação do testículo), epididimites (inflamação do epidídimo), além de auxiliar no tratamento de doenças andrógeno-dependentes como a hiperplasia prostática benigna, prostatite crônica (inflamação da próstata), adenoma e hérnia perineal;</li>

                                            <li>Auxílio no tratamento de doenças não relacionadas ao sistema reprodutivo, como diabetes e epilepsia;</li>

                                            <li>Possível redução de comportamentos indesejados como territorialismo, agressividade e marcação de território.</li>

                                            <li>Possível redução de animais abandonados nas ruas.</li>

                                        </ul>

                                    </p>

                                </div>

                            </div>

                            <div class="col-md-6">

                                <div class="p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                                    <div class="card">

                                        <h4 class="card-header">Perguntas Frequentes:</h4>

                                        <ol class="list-group list-group-flush">

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta0" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">É necessário pagar algo para castrar meu animal?</li>

                                            </a>

                                            <div class="collapse" id="pergunta0">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Não, o programa de castração é gratuito, o tutor tem apenas que lidar com os custos de transporte e cuidados pós-cirúrgicos.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta1" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">Com quantos anos posso solicitar a castração de meu animal?</li>

                                            </a>

                                            <div class="collapse" id="pergunta1">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> A partir de 18 anos você pode solicitar a castração de seu animal.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta2" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">Posso castrar mais de um animal?</li>

                                            </a>

                                            <div class="collapse" id="pergunta2">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> As vagas são limitadas e distribuídas para os moradores de Franco da Rocha baseado nos seguintes critérios:

                                                        <br><br>

                                                        Caso seja <span class="fw-bold">Protetor de Animais</span>: Têm direito a <span class="fw-bold">5</span> vagas para castração por mês;

                                                        <br>

                                                        Caso tenha <span class="fw-bold">Benefício Social</span>: Têm direito a <span class="fw-bold">2</span> vagas para castração por mês;

                                                        <br>

                                                        Para o restante da <span class="fw-bold">população</span>: Têm direito a apenas <span class="fw-bold">1</span> vaga para castração por mês;

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta3" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">O programa castra outros animais além de gatos e cachorros?</li>

                                            </a>

                                            <div class="collapse" id="pergunta3">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Não, o programa castra apenas <span class="fw-bold">gatos</span> e <span class="fw-bold">cachorros</span>.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta4" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">Como faço para solicitar uma castração para meu animal?</li>

                                            </a>

                                            <div class="collapse" id="pergunta4">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Clique no botão “Login” no menu do site, faça login caso já possua um cadastro, caso contrário clique em “Não possuo uma conta” e preencha as informações para se cadastrar. Após ter feito o login, vá em “Meus Animais” e cadastre seu animal, depois clique em “Solicitar Castração” e então sua solicitação estará em análise. Após ser aprovada, informações da clínica e horário serão enviadas ao e-mail cadastrado junto com as orientações que devem ser seguidas antes de ocorrer a cirurgia.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta5" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">Por onde eu vou receber as informações da castração?</li>

                                            </a>

                                            <div class="collapse" id="pergunta5">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> As informações de qual clínica será a responsável por realizar a castração e o endereço junto com o horário serão enviados para o e-mail que foi usado na hora de cadastrar sua conta.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta6" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">Como funciona o processo de castração?</li>

                                            </a>

                                            <div class="collapse" id="pergunta6">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> A castração é um processo simples onde o animal, após ser anestesiado, tem seus órgãos reprodutivos removidos, sendo eles no caso dos machos: os testículos, e no caso das fêmeas: o útero e os ovários. Depois da cirurgia ser feita, é orientado ao tutor os cuidados e remédios que devem ser dados ao animal para uma recuperação rápida.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta7" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">Onde será castrado meu animal?</li>

                                            </a>

                                            <div class="collapse" id="pergunta7">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Seu animal será castrado em uma das clínicas que têm parceria com a prefeitura de Franco da Rocha.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta8" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-2">Tem riscos de acontecer algo com meu animal durante a cirurgia?</li>

                                            </a>

                                            <div class="collapse" id="pergunta8">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Sim, como toda cirurgia, há riscos de acontecer algo durante o processo cirúrgico. Contudo a cirurgia de castração é simples e rotineira, sendo a raro o animal ter sequelas graves após a cirurgia.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta9" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-3">Tem algo que impeça o meu animal de ser castrado?</li>

                                            </a>

                                            <div class="collapse" id="pergunta9">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Sim, não poderá ser castrado caso o animal:

                                                        <br>

                                                        <ul>

                                                            <li>Esteja no período de acasalamento;</li>

                                                            <li>Não tenha seguido as orientações pré-cirúrgicas;</li>

                                                            <li>Tenha alergia a anestesia;</li>

                                                            <li>Esteja amamentando filhotes com menos de 45 dias;</li>

                                                            <li>For muito idoso (acima de 08 anos);</li>

                                                            <li>Tenha menos de 3 meses.</li>

                                                        </ul>

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta10" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-3">Existe alguma penalidade caso eu não compareça na clínica no dia da castração?</li>

                                            </a>

                                            <div class="collapse" id="pergunta10">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Sim, caso o tutor não entre em contanto com a clínica da castração justificando sua falta, será aplicada uma punição impedindo-o de solicitar a castração de seus animais durante a campanha de castração.

                                                    </p>

                                                </div>

                                            </div>

                                            <a class="list-group-item" data-bs-toggle="collapse" href="#pergunta11" role="button" aria-expanded="false" aria-controls="collapseExample">

                                                <li class="ms-3">O programa de castração de Franco da Rocha está disponível para moradores de outros municípios?</li>

                                            </a>

                                            <div class="collapse" id="pergunta11">

                                                <div class="card card-body border-primary">

                                                    <p>

                                                        <span class="fw-bold">R:</span> Não, o programa de castração é reservado apenas para moradores de Franco da Rocha.

                                                    </p>

                                                </div>

                                            </div>

                                        </ol>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL;?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**homeAdm**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

        <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-danger container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <p>

                            <h4>Olá Administrador(a)!</h4>

                            <br>

                            O administrador pode acessar as opções no menu acima para cadastrar clínicas e raças, consultar usuários e seus animais,

                            raças, clínicas e castrações, onde poderá editar ou excluir um registro deles caso seja necessário. Também há a opção de

                            aprovar as castrações, onde poderá analisar as informações do usuário e verificar se está tudo certo para escolher qual

                            clínica será a responsável por definir um dia e horário para realizar a castração.

                        </p>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL;?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL;?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL;?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**homeClinica**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-warning container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <p>

                            <h4>Bem-vindo(a) <?php echo $\_SESSION["dadosLogin"]->nome;?> !</h4>

                            <br>

                            A clínica poderá acessar no menu uma lista de castrações designadas e aprovadas pelo administrador para definir um dia e horário para sua realização.

                            Também há uma consulta de castrações que já têm seu dia e horário definidos, nessa tela a clínica deverá atualizar os <span class="fw-bold">status</span> da castração e colocar uma

                            <span class="fw-bold">observação</span>, caso seja necessário, conforme as seguintes situações:

                            <br><br>

                            <ul>

                                <li>“Animal Castrado”, caso tudo tenha ocorrido bem e o cirurgia foi bem sucedida;</li>

                                <li>“Tutor não compareceu”, caso o tutor não apareça no dia e horário marcado;</li>

                                <li>“Castração cancelada”, caso algum problema grave tenha acontecido antes da cirurgia e a castração tenha que ser cancelada;</li>

                                <li>“Castração remarcada”, caso algum problema tenha acontecido antes da cirurgia e a castração tenha que ser remarcada para um outro dia;</li>

                                <li>“Óbito”, caso o animal tenha falecido durante a cirurgia.</li>

                            </ul>

                        </p>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**homeUsuario**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum lacus mi, venenatis et scelerisque a, consectetur eu neque. Cras id rhoncus tortor, eget     aliquam erat. Aliquam erat volutpat. Pellentesque vehicula quam quis elementum maximus. Suspendisse ut congue nibh. Quisque vitae imperdiet nulla, eget pellentesque orci. In nunc mauris, tristique ut ex nec, volutpat pretium elit. Etiam sollicitudin ullamcorper elit in ultricies.

