



UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

BRUNO BRUGNEROTTO DE LARA, GABRIEL FRANCELINO VOIDALESKI, PEDRO
HENRIQUE LOPES

APPUNTURE: ATLAS DIGITAL EDUCATIVO DE ACUPUNTURA

CURITIBA

2025

BRUNO BRUGNEROTTO DE LARA, GABRIEL FRANCELINO VOIDALESKI, PEDRO
HENRIQUE LOPES

APPUNTURE: ATLAS DIGITAL EDUCATIVO DE ACUPUNTURA

Trabalho de Conclusão de Curso apresentado
como requisito parcial à obtenção do grau de
Tecnólogo em Análise e Desenvolvimento de
Sistemas, Setor de Ciências Exatas da Univer-
sidade Federal do Paraná..

Orientador: Prof. Dr. Paulo Sobreira Moraes

CURITIBA

2025

RESUMO

O *Appunture* é um atlas digital educativo voltado a estudantes e profissionais interessados em acupuntura, com o objetivo de facilitar o acesso a informações sobre os pontos de acupuntura corporal. Em um cenário onde a formação na área é heterogênea e o acesso a materiais de qualidade pode ser limitado, a ferramenta busca auxiliar no processo de aprendizagem e consulta, disponibilizando um atlas anatômico digital organizado por meridianos.

A plataforma foi desenvolvida para Android, apresentando 16 visualizações vetoriais (SVG) organizadas por meridianos e vistas (frontal e posterior). O sistema oferece funcionalidades de busca, detalhes dos 361 pontos principais, e um assistente baseado em Inteligência Artificial para auxiliar na correlação entre sintomas e pontos.

O desenvolvimento seguiu uma arquitetura “offline-first”, permitindo o acesso aos dados mesmo sem conexão à internet. Para auxiliar no mapeamento dos pontos sobre as imagens anatômicas, foi desenvolvida uma ferramenta interna de posicionamento de coordenadas.

É importante ressaltar que o *Appunture* não substitui a formação profissional adequada nem o acompanhamento de profissionais qualificados. A ferramenta foi concebida exclusivamente como um recurso educativo complementar de apoio ao estudo e à consulta de informações sobre acupuntura, não devendo ser utilizada para fins de diagnóstico, prescrição ou tratamento.

Palavras-chaves: Acupuntura; Atlas Digital; Aplicativo Educativo; Medicina Tradicional Chinesa; Tecnologia Educacional.

ABSTRACT

Appunture is an educational digital atlas designed to facilitate access to information about acupuncture points for students and professionals interested in the field. The Android application features a digital anatomical atlas with 16 vector views (SVG), organized by meridians. The system includes search functionalities, details of 361 main points, and an AI-based assistant to help correlate symptoms and points.

Developed with an “offline-first” architecture using React Native and Java Spring Boot, the application allows data access even without internet connection. An internal mapping tool was developed to assist in positioning point coordinates on the anatomical images.

It is important to note that *Appunture* does not replace proper professional training. The tool was designed exclusively as a complementary educational resource to support learning and information lookup about acupuncture, and should not be used for diagnosis, prescription, or treatment purposes.

Key-words: Acupuncture; Digital Atlas; Educational Application; Traditional Chinese Medicine; Educational Technology.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

QUADRO 4.1 – Exemplo de integração Spring AI	28
QUADRO 5.1 – Product Backlog do Appunture	35
QUADRO 5.2 – Estimativa por Pontos de Função	36
QUADRO 5.3 – Cronograma de Sprints	37

LISTA DE TABELAS

LISTA DE ABREVIATURAS E DE SIGLAS

API	Application Programming Interface
JWT	JSON Web Token
LLM	Large Language Model
MMKV	Memory Mapped Key-Value
MTC	Medicina Tradicional Chinesa
OMS	Organização Mundial da Saúde
RAG	Retrieval-Augmented Generation
RBAC	Role-Based Access Control
SDK	Software Development Kit
SQL	Structured Query Language
SVG	Scalable Vector Graphics
UML	Unified Modeling Language

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.1.1	Objetivos Específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	ACESSIBILIDADE COMO DIREITO E FERRAMENTA DE IGUALDADE NA SAÚDE	14
2.2	TECNOLOGIAS DIGITAIS COMO APOIO À FORMAÇÃO E AO ES-TUDO	15
2.3	APLICATIVOS DE ACUPUNTURA: EXPERIÊNCIAS E RECURSOS TECNOLÓGICOS	16
2.4	COMPARATIVO COM APLICATIVOS EXISTENTES	17
2.5	USABILIDADE E DESIGN CENTRADO NO USUÁRIO	18
2.6	NORMAS, DIRETRIZES E EVIDÊNCIAS CIENTÍFICAS	18
3	MOTIVAÇÕES	20
3.1	REGULAMENTAÇÃO E DISPUTA DE CATEGORIAS PROFISSIONAIS	20
3.2	FORMAÇÃO E FISCALIZAÇÃO INSUFICIENTES	20
3.3	SEGURANÇA DO PACIENTE E BIOSSEGURANÇA	21
3.4	DIAGNÓSTICO ENERGÉTICO: COMPLEXIDADE E SUBJETIVIDADE	21
4	TECNOLOGIAS E ARQUITETURA DO SISTEMA	23
4.1	VISÃO GERAL DA ARQUITETURA	23
4.1.1	Diagrama de Arquitetura	23
4.2	ARQUITETURA HÍBRIDA (OFFLINE + ONLINE)	23
4.2.1	Funcionamento Offline	23
4.2.2	Sincronização Inteligente	24
4.3	STACK TECNOLÓGICO DETALHADO	25
4.3.1	Frontend Mobile (React Native + Expo)	25
4.3.2	Backend API (Java + Spring Boot)	25
4.3.3	Banco de Dados	26
4.4	INTELIGÊNCIA ARTIFICIAL E SUPORTE À CONSULTA	27
4.4.1	Arquitetura RAG (Retrieval-Augmented Generation)	27
4.4.2	Integração com Spring AI	28
4.5	FERRAMENTAS DE DESENVOLVIMENTO E MAPEAMENTO	28
4.5.1	Point Mapper - Ferramenta de Mapeamento	28
4.5.2	Scripts de Migração e Integração	29
4.6	TECNOLOGIAS DE SUPORTE	29
4.6.1	Gerenciamento de Estado (Mobile)	29
4.6.2	Interface Web Administrativa (React)	29
5	METODOLOGIA	31
5.1	METODOLOGIA DE DESENVOLVIMENTO	31
5.1.1	Scrum	31

5.2	MODELAGEM DO PROJETO	32
5.2.1	Diagrama de Casos de Uso	32
5.2.2	Histórias de Usuário	33
5.2.2.1	Histórias de Usuário do Projeto	33
5.2.3	Diagrama Físico do Banco de Dados	34
5.3	PRODUCT BACKLOG	35
5.4	PLANEJAMENTO DE SPRINTS	36
5.4.1	Estimativa Geral por Pontos de Função (PF)	36
5.4.2	Cálculo de Esforço	36
5.5	FERRAMENTAS DE DESENVOLVIMENTO	37
5.5.1	React Native	37
5.5.2	Expo	37
5.5.3	Java	38
5.5.4	Spring Boot	38
5.5.5	Google Cloud Firestore	38
5.5.6	SQLite	38
6	APRESENTAÇÃO DO SISTEMA	39
6.1	ARQUITETURA DO SISTEMA	39
6.1.1	Arquitetura Híbrida (Offline + Online)	39
6.1.2	Comunicação e Sincronização	39
6.1.3	Assistente de IA para Busca Inteligente	40
6.1.4	Interface Interativa e Navegação	40
6.1.5	Backend e Web Admin	40
6.2	AUTENTICAÇÃO E SEGURANÇA	40
6.3	FUNCIONALIDADES DO SISTEMA	41
6.3.1	Tela Inicial e Onboarding	41
6.3.2	Cadastro e Autenticação via Firebase	41
6.3.3	Tela Principal e Navegação	42
6.3.4	Mapa Anatômico Interativo	42
6.3.5	Detalhes dos Pontos de Acupuntura	43
6.3.6	Sistema de Busca Inteligente	43
6.3.7	Assistente Inteligente com IA Generativa	44
6.3.8	Sistema de Favoritos e Anotações Pessoais	44
6.4	GESTÃO DE DADOS	45
6.5	EXPERIÊNCIA DO USUÁRIO E USABILIDADE	45
6.6	PROTÓTIPO DE TELAS	45
7	CONSIDERAÇÕES FINAIS	46
	REFERÊNCIAS	48
	APÊNDICE A – DIAGRAMA DE CASOS DE USO	50
	APÊNDICE B – HISTÓRIAS DE USUÁRIO	51

	APÊNDICE C – DIAGRAMA DE CLASSES	60
	APÊNDICE D – DIAGRAMAS DE SEQUÊNCIA	61
	APÊNDICE E – MODELO LÓGICO DE DADOS	62
	APÊNDICE F – ESPECIFICAÇÃO DE APIS	65
F.1	AUTENTICAÇÃO	65
F.2	PONTOS DE ACUPUNTURA	65
F.3	SINCRONIZAÇÃO	67
F.4	ASSISTENTE DE IA	68
	APÊNDICE G – ARQUITETURA RAG DO ASSISTENTE DE IA . .	69

1 INTRODUÇÃO

A prática da acupuntura no Brasil, embora cada vez mais difundida e procurada tanto no sistema público quanto no privado de saúde, ainda enfrenta desafios significativos no que diz respeito à regulamentação profissional, padronização de condutas, segurança técnica e formação dos praticantes. Trata-se de uma técnica milenar baseada na Medicina Tradicional Chinesa (MTC), que se expandiu no país por meio de cursos livres, técnicos e pós-graduações voltadas a diferentes profissionais da área da saúde. No entanto, a ausência de regulamentação federal clara e a disputa entre conselhos profissionais sobre quem pode exercer legalmente a acupuntura têm gerado um ambiente de insegurança jurídica, afetando não apenas os acupunturistas, mas também o acesso qualificado da população à prática.

Além das disputas institucionais, o cenário atual é marcado pela fiscalização fragmentada, formação heterogênea e riscos crescentes à biossegurança e à saúde do paciente, especialmente em ambientes informais. Segundo diretrizes da Organização Mundial da Saúde (OMS), o uso inadequado de agulhas, a falta de higienização correta ou a aplicação indevida de técnicas podem resultar em infecções, lesões internas e outros agravos que comprometem gravemente a integridade dos usuários da acupuntura (Organização Mundial da Saúde, 1999).

Outro desafio relevante é o caráter subjetivo e complexo do diagnóstico energético, próprio da MTC. O raciocínio clínico baseado em padrões de desarmonia (como deficiência de Qi, estagnação de sangue, entre outros) exige extenso treinamento e sensibilidade clínica. A falta de padronização e apoio didático dificulta a atuação de profissionais iniciantes ou com formação limitada, podendo comprometer a eficácia terapêutica.

Diante desse contexto, o uso de Tecnologias Digitais de Informação e Comunicação (TDIC) apresenta-se como uma ferramenta estratégica para apoiar a qualificação da prática da acupuntura. Soluções como aplicativos educativos e interativos podem auxiliar na visualização de pontos, organização de informações terapêuticas, revisão diagnóstica e adequação de conteúdos conforme o nível de formação do usuário, promovendo maior segurança, padronização e confiabilidade.

Neste cenário, propõe-se o desenvolvimento do *Appuncture*, uma plataforma educativa multiplataforma (Android, iOS e Web) que funciona como um atlas digital de acupuntura, voltada a estudantes e profissionais interessados na área. A ferramenta oferece acesso *offline* a informações sobre os 361 pontos dos meridianos principais, pontos extras, e um atlas anatômico composto por 15 visualizações vetoriais (SVG),

organizadas por meridianos e vistas (frontal e posterior). O aplicativo também disponibiliza um assistente baseado em Inteligência Artificial para auxiliar na correlação entre sintomas e pontos.

O *Appunture* foi desenvolvido com arquitetura “*offline-first*”, permitindo o acesso aos dados mesmo sem conexão à internet, com sincronização automática quando online. Para auxiliar no posicionamento dos pontos sobre as imagens anatômicas, foi desenvolvida uma ferramenta interna de mapeamento. Um painel administrativo web permite a atualização dos conteúdos do sistema.

