



**UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS JARDINS DE ANITA - ITAPAJÉ**

Relatório Final

Turma CD VI

Deep Learning

Itapajé, Janeiro, 2026



Conteúdo

1	Introdução	3
2	Metodologia	3
2.1	Modelo das Flores	3
2.1.1	Sobre os dados	3
2.1.2	Pré-processamento das Imagens	4
2.1.3	Arquitetura do Modelo	4
2.1.4	Treinamento	4
2.1.5	Procedimento de Inferência	5
2.2	Modelo da Flora	5
2.2.1	Sobre os dados	5
2.2.2	Pré-processamento das Imagens	5
2.2.3	Arquitetura do Modelo	5
2.2.4	Treinamento	6
2.3	Modelo dos pássaros	6
2.3.1	Sobre os dados	6
2.3.2	Pré-processamento dos Áudios	6
2.3.3	Extração de Espectrogramas Mel	6
2.3.4	Arquitetura do Modelo	7
2.3.5	Treinamento	7
3	Resultados	8
3.1	Modelo Flores	8
3.2	Modelo Flora	8
3.3	Modelo Pássaros	9
4	API de Inferência e Interface Web	9
4.1	API de Inferência	9
4.2	Interface Web	10
4.3	Página Inicial	10
4.4	Módulo de Identificação de Flores	10
4.5	Módulo de Identificação da Flora	10
4.6	Módulo de Classificação de Pássaros	12
4.7	Fluxo de Comunicação API → Interface	12
5	Conclusões	13



1 Introdução

O projeto *ItapajéBio* tem como objetivo desenvolver um sistema inteligente para identificação automática da biodiversidade do município de Itapajé, com foco na classificação de flores, espécies vegetais e cantos de pássaros característicos da região. A iniciativa integra ciência de dados, aprendizado profundo e conservação ambiental, oferecendo uma ferramenta acessível que contribui tanto para atividades de pesquisa quanto para ações educativas e de valorização do patrimônio natural local.

A plataforma busca facilitar a catalogação de espécies e o reconhecimento automatizado por parte de pesquisadores, estudantes, ecoturistas e moradores, promovendo uma experiência mais informativa e interativa acerca da fauna e flora itapajeense. Além do caráter tecnológico, o projeto também reforça a importância da educação ambiental e do ecoturismo como instrumentos de preservação, documentação e engajamento comunitário.

Este relatório apresenta o desenvolvimento dos modelos de inteligência artificial empregados na classificação de três tipos de dados: imagens de flores, imagens de espécies vegetais (flora) e sinais acústicos de pássaros. Para cada modalidade, foram adotadas arquiteturas de redes neurais distintas, visando acomodar as especificidades de cada domínio. No caso das flores, foi empregada a arquitetura *MobileNetV2*, uma CNN (*Convolutional Neural Network*) eficiente baseada em convoluções separáveis em profundidade, adequada para cenários com menor custo computacional e integração em aplicações web e dispositivos móveis. Para o reconhecimento de flora, utilizou-se a *MobileNetV3*, versão mais moderna da mesma família, com maior eficiência computacional e capacidade de generalização. Por fim, para o reconhecimento de pássaros a partir de áudio, implementou-se um modelo híbrido baseado em *CNN + LSTM*, aproveitando CNNs para extração de padrões espectrais e LSTMs para modelagem de dependências temporais.

Ao longo do relatório, são detalhados os procedimentos de coleta e preparação dos dados, definição arquitetural, experimentação e avaliação dos modelos, bem como as justificativas técnicas para as escolhas realizadas. São apresentadas também as métricas relevantes para cada tarefa, incluindo acurácia, precisão, revocação e *F1-score*, além de visualizações e matrizes de confusão que auxiliam na análise interpretativa dos resultados.

A consolidação desta etapa representa um avanço significativo no desenvolvimento do *ItapajéBio*, estabelecendo uma base sólida para futuras extensões, como a integração dos modelos em aplicações interativas, ampliação do escopo taxonômico e o emprego de estratégias de otimização em produção.

2 Metodologia

2.1 Modelo das Flores

2.1.1 Sobre os dados

A base de dados destinada à identificação de flores foi construída com foco na biodiversidade local do município de Itapajé (CE) e regiões de Caatinga, contemplando variações naturais de iluminação, distância de captura, ângulo, estágio fenológico e presença de ruídos de fundo (folhas, galhos, solo e sombras). O conjunto foi organizado em um problema de classificação multiclasse, no qual cada imagem pertence a exatamente uma espécie-alvo.

Ao final, a base contemplou **15** classes (espécies) e **255** imagens, distribuídas de forma **balanceada**. Para mitigar baixo aprendizado e melhorar a capacidade de generalização, foi utilizada estratégia de *data augmentation* durante o treinamento, ampliando a diversidade efetiva de exemplos sem a necessidade de novas coletas.