                        Donec cursus placerat commodo. Duis luctus arcu enim. Etiam lobortis fermentum aliquet. In lectus velit, aliquet tempor rhoncus et, efficitur sit amet nibh. Nulla quis odio id nulla rutrum pretium. Maecenas consequat consequat posuere. Nullam sed mattis sapien.

                        Suspendisse vitae magna sed neque vestibulum fermentum. Quisque a vulputate sapien. Pellentesque condimentum scelerisque feugiat. Pellentesque luctus quam orci, ut aliquam lorem iaculis sed. Morbi ac metus ac nisl fermentum volutpat eu in nulla. Curabitur congue quam et urna porttitor condimentum. Aliquam lorem erat, euismod vel velit a, imperdiet luctus magna. Suspendisse aliquam sapien non nisi eleifend, et suscipit ante auctor. Vivamus rhoncus enim nulla, in dignissim sem elementum ut. Phasellus feugiat volutpat tempor. Sed tristique enim nec elit ornare varius. Mauris congue mauris vitae dui elementum vulputate. Suspendisse consequat, tellus nec tincidunt pellentesque, nisl augue fringilla odio, eget egestas nunc dui vel eros.

                        Donec pharetra lacinia posuere. Integer lobortis dui vel suscipit ullamcorper. Morbi eget cursus ipsum, commodo mollis magna. Aliquam eu suscipit mi. Phasellus eu scelerisque est. Phasellus ac convallis leo. Integer maximus suscipit laoreet. Suspendisse non justo posuere nisi hendrerit efficitur. Nullam tortor justo, porttitor nec neque vel, lacinia commodo ipsum.

                        Phasellus dictum dolor sit amet ultrices blandit. Phasellus iaculis tempor pharetra. Pellentesque facilisis eget tellus vel interdum. Morbi tincidunt mi eget mi finibus commodo. Maecenas sit amet rutrum lorem. Sed in sem vel ante molestie luctus. Vivamus aliquam turpis id ante dignissim, et congue sapien aliquam. Mauris quis varius mi, id varius velit. Maecenas consequat non diam vel scelerisque. Suspendisse potenti. Fusce purus massa, porta eget odio eu, hendrerit semper nisi. In risus quam, molestie at porta nec, aliquet eget felis. Duis ultricies venenatis justo id sollicitudin. Mauris non convallis lorem.

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**infoUsuario**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

    <style rel="stylesheet" type="text/css">

        .corpo{

            grid-template-areas: 'header''corpo''footer';

            grid-template-rows: max-content auto 100px;

        }

    </style>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

        <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="container-fluid">

            <div class="bg-primary h-100 row align-items-center">

                <div class="container mx-auto p-3"style="grid-area:corpo;">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        Olá <?php echo $\_SESSION["dadosLogin"]->nome."!"; ?>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white">

                        <form action="<?php echo URL.'perfil'; ?>" class="p-sm-3 p-md-3 p-lg-4 p-3 px-0 row m-0" method="POST">

                            <div class="col-sm-6 form-group pe-sm-3 p-0">

                                <div class="row">

                                    <p>Nome: <?php echo $\_SESSION["dadosUsuario"]->nome;?></p>

                                </div>

                                <div class="row">

                                    <p>E-mail: <?php echo $\_SESSION["dadosUsuario"]->email;?></p>

                                </div>

                                <div class="row">

                                    <div class="col-6">

                                        <?php

                                            //Formatando o CPF

                                            $bloco1 = substr($\_SESSION["dadosUsuario"]->cpf,0,3);

                                            $bloco2 = substr($\_SESSION["dadosUsuario"]->cpf,3,3);

                                            $bloco3 = substr($\_SESSION["dadosUsuario"]->cpf,6,3);

                                            $digverificador = substr($\_SESSION["dadosUsuario"]->cpf,-2);

                                            $CPF\_Formatado = $bloco1.".".$bloco2.".".$bloco3."-".$digverificador;

                                        ?>

                                        <p>CPF: <?php echo $CPF\_Formatado?></p>

                                    </div>

                                    <div class="col-6">

                                        <?php

                                            //Formatando o Telefone

                                            $ddd = substr($\_SESSION["dadosUsuario"]->telefone,0,2);

                                            $bloco1 = substr($\_SESSION["dadosUsuario"]->telefone,2,4);

                                            $bloco2 = substr($\_SESSION["dadosUsuario"]->telefone,6,4);

                                            $Telefone\_Formatado = "(".$ddd.") ".$bloco1."-".$bloco2;

                                        ?>

                                        <p>Telefone: <?php if(empty($\_SESSION["dadosUsuario"]->telefone)) echo "-"; else echo $Telefone\_Formatado;?></p>

                                    </div>

                                </div>

                                <div class="row">

                                    <div class="col-6">

                                        <?php

                                            //Formatando o RG

                                            $bloco1 = substr($\_SESSION["dadosUsuario"]->rg,0,2);

                                            $bloco2 = substr($\_SESSION["dadosUsuario"]->rg,2,3);

                                            $bloco3 = substr($\_SESSION["dadosUsuario"]->rg,5,3);

                                            $digverificador = substr($\_SESSION["dadosUsuario"]->rg,-1);

                                            $RG\_Formatado = $bloco1.".".$bloco2.".".$bloco3."-".$digverificador;

                                        ?>

                                        <p>RG: <?php echo $RG\_Formatado?></p>

                                    </div>

                                    <div class="col-6">

                                        <?php

                                            //Formatando o Celular

                                            $ddd = substr($\_SESSION["dadosUsuario"]->celular,0,2);

                                            $bloco1 = substr($\_SESSION["dadosUsuario"]->celular,2,5);

                                            $bloco2 = substr($\_SESSION["dadosUsuario"]->celular,7,4);

                                            $Celular\_Formatado = "(".$ddd.") ".$bloco1."-".$bloco2;

                                        ?>

                                        <p>Celular: <?php echo $Celular\_Formatado;?></p>

                                    </div>

                                </div>

                            </div>

                            <div class="col-sm-6 form-group ps-sm-3 p-0">

                                <div class="row">

                                    <div class="col-6">

                                        <?php

                                            //Formatando o CEP

                                            $bloco1 = substr($\_SESSION["dadosUsuario"]->usucep,0,5);

                                            $bloco2 = substr($\_SESSION["dadosUsuario"]->usucep,-3);

                                            $CEP\_Formatado = $bloco1."-".$bloco2;

                                        ?>

                                        <p>CEP: <?php echo $CEP\_Formatado?></p>

                                    </div>

                                    <div class="col-6">

                                        <p>Número: <?php echo $\_SESSION["dadosUsuario"]->usunumero;?></p>

                                    </div>

                                </div>

                                <div class="row">

                                    <p>Bairro: <?php echo $\_SESSION["dadosUsuario"]->usubairro;?></p>

                                </div>

                                <div class="row">

                                    <p>Rua: <?php echo $\_SESSION["dadosUsuario"]->usurua;?></p>

                                </div>

                                <div class="row">

                                    <p>

                                        <?php

                                            if(empty($\_SESSION["dadosUsuario"]->nis))