É importante ressaltar que o *Appunture* não substitui a formação profissional adequada nem o acompanhamento de profissionais qualificados. A ferramenta busca ser um recurso complementar de apoio ao estudo e à consulta de informações sobre acupuntura, facilitando o acesso a informações organizadas e padronizadas.

1.1 OBJETIVOS

O principal objetivo deste projeto é desenvolver o *Appunture*, uma plataforma digital educativa voltada ao estudo da acupuntura. A ferramenta busca facilitar o acesso a informações sobre os pontos de acupuntura, oferecendo recursos visuais e de busca que possam auxiliar estudantes e profissionais em seu processo de aprendizagem e consulta.

1.1.1 Objetivos Específicos

Entre os objetivos específicos, destacam-se:

- **Visualização dos Pontos de Acupuntura:** Disponibilizar um atlas anatômico digital com 15 visualizações vetoriais (SVG), cobrindo vistas frontais e posteriores dos meridianos principais.
- **Assistente de Consulta:** Implementar um módulo baseado em IA para auxiliar na busca e correlação entre sintomas e pontos, utilizando os dados cadastrados no sistema.
- **Apoio ao Estudo:** Disponibilizar conteúdos organizados para auxiliar estudantes e profissionais no processo de aprendizagem.
- **Organização de Informações:** Disponibilizar informações sobre os 361 pontos clássicos e pontos extras, incluindo localização, indicações e funções.
- **Acesso Offline:** Permitir o acesso às informações mesmo sem conexão à internet.

- **Ferramenta de Mapeamento:** Desenvolver uma ferramenta auxiliar para posicionamento das coordenadas dos pontos sobre as imagens anatômicas.
- **Painel de Atualização:** Disponibilizar uma interface para atualização dos conteúdos do sistema.

Dessa forma, o *Appunture* propõe-se como um recurso digital complementar de apoio ao estudo da acupuntura, buscando facilitar o acesso a informações organizadas e contribuir para o processo de aprendizagem dos usuários interessados na área.

2 FUNDAMENTAÇÃO TEÓRICA

A construção de uma solução digital eficaz voltada ao ensino e ao estudo da acupuntura exige respaldo teórico multidisciplinar, abrangendo desde os fundamentos da acessibilidade à informação até os princípios de usabilidade e desenvolvimento de tecnologias educativas. Este capítulo busca contextualizar e embasar as escolhas metodológicas e funcionais do aplicativo *Appunture*, discutindo as bases conceituais que orientam sua proposta. Para isso, são explorados os aspectos relacionados à democratização do acesso à informação técnica em saúde, o papel das Tecnologias Digitais de Informação e Comunicação (TDICs) no apoio à formação e ao estudo, além da comparação com outras ferramentas disponíveis no mercado. A fundamentação aqui apresentada justifica a necessidade de um aplicativo como o *Appunture* e oferece suporte acadêmico às decisões adotadas no desenvolvimento do projeto.

2.1 ACESSIBILIDADE COMO DIREITO E FERRAMENTA DE IGUALDADE NA SAÚDE

A acessibilidade ao conhecimento e à prática segura em saúde deve ser compreendida como um direito fundamental, especialmente em contextos nos quais o acesso à formação e à informação de qualidade é desigual. No caso da acupuntura, esse desafio se intensifica pela diversidade de formações entre os praticantes — que podem ser médicos, fisioterapeutas, terapeutas integrativos ou técnicos em estética — e pela ausência de uma regulamentação federal unificada que determine diretrizes claras sobre a prática no Brasil. Essa lacuna acarreta diferentes formas de aprendizado, muitas vezes empíricas ou informalizadas, o que torna ainda mais necessária a disponibilização de conteúdos confiáveis e acessíveis.

Freire (1996) em sua abordagem pedagógica libertadora, enfatiza que o conhecimento só adquire sentido real quando é compartilhado, situado e contextualizado. Nesse sentido, a disseminação de informações em saúde — sobretudo em áreas complexas como a acupuntura — deve ir além do conteúdo técnico e considerar o contexto cultural, social e educacional de quem aprende. O acesso à informação, portanto, não pode depender apenas de condições econômicas, infraestrutura local ou formação prévia, pois isso aprofundaria as desigualdades já existentes entre os profissionais de saúde nas diversas regiões do país.

A inclusão digital torna-se uma aliada crucial nesse processo. Segundo Booth et al. (2021), tecnologias bem projetadas são capazes de reduzir barreiras estruturais no acesso à formação em saúde, desde que respeitem os diferentes níveis de letramento digital, necessidades pedagógicas e realidades locais dos usuários. Isso

significa que o design de soluções digitais deve ser centrado no usuário e incluir funcionalidades como interfaces intuitivas, conteúdos multimodais (texto, áudio, imagem, vídeo), acessibilidade para pessoas com deficiência e responsividade para dispositivos móveis.

Dados do *TIC Saúde 2023* revelam que mais de 65% dos profissionais de saúde no Brasil usam *smartphones* como ferramenta principal de acesso a informações clínicas (Comitê Gestor da Internet no Brasil, 2023). Isso reforça a urgência de produzir plataformas móveis eficazes, que funcionem bem mesmo em condições técnicas limitadas, como conexões lentas ou dispositivos com pouca capacidade de processamento.

Segundo Souza, Silva e Carvalho (2020), aplicativos de apoio ao estudo em saúde devem não apenas apresentar informações confiáveis, mas também oferecer recursos de interação e personalização, que favoreçam o aprendizado ativo e a construção de confiança do estudante em sua formação.

Dessa forma, um aplicativo como o proposto neste trabalho, que visa tornar o conhecimento da acupuntura mais acessível, confiável e intuitivo, cumpre um papel social importante: reduzir desigualdades no acesso à informação, apoiar a formação de estudantes e profissionais, e, conseqüentemente, contribuir para a qualificação da área.

2.2 TECNOLOGIAS DIGITAIS COMO APOIO À FORMAÇÃO E AO ESTUDO

A utilização de tecnologias digitais aplicadas à educação em saúde vem transformando não apenas os meios de acesso à informação, mas também a forma como os estudantes e profissionais constroem seu conhecimento. No campo da acupuntura, que exige domínio anatômico preciso e entendimento das interações energéticas do corpo humano, o uso de recursos visuais e interativos pode potencializar o aprendizado e a compreensão das técnicas.

Segundo Moraes e Cabral (2020), o uso de aplicativos, plataformas interativas e realidade aumentada na educação em saúde estimula a aprendizagem significativa, reduz o tempo de assimilação de conteúdos complexos e melhora a retenção do conhecimento. No caso da acupuntura, isso se torna ainda mais relevante ao considerar que muitos dos pontos utilizados não são visíveis a olho nu, exigindo visualização em profundidade, correlação com estruturas internas e treino constante.

Além disso, estudos como o de Oliveira, Santos e Lima (2022) destacam que alunos e profissionais da área de terapias integrativas relataram maior segurança em suas práticas após utilizar aplicativos educativos que oferecem representações tridimensionais e localização anatômica de pontos de acupuntura.

Portanto, investir no desenvolvimento de soluções digitais adaptadas à acupuntura não apenas moderniza o ensino, mas também amplia a compreensão do conteúdo. Tais recursos são especialmente relevantes para estudantes em formação e profissionais em início de carreira, que frequentemente enfrentam dificuldades em compreender corretamente os pontos e técnicas apresentados de forma teórica.

2.3 APLICATIVOS DE ACUPUNTURA: EXPERIÊNCIAS E RECURSOS TECNOLÓGICOS

O desenvolvimento de tecnologias na área da saúde deve considerar mais do que apenas conteúdo técnico. A forma como o usuário interage com a interface, compreende as informações e se orienta dentro do aplicativo é determinante para a sua eficácia. No contexto da acupuntura — que envolve a localização precisa de pontos no corpo, o entendimento de trajetos de meridianos e a aplicação de protocolos — a usabilidade se torna um pilar essencial.

De acordo com Norman (2013), um bom design deve “tornar possível o uso intuitivo”, ou seja, permitir que o usuário compreenda e utilize o sistema sem esforço cognitivo excessivo. Em aplicativos de saúde, isso significa que menus confusos, excesso de informações, termos técnicos não explicados ou falta de *feedback* visual comprometem a experiência do usuário e até mesmo colocam a segurança da prática em risco.

A Organização Mundial da Saúde (Organização Mundial da Saúde, 2020) recomenda que soluções tecnológicas em saúde digital adotem princípios de design centrado no usuário (DCU), considerando desde o início do desenvolvimento as reais necessidades, limitações e habilidades dos usuários. Isso é especialmente importante no caso de profissionais que atuam em contextos com restrições de tempo, estrutura ou formação técnica, como muitos terapeutas integrativos ou estudantes de cursos livres.

Além disso, Ferreira, Costa e Almeida (2021) ressaltam que aplicativos com interfaces acessíveis e organizadas promovem maior retenção do conhecimento e facilitam o aprendizado, especialmente em áreas que exigem visualização corporal anatômica como a fisioterapia e a acupuntura.

Por isso, o aplicativo proposto neste TCC prioriza:

- Interfaces limpas e intuitivas;
- Navegação fluida e segmentada por áreas do corpo;
- *Feedback* visual ao selecionar pontos de acupuntura;

- Compatibilidade com dispositivos móveis de diferentes capacidades;
- Inclusão de recursos de busca inteligente, filtros por sintomas e suporte ao idioma técnico com explicações simplificadas.

Esses elementos reforçam o compromisso com a acessibilidade e a qualidade da informação, pilares fundamentais quando se pensa em soluções tecnológicas que impactam diretamente a formação e o estudo na área da saúde.

2.4 COMPARATIVO COM APLICATIVOS EXISTENTES

Atualmente, existem diversos aplicativos disponíveis na Play Store que buscam auxiliar estudantes e profissionais da área da acupuntura por meio de recursos visuais e informativos. No entanto, ao analisar criticamente essas soluções, é possível perceber limitações significativas que comprometem sua eficácia como ferramentas de apoio à prática clínica e ao aprendizado, especialmente no contexto brasileiro.

O *Anatomy Learning*, por exemplo, é um aplicativo renomado para o estudo da anatomia em 3D. Sua qualidade gráfica e riqueza de detalhes são inegáveis, porém o aplicativo não possui foco específico em acupuntura, o que o torna pouco funcional para quem busca compreender os pontos e os meridianos energéticos. Além disso, seu conteúdo é voltado a usuários com formação técnica mais avançada, o que pode representar uma barreira para estudantes de cursos livres ou profissionais de áreas menos técnicas. Outro ponto problemático é seu desempenho em dispositivos de menor capacidade, com relatos de travamentos e lentidão.

Já o *Visual Acupuncture 3D* apresenta um escopo mais próximo ao esperado, oferecendo visualização de pontos e meridianos. No entanto, peca em aspectos de usabilidade. A interface é considerada pouco intuitiva por muitos usuários, com menus confusos, poluição visual e falta de clareza nos textos. Esses fatores dificultam seu uso durante a prática clínica, especialmente em momentos em que o profissional precisa localizar rapidamente um ponto ou revisar um protocolo. Além disso, a ausência de suporte ao idioma português reduz sua acessibilidade para profissionais brasileiros.

Outro aplicativo conhecido é o *Acupuncture 3D*, que também disponibiliza visualizações anatômicas tridimensionais focadas em acupuntura. Entretanto, o aplicativo carece de atualizações e apresenta conteúdos desatualizados em relação às nomenclaturas e práticas adotadas no Brasil. Sua interface é rígida e limitada, com dificuldade de navegação e ausência de funcionalidades complementares como busca por sintomas, indicação de protocolos ou explicações detalhadas para aplicação dos pontos.

Em contraste com essas soluções, o aplicativo proposto neste TCC se destaca por adotar uma abordagem centrada no usuário, com interface limpa, navegação fluida e foco no estudo e consulta rápida de informações sobre acupuntura. Ele foi pensado para funcionar bem mesmo em dispositivos mais simples, garantindo acessibilidade digital a um público mais amplo. Além disso, traz conteúdo em português, explicações didáticas e filtros de busca por sintomas, aproximando-se mais das reais necessidades dos estudantes e profissionais brasileiros da área.

