

Documentação do Sistema de Fidelidade da Açaíteria e Sorveteria SACIAR

Versão 1.1 – Outubro 2025

1. Introdução

Este documento descreve o sistema de fidelidade para a açaíteria e sorveteria SACIAR, que visa gerenciar clientes, pontos e recompensas. O sistema é composto por:

- **App em Flutter para tablet (Galaxy Tab E)** — usado pelos clientes para registrar CPF e acompanhar pontos.
- **Programa de gerenciamento em desktop (CustomTkinter)** — usado pelos funcionários para controlar clientes e pontos.
- **Backend em Python (FastAPI)** — centraliza toda a lógica do sistema e faz a comunicação com o banco de dados.
- **Banco de dados MongoDB (local)** — armazena informações dos clientes.

2. Objetivo do Sistema

- Registrar clientes e suas informações básicas (nome, CPF, telefone).
- Gerenciar pontos de fidelidade de cada cliente.
- Informar quando o cliente ganhou um açaí grátis.
- Permitir que os funcionários atualizem e controlem clientes e pontos.
- Exportar dados dos clientes para Excel para relatórios e controle externo.
- Garantir segurança e backup periódico dos dados.

3. Estrutura do Sistema

3.1 App em Flutter (Tablet)

- **Dispositivo:** Samsung Galaxy Tab E
- **Tamanho da tela:** 1280 x 800 px (9,6")
- **Funcionalidades:**
 - Digitar CPF do cliente e consultar no banco de dados
 - Criar cliente caso não exista (nome e telefone)

- Mostrar pontos e informar quando o cliente ganha um açaí grátis e quando poderá usar os pontos (3, 6, 8 e 10 pontos)
- Adicionar 1 ponto automaticamente caso o cliente não use seus pontos, com máximo de 10
- Comunicação com o backend via requisições HTTP
- **Wireframe (design):** [APP SACIAR](#)

3.2 Programa de Gerenciamento (Desktop)

- **Tecnologia:** Python com CustomTkinter
- **Funcionalidades:**
 - Listar clientes com informações: Nome, CPF, Telefone, Pontos
 - Adicionar, editar e remover clientes
 - Atualizar pontos manualmente
 - Filtrar clientes por nome ou CPF
 - Exportar lista de clientes para CSV ou Excel (implementação em Python usando biblioteca Pandas)
- **Segurança e Backup:**
 - Acesso restrito aos funcionários autorizados
 - Possibilidade de backup manual ou automático do banco de dados
- **Observação:** Não há integração direta com WhatsApp; mensagens serão enviadas manualmente.
- **Wireframes (design):** [Sistema de Gerenciamento SACIAR](#)

3.3 Backend

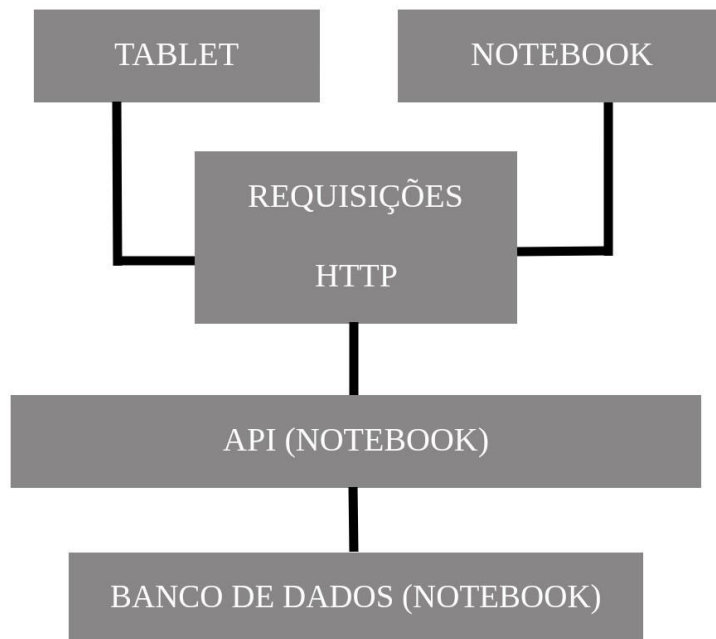
- **Tecnologia:** Python (FastAPI)
- **Funções:**
 - Endpoints para CRUD de clientes (GET, POST, PUT, DELETE)
 - Lógica de pontos e verificação de açaí grátis
 - Conexão com banco de dados MongoDB
 - Recebe requisições tanto do app quanto do desktop