                                            {

                                                echo "<input type='checkbox' disabled>";

                                                echo " Tenho benefício NIS: - ";

                                            }

                                            else

                                            {

                                                echo "<input type='checkbox' checked disabled>";

                                                //Formatando o NIS

                                                $bloco1 = substr($\_SESSION["dadosUsuario"]->nis,0,3);

                                                $bloco2 = substr($\_SESSION["dadosUsuario"]->nis,3,5);

                                                $bloco3 = substr($\_SESSION["dadosUsuario"]->nis,8,2);

                                                $digverificador = substr($\_SESSION["dadosUsuario"]->nis,-1);

                                                $NIS\_Formatado = $bloco1.".".$bloco2.".".$bloco3."-".$digverificador;

                                                echo " Tenho benefício NIS: $NIS\_Formatado";

                                            }

                                        ?>

                                    </p>

                                </div>

                                <div class="row">

                                    <p>

                                        <?php

                                            if(empty($\_SESSION["dadosUsuario"]->beneficio == 2)){

                                                echo "<input type='checkbox' disabled>";

                                                echo " Sou protetor de animais";

                                            }

                                            else{

                                                echo "<input type='checkbox' checked disabled>";

                                                echo " Sou protetor de animais";

                                            }

                                        ?>

                                    </p>

                                </div>

                            </div>

                            <div class="col row">

                                <div class="col-sm-3 mb-3">

                                    <a href="<?php echo URL.'alterar-senha/'.$\_SESSION["dadosLogin"]->idlogin; ?>" class="link-dark">Alterar minha senha</a>

                                </div>

                                <div class="col-sm-9">

                                    <a href="<?php echo URL.'meus-animais'; ?>" class="btn btn-success col-auto ">Meus animais</a>

                                </div>

                            </div>

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area:footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-usuario'; ?>" class="btn btn-success my-2 my-sm-0">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- MODAL: imagem protetor-->

    <div class="modal fade" id="modalImagem" tabindex="-1" data-bs-keyboard="false" aria-labelledby="staticBackdropLabel" aria-hidden="true" >

        <div class="modal-dialog modal-lg modal-dialog-centered">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title">Documento Comprovante de protetor</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

                </div>

                <div class="modal-body">

                    <img src="" class="img-fluid" id="img">

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Fechar</button>

                </div>

            </div>

        </div>

    </div>

    <!--/MODAL -->

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- ABRIR MODAL EDITAR -->

    <script>

        //Definindo os valores nos inputs da modal

        var modal = document.getElementById('modalImagem')

        modal.addEventListener('show.bs.modal', function (event) {

            // Button that triggered the modal

            var button = event.relatedTarget

            // Extract info from data-bs-\* attributes

            var img = button.getAttribute('data-bs-imagem')

            $("#img").prop("src","<?php echo URL.'recursos/img/docProtetores/'?>"+img);

        })

    </script>

</body>

</html>

**listaSolicitacao**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <!-- DataTables -->

    <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.1.3/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.12.1/css/dataTables.bootstrap5.min.css">

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <?php

            if($\_SESSION["dadosLogin"]->nivelacesso == 2){echo"<div class='bg-danger container-fluid' style='grid-area: corpo;''>";}

            else{echo"<div class='bg-warning container-fluid' style='grid-area: corpo;''>";}

        ?>

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Solicitações</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white table-responsive">

                        <table id="tbLista" class="table table-hover">

                            <thead>

                                <tr>

                                    <td>#</td>

                                    <td>Animal</td>

                                    <td>Tutor</td>

                                    <td>Solicitação</td>

                                </tr>

                            </thead>

                            <tbody>

                                <?php

                                    foreach ($dadosSolicitacao as $value)

                                    {

                                        echo"

                                            <tr>

                                                <td>Solicitação $value->idcastracao</td>

                                                <td>$value->aninome</td>

                                                <td>$value->nome</td>

                                                <td><a href='".URL."agendamento/$value->idcastracao' class='btn btn-info'>Analisar</a></td>

                                            </tr>

                                        ";

                                    }

                                ?>

                            </tbody>

                        </table>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'home-adm';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado -->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- DataTables -->

    <script type="text/javascript" charset="utf8" src="https://code.jquery.com/jquery-3.5.1.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.js"></script>

    <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.12.1/js/dataTables.bootstrap5.min.js"></script>

    <script>

        $(document).ready(function () {

            $('#tbLista').DataTable({

                "language": {

                    "url": "//cdn.datatables.net/plug-ins/1.11.5/i18n/pt-BR.json"

                }

            });

        });

    </script>

</body>

</html>

**Login**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Entrar na minha conta</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <form action="logar" class="container p-sm-3 p-md-3 p-lg-4 p-3" method="POST">

                            <div class="form-group">

                                <label for="email">E-mail:</label>

                                <input type="email" id="email" name="txtEmail" class="form-control" placeholder="" required>

                            </div>

                            <div class="form-group my-4">

                                <label for="senha">Senha:</label>

                                <input type="password" id="senha" name="txtSenha" class="form-control" placeholder="" required>

                            </div>

                            <div class="text-center">

                                <input type="submit" class="btn btn-success" value="Entrar">

                            </div>

                        </form>

                        <div class="row px-4 py-3">

                            <a href="<?php echo URL.'cadastra-tutor';?>" class="col-6 text-center">Cadastrar-se</a>

                            <a href="<?php echo URL.'esqueci-a-senha';?>" class="col-6 text-center">Esqueci minha senha</a>

                        </div>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'#inicio';?>" class="btn btn-success">Voltar</a>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**menu**

<nav class="navbar navbar-expand-md navbar-light bg-transparent mx-4 my-0" style="grid-area: header;">

  <a class="navbar-brand" href="<?php echo URL.'inicio';?>"><img src="<?php echo URL.'recursos/img/LogoCP (2).png'?>" style="height:80px;"></a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#conteudoNavbarSuportado" aria-controls="conteudoNavbarSuportado" aria-expanded="false" aria-label="Alterna navegação">

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="conteudoNavbarSuportado">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL; ?>">Início</a>

      </li>

      <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'sobre'; ?>">Sobre</a>

      </li>

      <li class="nav-item active">

        <a class="nav-link"  href="https://www.francodarocha.sp.gov.br/" target="\_blank">Prefeitura Franco da Rocha</a>

      </li>

    </ul>

    <form class="form-inline ms-auto my-2 my-lg-0">

      <a href="<?php echo URL.'login';?>" class="btn btn-success my-2 my-sm-0">Login</a>

    </form>

  </div>

</nav>

**menuADM**

<nav class="navbar navbar-expand-lg navbar-light bg-transparent mx-4 my-0" style="grid-area: header;">

  <a class="navbar-brand" href="<?php echo URL.'home-adm';?>"><img src="<?php echo URL.'recursos/img/LogoCP (2).png'?>" style="height: 80px;"></a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#conteudoNavbarSuportado" aria-controls="conteudoNavbarSuportado" aria-expanded="false" aria-label="Alterna navegação">

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="conteudoNavbarSuportado">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'home-adm'; ?>">Área do Administrador</a>

      </li>

      <li class="nav-item dropdown">

        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

          Cadastrar

        </a>

        <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

          <li><a class="dropdown-item" href="<?php echo URL.'cadastra-clinica'; ?>">Clínicas</a></li>

          <li><a class="dropdown-item" href="<?php echo URL.'cadastra-raca'; ?>">Raças</a></li>

        </ul>

      </li>

      <li class="nav-item dropdown">

        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

          Consultar

        </a>

        <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

          <li><a class="dropdown-item" href="<?php echo URL.'consulta-usuario'; ?>">Usuários</a></li>