2.1.2 Pré-processamento das Imagens

As imagens foram padronizadas para o formato RGB e preparadas para entrada em uma rede pré-treinada no ImageNet. Adotou-se redimensionamento e recorte central (*Resize + CenterCrop*) para obter entrada compatível com a arquitetura, mantendo consistência geométrica e evitando distorções severas.

Antes da inferência, as imagens foram normalizadas segundo médias e desvios padrão do ImageNet:

$$\tilde{I} = \frac{I - \mu_{\text{IN}}}{\sigma_{\text{IN}}}$$

Durante o treinamento, foi aplicado aumento de dados (*data augmentation*) com transformações moderadas (ex.: rotações leves, inversão horizontal e variações controladas de brilho/contraste), visando reduzir sobreajuste e aumentar robustez a condições reais de campo.

2.1.3 Arquitetura do Modelo

Para classificação de flores foi adotada a arquitetura **MobileNetV2**, uma rede convolucional eficiente baseada em *convoluções separáveis em profundidade* (*depthwise separable convolutions*) e blocos *inverted residual* com *linear bottlenecks*. Essa família de modelos é amplamente utilizada quando se busca bom desempenho com baixa latência, favorecendo implantação em aplicações web e dispositivos móveis.

O modelo foi inicializado com pesos pré-treinados no ImageNet (*transfer learning*). O cabeçalho original de classificação foi substituído por um bloco final compatível com o número de classes do problema ($N_f = 15$), permitindo adaptar a rede à tarefa de identificação das espécies locais. A decisão de classe para uma imagem I é obtida por:

$$\hat{y} = \arg \max_{c \in \mathcal{C}_f} p(c | I)$$

onde $p(\cdot | I)$ representa o vetor de probabilidades após *Softmax*. Para uso na aplicação, também foi disponibilizado retorno *Top-3*, permitindo ao usuário visualizar hipóteses alternativas em casos de grande similaridade morfológica entre espécies.

2.1.4 Treinamento

O treinamento foi realizado com a função de perda *Cross-Entropy*, adequada para classificação multiclasse:

$$\mathcal{L} = - \sum_{c \in \mathcal{C}_f} y_c \log(\hat{p}_c)$$

O conjunto foi dividido em proporção 80% **treino** e 20% **validação**, com separação por classes quando aplicável. A estratégia de *transfer learning* foi aplicada treinando-se inicialmente o cabeçalho adaptado e, quando necessário, realizando ajuste fino (*fine-tuning*) em parte das camadas finais do extrator convolucional, com taxa de aprendizado reduzida, a fim de refinar representações específicas para o domínio de flores da Caatinga.

Os principais hiperparâmetros empregados foram:



Hiperparâmetro	Valor
Arquitetura	MobileNetV2 (pré-treinada no ImageNet)
Tamanho de entrada	224×224 (após Resize + CenterCrop)
Função de perda	Cross-Entropy
Otimizador	Adam
Learning rate	10^{-3}
Weight decay	10^{-4}
Batch size	32

Tabela 1: Hiperparâmetros principais do treinamento do modelo de flores.

2.1.5 Procedimento de Inferência

Em produção, a inferência segue um pipeline determinístico: (i) leitura e validação da imagem; (ii) redimensionamento/recorte e normalização; (iii) execução do *forward* na MobileNetV2; (iv) aplicação de *Softmax*; e (v) retorno das predições *Top-3* com confidências. Essa estratégia foi escolhida por oferecer boa acurácia com baixa complexidade computacional, compatível com a interface do *ItapajéBio*.

2.2 Modelo da Flora

2.2.1 Sobre os dados

A base de dados utilizada para o reconhecimento de espécies vegetais foi construída a partir do Tree Species Identification Dataset, que contém imagens de diferentes espécies arbóreas sob condições ambientais variadas, e complementada com registros coletados na Secretaria de Meio Ambiente de Itapajé. Este segundo componente teve como objetivo garantir representatividade local, incorporando espécies nativas e adaptadas ao clima semiárido da região.

Ao final, a base contemplou cinquenta e três espécies distintas, incluindo árvores, arbustos e plantas ornamentais encontradas no território urbano e rural. A diversidade tanto intraespécie quanto nas condições de captura (iluminação, distância, enquadramento e estágio fenológico) tornou o problema mais realista e alinhado ao contexto de uso em campo.

2.2.2 Pré-processamento das Imagens

As imagens foram padronizadas para o formato RGB e redimensionadas para 384×384 pixels. Este tamanho foi escolhido para balancear o custo computacional da etapa convolucional e a preservação de informações estruturais relevantes, como padrões de folhas, texturas de casca e geometrias do caule.

