



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

PAULO DE TARSO VILARINHO CASTELO BRANCO FILHO

HECTORAI: ASSISTENTE MÉDICO PARA PRONTUÁRIOS COM AI

TERESINA
2026

PAULO DE TARSO VILARINHO CASTELO BRANCO FILHO

HECTORAI: ASSISTENTE MÉDICO PARA PRONTUÁRIOS COM AI

Trabalho de Conclusão de Curso (relatório de software) apresentado como exigência parcial para obtenção do diploma do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Orientador: Prof.^a Dr.^a Aline Montenegro Leal

TERESINA

2026

PAULO DE TARSO VILARINHO CASTELO BRANCO FILHO

HECTORAI: ASSISTENTE MÉDICO PARA PRONTUÁRIOS COM AI

Trabalho de Conclusão de Curso (relatório de software) apresentado como exigência parcial para obtenção do diploma do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Aprovado em: 17/06/2026.
BANCA EXAMINADORA

Prof.^a Dr.^a Aline Montenegro Leal (Orientadora)
Instituto Federal do Piauí (IFPI)

Prof.^a Dr.^a Nádia Mendes dos Santos
Instituto Federal do Piauí (IFPI)

Prof. Dr. Ricardo Martins Ramos
Instituto Federal do Piauí (IFPI)

Dedico este trabalho à minha esposa, aos meus pais e aos meus dois filhos, que são os motivos para eu nunca desistir e seguir lutando.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me permitido chegar até aqui, e à minha família, base de tudo o que sou. Em especial, deixo minha mais profunda gratidão aos meus pais, Paulo de Tarso Vilarinho Castelo Branco e Cristiane Maria Adad Amorim Castelo Branco, pelo apoio incondicional, pelo amor constante e por todos os sacrifícios que fizeram para que eu pudesse trilhar esse caminho. Sem o exemplo, o cuidado e o incentivo de vocês, nada disso seria possível.

Agradeço também à minha esposa, Amanda de Souza Martins Vilarinho, minha companheira há quase dez anos, por estar sempre ao meu lado, por acreditar em mim mesmo nos momentos em que eu duvidei, e por ser a minha maior incentivadora ao longo dessa jornada. E aos meus filhos, Maria Teresa Martins Vilarinho e Paulo de Tarso Vilarinho Castelo Branco Neto, que ainda são tão pequenos e mal sabem o quanto são importantes para mim. Vocês são a minha maior motivação para continuar lutando todos os dias.

Por fim, deixo meu sincero agradecimento a todos os professores que cruzaram meu caminho durante essa trajetória, verdadeiros mestres que contribuíram não apenas para minha formação acadêmica, mas também para meu crescimento pessoal. Um agradecimento especial à professora Aline Montenegro Leal, minha orientadora, por toda a paciência, dedicação e auxílio ao longo de cada etapa deste trabalho. Sua orientação foi fundamental para que eu conseguisse concluir esta jornada.

"É justo que muito custe o que muito vale"
Santa Teresa d'Ávila

RESUMO

O prontuário médico é um documento de caráter legal, sigiloso e científico que favorece a continuidade do cuidado, mas sua elaboração consome parte relevante da jornada profissional e contribui para a sobrecarga do médico. Este relatório teve como objetivo desenvolver e documentar o HectorAI, aplicação web que automatiza a produção de notas clínicas no formato SOAP a partir do áudio de consultas em português brasileiro. O trabalho foi conduzido por meio de pesquisa bibliográfica sobre documentação clínica e assistentes baseados em inteligência artificial, levantamento de requisitos, modelagem da arquitetura e desenvolvimento incremental do protótipo. A solução foi implementada em Ruby on Rails, com processamento assíncrono, transcrição automática de fala pelo Whisper e geração de texto estruturado pelo GPT-4o. Como resultado, obteve-se um protótipo funcional que permite cadastrar pacientes, agendar e gravar consultas, acompanhar o processamento, revisar a nota gerada e aprová-la para registro. A aplicação também registra as edições realizadas pelo médico para orientar gerações posteriores. Conclui-se que a arquitetura proposta integra, em um único fluxo, captura de áudio, transcrição e documentação clínica, oferecendo uma base técnica para reduzir o trabalho manual de registro. A avaliação em ambiente clínico real permanece como etapa futura.

Palavras-chave: assistente clínico de inteligência artificial; prontuário eletrônico; modelo de linguagem de grande porte; Ruby on Rails; nota SOAP.

ABSTRACT

The medical record is a legal, confidential, and scientific document that supports continuity of care, but producing it consumes a relevant part of the physician's workday and contributes to professional overload. This report aimed to develop and document HectorAI, a web application that automates the production of SOAP clinical notes from the audio of consultations conducted in Brazilian Portuguese. The work comprised a literature review on clinical documentation and artificial intelligence assistants, requirements elicitation, architecture modeling, and incremental prototype development. The solution was implemented with Ruby on Rails, asynchronous processing, automatic speech transcription by Whisper, and structured text generation by GPT-4o. The result was a functional prototype that allows physicians to register patients, schedule and record consultations, monitor processing, review the generated note, and approve it for registration. The application also records changes made by the physician to guide subsequent generations. The proposed architecture therefore integrates audio capture, transcription, and clinical documentation into a single workflow and provides a technical basis for reducing manual documentation work. Evaluation in a real clinical setting remains a future step.

Keywords: AI clinical assistant; electronic medical record; large language model; Ruby on Rails; SOAP note.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso do Médico	16
Figura 2 – Arquitetura do sistema	18
Figura 3 – Tela de <i>login</i>	19
Figura 4 – Tela de cadastro do médico	20
Figura 5 – Tela inicial	20
Figura 6 – Lista de pacientes	21
Figura 7 – Formulário de cadastro de paciente	22
Figura 8 – Detalhes do paciente	22
Figura 9 – Lista de consultas	23
Figura 10 – Lista de consultas com filtro por status	24
Figura 11 – Formulário de agendamento de consulta	24
Figura 12 – Detalhes da consulta no status <i>agendada</i>	25
Figura 13 – Tela de início da consulta com o gravador de áudio	26
Figura 14 – Consulta em transcrição	27
Figura 15 – Consulta em geração da nota SOAP	27
Figura 16 – Fluxo de estados de uma consulta	28
Figura 17 – Tela de revisão da nota SOAP	29
Figura 18 – Consulta aprovada	30

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CPF	Cadastro de Pessoas Físicas
CRM	Conselho Regional de Medicina
CSS	<i>Cascading Style Sheets</i>
ERB	<i>Embedded Ruby</i>
GPT	<i>Generative Pre-trained Transformer</i>
HTML	<i>Hypertext Markup Language</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
IA	Inteligência Artificial
JSON	<i>JavaScript Object Notation</i>
LGPD	Lei Geral de Proteção de Dados Pessoais
LLM	<i>Large Language Model</i> (Modelo de Linguagem de Grande Porte)
MVC	<i>Model-View-Controller</i>
SOAP	Subjetivo, Objetivo, Avaliação, Plano
SPA	<i>Single Page Application</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	11
1.1.1	Objetivo geral	11
1.1.2	Objetivos específicos	12
2	TECNOLOGIAS ENVOLVIDAS	13
2.1	TECNOLOGIAS DO SERVIDOR	13
2.2	TECNOLOGIAS DO CLIENTE WEB	13
2.3	TECNOLOGIAS DE INTELIGÊNCIA ARTIFICIAL	14
3	MODELAGEM DO PROJETO	15
3.1	LEVANTAMENTO DE REQUISITOS	15
3.1.1	Requisitos funcionais	15
3.1.2	Requisitos não funcionais	15
3.2	DIAGRAMA DE CASOS DE USO	16
3.3	ARQUITETURA DO SISTEMA	17
3.4	INTERFACE	19
4	CONCLUSÃO	31
	REFERÊNCIAS BIBLIOGRÁFICAS	32
	ANEXO A – DECLARAÇÃO DE ENTREGA DO CÓDIGO-FONTE	34

1 INTRODUÇÃO

O prontuário médico é instrumento central de continuidade do cuidado no Sistema Único de Saúde (SUS) e na prática privada. A Resolução nº 1.638/2002 do Conselho Federal de Medicina o define como o documento único que reúne informações sobre a saúde do paciente e a assistência prestada, com caráter legal, sigiloso e científico (Conselho Federal de Medicina, 2002). As justificativas apresentadas na própria resolução o destacam como instrumento de defesa do profissional em eventuais disputas judiciais ou ético-profissionais. A documentação clínica, no entanto, compete com o tempo da consulta. Pesquisa de Gardner et al. com 1.792 médicos em Rhode Island, publicada no *Journal of the American Medical Informatics Association*, mostrou que médicos que declaram tempo insuficiente para registrar a consulta apresentam chance 2,8 vezes maior de desenvolver burnout (GARDNER et al., 2019).