Essa comparação evidencia que, embora os aplicativos analisados ofereçam contribuições pontuais, nenhum deles contempla simultaneamente os pilares de acessibilidade, usabilidade, confiabilidade e contexto nacional. O desenvolvimento do aplicativo proposto surge, assim, como uma resposta direta às lacunas existentes no mercado digital de apoio ao estudo da acupuntura, promovendo inclusão e aprimoramento educacional.

2.5 USABILIDADE E DESIGN CENTRADO NO USUÁRIO

Para que a tecnologia seja de fato acessível e eficaz, é essencial que siga princípios de usabilidade e design centrado no usuário. Nielsen (1994) define usabilidade como a capacidade de um sistema ser usado por seus usuários com eficiência, efetividade e satisfação. Esse conceito é especialmente importante em contextos de formação em saúde, onde a curva de aprendizado precisa ser suavizada.

Segundo Siqueira, Barbosa e Furtado (2021), aplicações móveis voltadas para profissionais da saúde que incorporam princípios de design centrado no usuário têm maiores taxas de adesão e melhor impacto educacional. A adaptação do conteúdo conforme o perfil do usuário, como proposto no *Appuncture*, está em consonância com o modelo de usabilidade responsiva, recomendado por padrões como a norma ISO/IEC 25010 (International Organization for Standardization, 2011).

2.6 NORMAS, DIRETRIZES E EVIDÊNCIAS CIENTÍFICAS

O desenvolvimento de soluções tecnológicas voltadas à saúde deve considerar diretrizes oficiais e evidências científicas. A OMS (Organização Mundial da Saúde, 1999) publicou um guia para treinamento e segurança em acupuntura, enfatizando a necessidade de padronização de pontos, técnicas de punção e biossegurança.

Na área de software, normas como a ISO 9241-210 (International Organization for Standardization, 2019) e a ISO/IEC 25010 (International Organization for Standardization, 2011) orientam o desenvolvimento de interfaces acessíveis, seguras e eficientes, especialmente em sistemas críticos como os de saúde. A adoção dessas

diretrizes no *Appunture* garante não apenas a funcionalidade, mas também a segurança e a confiabilidade da ferramenta.

3 MOTIVAÇÕES

O desenvolvimento do aplicativo *Appunture* surge a partir de uma série de desafios estruturais enfrentados pela prática da acupuntura no Brasil. Estes desafios envolvem a regulamentação da profissão, a formação heterogênea dos profissionais, a fragilidade nos processos de fiscalização, os riscos à biossegurança e a complexidade diagnóstica da Medicina Tradicional Chinesa (MTC). A seguir, são apresentados os principais fatores motivadores que embasam a proposta deste projeto.

3.1 REGULAMENTAÇÃO E DISPUTA DE CATEGORIAS PROFISSIONAIS

A regulamentação da acupuntura no Brasil é um dos maiores entraves para sua prática estruturada e segura. Apesar de sua origem milenar na MTC e da crescente popularização a partir das décadas finais do século XX, ainda não há consenso nacional sobre quais categorias profissionais estão habilitadas a aplicar acupuntura. Isso tem gerado disputas entre conselhos de diferentes áreas, como medicina, fisioterapia, enfermagem, psicologia, odontologia e farmácia.

Enquanto o Conselho Federal de Medicina (CFM) defende a acupuntura como especialidade médica, conselhos de outras áreas reconhecem a prática multiprofissional e regulamentam internamente sua atuação. Essa disputa tem gerado insegurança jurídica e limitações à atuação de profissionais devidamente capacitados, ao mesmo tempo em que abre espaço para práticas irregulares.

A proposta do Projeto de Lei 5983/2019, atualmente em tramitação no Senado Federal, busca criar critérios nacionais para o exercício da acupuntura, independentemente da formação de base, desde que haja capacitação específica. Enquanto isso, soluções tecnológicas devem ser sensíveis a esse cenário, respeitando os limites legais e adaptando os conteúdos conforme o perfil profissional do usuário.

3.2 FORMAÇÃO E FISCALIZAÇÃO INSUFICIENTES

A ausência de um conselho exclusivo e de uma legislação unificada gera uma lacuna na fiscalização da prática da acupuntura no Brasil. Muitos profissionais atuam com formações diversas, que vão desde cursos livres até especializações, sem uma padronização mínima nacional. Além disso, terapeutas que não estão vinculados a conselhos de classe podem atuar sem supervisão ou responsabilização adequada.

Nesse contexto, um aplicativo como o *Appunture* pode oferecer suporte educativo, contribuindo para o estudo de pessoas com diferentes formações. Por meio da

diferenciação de perfis de usuários e de um sistema de conteúdos escalonados conforme a formação declarada, o aplicativo propõe uma experiência organizada, ética e ajustada à realidade do usuário.

3.3 SEGURANÇA DO PACIENTE E BIOSSEGURANÇA

A acupuntura, por envolver a inserção de agulhas no corpo humano, é considerada uma prática invasiva e requer protocolos rigorosos de biossegurança. A utilização inadequada de instrumentos ou a realização do procedimento por pessoas sem conhecimento técnico podem gerar complicações sérias, como infecções, lesões internas, hematomas e até pneumotórax.

Casos documentados, como o de uma paciente hospitalizada após perfuração pulmonar em São Paulo, demonstram os riscos reais da má prática. Muitos desses casos ocorrem em ambientes informais ou com baixa qualificação técnica.

A tecnologia pode ser uma aliada na disseminação de informações de qualidade. O aplicativo *Appunture* fornece informações claras e baseadas em diretrizes da OMS, apresentando dados sobre profundidade de punção, indicações, contraindicações e características especiais dos pontos, contribuindo para uma formação mais completa.

3.4 DIAGNÓSTICO ENERGÉTICO: COMPLEXIDADE E SUBJETIVIDADE

O raciocínio clínico na acupuntura, segundo os princípios da MTC, exige uma abordagem subjetiva e holística, muitas vezes desafiadora para iniciantes. Diagnósticos são baseados em padrões energéticos, observações da língua e do pulso, e uma análise minuciosa do estado geral do paciente.

Diferentemente da medicina ocidental, não se trata apenas de associar um sintoma a uma causa direta, mas de identificar síndromes energéticas como estagnação de Qi, excesso de calor, umidade interna, entre outras. O mesmo sintoma, como dor abdominal, pode ter interpretações energéticas distintas, dependendo do conjunto de sinais apresentados.

Nesse sentido, o aplicativo *Appunture* atua como um apoio ao estudo e à compreensão dos padrões da MTC, fornecendo descrições de padrões, relações entre sintomas e síndromes e, inclusive, oferecendo um sistema de busca inteligente por sintomas com suporte de NLP (Processamento de Linguagem Natural), que interpreta a linguagem do usuário e apresenta os pontos relacionados ao caso apresentado.

Diante dos desafios regulatórios, formativos e conceituais que permeiam o estudo da acupuntura no Brasil, torna-se essencial o desenvolvimento de ferramentas

tecnológicas que aliem conhecimento técnico e acessibilidade. O *Appunture* surge como resposta a essas motivações, buscando oferecer um recurso digital estratégico para apoiar o estudo da acupuntura com qualidade e base científica.

4 TECNOLOGIAS E ARQUITETURA DO SISTEMA

Este capítulo apresenta o *stack* tecnológico utilizado no desenvolvimento do *Appunture*, detalhando a arquitetura do sistema, as tecnologias empregadas e as decisões técnicas que fundamentam a implementação da solução.

4.1 VISÃO GERAL DA ARQUITETURA

O *Appunture* foi desenvolvido seguindo uma arquitetura híbrida distribuída, composta por quatro camadas principais:

- **Aplicativo Móvel (React Native + Expo):** Interface principal do usuário com estratégia “*Offline-First*”;
- **API Backend (Java + Spring Boot):** Servidor de aplicação responsável pela lógica de negócio, sincronização e Inteligência Artificial;
- **Interface Web Administrativa (React):** Painel de administração para gestão de conteúdo;
- **Ferramentas de Apoio (Python/HTML):** Scripts de migração de dados e ferramenta de mapeamento de pontos.

Esta arquitetura garante alta disponibilidade, performance otimizada e experiência consistente mesmo em ambientes com conectividade limitada.

4.1.1 Diagrama de Arquitetura

O *Appunture* foi desenvolvido seguindo uma arquitetura híbrida e modular, que integra funcionalidades *offline* e *online* para garantir alta disponibilidade, desempenho e usabilidade mesmo em ambientes com conexão limitada ou instável.

4.2 ARQUITETURA HÍBRIDA (OFFLINE + ONLINE)

4.2.1 Funcionamento Offline

O aplicativo móvel foi projetado com estratégia “*offline-first*”, garantindo que as funcionalidades de navegação e consulta de dados funcionem independentemente da conectividade:

Banco de Dados Local (SQLite):

- Armazena todos os 361 pontos de acupuntura dos meridianos principais;
- Contém pontos extras, sintomas e suas relações terapêuticas;
- Mantém favoritos e anotações do usuário;
- Estrutura otimizada para consultas rápidas;
- Garantia de performance instantânea e independência de rede.

Nota: A busca inteligente por sintomas utilizando o assistente de IA requer conexão com a internet. Quando *offline*, o usuário pode navegar manualmente pelo atlas anatômico e acessar os detalhes dos pontos já sincronizados.

Sistema de Cache Inteligente (MMKV):

- Armazena configurações de usuário e estados da aplicação;
- *Cache* de imagens SVG e recursos visuais;
- Dados de sessão e preferências de interface;
- Performance até 30x superior ao AsyncStorage tradicional.

4.2.2 Sincronização Inteligente

Quando conectado à internet, o sistema implementa sincronização bidirecional automática:

Sincronização Descendente (Backend → Mobile):

- Atualização automática de dados do atlas;
- *Download* incremental baseado em versionamento;
- Compressão de dados para otimizar transferência;
- Validação de integridade dos dados recebidos.

Sincronização Ascendente (Mobile → Backend):

- *Upload* de favoritos e anotações do usuário;
- *Backup* automático de preferências;
- *Logs* de uso para análise de performance;
- Controle de conflitos com *timestamp*.

4.3 STACK TECNOLÓGICO DETALHADO

4.3.1 Frontend Mobile (React Native + Expo)

Segundo a documentação oficial (Meta Platforms, 2023), “React Native combina as melhores partes do desenvolvimento nativo com React, uma biblioteca JavaScript de ponta para construção de interfaces de usuário”.

React Native 0.72+ com Expo SDK 49+:

- *Framework* multiplataforma para desenvolvimento nativo;
- Expo Router para navegação baseada em arquivos;
- Renderização otimizada de SVGs para o atlas anatômico;
- Compatibilidade total com Android 8+ e iOS 12+.

Atlas Anatômico Digital:

- 15 visualizações vetoriais (SVG) de alta fidelidade;
- Organizadas por meridianos e vistas (frontal e posterior);
- Zoom sem perda de qualidade;
- Interação precisa (toque) em áreas pequenas;
- Carga sob demanda das imagens para otimizar memória.

4.3.2 Backend API (Java + Spring Boot)

Segundo Walls (2016), “Spring Boot muda a forma como desenvolvemos aplicações Spring, oferecendo configuração automática e eliminando a necessidade de configurações XML complexas”.

Java 17 LTS e Spring Boot 3.2+:

- Linguagem robusta, fortemente tipada e de alta performance;
- *Framework* que simplifica o desenvolvimento de aplicações Java;
- Arquitetura baseada em injeção de dependência e inversão de controle;
- Integração nativa com serviços Google Cloud.

Spring AI - Integração com Inteligência Artificial:

Segundo a documentação oficial (Spring, 2024), Spring AI é o *framework* oficial do ecossistema Spring para integração com modelos de linguagem de grande escala (LLMs).

- *Framework* oficial do Spring para integração com modelos de IA;
- Suporte nativo a múltiplos provedores de LLM;
- Abstração unificada para chamadas de API de IA;
- Gerenciamento de *prompts* e contexto.

Spring Security + Firebase Auth:

Segundo a documentação do Firebase (Google, 2024a), o Firebase Authentication fornece serviços de backend, SDKs fáceis de usar e bibliotecas de UI prontas para autenticar usuários em aplicativos.

- Integração com Firebase Authentication para gestão de identidades;
- Validação de *tokens* JWT no *backend*;
- Controle de acesso baseado em *roles* (RBAC);
- Suporte a autenticação biométrica no dispositivo móvel.