Antes da inferência, as imagens foram normalizadas segundo as médias e desvios padrão do ImageNet, uma prática usual quando se emprega modelos pré-treinados:

$$\tilde{I} = \frac{I - \mu_{\text{IN}}}{\sigma_{\text{IN}}}$$

Além disso, um módulo de verificação de qualidade foi empregado para detectar problemas como desfoque, baixa resolução ou exposição inadequada antes da entrada no modelo, prevenindo previsões inválidas.

2.2.3 Arquitetura do Modelo

Para classificação foi adotada a arquitetura **MobileNetV3-Large**, uma rede convolucional otimizada para execução em dispositivos móveis, baseada em convoluções fatoradas e ativações



Hardswish. Para acomodar o número de espécies do conjunto, o cabeçalho original foi substituído por um bloco linear reduzido, composto por:

$$(\text{Linear}(d \rightarrow 256) \rightarrow \text{Hardswish} \rightarrow \text{Dropout} \rightarrow \text{Linear}(256 \rightarrow N))$$

onde d corresponde à dimensionalidade produzida pela etapa convolucional e N ao número total de espécies da base.

A utilização de pesos pré-treinados no ImageNet foi crucial, uma vez que a morfologia foliar apresenta características convolucionais semelhantes às de outros objetos naturais, permitindo boa transferência de conhecimento.

2.2.4 Treinamento

O modelo foi treinado utilizando a função de perda de *Cross-Entropy* com o otimizador Adam. A divisão do conjunto foi realizada em proporção 80% para treino e 20% para validação. Durante o treinamento não foi utilizada agregação temporal, ao contrário do caso dos pássaros, pois as entradas consistem em imagens únicas. A previsão final para cada imagem foi obtida via:

$$\hat{y} = \arg \max_{c \in \mathcal{C}} p(c | I)$$

onde $p(\cdot | I)$ corresponde ao vetor de probabilidades após a camada *Softmax*. Para fins de retorno na aplicação, foi utilizado o *Top-3* das espécies mais prováveis, permitindo ao usuário considerar hipóteses alternativas visualmente próximas.

2.3 Modelo dos pássaros

2.3.1 Sobre os dados

A construção da base de dados foi realizada a partir de uma pesquisa em plataformas especializadas em bioacústica, como os sites WikiAves e eBird, amplamente utilizados para o compartilhamento e catalogação de registros sonoros de aves. Além disso, foi incorporado material disponibilizado pela Universidade Federal do Ceará (UFC), proveniente do Projeto Fênix, voltado à identificação automática de aves por meio do canto.

A partir da integração dessas fontes, foi possível reunir gravações correspondentes a 130 espécies de aves, incluindo espécies nativas e espécies migratórias que ocorrem na região de estudo. Essa diversidade contribui para a robustez do modelo proposto, permitindo avaliar seu desempenho em um cenário realista, caracterizado por grande variabilidade acústica interespecies.

2.3.2 Pré-processamento dos Áudios

Cada gravação foi convertida para mono, normalizada e reamostrada para 22,050 Hz. Em seguida, os áudios foram segmentados em janelas deslizantes de duração fixa de 3 s, com *hop* de 1,5 s, resultando em sobreposição de 50%. Para assegurar consistência, segmentos menores que a janela-alvo foram preenchidos com *padding* de zeros.

2.3.3 Extração de Espectrogramas Mel

Para cada segmento foi computado um espectrograma Mel com os seguintes parâmetros:

- $n_{\text{fft}} = 2048$
- $\text{hop length} = 512$
- $n_{\text{mels}} = 128$



Relatório Final

Os espectrogramas foram posteriormente convertidos para escala dB por meio de:

$$S_{dB} = 10 \log_{10}(S_{mel})$$

e submetidos a normalização local via *z-score*:

$$\tilde{S} = \frac{S_{dB} - \mu}{\sigma + 10^{-6}}$$

onde μ e σ correspondem à média e ao desvio padrão do segmento. As matrizes resultantes foram armazenadas no formato `.npy` para uso no treinamento.