          <li><a class="dropdown-item" href="<?php echo URL.'consulta-clinica'; ?>">Clínicas</a></li>

          <li><a class="dropdown-item" href="<?php echo URL.'consulta-castracao'; ?>">Castrações</a></li>

          <li><a class="dropdown-item" href="<?php echo URL.'consulta-raca'; ?>">Raças</a></li>

        </ul>

      </li>

      <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'lista-solicitacao'; ?>">Solicitações</a>

      </li>

    </ul>

    <form class="form-inline ms-auto my-2 my-lg-0">

      <a href="<?php echo URL.'encerrar-sessao'; ?>" class="btn btn-success my-2 my-sm-0">Encerrar Sessão</a>

    </form>

  </div>

</nav>

**menuClinica**

<nav class="navbar navbar-expand-md navbar-light bg-transparent mx-4 my-0" style="grid-area: header;">

  <a class="navbar-brand" href="<?php echo URL.'home-clinica';?>"><img src="<?php echo URL.'recursos/img/LogoCP (2).png'?>" style="height:80px;"></a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#conteudoNavbarSuportado" aria-controls="conteudoNavbarSuportado" aria-expanded="false" aria-label="Alterna navegação">

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="conteudoNavbarSuportado">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'home-clinica'; ?>">Área da Clínica</a>

      </li>

      <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'consulta-castracao'; ?>">Consultar castrações</a>

      </li>

      <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'lista-solicitacao'; ?>">Lista Solicitações</a>

      </li>

    </ul>

    <form class="form-inline ms-auto my-2 my-lg-0">

      <a href="<?php echo URL.'encerrar-sessao'; ?>" class="btn btn-success my-2 my-sm-0">Encerrar Sessão</a>

    </form>

  </div>

</nav>

**menuControle**

<?php

//CONTROLE DE MENU

if(isset($\_SESSION["dadosLogin"])) //caso esteja logado e exista uma sessão

{

    switch($\_SESSION["dadosLogin"]->nivelacesso)

    {

        //caso tenha nível de acesso de usuário

        case 0: include\_once "menuLogado.php"; break;

        //caso tenha nível de acesso de clínica

        case 1: include\_once "menuClinica.php"; break;

        //caso tenha nível de acesso de Administrador

        case 2: include\_once "menuADM.php"; break;

    }

}

else{ include\_once "menu.php"; }

?>

**menuLogado**

<nav class="navbar navbar-expand-lg navbar-light bg-transparent mx-4 my-0" style="grid-area: header;">

<a class="navbar-brand" href="<?php echo URL.'home-usuario';?>"><img src="<?php echo URL.'recursos/img/LogoCP (2).png';?>" style="height:80px;"></a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#conteudoNavbarSuportado" aria-controls="conteudoNavbarSuportado" aria-expanded="false" aria-label="Alterna navegação">

    <span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="conteudoNavbarSuportado">

    <ul class="navbar-nav me-auto">

    <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'home-usuario'; ?>">Início</a>

    </li>

    <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'meus-animais'; ?>">Meus Animais</a>

    </li>

    <li class="nav-item active">

        <a class="nav-link"  href="<?php echo URL.'sobre'; ?>">Sobre</a>

    </li>

    <li class="nav-item active">

        <a class="nav-link"  href="https://www.francodarocha.sp.gov.br/">Prefeitura Franco da Rocha</a>

    </li>

    </ul>

    <form class="form-inline ms-auto my-2 my-lg-0">

        <a class="text-decoration-none alert p-0" href="<?php echo URL.'perfil'?>">

            <img src="<?php echo URL.'recursos/img/user.png';?>" alt="Perfil" width="40">

        </a>

        <span class="mx-3"></span>

        <a href="<?php echo URL.'encerrar-sessao';?>" class="btn btn-success my-2 my-sm-0">Encerrar Sessão</a>

    </form>

</div>

</nav>

**meusAnimais**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="container-fluid">

            <div class="bg-primary h-100 row align-items-center">

                <div class="container mx-auto p-3" style="grid-area:corpo;">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        Meus Animais

                    </div>

                    <div class="container bg-white pt-3">

                    <!-- Componentes aqui -->

                        <?php

                        foreach ($dadosAnimais as $value)

                        {

                            //Reescrevendo a espécie

                            $valorEspecie = $value->especie;

                            $value->especie = str\_replace("0","Canina", $value->especie);

                            $value->especie = str\_replace("1","Felina", $value->especie);

                            //Reescrevendo o sexo

                            $valorSexo = $value->sexo;

                            $value->sexo = str\_replace("0","Fêmea", $value->sexo);

                            $value->sexo = str\_replace("1","Macho", $value->sexo);

                            //Reescrevendo a pelagem

                            $valorPelagem = $value->pelagem;

                            $value->pelagem = str\_replace("0","Curta", $value->pelagem);

                            $value->pelagem = str\_replace("1","Média", $value->pelagem);

                            $value->pelagem = str\_replace("2","Alta", $value->pelagem);

                            //Reescrevendo o porte

                            $valorPorte = $value->porte;

                            $value->porte = str\_replace("0","Pequeno", $value->porte);

                            $value->porte = str\_replace("1","Médio", $value->porte);

                            $value->porte = str\_replace("2","Grande", $value->porte);

                            //Reescrevendo o Comunitário

                            $valorComunitario = $value->comunitario;

                            $value->comunitario = str\_replace("0","Não", $value->comunitario);

                            $value->comunitario = str\_replace("1","Sim", $value->comunitario);

                            echo

                            "

                            <!-- Começo de um animal -->

                                <div class='row align-items-center'>

                                    <div class='col-md-3 d-flex align-items-center'>

                                        <img src='".URL."recursos/img/Animais/$value->foto' alt='Imagem' class='mw-100' style='height: 200px; width: 300px;'>

                                    </div>

                                    <div class='col-md-7'>

                                        <div class='row'>

                                            <div class='col-md-7'>

                                                <div class='row'>

                                                    <p>

                                                        Nome:

                                                        ".$value->aninome."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Espécie:

                                                        ".$value->especie."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Sexo:

                                                        ".$value->sexo."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Pelagem:

                                                        ".$value->pelagem."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Porte:

                                                        ".$value->porte."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p class='mb-md-0'>

                                                        Animal Comunitário:

                                                        ".$value->comunitario."

                                                    </p>

                                                </div>

                                            </div>

                                            <div class='col-md-5'>

                                                <div class='row'>

                                                    <p>

                                                        Idade:

                                                        ".$value->idade."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p>

                                                        Cor:

                                                        ".$value->cor."

                                                    </p>

                                                </div>

                                                <div class='row'>

                                                    <p class='mb-0'>

                                                        Raça:

                                                        ".$value->raca."