4.3.3 Banco de Dados

SQLite (Mobile - Local):

- Banco relacional embarcado de alta performance;
- Zero configuração e manutenção;
- ACID *compliance* para integridade dos dados;
- Tamanho reduzido ideal para mobile.

Google Cloud Firestore (Backend - Remoto):

- Banco de dados NoSQL flexível, escalável e de alta performance;
- Sincronização em tempo real e suporte *offline* nativo;
- Modelo de dados baseado em documentos e coleções;
- Integração perfeita com o ecossistema Firebase e Google Cloud.

4.4 INTELIGÊNCIA ARTIFICIAL E SUPORTE À CONSULTA

Diferente de abordagens tradicionais baseadas apenas em palavras-chave, o *Appunture* implementa um sistema de RAG (*Retrieval-Augmented Generation*) no *backend*, oferecendo suporte contextualizado para consulta de informações.

4.4.1 Arquitetura RAG (Retrieval-Augmented Generation)

A arquitetura RAG (*Retrieval-Augmented Generation*) foi introduzida por Lewis et al. (2020) como uma abordagem que combina a recuperação de informações com a geração de texto, melhorando significativamente a precisão e confiabilidade das respostas em tarefas de conhecimento intensivo.

Modelo de IA:

O Google Gemini 1.5 Flash (Google, 2024b) é um modelo de linguagem multimodal otimizado para respostas rápidas e contextualizadas.

- Google Gemini 1.5 Flash via Spring AI;
- Modelo otimizado para respostas rápidas e contextualizadas;
- Integração nativa com o ecossistema Google Cloud.

Funcionamento do Sistema:

1. O sistema intercepta a pergunta do usuário;
2. Busca contexto relevante na base de dados (pontos e sintomas relacionados);
3. Envia um *prompt* enriquecido para o modelo de IA;
4. Retorna respostas precisas e contextualizadas, baseadas nos dados validados do sistema.

Benefícios da Abordagem RAG:

- Respostas baseadas em dados validados do sistema;
- Redução significativa de alucinações do modelo;
- Maior confiabilidade das informações apresentadas;
- Contextualização específica para acupuntura;
- Rastreabilidade das fontes utilizadas nas respostas.

4.4.2 Integração com Spring AI

O QUADRO 4.1 apresenta um exemplo simplificado da integração com Spring AI para o assistente de acupuntura:

```
@Service
public class AcupunctureAssistantService {

    private final ChatClient chatClient;
    private final PointRepository pointRepository;

    public String getEducationalAssistance(String userQuestion) {
        // Busca contexto relevante na base de dados
        List<Point> relevantPoints =
            pointRepository.findBySymptoms(userQuestion);

        // Monta prompt enriquecido com contexto
        String enrichedPrompt = buildPromptWithContext(
            userQuestion, relevantPoints);

        // Chama o modelo de IA via Spring AI
        return chatClient.prompt()
            .user(enrichedPrompt)
            .call()
            .content();
    }
}
```

Fonte: : Os autores (2025)

Quadro 4.1: Exemplo de integração Spring AI

4.5 FERRAMENTAS DE DESENVOLVIMENTO E MAPEAMENTO

Para garantir a precisão da localização dos pontos de acupuntura sobre o atlas anatômico digital, foi desenvolvida uma ferramenta interna denominada “*Point Mapper*”.

4.5.1 Point Mapper - Ferramenta de Mapeamento

Características:

- Interface Web para visualização dos 15 SVGs do atlas;
- Sistema de coordenadas percentuais para garantir responsividade;
- Funcionalidades de zoom e pan para precisão no posicionamento;
- Exportação de dados em formato JSON para integração com o banco;
- Validação cruzada com descrições anatômicas dos pontos.

Sistema de Coordenadas:

- Utilização de coordenadas percentuais (0-100%) ao invés de pixels;
- Garantia de responsividade em diferentes tamanhos de tela;
- Precisão milimétrica na localização dos pontos;
- Compatibilidade com diferentes resoluções de dispositivos.

4.5.2 Scripts de Migração e Integração

Scripts Python:

- Migração de dados entre formatos (JSON, CSV, SQL);
- Validação de integridade dos dados mapeados;
- Geração automática de *seeds* para o banco de dados;
- Integração com a ferramenta de mapeamento.

4.6 TECNOLOGIAS DE SUPORTE

4.6.1 Gerenciamento de Estado (Mobile)

Zustand:

- Biblioteca de estado minimalista e performática;
- *TypeScript-first* com inferência automática;
- *Persist middleware* para sincronização com *storage* local.

React Query:

- *Cache* inteligente de requisições HTTP;
- Invalidação automática e *refetch* estratégico;
- *Offline support* com *background sync*.

4.6.2 Interface Web Administrativa (React)

React 18+ com TypeScript:

- Biblioteca para interfaces reativas e componentizadas;
- *Hooks* para gerenciamento de estado eficiente;

- *Component library* customizada baseada em Material-UI.

Funcionalidades Administrativas:

- CRUD completo de pontos de acupuntura;
- Gestão de sintomas e relações terapêuticas;
- Gestão da base de conhecimento que alimenta a IA;
- *Dashboard* com métricas de uso;
- Sistema de permissões granular.

5 METODOLOGIA

Este capítulo apresenta as metodologias e ferramentas utilizadas para o desenvolvimento do aplicativo *Appunture*, abrangendo desde as práticas de gerenciamento de projeto até as técnicas de modelagem e as tecnologias empregadas na implementação da solução.

5.1 METODOLOGIA DE DESENVOLVIMENTO

“A engenharia de software é uma disciplina tecnológica que se preocupa com todos os aspectos da produção de software” (PRESSMAN; MAXIM, 2016).

Seguindo os princípios da engenharia de software moderna, buscou-se conciliar qualidade técnica e valor prático, conforme destaca Pressman e Maxim (2016), ao afirmar que o software deve ser desenvolvido de forma disciplinada, mas adaptável às necessidades do projeto. A metodologia permitiu ajustes rápidos com base em testes reais, além de facilitar a manutenção e escalabilidade da aplicação.

5.1.1 Scrum

Para a gestão do desenvolvimento do aplicativo *Appunture*, optou-se pela utilização do Scrum, um *framework* ágil amplamente adotado em projetos de software devido à sua flexibilidade e capacidade de adaptação a contextos complexos.

Conforme Schwaber e Sutherland (2020), o Scrum é “um *framework* leve que ajuda pessoas, times e organizações a gerar valor por meio de soluções adaptativas para problemas complexos”. Ele permite organizar o trabalho de maneira iterativa e incremental, promovendo entregas frequentes e contínua melhoria do produto.

Durante a construção do *Appunture*, adotaram-se ciclos de desenvolvimento quinzenais (*Sprints*), nos quais o time planejava, desenvolvia e entregava partes funcionais da aplicação. Essa prática segue o conceito apresentado por Sutherland (2014), segundo o qual “a definição de objetivos sequenciais que devem ser concluídos em um período definido” favorece foco, previsibilidade e evolução do produto.

Cada ciclo se iniciava com uma *Sprint Planning*, onde era definido o *backlog* da *sprint* — uma lista priorizada de funcionalidades e correções a serem entregues. O time se organizava de forma autogerenciada, decidindo coletivamente “quem faria o quê, como e quando”, respeitando a autonomia e a responsabilidade individual de cada membro da equipe (SCHWABER; SUTHERLAND, 2020).

No contexto deste projeto foram definidos os seguintes papéis:

- **Product Owner:** Bruno Brugnerotto de Lara
- **Scrum Master:** Gabriel Francelino Voidaleski
- **Desenvolvedor:** Pedro Henrique Lopes

Os eventos de *Daily Scrum* foram adaptados para ocorrer semanalmente, considerando a disponibilidade da equipe envolvida no projeto. Já o evento de *Sprint Retrospective* foi mantido ao final de cada *sprint*, promovendo momentos de reflexão e melhoria contínua do processo de desenvolvimento.

5.2 MODELAGEM DO PROJETO

A modelagem é uma etapa fundamental no desenvolvimento de sistemas, pois proporciona uma representação visual da estrutura e do comportamento da aplicação. Segundo Booch, Rumbaugh e Jacobson (2005), empresas que entregam sistemas de qualidade e atendem às necessidades dos usuários são aquelas que compreendem a importância da modelagem desde os estágios iniciais do projeto.

No desenvolvimento do aplicativo *Appunture*, a modelagem permitiu visualizar de forma macro os elementos do sistema, facilitando a compreensão das funcionalidades e dos fluxos de dados. Como destacam os autores Booch, Rumbaugh e Jacobson (2005), a construção de diagramas é essencial em sistemas orientados a objetos, pois oferece clareza sobre a estrutura e o funcionamento do software. Para isso, foi utilizada a UML (*Unified Modeling Language*) como linguagem padrão, uma vez que, de acordo com os mesmos autores, trata-se de uma abordagem consolidada e amplamente adotada para representar graficamente os componentes e relações de um sistema.

5.2.1 Diagrama de Casos de Uso

Enquanto o diagrama de classes oferece uma perspectiva estrutural do sistema, representando seus componentes de forma estática, ele não contempla a dinâmica das interações entre os elementos. Segundo Pressman (2011), é por meio dos diagramas de casos de uso que se visualiza como os usuários interagem com as funcionalidades do sistema, sendo uma ferramenta fundamental na modelagem orientada a objetos.

No projeto *Appunture*, o diagrama de casos de uso foi elaborado com o objetivo de representar os diferentes perfis de usuários — como profissionais da saúde, administradores e desenvolvedores — e suas respectivas interações com o aplicativo.

5.2.2 Histórias de Usuário

A prática de expressar requisitos por meio de histórias surgiu no contexto do desenvolvimento ágil, mais especificamente no processo *Extreme Programming*, conforme introduzido por Beck (2001). As histórias de usuário consistem em descrições concisas de funcionalidades escritas sob a ótica do próprio usuário, facilitando a comunicação entre a equipe de desenvolvimento e os *stakeholders*.

Segundo Bernardo (2014), a principal característica dessas histórias está na simplicidade e objetividade, permitindo que necessidades do cliente sejam descritas de forma clara e centrada em valor. No desenvolvimento do aplicativo *Appunture*, as funcionalidades foram documentadas com base nessa abordagem, sendo cada história composta por:

- Uma descrição no formato: SENDO (perfil), QUERO (função) PARA (objetivo);
- Seus respectivos critérios de aceitação.

Conforme Cohn (2004), os critérios de aceitação “definem os limites de uma história de usuário, sendo usados para confirmar quando uma história está concluída e funcionando como esperado”.

5.2.2.1 Histórias de Usuário do Projeto

1. Como usuário, desejo consultar pontos de acupuntura interativamente em um corpo humano, para localizar facilmente a região desejada.
 - Critérios de aceitação: Deve ser possível tocar/clicar no ponto anatômico e acessar seus detalhes. Deve haver destaque visual nos pontos ativos do corpo.
2. Como usuário, desejo buscar pontos por nome, localização, meridiano ou função terapêutica, para encontrar informações de forma rápida.
 - Critérios de aceitação: Deve aceitar buscas por texto. A pesquisa deve filtrar e exibir resultados dinâmicos.
3. Como usuário, desejo filtrar pontos com base em sintomas que estou estudando (ex: “ansiedade” + “insônia”), para encontrar pontos relacionados.
 - Critérios de aceitação: Deve aceitar múltiplos sintomas. Deve retornar pontos relacionados por frequência de associação na literatura.
4. Como usuário, desejo visualizar informações detalhadas sobre cada ponto, incluindo profundidade, punção, características especiais e funções.

- Critérios de aceitação: Todas as informações devem estar organizadas em seções. Imagens e descrições devem estar disponíveis *offline*.
5. Como usuário, desejo usar o app sem internet, para ter acesso ao conteúdo mesmo em locais sem sinal.
 - Critérios de aceitação: O app deve funcionar com banco local. Ao conectar-se, os dados são sincronizados com o *backend*.
 6. Como usuário, desejo registrar uma conta e fazer login, para salvar minhas preferências e histórico.
 - Critérios de aceitação: Deve permitir login com email/senha. O app deve diferenciar usuários por tipo (ex: estudante, profissional etc.).
 7. Como usuário, desejo usar um assistente inteligente para descrever sintomas com linguagem natural, para receber informações contextualizadas sobre pontos relacionados.
 - Critérios de aceitação: Quando *online*, deve utilizar IA Generativa (Spring AI + Google Gemini) para informações contextualizadas. Respostas devem ser baseadas na base de dados técnica do sistema.
 8. Como administrador, desejo acessar o painel web, para visualizar e gerenciar os dados do aplicativo.
 - Critérios de aceitação: Login restrito ao perfil administrador. Interface adaptada para *desktop*.
 9. Como administrador, desejo cadastrar e editar pontos de acupuntura, para manter o banco de dados atualizado.
 - Critérios de aceitação: Deve permitir adicionar nome, descrição, localização, meridiano, imagem etc. Interface amigável com validação dos campos.
 10. Como administrador, desejo consultar relatórios de uso, para entender como os usuários estão utilizando a plataforma.
 - Critérios de aceitação: Relatórios por data, localização, pontos mais acessados. Visualização em gráfico ou tabela.