2.3.4 Arquitetura do Modelo

A arquitetura CNN+LSTM aplicada inicialmente recebe tensores no formato $(1, H, W)$, onde H corresponde aos filtros Mel e W ao eixo temporal. A CNN é composta por quatro blocos convolucionais com *stride* 2, *Batch Normalization*, ativação ReLU e *Dropout*, conforme:

$$(\text{Conv2D} \rightarrow \text{BN} \rightarrow \text{ReLU} \rightarrow \text{Dropout}) \times 4$$

A convolução opera como mecanismo de extração de padrões espectrais locais. Após a etapa convolucional, a média global é computada ao longo da dimensão de frequência, produzindo uma sequência temporal de vetores que é então encaminhada a uma LSTM bidirecional de duas camadas:

$$\text{BiLSTM}(128 \rightarrow 256)$$

A saída temporal da LSTM é agregada via média ao longo do tempo:

$$\hat{z} = \frac{1}{T} \sum_{t=1}^T z_t$$

Por fim, uma camada totalmente conectada realiza a decisão de classe:

$$\hat{y} = \text{Softmax}(W\hat{z} + b)$$

2.3.5 Treinamento

O treinamento foi realizado utilizando a função de perda *Cross-Entropy* e o otimizador Adam com taxa de aprendizado inicial de 10^{-3} e *weight decay* de 10^{-4} . Um agendador do tipo *ReduceLROnPlateau* foi empregado para reduzir o *learning rate* quando a perda de validação estagnava.

O conjunto foi dividido em 80% para treino e 20% para validação. Durante o treinamento, cada segmento contribui individualmente para o erro, mas durante a inferência final o modelo agrega as probabilidades ao longo dos segmentos de uma mesma gravação via média:

$$p_{\text{final}} = \frac{1}{N} \sum_{i=1}^N p_i$$



3 Resultados

3.1 Modelo Flores

A avaliação do modelo de flores foi realizada em um conjunto de teste separado, mantendo o mesmo pipeline de pré-processamento descrito na Metodologia (redimensionamento/recorte e normalização compatível com ImageNet). Como se trata de um problema multiclasse com um único rótulo correto por imagem, a métrica central reportada foi a **acurácia**. Para complementar a análise, também foram consideradas métricas agregadas como *precision*, *recall* e *F1-score*, especialmente úteis quando há variação no suporte entre classes.

O classificador adotado foi baseado em **MobileNetV2** pré-treinada no ImageNet, com substituição da camada final para acomodar as 15 classes do problema. A estratégia de *transfer learning* permitiu explorar representações visuais robustas previamente aprendidas, reduzindo custo de treinamento e melhorando a generalização para imagens capturadas em condições reais de campo. Para a aplicação web, o retorno é apresentado como *Top-3*, oferecendo hipóteses alternativas em casos de ambiguidade morfológica.

Os resultados globais obtidos no conjunto de teste foram:

- **Acurácia: 0.9333**
- **Precision Macro: 0.9375**
- **F1-score Macro: 0.9329**
- **Recall Macro: 0.9333**

De forma geral, o desempenho observado indica que o modelo é capaz de capturar padrões discriminativos relevantes (formato e simetria das pétalas, textura e coloração predominante, presença de estruturas internas e padrões de contraste), mantendo robustez frente a variações de iluminação, escala e fundo. Os erros residuais concentram-se, em geral, em espécies com grande similaridade visual ou em imagens com o objeto parcialmente ocluído, baixa nitidez ou enquadramento desfavorável; nesses casos, o retorno em *Top-3* contribui para mitigar impacto prático, permitindo validação visual pelo usuário final.

3.2 Modelo Flora

A base utilizada continha 55 espécies distintas, incluindo árvores, arbustos, plantas ornamentais e espécies nativas relevantes ao bioma local. A variabilidade intraclasse — associada a iluminação, perspectivas, estágios fenológicos e diversidade morfológica — motivou o uso de modelos pré-treinados no ImageNet, com posterior *fine-tuning* supervisionado da camada classificadora.

No processo de adaptação, o modelo foi empregado como *feature extractor*, substituindo-se apenas a *head* final por blocos densos com *dropout* e ativação *Hardswish*, característicos da versão V3 da arquitetura. Essa estratégia permite que o modelo aproveite representações visuais previamente aprendidas, reduzindo o custo de treinamento e mitigando sobreajuste em bases de tamanho moderado.

A métrica principal utilizada foi a **acurácia**, uma vez que o problema é estritamente multiclasse com exatamente um rótulo correto por imagem. Os resultados obtidos no conjunto de teste foram:

- **Acurácia: 0.88**

Esse desempenho indica que o modelo é capaz de identificar corretamente a espécie em aproximadamente 88% das imagens, o que é expressivo considerando o número de classes e a variabilidade visual inerente à flora local. Além disso, a escolha da MobileNetV3 permite executar inferências em ambiente sem GPU com latências reduzidas, facilitando a integração em aplicações web e dispositivos móveis no contexto do *ItapajéBio*.



3.3 Modelo Pássaros

O modelo desenvolvido para identificação de espécies de pássaros a partir do canto obteve resultados expressivos considerando o número de classes e a variabilidade intraclasse do sinal acústico. A base final utilizada continha 130 espécies, padronizadas e balanceadas conforme descrito na seção de construção do dataset.