                                                    </p>

                                                </div>

                                            </div>

                                        </div>

                                        <div class='col'></div>

                                    </div>

                                    <div class='col-md-2 mt-2 mt-md-0'>

                                    ";

                                    if(!isset($value->status))

                                    {

                                        if($quantcastracoes->quantcastracoes > 0 && $\_SESSION["dadosUsuario"]->punicao == 0)

                                        {

                                            echo "

                                            <button class='btn btn-success w-100 mb-2' id='btnSolicitar' type='button' data-bs-target='#modalSolicitar' data-bs-toggle='modal' data-idanimal='$value->idanimal' data-idusuario='$value->idusuario'>

                                                Solicitar castração

                                            </button>

                                            ";

                                        }

                                        else if($\_SESSION["dadosUsuario"]->punicao == 1)

                                        {

                                            echo "<span class='btn bg-danger w-100 mb-2 text-white' style='cursor: default;'>Solicitação Bloqueada</span> ";

                                        }

                                        echo "

                                        <button class='btn btn-warning w-100 mb-2' id='btnEditar' type='button' data-bs-target='#modalEditar' data-bs-toggle='modal'

                                                data-idanimal='$value->idanimal' data-idusuario='$value->idusuario' data-nome='$value->aninome' data-especie='$valorEspecie'

                                                data-sexo='$valorSexo' data-cor='$value->cor' data-raca='$value->idraca' data-idade='$value->idade' data-pelagem='$valorPelagem'

                                                data-porte='$valorPorte' data-comunitario='$valorComunitario' data-foto='$value->foto'>

                                                Editar animal

                                        </button>

                                        <a class='btn btn-danger w-100' onclick='confirmar($value->idanimal)'>Excluir animal</a>

                                        ";

                                    }

                                    else

                                    {

                                        switch($value->status)

                                        {

                                            case 0:

                                                echo "<span class='btn btn-sm bg-warning w-100 my-3 text-white fw-bold' style='cursor: default;'>Solicitação em análise</span>";

                                            break;

                                            case 1:

                                                echo "<span class='btn btn-sm bg-success w-100 my-3 text-white fw-bold' style='cursor: default;'>Solicitação aprovada</span>";

                                            break;

                                            case 2:

                                                echo "<span class='btn btn-sm bg-success w-100 my-3 text-white fw-bold' style='cursor: default;'>Animal Castrado</span>";

                                            break;

                                            case 3:

                                                echo "<span class='btn btn-sm bg-danger w-100 my-3 text-white fw-bold' style='cursor: default;'>Solicitação recusada</span>";

                                            break;

                                            case 4:

                                                echo "<span class='btn btn-sm bg-danger w-100 my-3 text-white fw-bold' style='cursor: default;'>Tutor não compareceu</span>";

                                            break;

                                            case 5:

                                                echo "<span class='btn btn-sm bg-danger w-100 my-3 text-white fw-bold' style='cursor: default;'>Solicitação cancelada</span>";

                                            break;

                                            case 6:

                                                echo "<span class='btn btn-sm bg-success w-100 my-3 text-white fw-bold' style='cursor: default;'>Solicitação reagendada</span>";

                                            break;

                                            case 7:

                                                echo "<span class='btn btn-sm bg-dark w-100 my-3 text-white fw-bold' style='cursor: default;'>Animal foi a óbito</span>";

                                            break;

                                            default:

                                                echo "<span class='btn btn-sm bg-secondary w-100 my-3 text-white fw-bold' style='cursor: default;'>Ocorreu um erro</span>";

                                            break;

                                        }

                                    }

                                    echo "

                                    </div>

                                </div>

                                <hr>

                            <!-- Fim de um animal -->

                            ";

                        }

                        ?>

                        <div class="row align-items-center p-3">

                            <div class="col">

                                <a href="<?php echo URL.'cadastra-animal';?>" class="btn btn-success float-end">Cadastrar Animal</a>

                            </div>

                        </div>

                    <!-- Fim dos componentes -->

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area:footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <a href="<?php echo URL.'perfil'; ?>" class="btn btn-success my-2 my-sm-0">Voltar</a>

                </div>

            </div>

        </div>

        <!-- MODAL: solicitar castração -->

        <div class='modal fade' id='modalSolicitar' data-bs-keyboard='false' tabindex='-1' aria-labelledby='staticBackdropLabel' aria-hidden='true'>

            <div class='modal-dialog modal-dialog-centered'>

                <div class='modal-content'>

                    <form action="<?php echo URL.'solicitar-castracao';?>" method='post'>

                        <input type='hidden' id='idAnimalSolicita' name='idAnimal'>

                        <input type="hidden" name="idusuario" id="idusuario">

                        <div class='modal-header'>

                            <h5 class='modal-title' id='staticBackdropLabel'>Solicitar castração</h5>

                            <button type='button' class='btn-close' data-bs-dismiss='modal' aria-label='Close'></button>

                        </div>

                        <div class='modal-body'>

                            <label class='form-label' for='obhsCastracao'>Observação: (opcional)</label>

                            <textarea name='obsCastracao' id='obsCastr' rows='5' class='form-control' placeholder="Ex: tem alegria a dipirona"></textarea>

                        </div>

                        <div class='modal-footer'>

                            <button type='submit' class='btn btn-primary'>Enviar Solicitação</button>

                            <button type='button' class='btn btn-secondary' data-bs-dismiss='modal'>Cancelar</button>

                        </div>

                    </form>

                </div>

            </div>

        </div>

        <!-- /MODAL -->

        <!-- MODAL: editar animal-->

        <div class="modal fade" id="modalEditar" tabindex="-1" data-bs-backdrop="static" data-bs-keyboard="false" aria-labelledby="staticBackdropLabel" aria-hidden="true" >

            <div class="modal-dialog modal-lg modal-dialog-centered">

                <div class="modal-content">

                    <form method="post" id="formEdita" action="<?php echo URL.'atualizar-animal';?>" enctype="multipart/form-data">

                        <div class="modal-header">

                            <h5 class="modal-title">Editar</h5>

                            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close" ></button>

                        </div>

                        <div class="modal-body">

                            <input type="hidden" name="idanimal" id="idanimal">

                            <input type="hidden" name="idusuario" id="idusuario">

                            <div class="row">

                                <div class="col-md">

                                    <div class="row">

                                        <div class="col-12 mb-2">

                                            <label for="txtNome" class="form-label">Nome do Animal:</label>

                                            <input type="text" class="form-control" id="txtNome" name="txtNome" maxlength="50" required>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="tipoEspecie" class="form-label">Espécie:</label>

                                            <select id="tipoEspecie" name="tipoEspecie" class="form-select" onchange="carregarRaca(this)" required>

                                                <option>... SELECIONE A ESPÉCIE ...</option>

                                                <option value="0">Canina</option>

                                                <option value="1">Felina</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="numIdade" class="form-label">Idade:</label>

                                            <input type="number" class="form-control" id="numIdade" name="numIdade" min="0" max="100" required>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="slcSexo" class="form-label">Sexo:</label>

                                            <select id="slcSexo" name="slcSexo" class="form-select" required>

                                                <option value="0">Fêmea</option>

                                                <option value="1">Macho</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPelagem" class="form-label">Pelagem:</label>

                                            <select id="slcPelagem" name="slcPelagem" class="form-select" required>

                                                <option value="0">Curta</option>

                                                <option value="1">Média</option>

                                                <option value="2">Alta</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row">

                                        <div class="col-md-6 mb-2">

                                            <label for="txtCor" class="form-label">Cor:</label>

                                            <input type="text" class="form-control" id="txtCor" name="txtCor" maxlength="30" required>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcPorte" class="form-label">Porte:</label>

                                            <select id="slcPorte" name="slcPorte" class="form-select" required>

                                                <option value="0">Pequeno</option>

                                                <option value="1">Médio</option>

                                                <option value="2">Grande</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row mb-2">

                                        <div class="col-md-6 mb-2">

                                            <label for="listRaca" class="form-label">Raça:</label>

                                            <select name="racas" id="racas" class="form-select" required>

                                                <option value="">... SELECIONE A RAÇA ...</option>

                                            </select>

                                        </div>

                                        <div class="col-md-6 mb-2">

                                            <label for="slcComunitario" class="form-label">Animal Comunitário:</label>

                                            <select id="slcComunitario" name="slcComunitario" class="form-select" required>

                                                <option value="0">Não</option>

                                                <option value="1">Sim</option>

                                            </select>

                                        </div>

                                    </div>

                                    <div class="row text-center mb-3 aling-content-center">

                                        <div class="col-lg-6 justify-content-center aling-content-center">

                                            <label>Foto do animal:</label>

                                            <label id="labelImgAnimal" for="inputImgAnimal" class="form-label"></label>

                                            <img src="recursos/img/" alt="Foto do Animal" id="imgAnimal" for="inputImgAnimal" style="width:100%;">