5.2.3 Diagrama Físico do Banco de Dados

O diagrama físico de banco de dados é uma representação detalhada das tabelas que compõem o sistema, incluindo seus atributos, tipos de dados, chaves primárias e estrangeiras. Conforme observado por Garcia-Molina, Ullman e Widom (2009),

esse tipo de modelagem visa representar a estrutura real de armazenamento da informação, sendo vital para a implementação correta de um banco de dados relacional.

Para o *Appunture*, o diagrama físico foi desenvolvido com base no diagrama de classes UML, refletindo tanto o banco de dados local (SQLite), utilizado para acesso *offline* no dispositivo móvel, quanto o banco remoto (Firestore), hospedado no *backend* acessado via API.

5.3 PRODUCT BACKLOG

O QUADRO 5.1 apresenta o *Product Backlog* do projeto *Appunture*, com os itens priorizados e suas respectivas estimativas em Pontos de Função (PF).

ID	Item do Backlog	Plataforma	PF	Prioridade
1	Setup do projeto e bibliotecas iniciais	Todas	3	Alta
2	Sistema de navegação (tabs, stack, rotas protegidas)	Mobile/Web	5	Alta
3	Tela de login e registro com autenticação JWT	Todas	5	Alta
4	Tela de busca com filtro por sintomas ou nome de ponto	Mobile/Web	8	Alta
5	Assistente IA com RAG (Spring AI + Gemini)	Mobile/Backend	6	Alta
6	Tela de resultados com lista de pontos filtrados	Mobile/Web	8	Alta
7	Tela do corpo humano com hotspots interativos (SVG)	Mobile	14	Alta
8	Tela de detalhes de um ponto	Mobile/Web	6	Média
9	Funcionalidade de favoritar pontos	Mobile	3	Média
10	Anotações personalizadas por ponto	Mobile	3	Média
11	Banco de dados local (SQLite/MMKV) + versionamento	Mobile	4	Alta
12	Sistema de sincronização (favoritos, anotações)	Mobile/Backend	6	Alta
13	Backend RESTful (Java Firestore + JWT)	Backend	5	Alta
14	CRUD completo de pontos e sintomas (admin)	Web/Backend	6	Alta
15	Upload de imagens SVG e definição de coordenadas	Web Admin	3	Média
16	Controle de acesso por perfil (user/admin)	Todas	3	Alta
17	Deploy e testes finais (Web Mobile)	Todas	5	Média

Fonte: : Os autores (2025)

Quadro 5.1: Product Backlog do Appunture

5.4 PLANEJAMENTO DE SPRINTS

O desenvolvimento do *Appunture* foi estruturado com base na metodologia SCRUM, dividindo o projeto em 8 *sprints* quinzenais, ao longo de aproximadamente 15 semanas de trabalho efetivo. Cada *sprint* contempla entregas incrementais de funcionalidades, priorizando valor de uso e estrutura progressiva do sistema.

A equipe de desenvolvimento é composta por três integrantes, com disponibilidade média de 4,5 horas semanais por pessoa, totalizando 13,5 horas por semana, ou 27 horas por *sprint*. Dessa forma, o tempo estimado total do projeto é de aproximadamente 202,5 horas.

5.4.1 Estimativa Geral por Pontos de Função (PF)

O QUADRO 5.2 apresenta a distribuição das estimativas por funcionalidade:

Funcionalidade	Estimativa (PF)
Tela de busca com filtro por sintomas/finalidade	8
Tela de resultados com imagens	8
Tela do corpo humano com hotspots interativos (SVG)	14
Tela de detalhes de um ponto	6
Funcionalidade de favoritos e anotações	6
Backend com autenticação e CRUD de pontos/sintomas	8
Painel administrativo para cadastro e upload de imagens SVG	4
Sistema de sincronização (SQLite ↔ Backend)	6
Sistema de navegação e layout geral	5
Deploy, testes finais e refinamentos	5
Integração do Assistente IA com RAG	6
Total Estimado	72 PF

Fonte: : Os autores (2025)

Quadro 5.2: Estimativa por Pontos de Função

5.4.2 Cálculo de Esforço

- **Tempo disponível por sprint:** 3 pessoas \times 4,5 h/semana \times 2 semanas = 27 horas/sprint
- **Tempo total estimado:** 15 semanas \times 13,5 h/semana = 202,5 horas
- **Total de Pontos de Função estimado:** 72 PF, distribuídos ao longo de 7 sprints + 1 sprint de buffer

O QUADRO 5.3 apresenta o cronograma de *sprints*:

Sprint	Semana	PF	Entregas Planejadas
Sprint 1	0–1	4	Setup do projeto, autenticação, rotas iniciais
Sprint 2	2–3	8	Login/registro, persistência de token, navegação protegida
Sprint 3	4–5	10	Busca avançada, integração IA, exibição de resultados
Sprint 4	6–7	10	Tela de resultados com imagens + início da tela de detalhes
Sprint 5	8–9	10	Tela de corpo humano com SVG interativo
Sprint 6	10–11	10	Favoritos, anotações, sincronização offline e online
Sprint 7	12–13	12	Backend completo + painel administrativo com upload de SVG
Sprint 8	14–15	0	Testes finais, deploy mobile/web

Fonte: : Os autores (2025)

Quadro 5.3: Cronograma de Sprints

5.5 FERRAMENTAS DE DESENVOLVIMENTO

Nesta seção estão descritas as diferentes ferramentas que, juntas, viabilizaram a construção e o desenvolvimento do aplicativo *Appunture*.

5.5.1 React Native

React Native é um *framework* de desenvolvimento móvel criado pelo Facebook em 2015, que permite criar aplicações nativas para iOS e Android usando JavaScript e React. Segundo a documentação oficial (Meta Platforms, 2023), “React Native combina as melhores partes do desenvolvimento nativo com React, uma biblioteca JavaScript de ponta para construção de interfaces de usuário”.

O *framework* foi escolhido para este projeto por permitir desenvolvimento multiplataforma com alta performance, reduzindo significativamente o tempo de desenvolvimento e manutenção.

5.5.2 Expo

Expo é uma plataforma de código aberto para criar aplicações React Native universais. Segundo a documentação (Expo, 2023), “Expo é um conjunto de ferramentas e serviços construídos em torno do React Native que ajuda você a desenvolver, construir, implementar e iterar rapidamente em aplicações iOS, Android e web”.

A escolha do Expo se deu pela facilidade de configuração do ambiente de desenvolvimento, capacidade de testar aplicações em dispositivos reais através do Expo Go, e pelos serviços integrados como EAS Build para compilação na nuvem.

5.5.3 Java

Java é uma linguagem de programação orientada a objetos, robusta e segura, amplamente utilizada no desenvolvimento de sistemas corporativos. Segundo Deitel e Deitel (2017), “Java é a linguagem de programação mais popular do mundo, permitindo o desenvolvimento de aplicações seguras, portáteis e de alto desempenho”.

Para o *backend* do *Appunture*, Java foi escolhido por sua tipagem estática, que previne erros em tempo de compilação, e por sua robustez no tratamento de regras de negócio complexas e integrações com serviços em nuvem.

5.5.4 Spring Boot

Spring Boot é um *framework* que facilita a criação de aplicações baseadas em Spring autônomas e de nível de produção. Segundo Walls (2016), “Spring Boot muda a forma como desenvolvemos aplicações Spring, oferecendo configuração automática e eliminando a necessidade de configurações XML complexas”.

O Spring Boot foi adotado no projeto por sua capacidade de acelerar o desenvolvimento, integração nativa com serviços do Google Cloud (como Firestore e Firebase), e por fornecer recursos prontos para segurança (Spring Security) e monitoramento.

5.5.5 Google Cloud Firestore

O Firestore é um banco de dados NoSQL flexível e escalável para desenvolvimento móvel, web e de servidor, oferecido pelo Google Cloud Platform. Segundo a documentação oficial (Google, 2023), “o Firestore mantém seus dados em sincronia entre aplicativos clientes através de ouvintes em tempo real e oferece suporte *offline* para dispositivos móveis e web”.

Para o *Appunture*, o Firestore foi escolhido por sua estrutura de documentos flexível, ideal para armazenar dados hierárquicos de pontos de acupuntura, e por sua integração nativa com o ecossistema Firebase.

5.5.6 SQLite

SQLite é um mecanismo de banco de dados SQL embutido, que não requer servidor separado ou configuração. Segundo Kreibich (2010), “SQLite é uma biblioteca de software que implementa um mecanismo de banco de dados SQL transacional, autônomo, sem servidor e de configuração zero”.

No aplicativo móvel *Appunture*, SQLite foi implementado para garantir funcionamento *offline* completo, armazenando localmente todos os pontos de acupuntura, sintomas e dados do usuário.

6 APRESENTAÇÃO DO SISTEMA

Neste capítulo o *Appunture* é apresentado, pontuando os aspectos técnicos e as funcionalidades que constituem a aplicação.

6.1 ARQUITETURA DO SISTEMA

O *Appunture* foi desenvolvido seguindo uma arquitetura híbrida e modular que integra funcionalidades *offline* e *online* para garantir alta disponibilidade, desempenho e usabilidade mesmo em ambientes com conexão limitada ou instável. A arquitetura se divide em dois módulos principais: aplicativo móvel (React Native/Expo) e *backend* (Java + Spring Boot).

6.1.1 Arquitetura Híbrida (Offline + Online)

No núcleo do aplicativo móvel, o *Appunture* utiliza um banco de dados local SQLite para armazenar informações do atlas, incluindo os 361 pontos dos meridianos principais, pontos extras, sintomas, relações terapêuticas, favoritos e anotações do usuário. Esta abordagem garante que o aplicativo funcione integralmente em modo *offline*, permitindo consultas rápidas e uso em qualquer lugar, sem depender da conexão com a internet.

Quando a conexão estiver disponível, o sistema realiza sincronizações automáticas e seguras com um *backend* remoto, desenvolvido em Java com Spring Boot, que hospeda o banco de dados Firestore centralizado. Esta sincronização atualiza dados do atlas, envia registros do usuário e mantém o histórico sempre preservado e atualizado.

6.1.2 Comunicação e Sincronização

A comunicação entre o app e o *backend* é realizada via API RESTful, utilizando autenticação Firebase/JWT para controle seguro de acesso. O *backend* disponibiliza *endpoints* públicos para consulta de dados do atlas e *endpoints* privados para operações que envolvem dados do usuário, como sincronização e autenticação.

A sincronização implementa um mecanismo de controle de versões dos dados do atlas para evitar *downloads* desnecessários e garantir que o aplicativo trabalhe sempre com as informações mais recentes. O processo ocorre automaticamente em *background* sempre que uma conexão de internet é detectada.

6.1.3 Assistente de IA para Busca Inteligente

Para viabilizar a busca inteligente por sintomas, o aplicativo utiliza um assistente de inteligência artificial integrado ao *backend*. Esta funcionalidade requer conexão com a internet para processamento.

Arquitetura RAG (Retrieval-Augmented Generation): O sistema utiliza a arquitetura RAG integrada ao *backend* Java. O assistente inteligente, alimentado pelo Google Gemini 1.5 Flash via Spring AI, interpreta perguntas do usuário sobre sintomas e pontos de acupuntura, fornecendo respostas contextualizadas com a base de dados técnica do sistema. Esta abordagem reduz alucinações do modelo e aumenta a confiabilidade das informações apresentadas.

Navegação Offline: Quando sem conexão, o usuário pode navegar manualmente pelo atlas anatômico e acessar as informações dos pontos já sincronizados no banco de dados local. A busca inteligente por sintomas fica indisponível até que uma conexão seja restabelecida.