Os modelos de linguagem de grande porte (LLMs, do inglês *Large Language Models*), combinados a sistemas de reconhecimento automático de fala, viabilizaram a categoria dos assistentes clínicos automáticos baseados em Inteligência Artificial (IA). Em um levantamento realizado pela rede norte-americana Phyx Primary Care em 2025 com 116 provedores de atenção primária que utilizaram o assistente clínico automático Suki AI por mais de 30 dias, registrou uma redução de 41% no tempo de documentação e de 60% nas autoavaliações de burnout (Phyx Primary Care, 2025). No Brasil, há poucas alternativas nacionais: a maior parte das soluções é estrangeira, em língua inglesa e integrada a sistemas norte-americanos como Epic e Cerner, voltados ao médico institucional, e não ao autônomo de consultório particular. O HectorAI, objeto deste trabalho, busca preencher essa lacuna. Trata-se de uma aplicação web em Ruby on Rails 8 que recebe o áudio da consulta, transcreve via OpenAI Whisper, gera uma nota SOAP (Subjetivo, Objetivo, Avaliação, Plano) em português brasileiro via GPT-4o (*Generative Pre-trained Transformer*, versão 4o) e a disponibiliza para revisão e aprovação.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Desenvolver e documentar o HectorAI, aplicação web que automatiza a geração de notas clínicas em formato SOAP a partir do áudio de consultas médicas, com o objetivo de reduzir o tempo de documentação do médico autônomo brasileiro e melhorar a qualidade e o valor probatório do prontuário.

1.1.2 Objetivos específicos

- Projetar uma arquitetura web que orquestre um pipeline assíncrono de transcrição e geração de notas clínicas em conformidade com as exigências do Conselho Federal de Medicina;
- Implementar o fluxo completo da consulta, do áudio à nota SOAP revisada e aprovada pelo médico;
- Desenvolver um mecanismo de personalização que aprenda com as edições do médico e ajuste o estilo de redação.

2 TECNOLOGIAS ENVOLVIDAS

2.1 TECNOLOGIAS DO SERVIDOR

No desenvolvimento do servidor foi escolhido o *framework* Ruby on Rails, baseado na linguagem Ruby. Criada por Yukihiro Matsumoto em 1995, Ruby é uma linguagem interpretada, dinâmica e multiparadigma, com suporte a metaprogramação por meio de *open classes*. Rails é um *framework* web *open-source* criado por David Heinemeier Hansson em 2004, que adota o padrão *Model-View-Controller* (MVC) e os princípios *convention over configuration* e *don't repeat yourself*. Conforme as versões fixadas no repositório, o HectorAI utiliza Ruby 4.0.2 e Rails 8.1.3 (Ruby Core Team, 2026; Ruby on Rails, 2026b).

Para a persistência, adotou-se o SQLite, banco de dados relacional embarcado e autocontido, acessado pela *gem* adaptadora *sqlite3* 2.9.2 (SQLite Consortium, 2026; Sparklemotion, 2026). A autenticação dos médicos é feita pelo Devise 5.0.3, com os módulos de e-mail e senha, confirmação obrigatória e recuperação de senha (Heartcombo, 2026). O processamento assíncrono do *pipeline* de transcrição e geração de notas utiliza o Solid Queue 1.4.0; em produção, o cache utiliza o Solid Cache 1.0.10; e as mensagens do Action Cable utilizam o Solid Cable 3.0.12 em desenvolvimento e produção (Ruby on Rails, 2026c; Ruby on Rails, 2025b; Ruby on Rails, 2025a). O *upload* e o armazenamento do áudio das consultas são gerenciados pelo Active Storage, componente incluído no Rails e, portanto, na mesma versão 8.1.3 (Ruby on Rails, 2026a).

2.2 TECNOLOGIAS DO CLIENTE WEB

A camada de apresentação foi construída com o Hotwire (HTML Over The Wire), abordagem mantida pela 37signals que reúne tecnologias complementares para enviar HTML (*Hypertext Markup Language*) pré-renderizado pelo servidor em vez de transferir JSON (*JavaScript Object Notation*) e reconstruir a interface no navegador. Hotwire é o nome do conjunto, não um pacote instalado com versão própria (37signals, 2026).

No HectorAI, o Turbo é fornecido pela *gem* de integração *turbo-rails* 2.0.23, responsável pela navegação acelerada e pela atualização parcial da página via Turbo Streams sobre WebSocket (STEPHENSON; MAHKMALI; HANSSON, 2026; Hotwire, 2026b). O Stimulus é fornecido pela *gem* *stimulus-rails* 1.3.4 e adiciona comportamento JavaScript aos elementos HTML existentes (STEPHENSON; MAHKMALI; HANSSON, 2024; Hotwire, 2026a). Em conjunto, essas tecnologias substituem o fragmento de status da consulta conforme o processamento avança, sem recarga da página.

A estilização utiliza o Tailwind CSS 4.2.1, *framework* CSS (*Cascading Style*

Sheets) *utility-first* executado no projeto pelo pacote `tailwindcss-ruby` 4.2.1 e integrado ao fluxo de ativos do Rails pela *gem* `tailwindcss-rails` 4.4.0 (Tailwind Labs, 2026; `tailwindcss-ruby` contributors, 2026; HANSSON, 2025).

2.3 TECNOLOGIAS DE INTELIGÊNCIA ARTIFICIAL

A integração com modelos de IA é feita pelo RubyLLM, biblioteca Ruby que disponibiliza uma única interface para vários provedores de modelos de linguagem e integra-se ao Active Record por meio do método `acts_as_chat`, persistindo o histórico de conversas como entidade de domínio (PAOLINO, 2026). Sobre essa abstração, o HectorAI consome dois modelos da OpenAI: o Whisper, modelo de reconhecimento automático de fala treinado com 680 mil horas de áudio multilíngue (RADFORD et al., 2022), transcreve o áudio das consultas para o português brasileiro e o GPT-4o (*Generative Pre-trained Transformer*, versão 4o), modelo de linguagem multimodal lançado em 2024, recebe a transcrição da consulta e um *prompt* com instruções e devolve uma nota SOAP estruturada em formato JSON (OpenAI, 2024).

3 MODELAGEM DO PROJETO

3.1 LEVANTAMENTO DE REQUISITOS

O levantamento partiu da revisão de literatura sobre assistentes clínicos de IA na prática médica e de conversas informais com médicos autônomos sobre a rotina de documentação clínica. A lista a seguir reúne os requisitos obtidos nesse processo, divididos em funcionais e não funcionais.

3.1.1 Requisitos funcionais

- **[RF01]** A aplicação deve permitir que o médico se cadastre e se autentique.
- **[RF02]** A aplicação deve permitir que o médico cadastre, edite e consulte pacientes.
- **[RF03]** A aplicação deve restringir cada médico ao seu próprio conjunto de pacientes e consultas.
- **[RF04]** A aplicação deve permitir o agendamento de consultas vinculadas a um paciente.
- **[RF05]** A aplicação deve listar as consultas do médico, com filtro por status.
- **[RF06]** A aplicação deve permitir que o médico grave o áudio da consulta pelo navegador.
- **[RF07]** A aplicação deve transcrever o áudio e gerar uma nota clínica em formato SOAP.
- **[RF08]** A aplicação deve apresentar a nota SOAP gerada para revisão e edição pelo médico.
- **[RF09]** A aplicação deve permitir que o médico aprove a nota SOAP, tornando-a imutável.
- **[RF10]** A aplicação deve usar as edições feitas pelo médico para personalizar as gerações seguintes.