O modelo utilizado foi uma arquitetura **CNN+LSTM bidirecional**, escolhida por sua capacidade de capturar tanto padrões espectrais estáticos (via convolução) quanto dinâmicos no tempo (via recorrência), característica importante em vocalizações complexas.

A métrica principal de avaliação foi o *F1-score* por classe, devido ao caráter multiclasse e presença de classes com diferentes frequências. A Tabela de classificação apresenta os valores de *precision*, *recall* e *f1-score* para cada espécie individual e as médias macro e ponderada (*macro avg* e *weighted avg*), além da acurácia geral.

Um resumo dos resultados globais obtidos no conjunto de teste:

- **Acurácia:** 0.78
- **Macro F1-score:** 0.76
- **Weighted F1-score:** 0.78

A média *macro* é mais sensível às classes menos frequentes e indica que o modelo consegue generalizar razoavelmente mesmo para espécies com suporte reduzido, enquanto a *weighted* confirma que o desempenho nas classes de maior ocorrência foi ainda melhor.

Estes resultados indicam que o modelo é capaz de realizar identificação acústica de espécies de pássaros com desempenho sólido para um problema de 130 classes, fortemente desbalanceado e com variabilidade temporal relevante, sendo adequado para o uso na API de classificação implementada.

4 API de Inferência e Interface Web

Além do modelo de classificação, foi desenvolvida uma API de inferência capaz de servir o sistema para uso por terceiros, bem como uma interface web dedicada à interação com usuários não técnicos. Esta seção descreve o funcionamento da API, o fluxo de processamento da requisição e as funcionalidades da aplicação web, além de fornecer exemplos visuais de suas telas.

4.1 API de Inferência

A API foi implementada utilizando o framework **FastAPI** devido à sua baixa latência, suporte nativo à tipagem e facilidade de integração com modelos de *machine learning*. A API possui endpoints dedicados para cada domínio do projeto, notadamente para identificação de flores, para classificação de vocalizações de aves e para reconhecimento de espécies da flora (plantas não necessariamente floridas).

O endpoint responsável por vocalizações, por exemplo, recebe uma gravação sonora via formulário multipart, aplica todo o pipeline de pré-processamento descrito anteriormente e produz como saída um vetor de probabilidades associado às espécies:

```
POST /predict
Content-Type: multipart/form-data
file: <áudio>
```

O formato de resposta segue o padrão JSON:



```
{
  "quality": "processado",
  "top3": [
    {"scientific": "X", "confidence": p1},
    {"scientific": "Y", "confidence": p2},
    {"scientific": "Z", "confidence": p3}
  ]
}
```

Foram incorporados mecanismos de reamostragem, padronização e segmentação interna para garantir que tanto arquivos carregados localmente quanto gravações realizadas pelo usuário (via navegador) fossem aceitas independentemente da duração original ou do formato.

4.2 Interface Web

A aplicação web foi desenvolvida com **React**, utilizando componentes reutilizáveis para as diferentes modalidades de predição. A interface oferece ao usuário duas formas principais de interação:

1. envio de arquivos (*upload*)
2. aquisição ativa via sensores locais do dispositivo (câmera e microfone)

O sistema foi projetado com foco em acessibilidade, responsividade e reduzida complexidade de uso, buscando tornar o processo de identificação intuitivo para usuários finais sem conhecimento técnico.

4.3 Página Inicial

A página inicial (Figura 1) fornece uma visão geral do sistema com links diretos para os diferentes módulos disponíveis (flores, aves e flora) e uma breve explicação sobre seu funcionamento.

4.4 Módulo de Identificação de Flores

O módulo destinado à identificação de flores (Figura 2) permite o envio de imagens ou captura fotográfica mediante o uso da câmera do dispositivo. Após o envio, o sistema realiza a inferência via API e retorna as três espécies mais prováveis, acompanhadas de suas respectivas probabilidades. Quando disponível, um mapeamento para nomes populares é exibido para facilitar a interpretação por parte do usuário.

4.5 Módulo de Identificação da Flora

O módulo de flora (Figura 3) é voltado à identificação de espécies vegetais que nem sempre se encontram em estágio de floração evidente, tais como folhas, galhos, estruturas vegetativas ou aspectos gerais da planta. Assim como no módulo de flores, o usuário pode tanto enviar uma fotografia existente quanto capturar uma nova imagem utilizando a câmera do dispositivo.

Após a submissão, a imagem é processada no backend por um modelo específico para flora, baseado em arquiteturas de redes neurais convolucionais (MobileNet), treinado sobre imagens de plantas nativas da região. A API retorna as espécies mais prováveis e suas respectivas confidências, que são então apresentadas em uma interface similar à do módulo de flores, com destaque para a predição de maior confiança e uma lista das demais hipóteses.