6.1.4 Interface Interativa e Navegação

A interface do app é construída para promover uma experiência intuitiva e visualmente rica, com mapas anatômicos interativos baseados em SVG, que possibilitam a interação direta com os pontos de acupuntura. A navegação é estruturada para facilitar o acesso rápido às principais funcionalidades, como busca, visualização detalhada, favoritos e anotações, promovendo fluidez e usabilidade.

6.1.5 Backend e Web Admin

O *backend* centraliza a gestão dos dados oficiais, incluindo o cadastro e atualização dos pontos, sintomas e suas relações. A gestão do conteúdo é realizada através de APIs REST protegidas, com *endpoints* específicos para operações administrativas que requerem autenticação e permissões adequadas.

6.2 AUTENTICAÇÃO E SEGURANÇA

Para garantir a segurança dos usuários e a integridade dos dados, o sistema implementa autenticação via Firebase Authentication, integrada ao Spring Security no *backend*.

Métodos de Autenticação:

- Login com email e senha;
- Login social via Google (*Google Sign-In*);

- Validação de *tokens* JWT (Firebase ID Token) no *backend*.

Durante o processo de login, o Firebase Authentication gerencia a validação das credenciais e gera um *token* JWT assinado. Este *token* é utilizado para manter a sessão do usuário e validar o acesso aos recursos protegidos da API.

A sessão é gerenciada pelo aplicativo móvel, que armazena o *token* de forma segura no *storage* local do dispositivo. Desta forma, o usuário permanece logado mesmo após fechar e reabrir o aplicativo. A cada interação que requer comunicação com o *backend*, o *token* é enviado via cabeçalho HTTP para validação e controle de acesso baseado em *roles* (RBAC).

6.3 FUNCIONALIDADES DO SISTEMA

6.3.1 Tela Inicial e Onboarding

Quando o usuário inicia a utilização do aplicativo pela primeira vez, é apresentada uma sequência de telas de apresentação (*onboarding*) que explicam as principais funcionalidades e benefícios do *Appunture*. Esta experiência introdutória inclui informações sobre o uso *offline*, a busca inteligente e os mapas anatômicos interativos.

Na tela inicial, o usuário pode escolher entre realizar o processo de cadastro ou direcionar-se diretamente para a tela de login. O design foi pensado para transmitir profissionalismo e confiança, utilizando a identidade visual do projeto com cores que remetem à medicina tradicional chinesa e à modernidade tecnológica.

6.3.2 Cadastro e Autenticação via Firebase

O processo de autenticação do *Appunture* é inteiramente gerenciado pelo *Firebase Authentication*, serviço da Google que oferece segurança robusta e escalabilidade. A tela de cadastro foi projetada para ser simples e rápida, solicitando apenas as informações essenciais:

- **Nome completo:** Armazenado no *displayName* do perfil Firebase;
- **Email:** Identificador único do usuário no Firebase;
- **Senha:** Com validação de força mínima (6 caracteres);
- **Confirmação de senha:** Para evitar erros de digitação;
- **Aceite dos termos:** Concordância com os termos de uso e política de privacidade.

Ao submeter o formulário, o aplicativo utiliza a função `createUserWithEmailAndPassword` do Firebase SDK para criar a conta. Em seguida, o `displayName` é atualizado com o nome informado via `updateProfile`. Por fim, os dados do usuário são sincronizados com o *backend* Java através de uma chamada à API REST.

O sistema também oferece autenticação via *Google Sign-In*, implementada com o *expo-auth-session*. Neste fluxo, o usuário autoriza o acesso via OAuth 2.0, e o *token* retornado pelo Google é usado para criar uma credencial Firebase via `GoogleAuthProvider.credential`, completando a autenticação.

Após a autenticação (por qualquer método), o *Firebase ID Token* é obtido e armazenado de forma segura no dispositivo. Este *token* é enviado em todas as requisições ao *backend* para validação e autorização.

Todos os usuários têm acesso igualitário a todas as funcionalidades do atlas educativo, independentemente do método de cadastro utilizado.

6.3.3 Tela Principal e Navegação

A tela principal do aplicativo apresenta as principais funcionalidades de forma organizada e acessível. A navegação é estruturada por abas (*tabs*) na parte inferior da tela, permitindo acesso rápido às seções principais:

- **Início:** Tela inicial com acesso rápido às funcionalidades principais;
- **Buscar:** Sistema de busca por nome de ponto ou código;
- **Sintomas:** Lista de sintomas com pontos relacionados;
- **Meridianos:** Navegação por meridianos com mapas SVG;
- **Assistente:** Chat com IA para consultas contextualizadas;
- **Favoritos:** Lista de pontos favoritos pelo usuário;
- **Perfil:** Configurações do usuário e preferências.

6.3.4 Mapa Anatômico Interativo

Uma das funcionalidades mais destacadas do *Appunture* é o mapa anatômico interativo, implementado com tecnologia SVG (*Scalable Vector Graphics*). Este recurso permite que o usuário visualize o corpo humano de forma detalhada e interaja diretamente com os pontos de acupuntura mapeados.

Cada ponto é representado por um marcador visual que, ao ser tocado, exibe informações básicas sobre o ponto em um *tooltip*. O usuário pode então optar por acessar a tela de detalhes completa do ponto selecionado.

O mapa oferece 16 visualizações vetoriais de alta fidelidade, organizadas por meridianos:

- 12 meridianos principais: Pulmão (LU), Intestino Grosso (LI), Estômago (ST, ST2), Baço (SP), Coração (HT), Intestino Delgado (SI), Bexiga (BL), Rim (KI), Pericárdio (P), Triplo Aquecedor (TW), Vesícula Biliar (GB) e Fígado (LV);
- 2 vasos extraordinários: Vaso Governador (GV/Du Mai) e Vaso Conceção (CV/Ren Mai).

Cada mapa SVG é carregado sob demanda, permitindo navegação fluida entre as diferentes visualizações de frente e costas do corpo.

6.3.5 Detalhes dos Pontos de Acupuntura

A tela de detalhes apresenta informações completas sobre cada ponto de acupuntura, organizadas em seções para facilitar a consulta:

- **Identificação:** Nome em português, nome em chinês (pinyin), código internacional;
- **Localização:** Descrição anatômica precisa da localização do ponto;
- **Meridiano:** Identificação do meridiano ao qual o ponto pertence;
- **Profundidade:** Indicação da profundidade de punção recomendada;
- **Indicações:** Lista de condições clínicas para as quais o ponto é indicado;
- **Contraindicações:** Alertas sobre situações em que o ponto não deve ser utilizado;
- **Funções energéticas:** Descrição das funções segundo a MTC;
- **Características especiais:** Pontos de alarme, pontos de assentimento, etc.

6.3.6 Sistema de Busca Inteligente

O sistema de busca do *Appunture* utiliza inteligência artificial para oferecer resultados relevantes mesmo quando o usuário não conhece o nome exato do ponto que procura. A busca pode ser realizada por:

- Nome do ponto (em português ou pinyin);
- Localização anatômica;

- Sintomas ou condições clínicas;
- Meridiano;
- Funções terapêuticas.

O assistente de IA interpreta a intenção do usuário e retorna os pontos mais relevantes, mesmo com erros de digitação ou variações na escrita. Esta funcionalidade requer conexão com a internet.

6.3.7 Assistente Inteligente com IA Generativa

O assistente inteligente integrado ao *Appunture* permite que o usuário descreva sintomas em linguagem natural e receba informações educativas sobre pontos de acupuntura. A funcionalidade utiliza a arquitetura RAG (*Retrieval-Augmented Generation*):

Funcionamento do Sistema:

1. O sistema intercepta a pergunta do usuário;
2. Busca contexto relevante na base de dados (pontos e sintomas relacionados);
3. Envia um *prompt* enriquecido para o modelo Google Gemini 1.5 Flash via Spring AI;
4. Retorna respostas contextualizadas, baseadas nos dados validados do sistema.

Benefícios da Abordagem:

- Respostas contextualizadas com a base de dados técnica;
- Redução significativa de alucinações do modelo através da abordagem RAG;
- Maior confiabilidade das informações apresentadas;
- Rastreabilidade das fontes utilizadas nas respostas.

6.3.8 Sistema de Favoritos e Anotações Pessoais

O *Appunture* permite que o usuário salve pontos de acupuntura como favoritos para acesso rápido posterior. Além disso, é possível adicionar anotações pessoais a cada ponto, registrando observações de estudo, experiências ou lembretes.

Estas informações são armazenadas localmente no dispositivo e sincronizadas com o *backend* quando há conexão disponível, garantindo que o usuário não perca seus dados ao trocar de dispositivo.

6.4 GESTÃO DE DADOS

A gestão do conteúdo do *Appunture* é realizada através de APIs REST protegidas no *backend*, com *endpoints* específicos para operações administrativas (prefixo */admin/*). As principais funcionalidades disponíveis incluem:

- *Endpoints* CRUD para pontos de acupuntura;
- Gestão de sintomas e suas relações com pontos;
- *Upload* de imagens para o Firebase Storage;
- Seed de dados inicial via */admin/data/seed*;
- Controle de acesso baseado em *roles* (RBAC) com Firebase Custom Claims.

A documentação completa da API está disponível via Swagger/OpenAPI no *endpoint* */swagger-ui.html* do *backend*.

6.5 EXPERIÊNCIA DO USUÁRIO E USABILIDADE

O desenvolvimento da interface do *Appunture* seguiu princípios de design centrado no usuário, priorizando:

- **Simplicidade:** Interface limpa e intuitiva, sem excesso de informações;
- **Consistência:** Padrões visuais e de interação consistentes em todo o aplicativo;
- **Feedback:** Respostas visuais claras para todas as ações do usuário;
- **Acessibilidade:** Suporte a diferentes tamanhos de tela e orientações;
- **Performance:** Carregamento rápido e transições fluidas.

6.6 PROTÓTIPO DE TELAS

O fluxo de telas do aplicativo foi projetado para proporcionar uma navegação intuitiva entre as diferentes funcionalidades. O usuário pode transitar facilmente entre login, cadastro, tela inicial, busca, detalhes dos pontos, favoritos e anotações, seguindo um fluxo lógico e previsível.

7 CONSIDERAÇÕES FINAIS

O desenvolvimento do *Appunture* representou um desafio multidisciplinar que envolveu conhecimentos de engenharia de software, design de interfaces e compreensão básica do domínio da acupuntura. O resultado é uma ferramenta educativa que busca facilitar o acesso a informações sobre os pontos de acupuntura, servindo como recurso complementar de apoio ao estudo.

Os principais objetivos propostos no início do projeto foram alcançados:

- A implementação do aplicativo Android permite o acesso à ferramenta em dispositivos móveis;
- A arquitetura *offline-first* permite o uso do aplicativo mesmo em ambientes sem conectividade;
- O atlas anatômico digital com 16 visualizações vetoriais (SVG) auxilia na visualização dos pontos de acupuntura;
- O assistente baseado em IA auxilia na busca e correlação entre sintomas e pontos;
- A API REST administrativa permite a atualização dos conteúdos do sistema;
- A ferramenta de mapeamento auxiliou no posicionamento das coordenadas dos pontos sobre as imagens.

A adoção da metodologia Scrum permitiu um desenvolvimento iterativo e incremental, com entregas frequentes e ajustes baseados em *feedback* contínuo. A escolha das tecnologias — React Native com Expo para o aplicativo Android e Java com Spring Boot para o *backend* — mostrou-se adequada para os requisitos do projeto, oferecendo robustez, performance e facilidade de manutenção.

O *Appunture* surge em um contexto de crescimento das práticas integrativas e complementares no Brasil. A ferramenta busca contribuir como um recurso de apoio ao estudo, facilitando o acesso a informações organizadas sobre acupuntura. É importante ressaltar que o aplicativo não substitui a formação profissional adequada, mas pode servir como material complementar de consulta.

Como trabalhos futuros, sugerem-se:

- Implementação de recursos de realidade aumentada para visualização dos pontos em modelos 3D;

- Integração com sistemas de prontuário eletrônico;
- Desenvolvimento de módulos de avaliação e certificação;
- Expansão da base de conhecimento para incluir outras técnicas da MTC;
- Implementação de recursos de acessibilidade avançados;
- Tradução do conteúdo para outros idiomas;
- Aprimoramento contínuo do assistente IA com novos modelos e técnicas de RAG;
- Integração com *wearables* para monitoramento de tratamentos.