3.1.2 Requisitos não funcionais

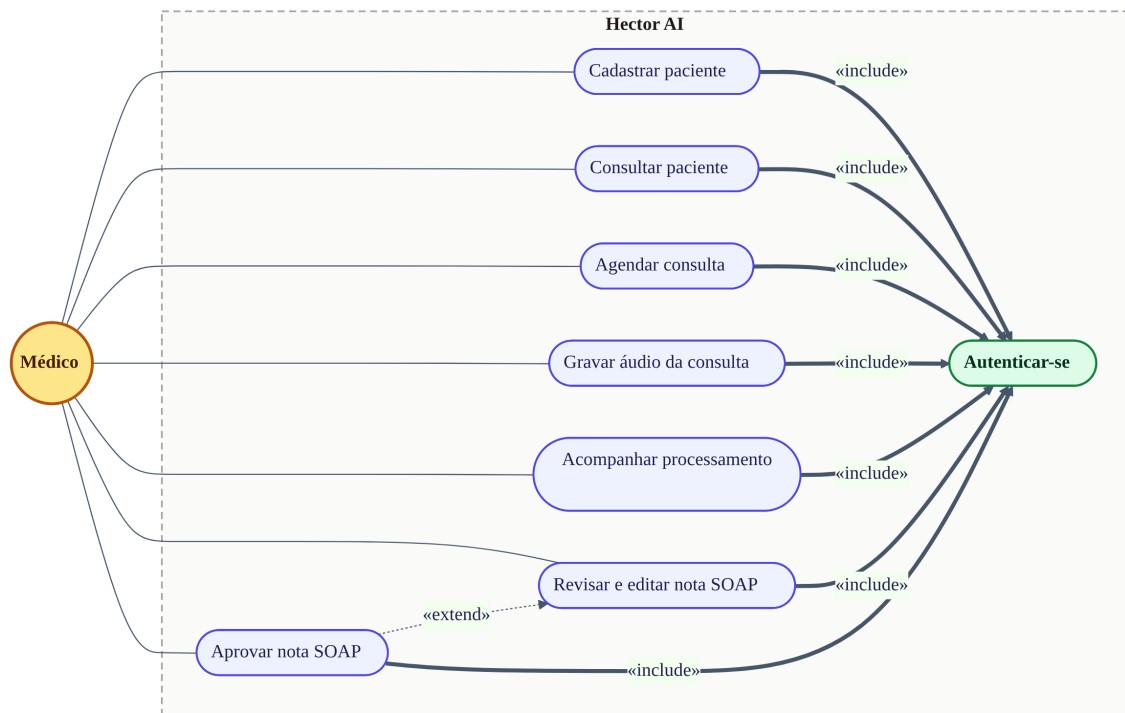
- **[RNF01]** A interface deve ser web e funcionar em navegadores modernos.
- **[RNF02]** A interface deve ser responsiva, operando em *desktops*, *notebooks* e *tablets*.

- **[RNF03]** O processamento do áudio deve ocorrer de forma assíncrona, sem bloquear o navegador.
- **[RNF04]** As mudanças de status da consulta devem aparecer ao médico em tempo real.
- **[RNF05]** A aplicação deve armazenar os dados pessoais dos pacientes de forma criptografada.
- **[RNF06]** O sistema deve registrar as transições de status de cada consulta e as razões de falha.
- **[RNF07]** O sistema deve ser construído sobre tecnologias com documentação consolidada.
- **[RNF08]** O código-fonte deve seguir boas práticas e ter testes automatizados nos fluxos críticos.

3.2 DIAGRAMA DE CASOS DE USO

O HectorAI possui um único ator humano, o Médico, responsável por gerenciar pacientes, registrar consultas e revisar as notas SOAP geradas pelo *pipeline* de IA. A Figura 1 apresenta o diagrama correspondente.

Figura 1 – Diagrama de casos de uso do Médico



Fonte: Elaborado pelo autor (2026).

O detalhamento dos casos de uso é apresentado a seguir:

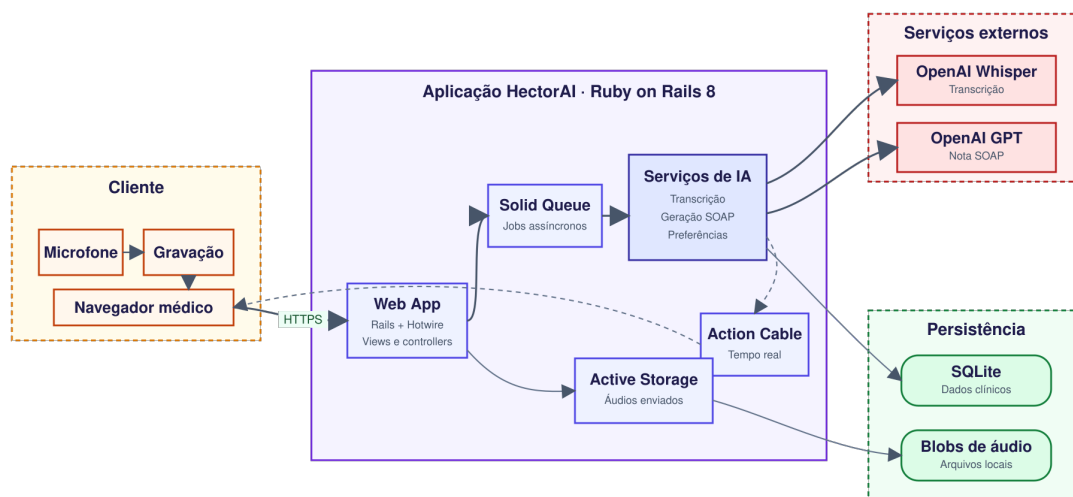
1. **Autenticar-se:** o médico se cadastra pelo formulário público, confirma o endereço de e-mail por meio do *link* enviado e, em seguida, autentica-se com as suas credenciais. A sessão estabelecida funciona como pré-requisito para o acesso aos demais recursos.
2. **Cadastrar paciente:** o médico preenche o formulário com os dados de identificação e o histórico médico relevante do paciente. Os campos sensíveis, a saber, nome, CPF (Cadastro de Pessoas Físicas) e e-mail, são criptografados no banco de dados pelo *Active Record Encryption*.
3. **Consultar paciente:** o médico acessa a lista dos seus pacientes, podendo filtrá-los por nome, e abre a visualização detalhada com os dados cadastrais e o histórico de consultas associadas.
4. **Agendar consulta:** o médico seleciona um dos seus pacientes ativos, define data, hora e, opcionalmente, uma nota livre. A consulta é criada como agendada, ainda sem áudio associado.
5. **Gravar áudio da consulta:** Na tela de início da consulta, o médico ativa o gravador embutido no navegador, que utiliza a *API (Application Programming Interface) MediaRecorder* para capturar o áudio. Ao confirmar a gravação, o arquivo é enviado ao servidor, anexado à consulta via *Active Storage* e a consulta transita para *enviando* e, na sequência, para *transcrevendo*.
6. **Acompanhar processamento:** Durante a transcrição e a geração da nota, o médico vê o status atualizado em tempo real na própria tela de detalhes da consulta, sem necessidade de recarregar a página, graças aos *Turbo Streams* transmitidos pelo *Action Cable*.
7. **Revisar e editar nota SOAP:** quando a consulta chega ao status *em revisão*, o médico visualiza a nota gerada com as seções S (Subjetivo), O (Objetivo), A (Avaliação) e P (Plano), e pode editar o texto livremente em um campo de edição dentro da própria tela de detalhes da consulta. As edições intermediárias podem ser salvas como rascunho.
8. **Aprovar nota SOAP:** Após a revisão, o médico aprova a nota, o que registra o instante da aprovação e a torna imutável. Se houver edições registradas, o sistema dispara em segundo plano o *job* de aprendizado de preferências, que usará as alterações como insumo para personalizar gerações futuras.

3.3 ARQUITETURA DO SISTEMA

O HectorAI adota a arquitetura monolítica do Ruby on Rails, organizada em torno do padrão *Model-View-Controller*. A esse núcleo somam-se filas de *jobs* para o

processamento assíncrono do áudio e Turbo Streams para a atualização incremental da interface. A Figura 2 apresenta uma visão geral dos componentes.

Figura 2 – Arquitetura do sistema



Fonte: Elaborado pelo autor (2026).

O navegador do médico é o único cliente do sistema. A camada de apresentação combina *views* ERB (*Embedded Ruby*) renderizadas pelo servidor, atualizações parciais via Turbo Frames e Turbo Streams e comportamentos pontuais em Stimulus, com destaque para o *controller* que orquestra o gravador de áudio. A captura do áudio é feita pela API *MediaRecorder* do navegador, e o *upload* ocorre por *multipart* sobre HTTPS (*Hypertext Transfer Protocol Secure*).

No servidor, os *controllers* do Rails recebem as requisições, persistem os dados via Active Record e delegam o trabalho custoso a *jobs* processados pelo Solid Queue. O primeiro *job* envia o áudio à API Whisper da OpenAI e devolve a transcrição; o segundo encaminha essa transcrição ao GPT-4o, recebe a nota SOAP estruturada e a persiste no prontuário. Um terceiro *job* extrai, dos textos editados pelo médico, os padrões de escrita a serem replicados nas próximas anotações. As transições de status da consulta seguem uma máquina de estados explícita, e cada mudança é propagada ao navegador via Action Cable, dando suporte ao acompanhamento em tempo real descrito na seção anterior.