3.4 Banco de Dados

- **Tecnologia:** MongoDB (local, rodando no notebook)
- **Coleção:** clientes
- **Estrutura do documento:**

```
{  
  "_id": ObjectId,  
  "nome": "Nome do Cliente",  
  "cpf": "123.456.789-00",  
  "telefone": "(11) 99999-9999",  
  "pontos": 0  
}
```

4. Fluxo do Sistema



(Representação de como os sistemas interagem entre si)

5. Regras de Negócio

- Cada cliente possui uma pontuação inicial de 0 pontos.
- Cada compra adiciona 1 ponto ao cliente.
- Quando a soma de pontos + 1 atingir 10 pontos, o cliente ganha um açaí grátis e, caso opte por usar os pontos, eles são zerados.
- O cliente pode optar por usar os pontos atuais para ganhar o açaí antes de completar 10 pontos, zerando sua pontuação, sendo os possíveis pontos 3, 6 e 8.
- Todas as regras de pontos são calculadas no backend.

6. Segurança e Backup

- **Autenticação do sistema de gerenciamento:**
O sistema de gerenciamento terá uma única conta de acesso, armazenada no banco de dados com senha segura. Ao iniciar o sistema, o funcionário deverá fazer login para acessar os dados completos dos clientes. Isso garante que apenas pessoas autorizadas possam manipular os registros e pontuações.
- **Proteção do MongoDB:**
O banco de dados local MongoDB terá autenticação habilitada. Somente usuários cadastrados e com senha poderão acessar os dados. Dessa forma, mesmo que alguém tente acessar o banco diretamente, será necessário fornecer credenciais válidas.
- **Acesso do app:**
O app em tablet não possui autenticação e só poderá consultar os dados relacionados ao cliente que inserir o CPF. Ele **não tem acesso a todos os registros**, garantindo que dados de outros clientes permaneçam protegidos.
- **Responsabilidade sobre segurança física e digital:**
A proteção do notebook onde o banco e o sistema de gerenciamento estão hospedados é de responsabilidade do dono do equipamento. O sistema garante que, enquanto o notebook estiver seguro, os dados não poderão ser acessados indevidamente.
- **Backup dos dados:**
É recomendada a realização periódica de backups do banco de dados MongoDB, de preferência automáticos, para evitar perda de informações em caso de falha de hardware ou outro incidente.
- **Proteção dos endpoints do backend:**
Os endpoints (portas de entrada do backend) que retornam todos os dados dos

clientes só podem ser acessados pelo sistema de gerenciamento, garantindo que terceiros não consigam obter informações sensíveis através da API.

7. Glossário / Legenda

- **Wireframe:** Representação visual simplificada das telas do sistema, mostrando layout, botões e campos sem necessidade de funcionalidade.
- **MongoDB:** Banco de dados NoSQL que armazena informações em documentos JSON.
- **CRUD:** Sigla para Create, Read, Update, Delete — operações básicas de banco de dados.
- **Backend:** Parte do sistema que processa a lógica, recebe requisições e interage com o banco de dados.
- **API:** Interface que permite a comunicação entre o app, desktop e o backend.
- **Exportar para Excel:** Funcionalidade que permite gerar relatórios em planilhas a partir dos dados do sistema.
- **Backup:** Cópia de segurança dos dados armazenados no banco, podendo ser manual ou automática.

8. Observações Finais

- O envio de mensagens para clientes via WhatsApp será feito manualmente, sem integração automática.
- O sistema foi pensado para ser simples, rápido de usar e de fácil manutenção, atendendo a necessidade da SACIAR de gerenciar clientes e pontos localmente.
- Futuras melhorias podem incluir migração do MongoDB para nuvem e envio automático de mensagens.