Em síntese, o *Appunture* representa uma contribuição para o ecossistema de ferramentas educativas voltadas à acupuntura. A plataforma busca facilitar o acesso a informações organizadas e servir como recurso complementar de apoio ao estudo. Espera-se que a ferramenta possa evoluir com base no feedback dos usuários, mantendo seu foco na acessibilidade e na qualidade da informação disponibilizada.

REFERÊNCIAS

- BECK, K. *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley, 2001.
- BERNARDO, K. *Histórias de Usuário: A Arte de Escrever Requisitos para Software*. São Paulo: Novatec, 2014.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: Guia do Usuário*. 2. ed. Rio de Janeiro: Elsevier, 2005.
- BOOTH, R. G. et al. How the nursing profession should adapt for a digital future. *BMJ*, v. 373, p. n1190, 2021.
- COHN, M. *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley, 2004.
- Comitê Gestor da Internet no Brasil. *TIC Saúde 2023: Pesquisa sobre o uso das tecnologias de informação e comunicação nos estabelecimentos de saúde brasileiros*. São Paulo, 2023.
- DEITEL, P.; DEITEL, H. *Java: Como Programar*. 10. ed. São Paulo: Pearson, 2017.
- Expo. *Expo Documentation*. [S.l.], 2023. Disponível em: <<https://docs.expo.dev/>>. Acesso em: 15 nov. 2025.
- FERREIRA, L. P.; COSTA, M. B.; ALMEIDA, J. P. Usabilidade em aplicativos de saúde: diretrizes para desenvolvimento. *Revista de Informática Aplicada*, v. 17, n. 2, p. 112–125, 2021.
- FREIRE, P. *Pedagogia da autonomia: saberes necessários à prática educativa*. São Paulo: Paz e Terra, 1996.
- GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, J. *Database Systems: The Complete Book*. 2. ed. [S.l.]: Pearson, 2009.
- Google. *Cloud Firestore Documentation*. [S.l.], 2023. Disponível em: <<https://firebase.google.com/docs/firestore>>. Acesso em: 15 nov. 2025.
- Google. *Firebase Authentication Documentation*. [S.l.], 2024. Disponível em: <<https://firebase.google.com/docs/auth>>. Acesso em: 15 nov. 2025.
- Google. *Gemini API Documentation*. [S.l.], 2024. Disponível em: <<https://ai.google.dev/gemini-api/docs>>. Acesso em: 15 nov. 2025.
- International Organization for Standardization. *ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*. 2011.
- International Organization for Standardization. *ISO 9241-210:2019 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems*. 2019.

- KREIBICH, J. A. *Using SQLite*. Sebastopol: O'Reilly Media, 2010.
- LEWIS, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, v. 33, p. 9459–9474, 2020.
- Meta Platforms. *React Native Documentation*. [S.l.], 2023. Disponível em: <https://reactnative.dev/docs/getting-started>. Acesso em: 15 nov. 2025.
- MORAES, E. H.; CABRAL, A. C. F. Tecnologias digitais na educação em saúde: uma revisão sistemática. *Revista Brasileira de Educação Médica*, v. 44, n. 2, 2020.
- NIELSEN, J. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1994.
- NORMAN, D. A. *The Design of Everyday Things: Revised and Expanded Edition*. New York: Basic Books, 2013.
- OLIVEIRA, A. S.; SANTOS, R. C. d.; LIMA, F. R. Aplicativos educativos para acupuntura: análise de usabilidade e eficácia. *Revista Brasileira de Práticas Integrativas e Complementares em Saúde*, v. 12, n. 1, p. 45–58, 2022.
- Organização Mundial da Saúde. *Guidelines on basic training and safety in acupuncture*. Geneva, 1999.
- Organização Mundial da Saúde. *WHO guideline: recommendations on digital interventions for health system strengthening*. Geneva, 2020.
- PRESSMAN, R. S. *Engenharia de Software: Uma Abordagem Profissional*. 7. ed. Porto Alegre: McGraw-Hill, 2011.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: Uma Abordagem Profissional*. 8. ed. Porto Alegre: McGraw-Hill Education, 2016.
- SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide*. [S.l.]: Scrum.org, 2020.
- SIQUEIRA, L. A.; BARBOSA, S. D. J.; FURTADO, E. Design centrado no usuário em aplicações móveis de saúde. *Revista de Design, Tecnologia e Sociedade*, v. 8, n. 1, p. 78–95, 2021.
- SOUZA, M. T. d.; SILVA, M. D. d.; CARVALHO, R. d. Aplicativos móveis para profissionais de saúde: uma revisão integrativa. *Revista Brasileira de Enfermagem*, v. 73, n. 5, 2020.
- Spring. *Spring AI Documentation*. [S.l.], 2024. Disponível em: <https://docs.spring.io/spring-ai/reference/>. Acesso em: 15 nov. 2025.
- SUTHERLAND, J. *Scrum: A arte de fazer o dobro do trabalho na metade do tempo*. São Paulo: LeYa, 2014.
- WALLS, C. *Spring Boot in Action*. Shelter Island: Manning Publications, 2016.

APÊNDICE A – DIAGRAMA DE CASOS DE USO

APÊNDICE B – HISTÓRIAS DE USUÁRIO

- HU01 – Buscar pontos por sintoma
- HU02 – Visualizar detalhes do ponto
- HU03 – Navegar pelo atlas anatômico
- HU04 – Favoritar ponto de acupuntura
- HU05 – Adicionar anotação pessoal
- HU06 – Sincronizar dados entre dispositivos
- HU07 – Consultar assistente de IA
- HU08 – Mapear ponto no atlas (administrador)

HU01 – BUSCAR PONTOS POR SINTOMA

SENDO um estudante ou profissional de acupuntura

QUERO buscar pontos de acupuntura por sintomas

PARA encontrar rapidamente os pontos indicados para tratamento

DESENHO DA(S) TELA(S):

CRITÉRIOS DE ACEITAÇÃO

1. Deve aceitar entrada de texto livre descrevendo sintomas
2. Deve utilizar IA para interpretar a intenção do usuário
3. Deve ordenar resultados por relevância
4. Deve requerer conexão com a internet
5. Deve exibir indicações de cada ponto nos resultados

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

Critério de contexto (Válido como premissa para todos os critérios):

Dado que desejo buscar pontos de acupuntura

E acessei a tela de busca do aplicativo

1. Deve aceitar entrada de texto livre descrevendo sintomas

Dado que desejo buscar por um sintoma

Quando digito “dor de cabeça” no campo de busca

Então o sistema realiza a busca e exibe os pontos relacionados

2. Deve utilizar IA para interpretar a intenção

Dado que digitei “dor de cabesa” com erro ortográfico

Quando seleciono buscar

Então o assistente de IA identifica a intenção

E retorna resultados relevantes baseados no contexto

3. Deve ordenar resultados por relevância

Dado que a busca retornou múltiplos pontos

Quando visualizo os resultados

Então os pontos mais relevantes aparecem primeiro

E cada resultado exibe um *score* de relevância

4. Deve requerer conexão com a internet

Dado que o dispositivo está sem conexão com a internet

Quando tento realizar uma busca inteligente

Então o sistema informa que é necessária conexão

E sugere navegação manual pelo atlas

5. Deve exibir indicações de cada ponto nos resultados

Dado que a busca retornou resultados

Quando visualizo a lista de pontos

Então cada ponto exibe seu código, nome e principais indicações

HU02 – VISUALIZAR DETALHES DO PONTO

SENDO um usuário do aplicativo

QUERO visualizar os detalhes completos de um ponto de acupuntura

PARA conhecer sua localização, indicações e técnicas de punção

DESENHO DA(S) TELA(S):

CRITÉRIOS DE ACEITAÇÃO

1. Deve exibir todas as informações do ponto (código, nomes, meridiano)
2. Deve exibir localização anatômica detalhada

3. Deve exibir indicações clínicas baseadas no padrão WHO
4. Deve exibir funções energéticas segundo a MTC
5. Deve mostrar a posição do ponto no atlas SVG
6. Deve permitir favoritar o ponto
7. Deve permitir adicionar anotações pessoais

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1. Deve exibir todas as informações do ponto

Dado que selecionei o ponto LU-7 na lista de resultados

Quando a tela de detalhes é carregada

Então o sistema exibe: código (LU-7), nome em português (Lieque), nome em pinyin (Liéque), meridiano (Pulmão)

2. Deve exibir localização anatômica detalhada

Dado que estou visualizando os detalhes do ponto

Quando verifico a seção de localização

Então o sistema exibe a descrição anatômica conforme padrão WHO

E indica a profundidade de punção recomendada

3. Deve mostrar a posição do ponto no atlas SVG

Dado que estou visualizando os detalhes do ponto

Quando verifico a seção do atlas

Então o sistema exibe a vista anatômica correspondente

E destaca a posição exata do ponto no corpo

HU03 – NAVEGAR PELO ATLAS ANATÔMICO

SENDO um usuário do aplicativo

QUERO navegar pelas diferentes vistas do atlas anatômico interativo

PARA localizar visualmente os pontos de acupuntura no corpo humano

DESENHO DA(S) TELA(S):

CRITÉRIOS DE ACEITAÇÃO

1. Deve permitir selecionar entre as 15 vistas disponíveis

2. Deve exibir os pontos mapeados como marcadores clicáveis
3. Deve permitir aplicar *zoom* e arrastar a imagem
4. Deve exibir detalhes ao tocar em um marcador
5. Deve funcionar em modo *offline* (navegação manual)

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1. Deve permitir selecionar entre as 15 vistas disponíveis

Dado que acessei a tela do atlas anatômico

Quando visualizo as opções de vistas

Então o sistema exibe: corpo anterior, corpo posterior, cabeça frontal, cabeça lateral, mão dorsal, mão palmar, pé dorsal, pé plantar, entre outras

2. Deve exibir os pontos mapeados como marcadores clicáveis

Dado que selecionei a vista “corpo anterior”

Quando a imagem SVG é carregada

Então o sistema exibe marcadores nos pontos que possuem coordenadas mapeadas

E cada marcador é interativo

3. Deve exibir detalhes ao tocar em um marcador

Dado que visualizo os marcadores no atlas

Quando toco em um marcador de ponto

Então o sistema exibe um *tooltip* com código e nome do ponto

E permite navegar para a tela de detalhes

HU04 – FAVORITAR PONTO DE ACUPUNTURA

SENDO um usuário autenticado

QUERO marcar pontos de acupuntura como favoritos

PARA acessá-los rapidamente em consultas futuras

CRITÉRIOS DE ACEITAÇÃO

1. Deve permitir favoritar/desfavoritar com um toque
2. Deve salvar favoritos no banco local imediatamente

3. Deve sincronizar favoritos quando houver conexão
4. Deve listar todos os favoritos em tela dedicada

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1. Deve permitir favoritar/desfavoritar com um toque

Dado que estou visualizando os detalhes de um ponto

Quando toco no ícone de coração/estrela

Então o sistema alterna o estado de favorito do ponto

E atualiza o ícone visualmente

2. Deve salvar favoritos no banco local imediatamente

Dado que favoritei um ponto

Quando fecho e reabro o aplicativo

Então o ponto permanece marcado como favorito

3. Deve sincronizar favoritos quando houver conexão

Dado que o dispositivo está conectado à internet

Quando adiciono ou removo um favorito

Então o sistema sincroniza a alteração com o *backend*

E mantém consistência entre dispositivos

HU05 – ADICIONAR ANOTAÇÃO PESSOAL

SENDO um usuário autenticado

QUERO adicionar anotações pessoais aos pontos de acupuntura

PARA registrar observações de estudo e experiências clínicas

CRITÉRIOS DE ACEITAÇÃO

1. Deve permitir criar anotação em texto livre
2. Deve associar anotação ao ponto específico
3. Deve permitir editar e excluir anotações
4. Deve exibir data de criação/modificação
5. Deve sincronizar anotações entre dispositivos

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1. Deve permitir criar anotação em texto livre

Dado que estou na tela de detalhes do ponto LU-5

Quando seleciono “adicionar anotação”