A persistência ocorre em duas camadas do Rails. O Active Record, com banco SQLite, armazena as entidades de domínio (*Doctor*, *Patient*, *Consultation* e *MedicalRecord*) e aplica criptografia no banco de dados aos atributos pessoais do paciente. O Active Storage cuida dos arquivos de áudio, mantidos no disco local até a aprovação da consulta, quando podem ser descartados por rotina de limpeza. Os

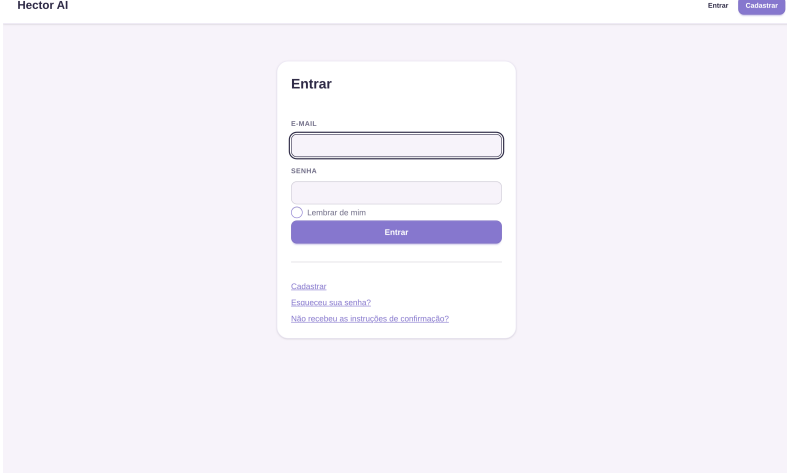
serviços externos da OpenAI (Whisper, para reconhecimento de fala, e GPT-4o, para geração da nota SOAP) ficam encapsulados em *service objects*, isolando o restante da aplicação dos detalhes de integração com cada modelo.

3.4 INTERFACE

Nesta seção, é discutido o fluxo de utilização do HectorAI a partir da apresentação das principais telas desenvolvidas no cliente Web. As capturas foram obtidas em ambiente de desenvolvimento, com dados fictícios.

O fluxo da aplicação começa na tela de *login* (Figura 3), ponto de entrada do sistema. O médico informa o e-mail e a senha já cadastrados para iniciar a sessão, sem a qual nenhum outro recurso fica acessível.

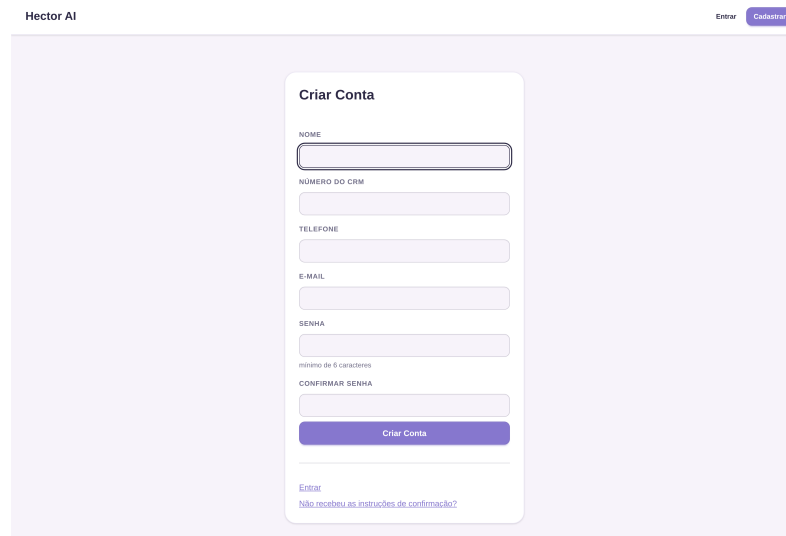
Figura 3 – Tela de *login*

A imagem mostra a interface de login do sistema Hector AI. No topo, há o nome 'Hector AI' à esquerda e os links 'Entrar' e 'Cadastrar' à direita. O formulário centralizado, intitulado 'Entrar', contém campos para 'E-MAIL' e 'SENHA'. Abaixo do campo de senha, há uma opção 'Lembrar de mim' com um botão de rádio desativado. Um botão azul 'Entrar' está posicionado abaixo do formulário. Na base do formulário, há links para 'Cadastrar', 'Esqueceu sua senha?' e 'Não recebeu as instruções de confirmação?'. O fundo da página é uma cor sólida de lavanda claro.

Fonte: Elaborado pelo autor (2026).

Quem ainda não possui conta acessa o formulário de cadastro (Figura 4), no qual informa nome, CRM (Conselho Regional de Medicina), telefone, e-mail e senha. A conta só é ativada após a confirmação do endereço de e-mail por meio do *link* enviado pelo Devise.

Figura 4 – Tela de cadastro do médico

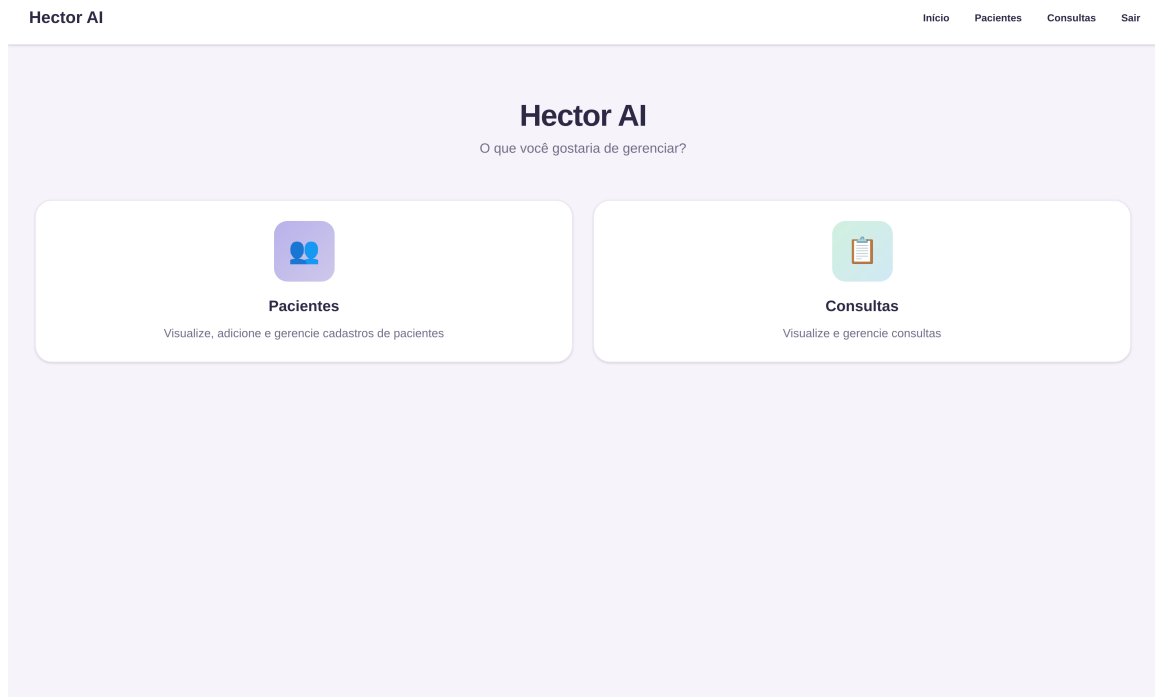


A interface de cadastro do médico, intitulada "Criar Conta", apresenta um formulário com os seguintes campos: NOME, NÚMERO DO CRM, TELEFONE, E-MAIL, SENHA (com a restrição "mínimo de 6 caracteres") e CONFIRMAR SENHA. Um botão "Criar Conta" em azul está posicionado abaixo dos campos. Na base do formulário, há links para "Entrar" e "Não recebeu as instruções de confirmação?". No topo da página, o nome "Hector AI" aparece à esquerda, e os botões "Entrar" e "Cadastre" à direita.

Fonte: Elaborado pelo autor (2026).

Concluída a autenticação, o usuário é direcionado à tela principal (Figura 5). Dois cartões clicáveis no corpo da página dão acesso aos fluxos centrais da aplicação, o gerenciamento de pacientes e o de consultas. A barra superior, presente em todas as telas autenticadas, mantém esses atalhos sempre disponíveis e oferece a opção de encerrar a sessão.

Figura 5 – Tela inicial



Fonte: Elaborado pelo autor (2026).

O fluxo de pacientes começa pela listagem (Figura 6), em que cada paciente aparece em um cartão com a inicial do nome e os principais dados de contato. Um

campo de busca no topo permite filtrar a grade pelo nome, enquanto o botão “Novo Paciente”, no canto superior direito, dá acesso ao formulário de cadastro.

Figura 6 – Lista de pacientes

Hector AI

InicioPacientesConsultasSair

Pacientes

Novo Paciente

Buscar por nome...

