

# Memorial Descritivo de Entrega de Software

## 1. Introdução

O projeto foi desenvolvido com o objetivo de tornar eletrônico o cadastro de pacientes na rede de saúde a fim de agilizar e simplificar o processo que é feito atualmente no prontuário físico. O mesmo compõe nota na matéria de desenvolvimento rápido e aplicações em python do docente Roney Camargo.

## 2. Escopo do Projeto

O software conta com as funções de cadastro de paciente, armazenamento de dados, consultas e edição de informações e receituário de medicamento personalizado do paciente. Funções essas similares ao funcionamento de um CRUD.

Funcionalidades como o resultado de exames foi descartado do projeto.

## 3. Especificações Funcionais

Tela de login:

- Login - forma de identificação para entrar no sistema, a entrada é uma string.
- Senha - forma de segurança para conectar no sistema, a entrada é uma string de números.

Cadastro do Paciente:

- CPF – Forma de identificação única, a entrada é uma string de números.
- Nome – Forma de direcionamento, a entrada é uma string de letras.
- Nascimento – Data de nascimento da pessoa, pode ser preenchida manualmente em formato de data ou selecionado via calendário disponível no sistema.
- Telefone – Meio para contato, a entrada string de número.
- Endereço - Local para contato da pessoa, string com números e letras.
- Sexo - Identificação pessoal de sexo, entrada é um campo de seleção com três opções.
- Observações - Campo livre para preenchimento de qualquer informação que sentir necessária, entrada em string.
- Medicamento – Campo para adicionar os medicamentos utilizados por aquele paciente, entrada em string.

- Dosagem – Campo para quantidade manipulada do medicamento anteriormente preenchido, entrada em string.
- Peso: campo para adicionar o peso em quilogramas (kg), entrada string.
- Altura: campo para adicionar altura em centímetros (cm), entrada string.

Saída gerada: Após a conclusão do cadastro do paciente é retornado uma mensagem de êxito caso o cadastro esteja correto ou uma mensagem de erro caso falte alguma informação, finalizado com êxito os dados são conglomerados no banco de dados do sistema, ficando disponíveis para consulta, edição ou exclusão.

### Consulta:

- CPF de Paciente – Campo para identificação do paciente que será marcada a consulta, entrada em string.
- Motivo – O que levou a marcação da consulta do paciente, entrada em string.
- Tipo do Médico - Especialidade do profissional mais indicada para atendimento do paciente, entrada é uma seleção com seis opções de escolha.
- Data – Data da consulta, pode ser preenchida manualmente em formato de data ou selecionado via calendário disponível no sistema.
- Horário - Intervalo de horário para realização da consulta, entrada é uma seleção com oito opções de escolha.

Saída gerada: Após a conclusão da marcação de consultas, é retornado uma mensagem de êxito e fica documentado no sistema a visualização do resultado da marcação da consulta.

## 4. Tecnologias Utilizadas

A linguagem usada no desenvolvimento do projeto foi Python, o framework utilizado foi o CustomTkinter e as bibliotecas utilizadas foram: tkinter, tkcalendar e sqlite3.

## 5. Requisitos de Sistema

Uma cpu que suporte rodar projetos Python, normalmente qualquer máquina atual consegue realizar essa tarefa.

## 6. Procedimentos de Instalação

Utilizar algum programa que rode projetos na linguagem Python. Já dentro do programa, para rodar o sistema será necessário a importação das três bibliotecas (CustomTkinter, tkcalendar e sqlite3) que é feita toda no próprio código através dos comandos “pip install customtkinter e pip install tkcalendar”.

## 7. Manual do Usuário

### Tela de Login:

Ao executar o sistema será observado uma tela de login que deverá ser preenchida para entrar no sistema.

### Tela Inicial:

No momento em que iniciar o sistema você verá uma tela com campos preenchíveis.

Existem duas abas principais: "Paciente" e "Consulta".

Na aba "Paciente", você pode adicionar, buscar, atualizar e deletar informações sobre os pacientes.

Na aba "Consulta", você pode visualizar o histórico de consultas, buscar consultas por CPF do paciente e desmarcar consultas.

### Funcionalidades:

**Adicionar Paciente:** Preencha os campos obrigatórios e clique no botão "Adicionar Paciente" para adicionar um novo paciente ao banco de dados.

**Buscar Paciente:** Insira o CPF do paciente e clique no botão "Buscar Paciente" para recuperar suas informações e preencher os campos correspondentes.