                                        </div>

                                        <div class="col-lg-6 justify-content-center aling-content-center">

                                            <label id="labelImgAnimal" for="inputImgAnimal" class="form-label">Trocar foto:</label>

                                            <input type="file" name="foto" id="inputImgAnimal" class="form-control" accept="image/\*">

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <div class="modal-footer">

                            <button type="submit" class="btn btn-success">Editar</button>

                            <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Cancelar</button>

                        </div>

                    </form>

                </div>

            </div>

        </div>

        <!--/MODAL -->

        <!-- /CORPO -->

    </div>

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

    <!-- EXTENSÃO JQUERY PARA O AJAX -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js" integrity="sha512-894YE6QWD5I59HgZOGReFYm4dnWc1Qt5NtvYSaNcOP+u1T9qYdvdihz0PPSiiqn/+/3e7Jo4EaG7TubfWGUrMQ==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

    <!-- JS SweetAlert 2-->

    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script src="sweetalert2.all.min.js"></script>

    <!-- ABRIR MODAL SOLICITAR -->

    <script>

        var exampleModal = document.getElementById('modalSolicitar')

        exampleModal.addEventListener('show.bs.modal', function (event) {

            // Button that triggered the modal

            var button = event.relatedTarget

            // Extract info from data-bs-\* attributes

            var idanimal = button.getAttribute('data-idanimal')

            var idusuario = button.getAttribute('data-idusuario')

            $("#idAnimalSolicita").val(idanimal);

            $("#idusuario").val(idusuario)

        });

    </script>

    <!-- ABRIR MODAL EDITAR -->

    <script>

        //Definindo os valores nos inputs da modal

        var modal = document.getElementById('modalEditar')

        modal.addEventListener('show.bs.modal', function (event) {

            // Button that triggered the modal

            var button = event.relatedTarget

            // Extract info from data-bs-\* attributes

            var idusuario = button.getAttribute('data-idusuario')

            var idanimal = button.getAttribute('data-idanimal')

            var nome = button.getAttribute('data-nome')

            var especie = parseInt(button.getAttribute('data-especie'))

            var sexo = parseInt(button.getAttribute('data-sexo'))

            var cor = button.getAttribute('data-cor')

            var idade = button.getAttribute('data-idade')

            var pelagem = parseInt(button.getAttribute('data-pelagem'))

            var foto = button.getAttribute('data-foto')

            if (foto == "")

            {

                foto = "imagem\_exemplo.jpg";

            }

            $(document).ready(function()

            {

                $("#racas").empty(); //limpar todos antes de carregar

                $.ajax({

                    url: '<?php echo URL;?>carregar-raca/'+ especie,

                    success: function(data) {

                        $("#racas").append(data);

                        try{

                            $("#racas option").filter("[value="+raca+"]").attr("selected",true)

                        }

                        catch{}

                    }

                });

            })

            var raca = button.getAttribute('data-raca')

            var porte = button.getAttribute('data-porte')

            var comunitario = button.getAttribute('data-comunitario')

            $("#idusuario").val(idusuario)

            $("#idanimal").val(idanimal)

            $("#txtNome").val(nome)

            $("#tipoEspecie option").filter("[value="+especie+"]").attr("selected",true)

            $("#slcSexo option").filter("option[value="+sexo+"]").attr("selected",true)

            $("#slcPelagem option").filter("option[value="+pelagem+"]").attr("selected",true)

            $("#slcPorte option").filter("option[value="+porte+"]").attr("selected",true)

            $("#slcComunitario option").filter("option[value="+comunitario+"]").attr("selected",true)

            $("#txtCor").val(cor)

            $("#numIdade").val(idade)

            $("#imgAnimal").prop("src","<?php echo URL.'recursos/img/Animais/';?>"+foto);

        })

    </script>

    <script>

        //Carregar raças

        function carregarRaca(id)

        {

            //limpar todos antes de carregar

            $("#racas").empty();

            $.ajax({

                url: '<?php echo URL;?>carregar-raca/'+ id.value,

                success: function(data) {

                    $("#racas").append(data);

                }

            });

        }

    </script>

    <!-- SCRIPT PARA MOSTRAR AS FOTOS EM TEMPO REAL -->

    <script>

        inputImgAnimal.onchange = evt => {

            const [file] = inputImgAnimal.files

            if (file) {

                var url = URL.createObjectURL(file);

                imgAnimal.src = url;

            }

        }

    </script>

    <!-- SCRIPT CONFIRMAÇÃO PARA EXCLUIR O ANIMAL -->

    <script>

        function confirmar(id)

        {

            Swal.fire({

                title: 'Você tem certeza que deseja excluir?',

                text: "Você não será capaz de desfazer esta ação!",

                icon: 'warning',

                showCancelButton: true,

                confirmButtonColor: '#3085d6',

                cancelButtonColor: '#d33',

                confirmButtonText: 'Excluir',

                cancelButtonText: 'Cancelar'

                }).then((result) => {

                if (result.isConfirmed) {

                    Swal.fire(

                        'Excluído!', //title:

                        'Animal apagado com sucesso.', //text:

                        'success', //icon:

                    ).then(()=> {

                        window.location='<?php echo URL;?>excluir-animal/'+id;

                        }

                    )}

            })

        }

    </script>

</body>

</html>

**paginaNaoEncontrada**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <?php

        if(isset($\_SESSION["dadosLogin"]))

        {

            switch($\_SESSION["dadosLogin"]->nivelacesso)

            {

                case 0:

                    echo "<div class='bg-primary container-fluid' style='grid-area: corpo;'>";

                break;

                case 1:

                    echo "<div class='bg-warning container-fluid' style='grid-area: corpo;'>";

                break;

                case 2:

                    echo "<div class='bg-danger container-fluid' style='grid-area: corpo;'>";

                break;

            }

        }

        else

        echo "<div class='bg-primary container-fluid' style='grid-area: corpo;'>";

        ?>

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Ops... Parece que esta página não existe</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 px-0 bg-white text-center" style="height:400px;">

                        <div class="row align-items-center m-0 h-100 w-100">

                            <?php

                            if(isset($\_SESSION["dadosLogin"]->nivelacesso))

                            {

                                switch($\_SESSION["dadosLogin"]->nivelacesso)

                                {

                                    case 0:

                                        echo "<a href='".URL."home-usuario' class='stretched-link'>Voltar para a página inicial</a>";

                                    break;

                                    case 1:

                                        echo "<a href='".URL."home-clinica' class='stretched-link'>Voltar para a página inicial</a>";

                                    break;

                                    case 2:

                                        echo "<a href='".URL."home-adm' class='stretched-link'>Voltar para a página inicial</a>";

                                    break;

                                    default:

                                        echo "<a href='".URL."inicio' class='stretched-link'>Voltar para a página inicial</a>";

                                }

                            }

                            else

                            echo "<a href='".URL."inicio' class='stretched-link'>Voltar para a página inicial</a>";

                            ?>

                        </div>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                    <?php

                        if(isset($\_SESSION["dadosLogin"]))

                        {

                            switch($\_SESSION["dadosLogin"]->nivelacesso)

                            {

                                case 0:

                                    echo "<a href='".URL."home-usuario' class='btn btn-success'>Voltar</a>";

                                break;

                                case 1:

                                    echo "<a href='".URL."home-clinica' class='btn btn-success'>Voltar</a>";

                                break;

                                case 2:

                                    echo "<a href='".URL."home-adm' class='btn btn-success'>Voltar</a>";