Buscar

A

Alice Morgan

DATA DE NASCIMENTO

14/03/1985

CPF

123.456.789-01

TELEFONE

(11) 98765-1234

E-MAIL

alice.morgan@example.com

B

Bruno Costa

DATA DE NASCIMENTO

22/07/1990

CPF

234.567.890-12

TELEFONE

(21) 97654-3210

E-MAIL

bruno.costa@example.com

C

Clara Chen

DATA DE NASCIMENTO

05/11/1978

CPF

345.678.901-23

TELEFONE

(31) 96543-2109

E-MAIL

clara.chen@example.com

D

Diego Ramirez

DATA DE NASCIMENTO

30/01/1995

CPF

456.789.012-34

TELEFONE

(41) 95432-1098

E-MAIL

diego.ramirez@example.com

E

Elena Petrov

DATA DE NASCIMENTO

18/09/1982

CPF

567.890.123-45

TELEFONE

(51) 94321-0987

E-MAIL

elena.petrov@example.com

F

Felipe Andrade

DATA DE NASCIMENTO

03/12/2000

CPF

678.901.234-56

TELEFONE

(61) 93210-9876

E-MAIL

felipe.andrade@example.co

G

Grace Okonkwo

DATA DE NASCIMENTO

11/06/1993

CPF

789.012.345-67

TELEFONE

(71) 92109-8765

E-MAIL

grace.okonkwo@example.cc

H

Hiroshi Tanaka

DATA DE NASCIMENTO

27/04/1971

CPF

890.123.456-78

TELEFONE

(81) 91098-7654

E-MAIL

hiroshi.tanaka@example.com

I

Isabela Santos

DATA DE NASCIMENTO

09/08/1988

CPF

901.234.567-89

TELEFONE

(91) 90987-6543

E-MAIL

isabela.santos@example.co

J

James Wright

DATA DE NASCIMENTO

15/02/1966

CPF

012.345.678-90

TELEFONE

(11) 99876-5432

E-MAIL

james.wright@example.com

K

Karina Sousa

DATA DE NASCIMENTO

21/10/1997

CPF

111.222.333-44

TELEFONE

(21) 98765-0123

E-MAIL

karina.sousa@example.com

L

Liam O'Brien

DATA DE NASCIMENTO

08/05/1974

CPF

222.333.444-55

TELEFONE

(31) 97654-9012

E-MAIL

liam.obrien@example.com

A

Alice Morgan

DATA DE NASCIMENTO

14/03/1985

CPF

123.456.789-01

TELEFONE

(11) 98765-1234

E-MAIL

alice.morgan@example.com

B

Bruno Costa

DATA DE NASCIMENTO

22/07/1990

CPF

234.567.890-12

TELEFONE

(21) 97654-3210

E-MAIL

bruno.costa@example.com

C

Clara Chen

DATA DE NASCIMENTO

05/11/1978

CPF

345.678.901-23

TELEFONE

(31) 96543-2109

E-MAIL

clara.chen@example.com

D

Diego Ramirez

DATA DE NASCIMENTO

30/01/1995

CPF

456.789.012-34

TELEFONE

(41) 95432-1098

E-MAIL

diego.ramirez@example.com

E

Elena Petrov

DATA DE NASCIMENTO

18/09/1982

CPF

567.890.123-45

TELEFONE

(51) 94321-0987

E-MAIL

elena.petrov@example.com

F

Felipe Andrade

DATA DE NASCIMENTO

03/12/2000

CPF

678.901.234-56

TELEFONE

(61) 93210-9876

E-MAIL

felipe.andrade@example.co

G

Grace Okonkwo

DATA DE NASCIMENTO

11/06/1993

CPF

789.012.345-67

TELEFONE

(71) 92109-8765

E-MAIL

grace.okonkwo@example.cc

H

Hiroshi Tanaka

DATA DE NASCIMENTO

27/04/1971

CPF

890.123.456-78

TELEFONE

(81) 91098-7654

E-MAIL

hiroshi.tanaka@example.com

I

Isabela Santos

DATA DE NASCIMENTO

09/08/1988

CPF

901.234.567-89

TELEFONE

(91) 90987-6543

E-MAIL

isabela.santos@example.co

J

James Wright

DATA DE NASCIMENTO

15/02/1966

CPF

012.345.678-90

TELEFONE

(11) 99876-5432

E-MAIL

james.wright@example.com

K

Karina Sousa

DATA DE NASCIMENTO

21/10/1997

CPF

111.222.333-44

TELEFONE

(21) 98765-0123

E-MAIL

karina.sousa@example.com

L

Liam O'Brien

DATA DE NASCIMENTO

08/05/1974

CPF

222.333.444-55

TELEFONE

(31) 97654-9012

E-MAIL

liam.obrien@example.com

Fonte: Elaborado pelo autor (2026).

O formulário de cadastro de paciente (Figura 7) reúne os dados pessoais e

o histórico médico relevante. Os campos sensíveis, como nome, CPF e e-mail, são criptografados no banco pelo Active Record *Encryption* antes da persistência.

Figura 7 – Formulário de cadastro de paciente

Hector AI

Inicio Pacientes Consultas Sair

Novo Paciente

NOME

Nome completo

DATA DE NASCIMENTO

mm/dd/yyyy

CPF

000.000.000-00

TELEFONE

Telefone (opcional)

E-MAIL

E-mail (opcional)

HISTÓRICO MÉDICO

Histórico médico relevante, alergias, condições...

Cadastrar Paciente

Cancelar

Fonte: Elaborado pelo autor (2026).

Ao clicar em um cartão da listagem, abre-se a página de detalhes do paciente (Figura 8), com os dados cadastrais, o histórico médico relevante e a tabela das consultas anteriores associadas. Dessa mesma tela, o botão “Editar” no canto superior permite atualizar os dados do paciente.

Figura 8 – Detalhes do paciente

Hector AI

Inicio Pacientes Consultas Sair

Alice Morgan

Editar

Arquivar

Voltar

Dados do Paciente

DATA DE NASCIMENTO

14/03/1985

CPF

123.456.789-01

TELEFONE

(11) 98765-1234

E-MAIL

alice.morgan@example.com

HISTÓRICO MÉDICO

Type 2 diabetes diagnosed 2018. Metformin 500mg twice daily. Seasonal allergies.

Histórico de Consultas

Data	Status	Anotações
02/06/2026 às 14:30	Agendada	Primeira consulta de retorno
01/06/2026 às 21:54	Transcrevendo	
01/06/2026 às 21:34	Gerando	
01/06/2026 às 21:04	Em revisão	
01/06/2026 às 20:04	Aprovada	

Fonte: Elaborado pelo autor (2026).

O segundo grande processo da aplicação é o de consultas, cuja listagem é apresentada na Figura 9. As consultas aparecem em formato tabular, com paciente, data agendada, status e *link* para a tela de detalhes.

Figura 9 – Lista de consultas

Hector AI

Início

Pacientes

Consultas

Sair

Consultas

Todas

Agendada

Enviando

Transcrevendo

Em revisão

Aprovada

Nova Consulta

Paciente	Agendada para	Status	
Alice Morgan	01/06/2026 20:04	Aprovada	Visualizar
Alice Morgan	01/06/2026 21:04	Em revisão	Visualizar
Alice Morgan	01/06/2026 21:34	Gerando	Visualizar
Alice Morgan	01/06/2026 21:54	Transcrevendo	Visualizar
Clara Chen	04/06/2026 16:15	Agendada	Visualizar
Bruno Costa	03/06/2026 09:00	Agendada	Visualizar
Alice Morgan	02/06/2026 14:30	Agendada	Visualizar

Fonte: Elaborado pelo autor (2026).

Sobre a tabela, uma fileira de filtros em pílula permite restringir a visualização por status. A Figura 10 ilustra o filtro de revisão acionado, situação em que apenas as consultas em revisão permanecem visíveis.

Figura 10 – Lista de consultas com filtro por status

Hector AI

Início

Pacientes

Consultas

Sair

Consultas

Nova Consulta

Todas

Agendada

Enviando

Transcrevendo

Em revisão

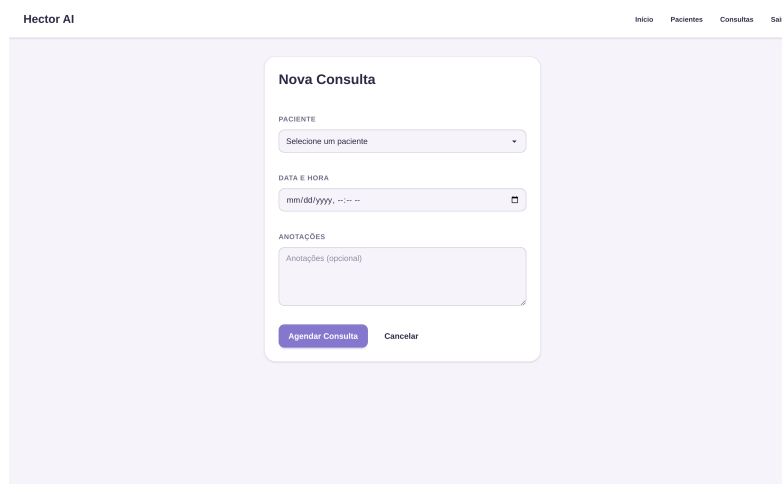
Aprovada

Paciente	Agendada para	Status	
Alice Morgan	01/06/2026 21:04	Em revisão	Visualizar

Fonte: Elaborado pelo autor (2026).