Relatório Final

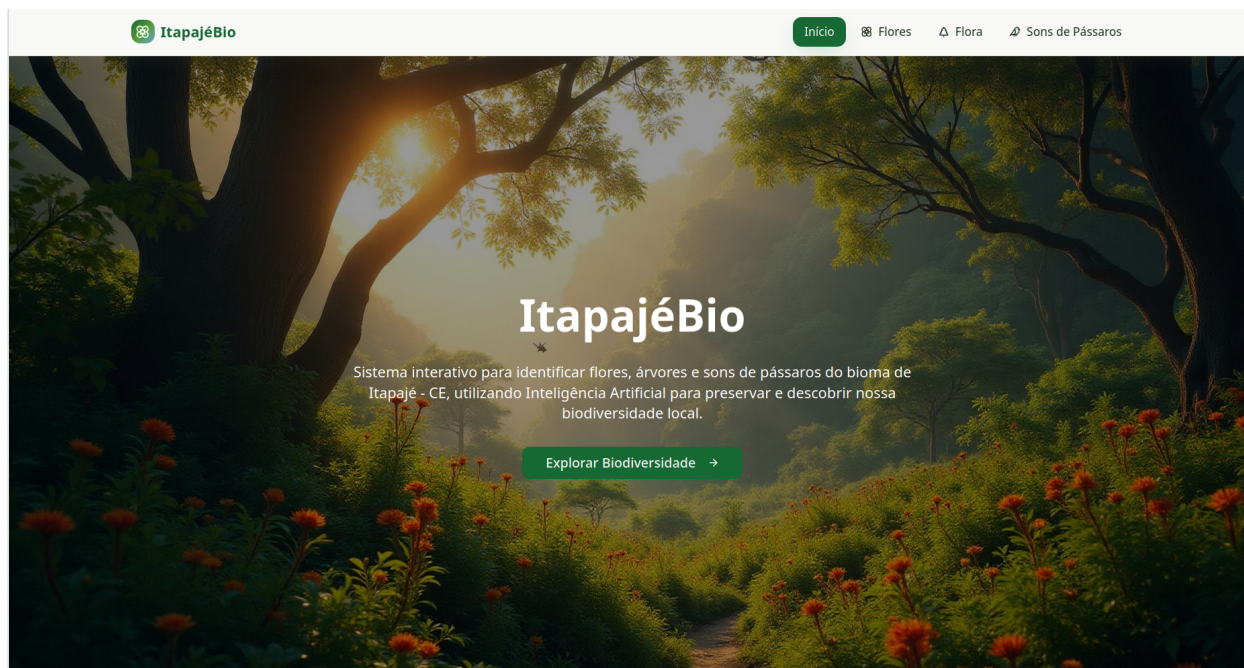


Figura 1: Página Inicial

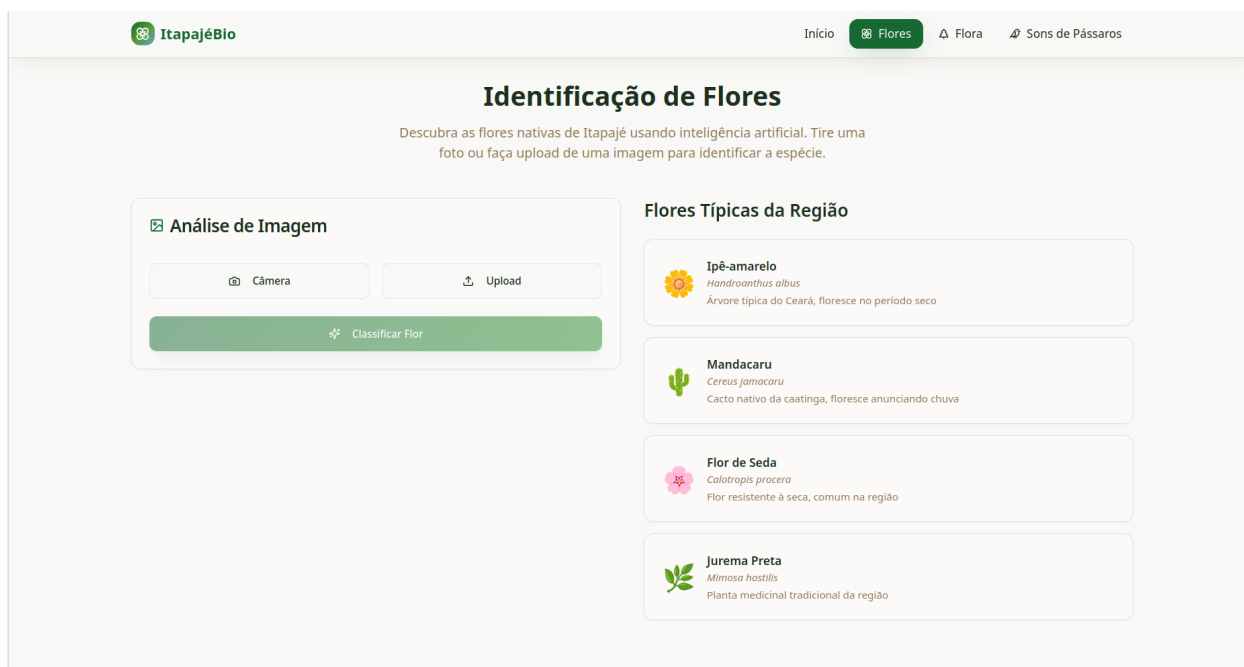


Figura 2: Página de classificação de Flores

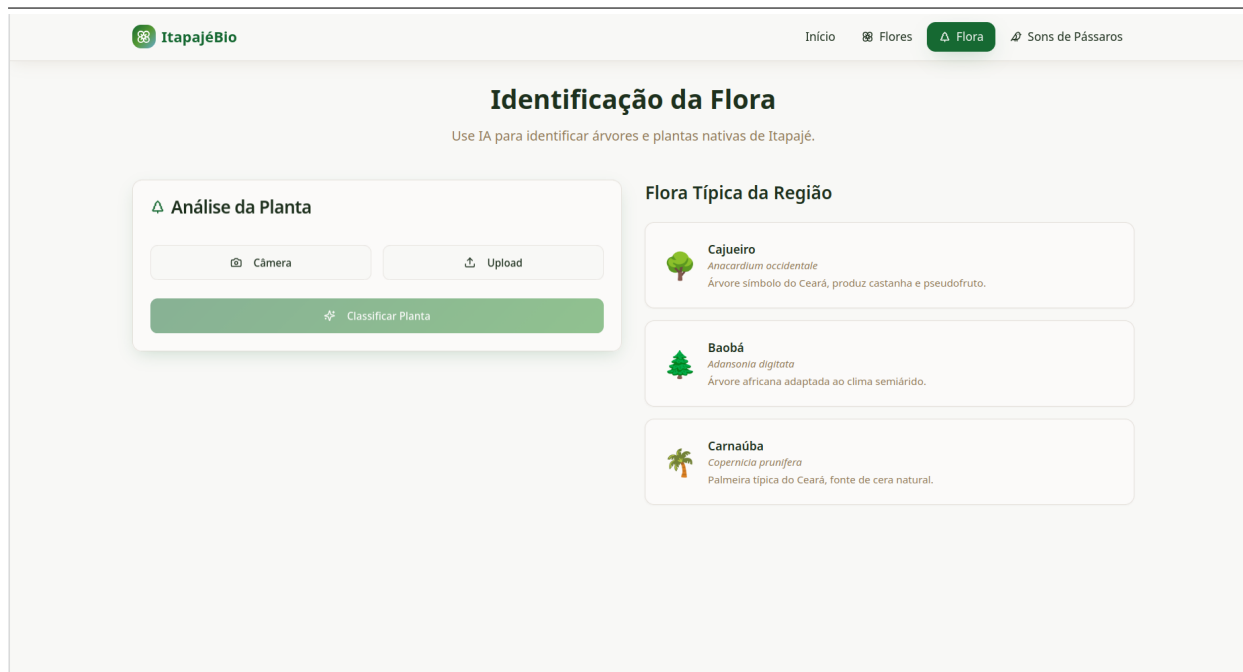


Figura 3: Página de classificação de Flora

4.6 Módulo de Classificação de Pássaros

O módulo dedicado às aves (Figura 4) integra um pipeline mais complexo devido à necessidade de processar áudios ambientais. Neste módulo, o usuário pode gravar sons diretamente via microfone ou enviar arquivos de áudio. Uma visualização em forma de *waveform* é exibida para indicar a forma temporal do sinal capturado, permitindo ao usuário verificar se houve captura adequada do canto.

Após o envio, o backend segmenta o áudio, extrai espectrogramas Mel, calcula as probabilidades por segmento e, finalmente, agrega as saídas temporais por meio de uma média, produzindo uma classificação final. O resultado é apresentado ao usuário em formato de *Top-3* com percentuais de confiança.

4.7 Fluxo de Comunicação API → Interface

O fluxo de comunicação entre interface e API pode ser resumido da seguinte forma:

1. aquisição do dado (imagem ou áudio) no navegador;
2. construção de formulário multipart contendo o arquivo;
3. envio da requisição via HTTP POST ao endpoint correspondente;
4. pré-processamento no backend (normalização, segmentação, extração de atributos);
5. execução do modelo de inferência (flores, flora ou aves);
6. agregação e formatação da resposta em JSON;
7. exibição do resultado ao usuário com as espécies mais prováveis e suas confidências.