                                break;

                                default:

                                    echo "<a href='".URL."inicio' class='btn btn-success'>Voltar</a>";

                            }

                        }

                        else

                        echo "<a href='".URL."inicio' class='btn btn-success'>Voltar</a>";

                    ?>

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>/recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> -->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>/recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**Sobre**

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <?php include\_once "head.php";?>

</head>

<body>

    <!-- CORPO -->

    <div class="container-fluid d-grid min-vh-100 corpo">

    <?php /\*Controle de menu!\*/ include\_once "menuControle.php";?>

        <div class="bg-primary container-fluid" style="grid-area: corpo;">

            <div class="row h-100 align-items-center">

                <div class="p-3">

                    <div class="container bg-dark text-light font-weight-bold p-3">

                        <h5 class="m-0">Sobre a castração em Franco da Rocha</h5>

                    </div>

                    <div class="container p-sm-3 p-md-3 p-lg-4 p-3 bg-white">

                        <h3>

                            Título

                        </h3>

                        <p>

                            Texto

                        </p>

                    </div>

                </div>

            </div>

        </div>

        <div class="container-fluid bg-dark" style="grid-area: footer;">

            <div class="row h-100 align-items-center">

                <div class="px-5">

                </div>

            </div>

        </div>

    </div>

    <!-- /CORPO -->

    <!-- EXTENSÃO BOOTSTRAP -->

    <script src="<?php echo URL; ?>recursos/js/jquery-3.3.1.slim.min.js"></script>

    <!--<script src="<?php echo URL; ?>recursos/js/popper.min.js"></script> Ultrapassado-->

    <script src="<?php echo URL; ?>recursos/js/bootstrap.min.js"></script>

    <script src="<?php echo URL;?>recursos/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**Index**

<?php

session\_start();

include\_once "controller/Controller.php";

include\_once "controller/AnimalController.php";

include\_once "controller/UsuarioController.php";

include\_once "controller/ClinicaController.php";

//Definindo uma constante para a URL do site

define("URL","http://localhost/CastraPet/");

if($\_GET)

{

    //Pegando a URL e apagando a "/" no final dela.

    $url = $\_GET["url"];

    $url = explode("/",$url);

    //Definindo os nomes das telas que vão aparecer na URL

    switch($url[0])

    {

        // PÁGINA INICIAL

        case "inicio":

            $direciona = new Controller();

            $direciona->abrirInicio();

        break;

        // CADASTRO TUTOR

        case "cadastra-tutor";

            $direciona = new Controller();

            $direciona->abrirCadastro();

        break;

        case "cadastrar-tutor";

            $direciona = new UsuarioController();

            $direciona->cadastrarUsuario();

        break;

        // PÁGINAS IMPORTANTES

        case "sobre":

            $teste = new Controller();

            $teste->abrirSobre();

        break;

        // LOGIN

        case "login":

            $direciona = new Controller();

            $direciona->abrirLogin();

        break;

        case "logar":

            $usuario = new UsuarioController();

            $usuario->logar();

        break;

        case "esqueci-a-senha":

            $direciona = new Controller();

            $direciona->abrirEsqSenha();

        break;

        case "recuperar-senha":

            $direciona = new UsuarioController();

            $direciona->recuperarSenha();

        break;

        case "codigo-de-recuperacao":

            $direciona = new Controller();

            $direciona->abrirRecuperacao($url[1]);

        break;

        case "confirmar-recuperacao":

            $direciona = new UsuarioController();

            $direciona->confirmarCodigo();

        break;

        case "redefinir-senha":

            $direciona = new UsuarioController();

            $direciona->redefinirSenha();

        break;

        // USUÁRIO

        case "home-usuario":

            $usuario = new Controller();

            $usuario->abrirHomeUsuario();

        break;

        case "perfil":

            $usuario = new Controller();

            $usuario->abrirPerfil();

        break;

        case "alterar-senha":

            $usuario = new Controller();

            $usuario->abrirAlterarSenha($url[1]);

        break;

        case "cadastra-animal":

            $animal = new Controller();

            $animal->abrirCadAnimal();

        break;

        case "carregar-raca":

            $raca = new AnimalController();

            $raca->carregarRaca($url[1]);

        break;

        case "cadastrar-animal":

            $animal = new AnimalController();

            $animal->cadastrarAnimal();

        break;

        case "meus-animais":

            $usuario = new Controller();

            $usuario->abrirMeusAnimais();

        break;

        case "atualizar-animal":

            $usuario = new AnimalController();

            $usuario->atualizarAnimal($url[1],$url[2]);

        break;

        case "excluir-animal":

            $usuario = new AnimalController();

            $usuario->excluirAnimal($url[1],$url[2],$url[3]);

        break;

        case "solicitar-castracao":

            $usuario = new UsuarioController();

            $usuario->solicitarCastracao();

        break;

        case"excluir-tutor":

            $usuario = new UsuarioController();

            $usuario->excluir($url[1],$url[2]);

        break;

        // ADM

        case "home-adm":

            $adm = new Controller();

            $adm->abrirHomeAdm();

        break;

        #CADASTROS

        case "cadastra-raca":

            $raca = new Controller();

            $raca->abrirCadRaca();

        break;

        case "cadastrar-raca":

            $raca = new AnimalController();

            $raca->cadastrarRaca();

        break;

        case "cadastra-clinica":

            $clinica = new Controller();

            $clinica->abrirCadClinica();

        break;

        case "cadastrar-clinica":

            $clinica = new ClinicaController();

            $clinica->cadastrarClinica();

        break;

        #ATUALIZAÇÕES

        case "atualiza-tutor":

            $usuario = new UsuarioController();

            $usuario->atualizarUsuario();

        break;

        case "atualiza-raca":

            $raca = new AnimalController();

            $raca->atualizarRaca();

        break;

        #CONSULTAS

        case "consulta-usuario":

            $adm = new Controller();

            $adm->abrirConsultaUsuario();

        break;

        case "consulta-clinica":

            $adm = new Controller();

            $adm->abrirConsultaClinica();

        break;

        case "consulta-castracao":

            $adm = new Controller();

            $adm->abrirConsultaCastracao();

        break;

        case "consulta-animais":

            $adm = new Controller();

            $adm->abrirConsultaAnimais($url[1]);

        break;

        case "consulta-raca":

            $adm = new Controller();

            $adm->abrirConsultaRaca();

        break;

        #EXCLUSÕES

        case "excluir-raca":

            $raca = new AnimalController();

            $raca->excluirRaca($url[1]);

        break;

        // CASTRAÇÃO - vizualização, confirmação e atualização

        case "lista-solicitacao":

            $adm = new Controller();

            $adm->abrirListaSolicitacao();

        break;

        case "agendamento":

            $adm = new Controller();

            $adm->abrirAgendamento($url[1]);

        break;

        case "agendar-clinica":

            $castracao = new UsuarioController();

            $castracao->agendarClinicaCastracao();

        break;

        case "agendar":

            $castracao = new ClinicaController();

            $castracao->agendarDataCastracao();

        break;

        case "atualizar-castracao":

            $castracao = new UsuarioController();

            $castracao->atualizarCastracao();

        break;

        // CLÍNICA

        case "home-clinica":

            $clinica = new Controller();

            $clinica->abrirHomeClinica();

        break;

        case "atualizar-clinica":

            $clinica = new ClinicaController();

            $clinica->atualizarClinica();

        break;

        case "excluir-clinica":

            $clinica = new ClinicaController();

            $clinica->excluirClinica($url[1], $url[2]);

        break;

        // LOGOUT

        case "encerrar-sessao":

            $login = new UsuarioController();

            $login->sair();

        break;

        // TESTE

        case "vazio":

            $teste = new Controller();

            $teste->abrirTeste();

        break;

        default:

            // URL INVÁLIDA

            $direciona = new Controller();

            $direciona->paginaNaoEncontrada();

    }

}

else

{

    //testee

    //ABRIR PÁGINA INICIAL

    $direciona = new Controller();

    $direciona->abrirInicio();

}

?>

# REFERÊNCIAS

COSTA, Sérgio Souza.*Engenharia de Software - Conceitos e Modelos de Desenvolvimento*. 15 nov. 2015. Apresentação em Slideshare. 112 slides. color. Disponível em: <<https://pt.slideshare.net/skosta/aula-engenharia-de-software>>. Acesso em: 14 novembro. 2021.

