

## **Documentação da Aplicação "Know Your Fan - FURIA"**

**Desenvolvido para processo seletivo**

<b>Data:</b>	<b>02/04/2025</b>
<b>Elaborado por:</b>	<b>Felipe Augusto Sousa de Almada</b>
<b>Versão:</b>	<b>1.0</b>

## Resumo

Know Your Fan é uma aplicação web desenvolvida com **Streamlit** que permite o cadastro e análise de fãs da organização FURIA, utilizando validação documental por **OCR** e integração com a API do Twitter.

O sistema realiza:

- Coleta de dados pessoais do fã (**nome, CPF, e-mail, etc.**).
- Validação automática de documentos, extraindo texto via **OCR** e comparando com os dados cadastrados.
- Análise do perfil do Twitter, verificando se o usuário:
  - Menciona a FURIA na bio.
  - Publicou tweets sobre a organização.
- Coleta de dados e Registro de atividades em e-sports (eventos e compras).
- Geração de recomendações personalizadas com **IA (Gemini)**, com base no perfil do fã.

A aplicação organiza essas informações em abas, permitindo que cada etapa seja realizada de forma independente e intuitiva. Todos os dados coletados são salvos localmente em um arquivo **.csv** e **.json**.

## **Sumário**

1. Objetivos do Sistema
2. Estrutura da Interface
3. Segurança e Privacidade
4. Tecnologias Utilizadas
5. Arquivos Gerados
6. Fluxo de Cadastro
7. Requisitos para rodar a aplicação

## 1. Objetivos do Sistema

- Realizar cadastro de fãs com dados pessoais e sociais.
  - Validar a identidade via **OCR** de documentos (RG/CNH).
  - Analisar o engajamento no Twitter, verificando relação com a FURIA.
  - Gerar recomendações personalizadas baseadas em interações com eventos e compras.
- 

## 2. Estrutura da Interface

A interface é dividida em quatro abas principais:

### 2.1 Cadastro

- Coleta: nome, CPF, endereço e e-mail.
- Campos obrigatórios: Nome, CPF e E-mail.
- Os dados são armazenados em `st. session_state` para uso nas demais etapas.

### 2.2 Documento

- Upload de imagens (frente e verso) de documentos.
- Utiliza o Tesseract **OCR** para extrair texto.
- Valida se o nome e **CPF** aparecem no texto extraído.
- As imagens são salvas localmente no diretório `documentos/`.

### 2.3 Redes Sociais

- O usuário informa um perfil do Twitter.
- **São analisados:**
  - Se menciona "FURIA" na bio;
  - Se segue a conta oficial da FURIA;
  - Tweets recentes e contas seguidas.
- **Requisições feitas com a Twitter API v2 via token Bearer.**

### 2.4 Atividades

- Coleta informações sobre eventos participados e compras feitas.

- Usa a Google Gemini API (via `google.generativeai`) para gerar sugestões com base no perfil do fã.

---

### 3. Segurança e Privacidade

- Chaves sensíveis (`genai_api_key`, `bearer_api_token`) são acessadas via `st.secrets`.
- Os documentos são salvos com nomes únicos (baseados no CPF).
- Dados do usuário são salvos em `dados_fas.csv` apenas após o clique no botão "Salvar Cadastro Completo".

---

### 4. Tecnologias Utilizadas

Componente	Descrição
Streamlit	Interface interativa em Python
Pandas	Manipulação de dados tabulares (CSV)
Tesseract OCR	Extração de texto de imagens
PIL (Pillow)	Processamento de imagens
Twitter API v2	Análise de perfis, tweets e relações sociais
Google Gemini API	Geração de conteúdo recomendativo via IA

---

### 5. Arquivos Gerados

- CSV: **`dados_fas.csv`** — armazena os dados consolidados dos fãs.
- Imagens: armazenadas em **`documentos/{cpf}_frente.jpg`** e **`documentos/{cpf}_verso.jpg`**.

---

### 6. Fluxo de Cadastro

1. O usuário preenche os dados pessoais.
2. Faz o upload dos documentos.
3. A aplicação valida o nome e **CPF** via **OCR**.
4. Insere seu Twitter para análise.

5. Informa atividades relacionadas a e-sports.
6. Gera recomendações personalizadas.
7. Salva todos os dados em arquivo **CSV** e **JSON** ao final.

---

## 7. Requisitos para rodar a aplicação

### 1. Instalar o Python

- Baixe e instale o Python 3.8+.
- Certifique-se de marcar a opção **"Add Python to PATH"** na instalação.

### 2. Instalar o Tesseract OCR

- Faça o download do instalador em: <https://github.com/tesseract-ocr/tesseract>
- Instale e copie o caminho da instalação, ex:  
C:\Program Files\Tesseract-OCR\tesseract.exe

### 3. Instalar dependências Python

- No terminal ou prompt de comando, crie um ambiente virtual (opcional, mas recomendado) e instale os pacotes:

***pip install streamlit pandas pillow pytesseract unidecode requests google-generativeai***

### 4. Credenciais necessárias

Crie um arquivo chamado .streamlit/secrets.toml no mesmo diretório da aplicação com o seguinte conteúdo (substitua com suas chaves reais):

[general]

**genai\_api\_key** = "SUA\_CHAVE\_DO\_GENAI"

**bearer\_api\_token** = "SEU\_BEARER\_TOKEN\_TWITTER"

A **genai\_api\_key** pode ser obtida em: <https://makersuite.google.com/app>

O **bearer\_api\_token** é a chave da API v2 do Twitter.

## 5. Rodar o aplicativo

No terminal, execute:

***streamlit run nome\_do\_arquivo.py***

Substitua nome\_do\_arquivo.py pelo nome real do seu script, por exemplo:

***streamlit run app.py***

---

## 6. Arquivos gerados automaticamente

- dados\_fas.csv: registros dos usuários cadastrados.
  - Pasta documentos/: onde os arquivos de imagem dos documentos são salvos.
- 

## 7. Recomendações

- Use imagens legíveis e claras para melhor extração de texto (OCR).
- Certifique-se de que o CPF e o nome estejam corretos e constem no documento enviado.
- A análise de Twitter pode falhar se o perfil for privado ou o token estiver incorreto.