O botão “Nova Consulta” abre o formulário de agendamento (Figura 11), em que o médico seleciona o paciente entre os seus cadastrados, define data e hora e pode incluir uma nota livre opcional. A consulta é criada no status *agendada*, ainda sem áudio associado.

Figura 11 – Formulário de agendamento de consulta



Hector AI

Início Pacientes Consultas Sair

Nova Consulta

PACIENTE

Selecione um paciente

DATA E HORA

mm/dd/yyyy, --:-- --

ANOTAÇÕES

Anotações (opcional)

Agendar Consulta Cancelar

Fonte: Elaborado pelo autor (2026).

Concluído o agendamento, o médico é direcionado para a tela de detalhes da consulta no status *agendada* (Figura 12). A tela exibe os dados do agendamento e um botão em destaque que dá início à consulta propriamente dita.

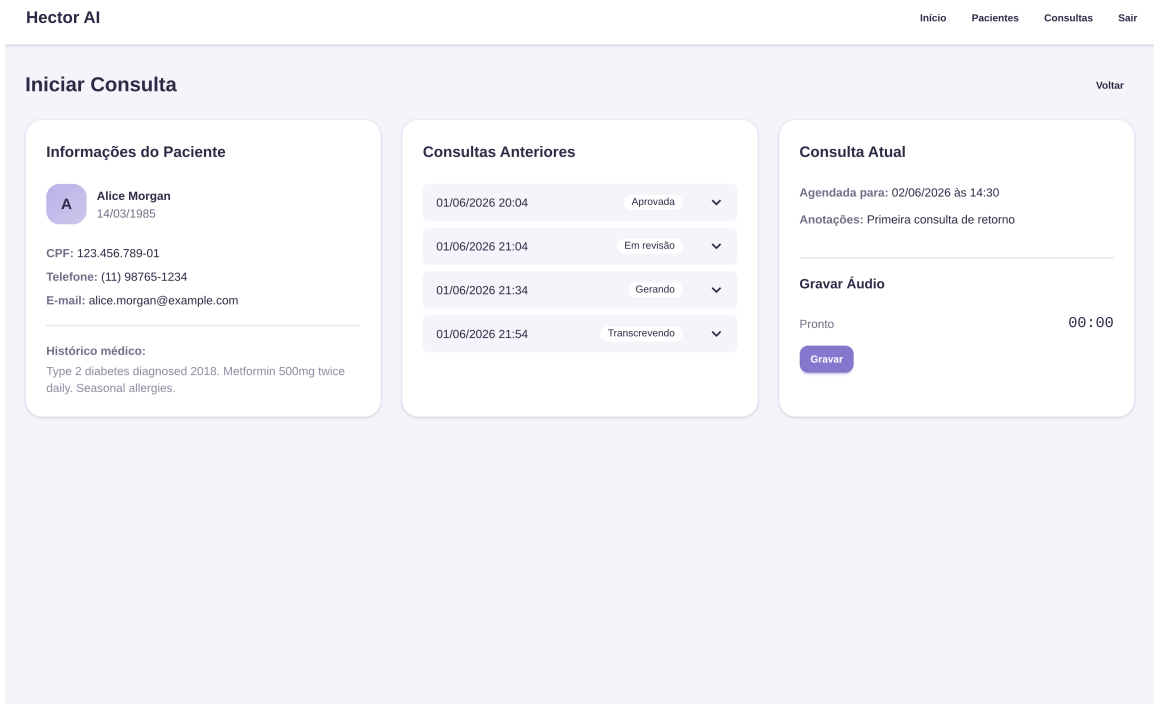
Figura 12 – Detalhes da consulta no status *agendada*

A interface da aplicação Hector AI apresenta uma barra de navegação superior com os links 'Início', 'Pacientes', 'Consultas' e 'Sair'. O título 'Hector AI' está no canto superior esquerdo. O conteúdo principal é dividido em duas seções. A primeira seção, intitulada 'Consulta', contém as seguintes informações: 'Paciente: Alice Morgan', 'Agendada para: 02/06/2026 às 14:30', 'Anotações: Primeira consulta de retorno' e 'Status: Agendada'. Abaixo dessas informações, há um botão azul com o texto 'Iniciar Consulta'. A segunda seção, localizada abaixo da primeira, contém um único botão com o texto 'Agendada'.

Fonte: Elaborado pelo autor (2026).

Ao acionar esse botão, a aplicação leva à tela de início da consulta (Figura 13), organizada em três seções: à esquerda, as informações do paciente; ao centro, o histórico das consultas anteriores com as respectivas notas SOAP; à direita, o gravador de áudio, pelo qual o médico inicia, pausa e descarta a gravação. Ao confirmar, o áudio é enviado ao servidor por requisição assíncrona disparada pelo *controller* Stimulus correspondente.

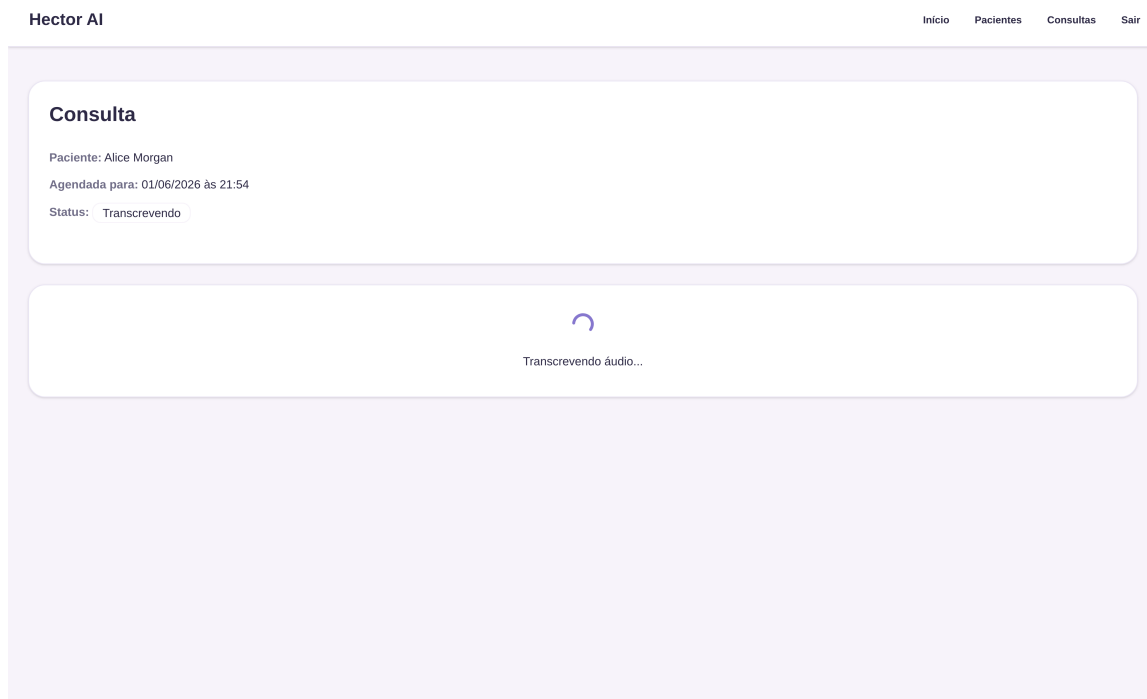
Figura 13 – Tela de início da consulta com o gravador de áudio



Fonte: Elaborado pelo autor (2026).

Enviado o áudio, a consulta passa a percorrer uma máquina de estados cujas transições aparecem na interface em tempo real, sem que o médico recarregue a página. O primeiro estado é *transcrevendo* (Figura 14), no qual um indicador de carregamento substitui a área da nota SOAP enquanto o áudio é encaminhado à API Whisper para reconhecimento de fala.

