Universidade Estácio de Sá

curso Análise e Desenvolvimento de sistemas

UNIDADE: NOVA AMÉRICA

**TRABALHO DE SEMESTRE**

**EM desenvolvimento rápido**

**de aplicações em python**

Rio de Janeiro - RJ

Maio / 2024

202302198521 – Fabíola Dutra dos Santos

**Trabalho de semestre**

**em Desenvolvimento Rápido de Aplicações em Python**

Trabalho de Desenvolvimento Rápido de Aplicações em Python apresentado a Universidade Estácio de Sá, como exigência para avaliação na disciplina Desenvolvimento Rápido de Aplicações em Python - ARA0095

Orientador:

Prof. Ronaldo Candido dos Santos

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc167798222)

[1.1 DESCRIÇÃO DO PROBLEMA 3](#_Toc167798223)

[1.2 OBJETIVOS 3](#_Toc167798224)

[2 DESENVOLVIMENTO 4](#_Toc167798225)

[2.1 MODELAGEM DE DADOS: IMPLEMENTAÇÃO DO BANCO DE DADOS 4](#_Toc167798226)

[2.2 DETALHES DA IMPLEMENTAÇÃO 5](#_Toc167798227)

[2.3 DESIGN DE INTERFACE COM O USUÁRIO 6](#_Toc167798228)

[3 CONCLUSÃO 14](#_Toc167798229)

[REFERÊNCIAS 15](#_Toc167798230)

# INTRODUÇÃO

Usar do Desenvolvimento Rápido de Aplicações em Python para esboçar um modelo de cadastro de pets para o sistema de agendamento para um pet shop que atende o serviço de banho e tosa. O objetivo é criar um modelo simples, porém com eficiência e de uso intuitivo para o usuário e administrador do sistema em sua loja, sem pedir por muitos dados pessoais de cada cliente, porém mantendo a segurança das informações fornecidas.

## DESCRIÇÃO DO PROBLEMA

Para esse projeto, o objetivo e suas funções já eram bem claras, o impasse foi no momento de definir, no momento da construção do código fonte, quais as estruturas seriam necessárias: se somente uma janela de login e uma janela de cadastro. Porém depois foi percebida a falta de uma janela para exibir os dados de cadastro do cliente e as informações atreladas a ele, logo depois do login. E junto a isso, todas as soluções para a construção do código, que atendesse as necessidades do projeto pensado.

O projeto pediu diversos imports e bibliotecas e vários métodos em Def para criar todas as janelas, condicionais para comparações e duas tabelas para o banco de dados.

## OBJETIVOS

Criar tela de login e cadastro para clientes de um pet shop que querem cadastrar seus pets para o atendimento de banho e tosa em uma loja. Por meio de tela de cadastro e login, o tutor/responsável do pet poderia cadastrar seu pet, que já ficaria atrelado aos dados do tutor, e o tutor poderia adicionar mais pets para seu cadastro.

Em um desenvolvimento mais abrangente do projeto, o tutor poderia adicionar a raça e espécie dos pets, e descrever necessidades específicas de cada pet, ou adicionar mais um responsável para animal, consultar dias e horários disponíveis para agendamentos de serviços e agendar com mais rapidez e facilidade.

O objetivo principal é facilitar o cadastro dos animais e as informações e necessidades, orientadas pelos próprios tutores. Diminuir a dependência dos tutores para agendamentos somente em horário de funcionamento da loja, criar a facilidade do cliente em ver os dias e horários disponíveis mais convenientes, agendar para atendimento recorrente, evitaria a sobreposição de horários em agendas, e tiraria a necessidade de uma pessoa sempre para atendimento dos agendamentos, focando nas confirmações das agendas, e organização das rotinas do local de trabalho para recepção dos clientes para o atendimento.

# DESENVOLVIMENTO

O projeto começou a ser desenvolvido pelo Frontend, desenvolvendo as janelas de cadastro e janelas de login. Para isso houve a necessidade da importação de bibliotecas e ferramentas como o TKinter e o Custom Tkinter.

Foi usado métodos e classes para uma usabilidade e organização melhor do código e ser legível de maneira mais universal na orientação a objetos.

As telas principais para cadastro e login foram criadas com formas mais arredondas e temas claros para serem mais amigáveis para o usuário

Depois de criar as duas telas para cadastros e login, foi observada a necessidade de uma tela principal

Foi criada também a necessidade de apagar os itens escritos nas telas anteriores, e limpar as telas na intermediação e um botão de retorno, que leve de volta a tela inicial, e caixas de pop up com avisos para o usuário.

A maior dificuldade atual é o posicionamento e alinhamento de imagens de referência, títulos e frases de indicação e orientação das telas de cadastro e login, sem que atrapalhe as telas seguintes ou desconfigure o layout das janelas.

## MODELAGEM DE DADOS: IMPLEMENTAÇÃO DO BANCO DE DADOS

Para a implementação do banco de dados foi optado por criar uma classe chamada Backend, usando o SQLite3, que também precisou ser importado.

Foram criadas duas tabelas: para o cadastro dos usuários e para o cadastro dos pets relacionados aos usuários, e que permitirá o cadastro de mais de um pet e alteração de dados.

A tabela de usuários armazena nome, e-mail em PRIMARY KEY e que será usado para o login, e senha que é armazenada com criptografia usando a biblioteca haslib.