**Atualizar Paciente:** Preencha os campos desejados e clique no botão "Atualizar Paciente" para atualizar as informações do paciente com base no CPF fornecido.

**Deletar Paciente:** Insira o CPF do paciente e clique no botão "Deletar Paciente" para remover o paciente do banco de dados.

**Histórico de Consulta:** Adicione consultas associadas a um paciente, fornecendo os detalhes da consulta e clicando no botão "Histórico de Consulta".

**Buscar Consulta:** Insira o CPF do paciente e clique no botão "Buscar Consulta" para recuperar informações sobre consultas agendadas para esse paciente.

**Desmarcar Consulta:** Insira o CPF do paciente e clique no botão "Desmarcar Consulta" para remover uma consulta agendada para esse paciente.

### Encerramento:

Feche a tela quando terminar de usar o sistema.

## 8. Testes e Validação

Teste Unitário: Foi testado cada função separadamente, por exemplo, no campo CPF foram realizados testes para os tipos de entrada e saída, no campo "Nascimento" foi testado a integração com a biblioteca "tkcalendar" para entrada de dados, no geral os campos seguintes representam o mesmo retorno pois seguindo a sequencia logica a resposta será similar ao anteriores.

**Teste de Integração:** Foi realizado o teste com o cadastro de um paciente completo e em seguida foi utilizado a função de buscar paciente para verificar se os dados anteriormente cadastrados, estavam corretos no sistema.

**Teste de Sistema:** Concluído o cadastro de um paciente, foi então feita a edição de alguns dados e em seguida marcação de consulta e busca de consulta do mesmo.

**Teste de Aceitação:** Foi realizado um teste de aceitação, utilizando todas as funções do sistema, testando erros e acertos para possíveis bugs.

## **9. Documentação do Código**

### **Documentação do Sistema de Saúde**

#### **Introdução**

O Sistema de Saúde é uma aplicação de gerenciamento de informações de pacientes e consultas, desenvolvida em Python utilizando a biblioteca CustomTkinter para a interface gráfica.

#### **Funcionalidades Principais**

##### **1. Gerenciamento de Pacientes**

- Adicionar novo paciente ao sistema.
- Buscar informações de pacientes existentes.
- Atualizar informações de pacientes.
- Deletar pacientes do sistema.

##### **2. Gerenciamento de Consultas**

- Adicionar consultas associadas a um paciente.
- Buscar consultas agendadas para um paciente.
- Desmarcar consultas agendadas.

#### **Requisitos**

- Python 3.x
- Bibliotecas: CustomTkinter, tkcalendar, sqlite3

#### **Como Executar**

Instale as dependências utilizando o seguinte comando:

```
pip install customtkinter tkcalendar
```

Execute o código Python:

```
python nome_do_arquivo.py
```

## Interface do Usuário

O sistema possui uma interface gráfica com duas abas principais:

- **Paciente:** Para gerenciamento de informações de pacientes.
- **Consulta:** Para gerenciamento de consultas médicas.

## Banco de Dados

O sistema utiliza um banco de dados SQLite chamado "sys\_saude.db" para armazenar informações de pacientes e consultas.

## Testes

O código pode ser testado utilizando diferentes abordagens de teste, incluindo:

- Testes Unitários
- Testes de Integração
- Testes de Sistema
- Testes de Aceitação

## Limitações

- Não há implementação de autenticação de usuários.
- A interface gráfica pode ser melhorada para uma melhor experiência do usuário.
- Não há validação de entrada de dados.

## 10. Planos de Manutenção

No futuro temos o intuito de aumentar a capacidade do software como um todo para a vinda de mais usuários. Para que não ocorram erros futuros pela falta de espaço ou capacidade de atendimento temos como objetivo ampliar o suporte do sistema atual.

## 11. Referências

Sistema de prontuário eletrônico usado em algumas unidades de saúde do estado da Bahia:

<https://www.hcpa.edu.br/institucional/institucional-apresentacao/tecnologia-da-informacao-e-comunicacao/institucional-sistema-aghuse>.

## 12. Considerações Finais

O desenvolvimento desse projeto nos estimulou a utilizar o framework TKinter, além

das bibliotecas tkcalendar e sqlite3.

O propósito inicial do projeto para nós como equipe foi alcançado, sendo possível o cadastro/edição de um paciente como prontuário eletrônico além da marcação e busca de informações sobre a consulta do mesmo no próprio sistema.

No futuro acredito que seria sensato a adição de outras funções no sistema, para abranger o atendimento e as possibilidades dentro do prontuário eletrônico.