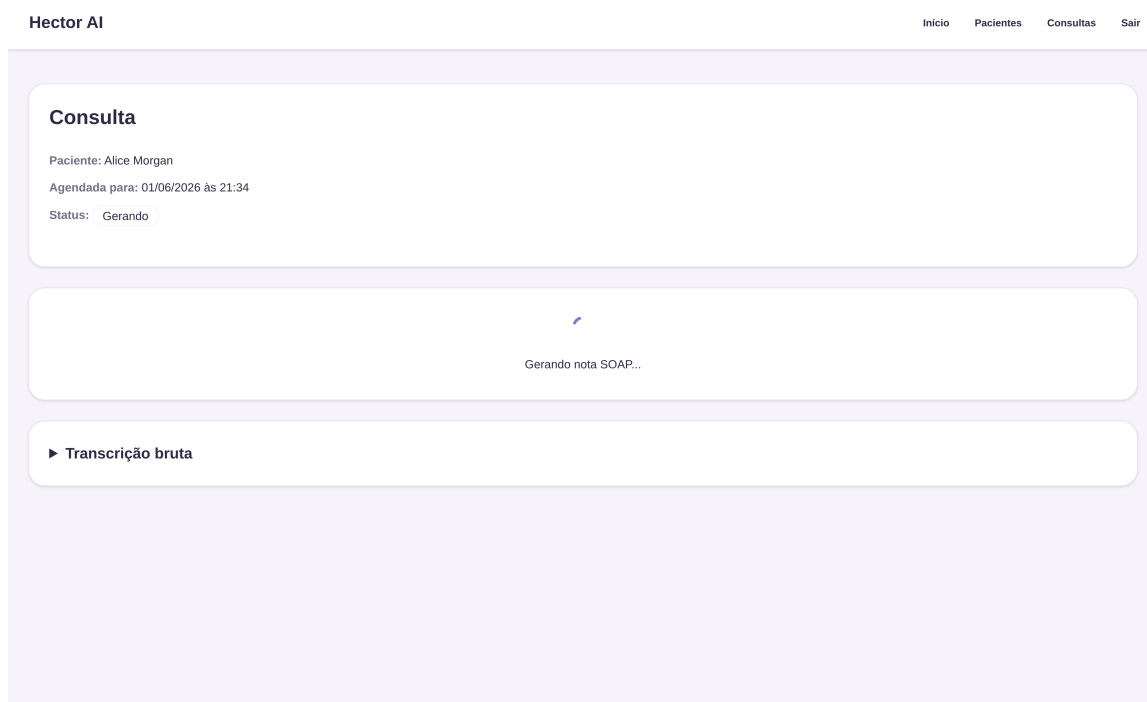
Figura 14 – Consulta em transcrição



Fonte: Elaborado pelo autor (2026).

Devolvida a transcrição, o status migra para *gerando* (Figura 15), em que o texto reconhecido é submetido ao GPT-4o para a produção da nota SOAP. Nesse mesmo momento, a transcrição bruta entregue pelo Whisper passa a ficar acessível em uma seção colapsável no rodapé da tela.

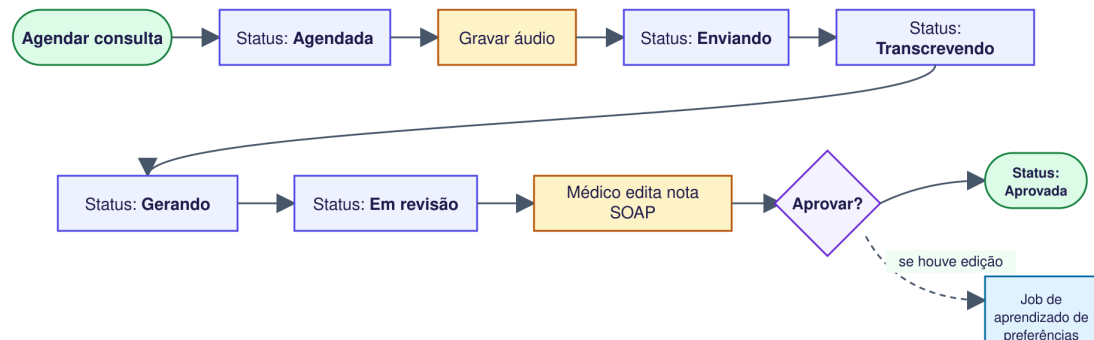
Figura 15 – Consulta em geração da nota SOAP



Fonte: Elaborado pelo autor (2026).

A Figura 16 sintetiza, em forma de diagrama de estados, o ciclo completo de uma consulta no HectorAI, desde o agendamento até a aprovação, e explicita também as transições para os estados de erro.

Figura 16 – Fluxo de estados de uma consulta



Fonte: Elaborado pelo autor (2026).

Quando o *pipeline* de IA termina, a consulta entra no status *em revisão* (Figura 17). A nota SOAP é apresentada em um editor de texto livre dividido nas seções S (Subjetivo), O (Objetivo), A (Avaliação) e P (Plano), e o médico pode salvá-la como rascunho a qualquer momento. Abaixo do editor, um botão verde “Aprovar” formaliza a aprovação, e a transcrição bruta entregue pelo Whisper continua disponível em uma seção colapsável, recurso útil para auditar a fonte caso a nota gerada divirja do que foi dito na consulta.

Figura 17 – Tela de revisão da nota SOAP

Hector AI

[Início](#) [Pacientes](#) [Consultas](#) [Sair](#)

Consulta

Paciente: Alice Morgan

Agendada para: 01/06/2026 às 21:04

Status: Em revisão

Revisar nota SOAP

Nota SOAP

S (Subjetivo):

Paciente do sexo feminino, 40 anos, refere cefaleia frontal há três dias, de intensidade moderada, acompanhada de fotofobia leve. Nega febre, vômitos ou perda de força.

O (Objetivo):

Bom estado geral, corada, hidratada, afebril. PA: 128/82 mmHg. FC: 78 bpm. Exame neurológico sem alterações. Pares cranianos preservados.

A (Avaliação):

Cefaleia tensional. Sem sinais de alerta para causas secundárias no exame atual.

P (Plano):

Orientação sobre higiene do sono e hidratação. Dipirona 500 mg via oral até de 6/6 horas em caso de dor. Retorno em 15 dias ou antes se piora.

Salvar Rascunho

Aprovar

► Transcrição bruta

Fonte: Elaborado pelo autor (2026).

Após a aprovação, a consulta passa ao status *aprovada* (Figura 18). A nota SOAP fica em modo somente leitura, a data e a hora da aprovação são registradas no cabeçalho e, em segundo plano, dispara-se o *job* de aprendizado de preferências sempre que houver edições no texto gerado pela IA.

Figura 18 – Consulta aprovada

Hector AI

Início

Pacientes

Consultas

Sair

Consulta

Paciente: Alice Morgan

Agendada para: 01/06/2026 às 20:04

Status:

Aprovada

Nota SOAP

Aprovada

01/06/2026 às 21:59

S (Subjetivo):

Paciente do sexo feminino, 40 anos, refere cefaleia frontal há três dias, de intensidade moderada, acompanhada de fotofobia leve. Nega febre, vômitos ou perda de força.

O (Objetivo):

Bom estado geral, corada, hidratada, afebril. PA: 128/82 mmHg. FC: 78 bpm. Exame neurológico sem alterações. Pares cranianos preservados.

A (Avaliação):

Cefaleia tensional. Sem sinais de alerta para causas secundárias no exame atual.

P (Plano):

Orientação sobre higiene do sono e hidratação. Dipirona 500 mg via oral até de 6/6 horas em caso de dor. Retorno em 15 dias ou antes se piora.

► Transcrição bruta

Fonte: Elaborado pelo autor (2026).

4 CONCLUSÃO

Este trabalho apresentou o HectorAI, uma aplicação Web monolítica feita em Ruby on Rails 8 que automatiza a geração de notas clínicas em formato SOAP a partir do áudio das consultas médicas. O *pipeline* de inteligência artificial foi validado utilizando a OpenAI Whisper para a transcrição do áudio e o GPT-4o para a geração do texto estruturado, provando ser uma forma ágil e confiável de entregar ao médico notas em português brasileiro prontas para revisão. A execução assíncrona dos *jobs* é bem estruturada por meio do Solid Queue, que garante uma camada extra de fluidez sob uma interface atualizada incrementalmente pelo Hotwire. A arquitetura da aplicação seguiu as convenções do Ruby on Rails e práticas modernas de código limpo, que garantem a fácil integração de novas funcionalidades ao projeto no futuro.

Por limitações de tempo, não foi possível realizar a implantação do HectorAI em consultórios médicos durante o desenvolvimento deste relatório, de modo que os resultados do uso da ferramenta em ambiente clínico real não serão apresentados. Caso essa etapa fosse possível, o trabalho seria enriquecido por métricas práticas sobre tempo de documentação, aceitação dos médicos e qualidade das notas geradas em situações reais de atendimento, tornando a avaliação da solução mais completa.

Como trabalhos futuros, para além das notas SOAP, sugere-se a extensão do *pipeline* para formatos de documentação clínica complementares, como evoluções de prontuário em consultas de seguimento e laudos breves, para que a solução cubra um espectro mais amplo da rotina do consultório. O escopo da solução proposta aqui entrega a geração da nota clínica, mas o médico ainda precisa transitar entre o HectorAI e o sistema de prontuário eletrônico de sua clínica para finalizar o registro do atendimento, de modo que a integração com esses sistemas já adotados no mercado brasileiro é também proposta. Como sugerido anteriormente, também se propõe a adequação plena à Lei Geral de Proteção de Dados Pessoais, com migração da infraestrutura de produção para provedores nacionais de nuvem, extensão da criptografia aos arquivos de áudio armazenados via Active Storage e implementação das rotinas de anonimização, remoção definitiva de dados e consentimento explícito do paciente, para efetivamente mitigar os riscos do tratamento de dados clínicos sensíveis.