As tabelas se relacionam 1:N : Um para Muitos.

Desenho com traços pretos em fundo branco

Descrição gerada automaticamente com confiança média

Cada pet tem uma FOREING KEY e um Id, a quantidade de pets é ilimitada, mesmo que com os mesmos nomes, mas Usuário é único por e-mail. Todo pet leva o Id do Usuário relacionado, sendo ‘Pet’ a entidade fraca, por não existir sozinha, e ‘Usuário’ a entidade forte, e também se usou o comando ‘CONSTRAINT’ na tabela de Pets, para permitir apenas um Id de Usuário para o Pet relacionado.

## DETALHES DA IMPLEMENTAÇÃO

Para o desenvolvimento desse projeto foi decidido separar em duas classes de nomes bem objetivos para orientação clara no código:

Backend, onde foram desenvolvidas e organizadas as tabelas do banco de dados. Foram criadas duas tabelas, tabela para clientes e tabelas para pets, relacionadas.

Para a implementação do banco de dados, além da importação do ‘SQLite3’, também foi necessária a importação da biblioteca ‘hashlib’ para a criptografia das senhas dos usuários.

Frontend, onde foram desenvolvidas as janelas e telas para interação com o usuário.

Foi usado o framework ‘TKinter’, padrão do próprio Python, para a criação gráfica da interface de usuário (GUI). Do ‘TKinter’ foi usado o módulo ‘tkinter’ para funções e classes básicas para a criação das janelas de login e cadastro, os botões e os campos de entrada de dados do usuário e seus pets.

Para a criação de uma interface de aparência mais moderna e atualizada, foi importada a extensão ‘customTKinter’, que facilitou a personalização das janelas, botões e caixas de entrada. Para esses, foi escolhido bordas arredondadas e cores suaves, design que acompanhou as caixas de mensagens e seus botões, as bordas dos frames e janelas. Essas caixas de mensagens, também chamadas de pop-ups, foram criadas com o módulo ‘tkinter.messagebox’, que fornece a função para exibir essas caixas com mensagens e alertas. E para essas janelas terem função no Windows, foi usado o módulo ‘ctypes’, biblioteca’user32’ e função ‘MessageBoxW’. Essas mesmas caixas exibem mensagens para alertar sobre o não preenchimento de campos de cadastro, senhas em tamanho menor que a pedida na caixa de entrada para senhas e também senhas não idênticas, logins inválidos, sucesso no cadastro do usuário, tela de exibição e correção e exclusão dos nomes dos pets.

Para lidar com a ilustração da tela de login, houve a necessidade da extensão ‘tkinter.photoImage.

Na tela de cadastro, exibição, correção e exclusão dos nomes dos pets foi usada a extensão ‘CTKTable’, facilitando a apresentação de várias células na mesma tela, o que facilita a usabilidade do usuário.

## DESIGN DE INTERFACE COM O USUÁRIO



Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente

# CONCLUSÃO

O projeto foi desenvolvido pensando em facilitar a rotina de agendamentos de banho e tosa, dado mais autonomia para o cliente e facilitar a loja ou o profissional, visto que também poderia ser adotado por profissionais autônomos. O design foi pensado para ser amigável, limpo e intuitivo, o que ajuda o usuário a não ter dificuldades de encontrar os campos e botões que executam as finalidades do projeto.

Foi desenvolvido como um projeto de execução e aparência simples, mas conta com mais métodos, funções e bibliotecas imaginadas no início do desenvolvimento, exatamente para que toda a aplicação rode sem problemas de execução, e pensando na mínima segurança do possível usuário, salvo proporções e necessidades.

Uma possível evolução dele, poderia ser o desenvolvimento para um aplicativo para smartphones, que poderiam integrar e buscar por GPS profissionais ou locais para atendimento cadastrados no aplicativos, e o cliente ter a liberdade de buscar a data e horário mais convenientes para ele, sem a necessidade de entrar em contato com cada estabelecimento ou profissional e aguardar as respostas de cada um.

Com possíveis expansões de funcionalidades da aplicações, a automatização desse serviço facilitaria o dia a dia de quem busca um serviço de atendimento para seu pet e de quem atende, tendo mais tempo livre para outras rotinas.

# REFERÊNCIAS

Stack Overflow. “[How to compare plain text password to hashed password using bcrypt?](https://stackoverflow.com/questions/9559549/how-to-compare-plain-text-password-to-hashed-password-using-bcrypt)”

Disponível em: < https://stackoverflow.com/questions/9559549/how-to-compare-plain-text-password-to-hashed-password-using-bcrypt>

@setprogramacao. Sistema de Login Completo. Youtube, Abril/2023. Disponível em: <https://www.youtube.com/@setprogramacao>

REIS, Fábio dos. Interface Gráfica com Tkinter em Python. Youtube, Março/2024. Disponível em: <https://www.youtube.com/@bosontreinamentos>

Reis, Fábio dos. Bóson Treinamentos. Disponível em:

<https://www.bosontreinamentos.com.br>. Acesso em: 14 maio 2024.

NORMAS ABNT. Normas para elaboração de trabalhos acadêmicos. Disponível em: <https://www.normasabnt.org/>. Acesso em: 28 mai. 2021.