A utilização de um backend desacoplado permite a substituição ou atualização dos modelos de maneira transparente para o usuário final, além de facilitar a integração com futuros serviços externos e a expansão do sistema para novos domínios de biodiversidade.



Relatório Final

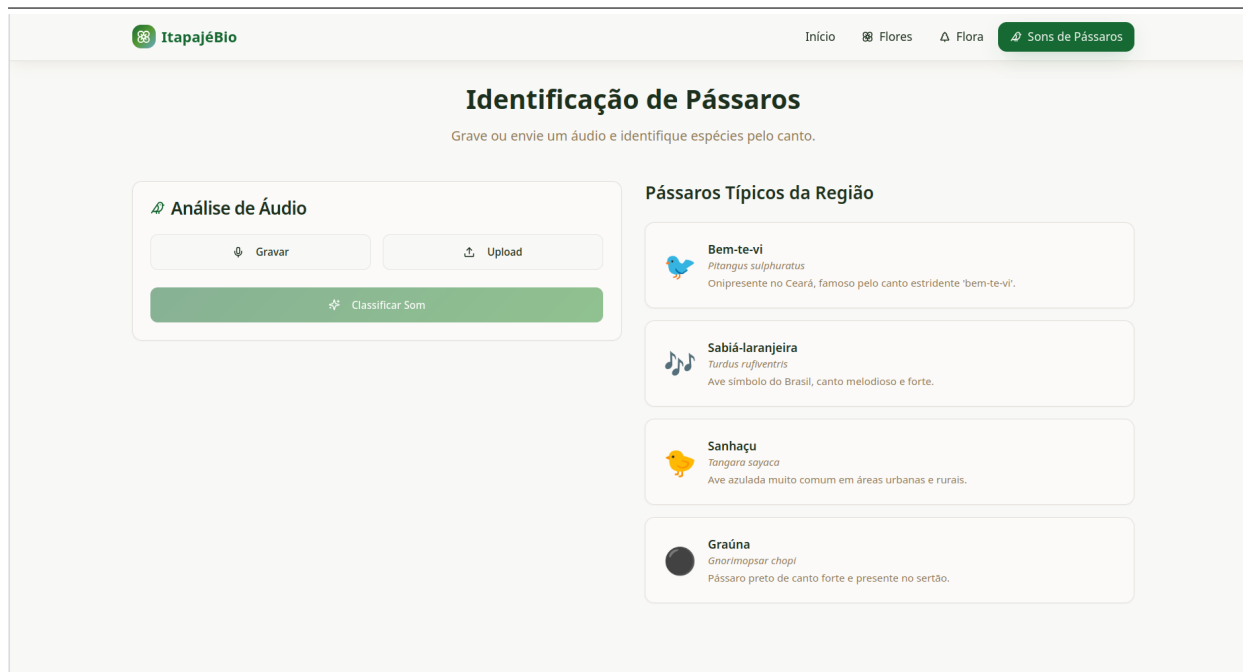


Figura 4: Página de classificação de Pássaros

5 Conclusões

O desenvolvimento dos modelos apresentados demonstrou que é tecnicamente viável realizar a identificação automática de elementos da biodiversidade regional utilizando arquiteturas de aprendizado profundo otimizadas tanto para precisão quanto para eficiência de inferência. Foram integradas três modalidades distintas de classificação — imagens de flores, imagens de espécies vegetais (flora) e sinais acústicos de pássaros — cada uma exigindo adaptações arquiteturais próprias. As arquiteturas MobileNetV2 e MobileNetV3 mostraram-se adequadas para as classificações visuais, alcançando desempenho elevado sob custos computacionais reduzidos, enquanto a combinação CNN+LSTM se mostrou apropriada para a modelagem temporal de vocalizações de aves, obtendo métricas estáveis em um problema de 130 classes. Os resultados obtidos, incluindo acurácia de 88% na classificação da flora e desempenho competitivo no domínio acústico, validam a abordagem proposta e indicam maturidade suficiente para uso aplicado.

Além dos modelos de classificação, foi desenvolvida uma API de inferência unificada e uma interface web interativa, permitindo que os modelos sejam acessados por usuários não técnicos e integrados em diferentes dispositivos, incluindo computadores e smartphones. Essa camada adicional de software transforma o conjunto de modelos em uma ferramenta de apoio ao reconhecimento, documentação e educação ambiental no contexto do município de Itapajé. Como desdobramento futuro, o sistema pode ser expandido com novos domínios biológicos, ampliação taxonômica, coleta contínua de dados em campo e estratégias de melhoria contínua dos modelos, consolidando o *ItapajéBio* como uma plataforma de referência para iniciativas de conservação e valorização da biodiversidade local.