*Diagramas de Navegação*. IBM, 2021. Disponível em: <<https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=models-browse-diagrams>>. Acesso em: 21 fevereiro. 2022.

GODÓY, Fernando. *3ª Etapa TCC – Estudo de Viabilidade*. Fernando Godoy, maio, 2011.  Disponível em: [<https://fernandogodoy.wordpress.com/2011/05/01/3%C2%AA-etapa-tcc-estudo-de-viabilidade/](https://fernandogodoy.wordpress.com/2011/05/01/3%C2%AA-etapa-tcc-estudo-de-viabilidade/%23:~:text=Viabilidade%20t%C3%A9cnica%3A%20Esse%20estudo%20visa,n%C3%A3o%20as%20necessidades%20do%20cliente.%3e.%20)> Acesso em: 21 fevereiro. 2022.

*Conceito de Requisitos*. Conceito De, 2016. Disponível em: < [https://conceito.de/requisito](https://conceito.de/requisito#:~:text=Requisito%20%C3%A9%20um%20conceito%20que,para%20o%20desenvolvimento%20de%20algo)>. Acesso em: 21 fevereiro. 2022.

VENTURA, Plínio. *O que é Requisito Funcional*. Até o Momento, 11 maio, 2016. Disponível em: <<https://www.ateomomento.com.br/o-que-e-requisito-funcional/>>. Acesso em: 21 fevereiro. 2022.

Guedes, Marylene. *Ciclo de vida do software: por que é importante saber?.* Treina Web, 2017. Disponível em: <<https://www.treinaweb.com.br/blog/ciclo-de-vida-software-por-que-e-importante-saber>>.  Acesso em: 28 fevereiro. 2022.

Braz, Fabricio. *Requisitos de Segurança de Software*. Software Seguro, 29 outubro, 2012. Disponível em: <<http://softwareseguro.blogspot.com/2012/10/requisitos-de-seguranca-de-software.html>>. Acesso em: 20 fevereiro. 2022.

Significados. *Significado de Escopo*. Significados, c2011.

Disponível: <<https://www.significados.com.br/escopo/>>. Acesso em: 28 fevereiro. 2022.

Silva, Andreia. EAP (Estrutura Analítica do Projeto*): o que é, como fazer e qual a diferença entre EAP e Cronograma*. EUAX, 11 de dezembro. 2018. Disponível em: <<https://www.euax.com.br/2018/12/eap-estrutura-analitica-projeto/>>. Acesso em: 28 de fevereiro. 2022.

Silva, Douglas. *Matriz de responsabilidade: o que é e por que usar em seus projetos?*. Zendesk, 4 junho 2021. Disponível em: <<https://www.zendesk.com.br/blog/matriz-de-responsabilidade/>>. Acesso em: 28 fevereiro. 2022.

Espinha, Roberto Gil. *Gráfico de Gantt: o que é, para que serve e como fazer*. Artia, 15 fevereiro. 2021. Disponível em: <<https://artia.com/blog/grafico-de-gantt-o-que-e-para-que-serve-e-como-montar-o-seu/>>. Acesso em: 28 fevereiro. 2022.

*O que é um diagrama UML?*. Lucidchart, 2021. Disponívelç em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml>>. Acesso em: 28 fevereiro. 2022.

*Metodologia Ágil e Scrum*. Metodologia ágil. Disponível em: <<https://metodologiaagil.com/> >. Acesso em: 01 março. 2022.

*Ciclos de Vida do Software*. DevMedia. Disponível em: <<https://www.devmedia.com.br/ciclos-de-vida-do-software/21099>>. Acesso em: 01 março. 2022.

Macêdo, Diego. *Modelos de Ciclo de Vida*. Diego Macêdo. Disponível em: <<https://www.google.com/amp/s/www.diegomacedo.com.br/modelos-de-ciclo-de-vida/amp/>>. Acesso em: 01 março. 2022.

*UML Use Case Diagram Tutorial*. Lucidchart. Disponível em: <<https://www.lucidchart.com/pages/uml-use-case-diagram/#section_0>>. Acesso em: 12 fevereiro. 2022.

*O que é um diagrama de classe UML?*. Lucidchart. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>>. Acesso em: 12 fevereiro. 2022.

*UML Activity Diagram Tutorial.* Lucidchart. Disponível em: <<https://www.lucidchart.com/pages/uml-activity-diagram>>. Acesso em: 12 fevereiro. 2022.

*O que é um diagrama de sequência UML?.* Lucidchart. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml>>. Acesso em: 12 fevereiro. 2022.

Ávila, Rafael. *O que é o diagrama de rede e como usar em um cronograma de projeto.* Academia do Consultor. Disponível em: <<https://academiadoconsultor.com.br/diagrama-de-rede-o-que-e-como-usar>>. Acesso em: 12 fevereiro. 2022.

Maniero. *Qual a diferença entre MER (Modelo de Entidade Relacionamento) e DER (Diagrama de Entidade Relacionamento)?*. Stack Overflow. Disponível em: <<https://pt.stackoverflow.com/questions/328342/qual-a-diferen%C3%A7a-entre-mer-modelo-de-entidade-relacionamento-e-der-diagrama-d>>. Acesso em: 12 fevereiro. 2022.

*Significado de Metodologia.* Significados. Disponível em: <<https://www.significados.com.br/metodologia/>>. Acesso em: 12 fevereiro. 2022.

Equipe editorial de Conceito De. Conceito de manual do utilizador. Conceito.de. Disponível em: <https://conceito.de/manual-do-utilizador<https://conceito.de/manual-do-utilizador>> Acesso em: 12 fevereiro. 2022.

*Qual a importância do Manual Técnico para o seu produto?.* PJ Consultoria. Disponível em: <https://pjufmg.com.br/qual-a-importancia-do-manual-tecnico-para-meu-produto/>. Acesso em: 12 fevereiro. 2022.

Visual studio code: https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar

USBWebserver:  https://www.usbwebserver.net/webserver/

DIA: https://sourceforge.net/projects/dia-installer/

Draw.io: https://www.techtudo.com.br/tudo-sobre/drawio.html

Photoshop CS6: https://www.vivadecora.com.br/pro/photoshop/

Google Forms: https://blog.safetec.com.br/colaboracao/como-usar-google-forms/

Trello: https://www.techtudo.com.br/listas/2019/10/como-funciona-o-trello-saiba-tudo-sobre-programa-para-organizar-projetos.ghtml

Excel: https://www.infoescola.com/informatica/introducao-ao-excel-e-operacoes-basicas/

PowerPoint: https://mgtek.com.br/lages/blog/microsoft-powerpoint/

HTML: https://www.totvs.com/blog/developers/o-que-e-html/

CSS: https://www.totvs.com/blog/developers/o-que-e-css/

PHP: https://www.php.net/manual/pt\_BR/intro-whatis.php

GitHub: <https://docs.github.com/pt/get-started/quickstart/hello-world>

Canva: https://www.canva.com/pt\_br/about/