Ademais, este relatório apresenta um modelo replicável de desenvolvimento de assistentes clínicos de inteligência artificial voltados à prática médica em português, visto que apesar do foco voltado para a geração de notas SOAP, é notória a facilidade de expansão dessa solução para outros contextos de documentação clínica. Com o desenvolvimento em ferramentas de código aberto, espera-se que a solução proposta seja um protótipo acessível e de valor para a comunidade de médicos e pesquisadores brasileiros que pretendam discutir o tema a partir de uma implementação concreta.

REFERÊNCIAS BIBLIOGRÁFICAS

37signals. *HTML Over The Wire*. 2026. Página institucional. Disponível em: <<https://hotwired.dev/>>. Acesso em: 17 jun. 2026.

Conselho Federal de Medicina. Resolução, *Resolução CFM nº 1.638, de 10 de julho de 2002*: Define prontuário médico e torna obrigatória a criação da comissão de revisão de prontuários nas instituições de saúde. Brasília: Conselho Federal de Medicina, 2002. Disponível em: <https://sistemas.cfm.org.br/normas/arquivos/resolucoes/BR/2002/1638_2002.pdf>. Acesso em: 3 maio 2026.

GARDNER, R. L.; COOPER, E.; HASKELL, J.; HARRIS, D. A.; POPLAU, S.; KROTH, P. J.; LINZER, M. Physician stress and burnout: the impact of health information technology. *Journal of the American Medical Informatics Association*, v. 26, n. 2, p. 106–114, 2019. ISSN 1527-9995. DOI: 10.1093/jamia/ocy145. Disponível em: <<https://academic.oup.com/jamia/article/26/2/106/5230918>>. Acesso em: 8 maio 2026.

HANSSON, D. H. *tailwindcss-rails: version 4.4.0*. 2025. Pacote de software, versão 4.4.0, RubyGems.org. Disponível em: <<https://rubygems.org/gems/tailwindcss-rails/versions/4.4.0>>. Acesso em: 17 jun. 2026.

Heartcombo. *Devise: version 5.0.3*. 2026. Código-fonte, versão 5.0.3, GitHub. Disponível em: <<https://github.com/heartcombo/devise/tree/v5.0.3>>. Acesso em: 17 jun. 2026.

Hotwire. *Stimulus Handbook: A JavaScript framework with modest ambitions*. Chicago, 2026. Documentação oficial. Disponível em: <<https://stimulus.hotwired.dev/handbook/introduction>>. Acesso em: 19 maio 2026.

Hotwire. *Turbo Handbook*. Chicago, 2026. Documentação oficial. Disponível em: <<https://turbo.hotwired.dev/handbook/introduction>>. Acesso em: 17 maio 2026.

OpenAI. *GPT-4o Model: OpenAI API Documentation*. San Francisco, 2024. Documentação oficial da API. Disponível em: <<https://developers.openai.com/api/docs/models/gpt-4o>>. Acesso em: 27 maio 2026.

PAOLINO, C. *RubyLLM: One beautiful Ruby API for GPT, Claude, Gemini, and more*. 2026. Documentação oficial do software. Disponível em: <<https://rubyllm.com/>>. Acesso em: 23 maio 2026.

Phyx Primary Care. Whitepaper, *Phyx Primary Care Report: Ambient AI Assistant Transforms the Primary Care Experience*. Redwood City, CA: Suki AI, 2025. Disponível em: <<https://www.suki.ai/whitepapers/phyx-primary-care-report-ambient-ai-assistant-transforms-the-primary-care-experience/>>. Acesso em: 12 maio 2026.

RADFORD, A.; KIM, J. W.; XU, T.; BROCKMAN, G.; MCLEAVEY, C.; SUTSKEVER, I. Robust speech recognition via large-scale weak supervision. *arXiv preprint*, arXiv:2212.04356, n. eess.AS, 2022. DOI: 10.48550/arXiv.2212.04356. Disponível em: <<https://arxiv.org/abs/2212.04356>>. Acesso em: 29 maio 2026.

Ruby Core Team. *Ruby Releases*. 2026. Página institucional. Disponível em: <<https://www.ruby-lang.org/en/downloads/releases/>>. Acesso em: 17 jun. 2026.

Ruby on Rails. *Solid Cable: version 3.0.12*. 2025. Código-fonte, versão 3.0.12, GitHub. Disponível em: <https://github.com/rails/solid_cable/tree/v3.0.12>. Acesso em: 17 jun. 2026.

Ruby on Rails. *Solid Cache: version 1.0.10*. 2025. Código-fonte, versão 1.0.10, GitHub. Disponível em: <https://github.com/rails/solid_cache/tree/v1.0.10>. Acesso em: 17 jun. 2026.

Ruby on Rails. *Active Storage: Rails 8.1.3 API Documentation*. Chicago, 2026. Documentação oficial da API. Disponível em: <<https://api.rubyonrails.org/v8.1.3/classes/ActiveStorage.html>>. Acesso em: 17 jun. 2026.

Ruby on Rails. *Ruby on Rails 8.1.3 API Documentation*. Chicago, 2026. Documentação oficial da API. Disponível em: <<https://api.rubyonrails.org/v8.1.3/>>. Acesso em: 17 jun. 2026.

Ruby on Rails. *Solid Queue: version 1.4.0*. 2026. Código-fonte, versão 1.4.0, GitHub. Disponível em: <https://github.com/rails/solid_queue/tree/v1.4.0>. Acesso em: 17 jun. 2026.

Sparklemotion. *SQLite3 Ruby: version 2.9.2*. 2026. Código-fonte, versão 2.9.2, GitHub. Disponível em: <<https://github.com/sparklemotion/sqlite3-ruby/tree/v2.9.2>>. Acesso em: 17 jun. 2026.

SQLite Consortium. *About SQLite*. 2026. Página institucional. Disponível em: <<https://www.sqlite.org/about.html>>. Acesso em: 17 jun. 2026.

STEPHENSON, S.; MAHKMALI, J.; HANSSON, D. H. *stimulus-rails: version 1.3.4*. 2024. Pacote de software, versão 1.3.4, RubyGems.org. Disponível em: <<https://rubygems.org/gems/stimulus-rails/versions/1.3.4>>. Acesso em: 17 jun. 2026.

STEPHENSON, S.; MAHKMALI, J.; HANSSON, D. H. *turbo-rails: version 2.0.23*. 2026. Pacote de software, versão 2.0.23, RubyGems.org. Disponível em: <<https://rubygems.org/gems/turbo-rails/versions/2.0.23>>. Acesso em: 17 jun. 2026.

Tailwind Labs. *Tailwind CSS: version 4.2.1*. 2026. Código-fonte, versão 4.2.1, GitHub. Disponível em: <<https://github.com/tailwindlabs/tailwindcss/releases/tag/v4.2.1>>. Acesso em: 17 jun. 2026.

tailwindcss-ruby contributors. *tailwindcss-ruby: version 4.2.1*. 2026. Pacote de software, versão 4.2.1, RubyGems.org. Disponível em: <<https://rubygems.org/gems/tailwindcss-ruby/versions/4.2.1>>. Acesso em: 17 jun. 2026.

ANEXO A – DECLARAÇÃO DE ENTREGA DO CÓDIGO-FONTE

Documento emitido pela Coordenação do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central, atestando a entrega do código-fonte do Relatório de Software (Trabalho de Conclusão de Curso).



Ministério da Educação
Instituto Federal de Educação, Ciência e Tecnologia do Piauí
IFPI - CAMPUS TERESINA CENTRAL
Praça da Liberdade, 1597, Centro, TERESINA / PI, CEP 64.000-040
Fone: Site: www.ifpi.edu.br

DECLARAÇÃO 15/2026 - CCADS/DIAS/DENS/DG-TERCENT/CATCE/IFPI

TERESINA, 4 de junho de 2026.

Declaro, para os devidos fins, que o aluno do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Paulo de Tarso Vilarinho Castelo Branco Filho, matrícula 20181ADS0450, entregou à coordenação do referido curso o código-fonte referente ao relatório de software (Trabalho de Conclusão de Curso).

ALINE MONTENEGRO LEAL

Coordenadora do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

Documento assinado eletronicamente por:

■ **Aline Montenegro Leal**, COORDENADOR(A) DE CURSO - FUC0001 - CCADS-IFPI - CAMPUS TERESINA CENTRAL, em 04/06/2026 19:29:12.

Este documento foi emitido pelo SUAP em 04/06/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpi.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 457057

Código de Autenticação: c97648883c