Então o sistema exibe campo de texto para digitação

E permite salvar a anotação

2. Deve permitir editar e excluir anotações

Dado que já tenho uma anotação salva no ponto

Quando toco na anotação existente

Então o sistema exibe opções de editar ou excluir

HU06 – SINCRONIZAR DADOS ENTRE DISPOSITIVOS

SENDO um usuário autenticado

QUERO que meus dados sejam sincronizados automaticamente

PARA não perder anotações e favoritos ao trocar de dispositivo

CRITÉRIOS DE ACEITAÇÃO

1. Deve sincronizar automaticamente quando houver conexão
2. Deve salvar dados localmente quando *offline*
3. Deve resolver conflitos mantendo versão mais recente
4. Deve notificar status da sincronização
5. Não deve impactar performance do aplicativo

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1. Deve sincronizar automaticamente quando houver conexão

Dado que o dispositivo está conectado à internet

Quando abro o aplicativo

Então o sistema verifica e sincroniza dados automaticamente

2. Deve salvar dados localmente quando offline

Dado que o dispositivo está sem conexão

Quando adiciono um favorito ou anotação

Então o sistema salva no banco SQLite local

E marca para sincronização posterior

3. Deve resolver conflitos mantendo versão mais recente

Dado que editei a mesma anotação em dois dispositivos

Quando ocorre a sincronização

Então o sistema mantém a versão com *timestamp* mais recente

HU07 – CONSULTAR ASSISTENTE DE IA

SENDO um usuário do aplicativo

QUERO consultar um assistente de IA sobre acupuntura

PARA obter orientações educacionais sobre pontos e tratamentos

CRITÉRIOS DE ACEITAÇÃO

1. Deve aceitar perguntas em linguagem natural
2. Deve fornecer respostas baseadas na base de dados do sistema (RAG)
3. Deve incluir *disclaimer* sobre uso educacional
4. Deve estar disponível apenas com conexão à internet
5. Deve referenciar os pontos mencionados nas respostas

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1. Deve aceitar perguntas em linguagem natural

Dado que acessei a tela do assistente de IA

Quando digito “quais pontos são indicados para ansiedade?”

Então o sistema processa a pergunta e gera uma resposta

2. Deve fornecer respostas baseadas na base de dados do sistema

Dado que fiz uma pergunta ao assistente

Quando a resposta é gerada

Então o sistema utiliza dados dos pontos cadastrados no sistema

E menciona pontos específicos com seus códigos

3. Deve incluir disclaimer sobre uso educacional

Dado que recebi uma resposta do assistente

Quando visualizo o conteúdo

Então a resposta inclui aviso de que é apenas para fins educacionais

E não substitui orientação profissional

HU08 – MAPEAR PONTO NO ATLAS (ADMINISTRADOR)

SENDO um administrador do sistema

QUERO mapear as coordenadas de um ponto no atlas SVG

PARA permitir a visualização correta do ponto pelos usuários

DESENHO DA(S) TELA(S):

CRITÉRIOS DE ACEITAÇÃO

1. Deve permitir selecionar um ponto da lista
2. Deve permitir selecionar a vista do atlas
3. Deve permitir clicar na imagem para definir coordenadas
4. Deve exibir prévia do marcador na posição selecionada
5. Deve salvar coordenadas no formato percentual
6. Deve exportar mapeamentos em formato JSON

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1. Deve permitir selecionar um ponto da lista

Dado que acessei a ferramenta Point Mapper

Quando seleciono o meridiano “LU” e o ponto “LU-1”

Então o sistema carrega as informações do ponto

2. Deve permitir clicar na imagem para definir coordenadas

Dado que selecionei um ponto e uma vista do atlas

Quando clico na posição anatômica correta na imagem

Então o sistema registra as coordenadas X e Y em percentual

3. Deve exportar mapeamentos em formato JSON

Dado que mapeei múltiplos pontos

Quando seleciono “exportar”

Então o sistema gera arquivo JSON com todas as coordenadas

E salva no formato compatível com o seed do sistema

APÊNDICE C – DIAGRAMA DE CLASSES

O sistema utiliza uma arquitetura baseada em Firestore (NoSQL) no *backend* e SQLite no aplicativo móvel para armazenamento local. As principais entidades são descritas a seguir.

CLASSES DO BACKEND (FIRESTORE)

- **FirestoreUser:** Representa os usuários do sistema.
 - id: String (Document ID do Firestore)
 - firebaseUid: String (UID do Firebase Authentication)
 - email: String
 - name: String
 - role: String (USER, ADMIN)
 - enabled: Boolean
 - profileImageUrl: String (URL da foto de perfil)
 - favoritePointIds: List<String> (IDs dos pontos favoritos)
 - createdAt, updatedAt: LocalDateTime
- **FirestorePoint:** Representa um ponto de acupuntura.
 - id: String (Document ID)
 - code: String (ex: VG20, E36)
 - name: String (nome do ponto)
 - description: String (descrição detalhada)
 - meridian: String (meridiano associado)
 - location: String (localização anatômica)
 - indication: String (indicações terapêuticas)
 - coordinates: Map<String, Double> (coordenadas x, y no mapa)
 - imageUrls: List<String> (URLs das imagens no Firebase Storage)
 - imageThumbnailMap: Map<String, String> (mapeamento original → thumb-nail)
 - symptomIds: List<String> (IDs dos sintomas relacionados)
 - tags: List<String> (tags para busca)

- category: String (categoria do ponto)
 - favoriteCount, viewCount: Integer (estatísticas de uso)
 - createdAt, updatedAt: LocalDateTime
 - createdBy: String (UID do criador)
- **FirestoreSymptom:** Representa sintomas que podem ser associados aos pontos.
 - id: String (Document ID)
 - name: String (nome do sintoma)
 - description: String (descrição detalhada)
 - category: String (ex: Dor, Digestivo, Emocional)
 - tags: List<String> (tags para classificação)
 - pointIds: List<String> (IDs dos pontos relacionados)
 - severity: Integer (1-10, gravidade)
 - useCount: Integer (estatística de uso em buscas)
 - createdAt, updatedAt: LocalDateTime

CLASSES DO MOBILE (SQLITE LOCAL)

- **LocalPoint:** Cache local dos pontos de acupuntura.
 - id: String
 - name, chinese_name, meridian, location: String
 - functions, indications, contraindications: String
 - code: String
 - favorite_count: Integer
 - synced: Boolean (status de sincronização)
 - last_sync: String (timestamp da última sincronização)
- **Favorite:** Relação entre usuário e pontos favoritos.
 - id: Integer (autoincrement)
 - point_id, user_id: String
 - synced: Boolean
 - operation: String (UPSERT, DELETE)

- created_at, updated_at: String
- **Note:** Anotações pessoais do usuário sobre pontos.
 - id: Integer (autoincrement)
 - point_id, user_id: String
 - content: String (texto da anotação)
 - synced: Boolean
 - created_at, updated_at: String
- **SyncOperation:** Fila de operações pendentes de sincronização.
 - id: String (identificador único da operação)
 - entity_type: String (favorite, note, point, symptom, image)
 - operation: String (UPSERT, DELETE)
 - data: String (JSON com os dados)
 - status: String (pending, in_progress, retry, completed, failed)
 - retry_count: Integer
 - created_at: String

APÊNDICE D – DIAGRAMAS DE SEQUÊNCIA

APÊNDICE E – MODELO LÓGICO DE DADOS

ESTRUTURA DO BANCO DE DADOS LOCAL (SQLITE)

O aplicativo móvel utiliza SQLite para armazenamento local dos dados, permitindo funcionamento *offline*. A estrutura foi projetada para sincronização bidirecional com o Firestore do *backend*.

Listing E.1 – Estrutura do banco SQLite local

```

1  -- Pontos de acupuntura (cache local)
2  CREATE TABLE points (
3      id TEXT PRIMARY KEY,
4      code TEXT,
5      name TEXT NOT NULL,
6      chinese_name TEXT,
7      meridian TEXT NOT NULL,
8      location TEXT NOT NULL,
9      functions TEXT,
10     indications TEXT,
11     contraindications TEXT,
12     image_path TEXT,
13     coordinates TEXT,
14     favorite_count INTEGER,
15     synced INTEGER DEFAULT 1,
16     last_sync DATETIME
17 );
18
19 -- Sintomas relacionados aos pontos
20 CREATE TABLE symptoms (
21     id TEXT PRIMARY KEY,
22     name TEXT NOT NULL,
23     synonyms TEXT,
24     category TEXT,
25     use_count INTEGER,
26     synced INTEGER DEFAULT 1,
27     last_sync DATETIME
28 );
29
30 -- Relacao N:N entre sintomas e pontos
31 CREATE TABLE symptom_points (
32     symptom_id TEXT NOT NULL,
33     point_id TEXT NOT NULL,
34     efficacy_score REAL DEFAULT 1.0,
35     PRIMARY KEY(symptom_id, point_id)
36 );

```

```

37
38 -- Favoritos do usuario
39 CREATE TABLE favorites (
40     id INTEGER PRIMARY KEY AUTOINCREMENT,
41     point_id TEXT NOT NULL,
42     user_id TEXT NOT NULL,
43     synced INTEGER DEFAULT 0,
44     operation TEXT NOT NULL DEFAULT 'UPSERT',
45     created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
46     updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
47     UNIQUE(point_id, user_id)
48 );
49
50 -- Anotacoes pessoais do usuario
51 CREATE TABLE notes (
52     id INTEGER PRIMARY KEY AUTOINCREMENT,
53     point_id TEXT NOT NULL,
54     user_id TEXT NOT NULL,
55     content TEXT NOT NULL,
56     synced INTEGER DEFAULT 0,
57     created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
58     updated_at DATETIME DEFAULT CURRENT_TIMESTAMP
59 );
60
61 -- Historico de buscas
62 CREATE TABLE search_history (
63     id INTEGER PRIMARY KEY AUTOINCREMENT,
64     query TEXT NOT NULL,
65     type TEXT NOT NULL,
66     created_at DATETIME DEFAULT CURRENT_TIMESTAMP
67 );
68
69 -- Status de sincronizacao por tabela
70 CREATE TABLE sync_status (
71     table_name TEXT PRIMARY KEY,
72     last_sync DATETIME NOT NULL,
73     status TEXT NOT NULL DEFAULT 'success'
74 );
75
76 -- Fila de sincronizacao de imagens
77 CREATE TABLE image_sync_queue (
78     id INTEGER PRIMARY KEY AUTOINCREMENT,
79     point_id TEXT NOT NULL,
80     image_uri TEXT NOT NULL,
81     payload TEXT,
82     status TEXT NOT NULL DEFAULT 'pending',
83     retry_count INTEGER NOT NULL DEFAULT 0,

```

```
84 |     last_attempt DATETIME ,  
85 |     created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
86 | );
```

APÊNDICE F – ESPECIFICAÇÃO DE APIS

F.1 AUTENTICAÇÃO

POST /api/auth/login

Descrição: Autentica um usuário no sistema.

Request Body:

```
1 {  
2   "email": "usuario@email.com",  
3   "password": "senha123"  
4 }
```

Response (200 OK):

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIs...",  
3   "user": {  
4     "id": 1,  
5     "name": "Usuario",  
6     "email": "usuario@email.com",  
7     "profileType": "professional"  
8   }  
9 }
```

POST /api/auth/register

Descrição: Registra um novo usuário no sistema.

Request Body:

```
1 {  
2   "name": "Novo Usuario",  
3   "email": "novo@email.com",  
4   "password": "senha123",  
5   "profileType": "student"  
6 }
```

F.2 PONTOS DE ACUPUNTURA

GET /api/points

Descrição: Retorna lista de pontos de acupuntura.

Query Parameters:

- meridian: Filtrar por código do meridiano (ex: LU, LI, ST)
- search: Termo de busca em nome ou indicações
- symptoms: Lista de sintomas separados por vírgula
- page: Número da página (paginação)
- limit: Quantidade por página (padrão: 20)

Response (200 OK):

```

1 {
2   "data": [
3     {
4       "id": 1,
5       "code": "LU-1",
6       "namePt": "Zhongfu",
7       "namePinyin": "Zhongfu",
8       "meridian": "LU",
9       "location": "Na regioa anterolateral...",
10      "indications": ["tosse", "asma", "dor toracica"]
11    }
12  ],
13  "pagination": {
14    "page": 1,
15    "limit": 20,
16    "total": 362
17  }
18 }
```

GET /api/points/{code}

Descrição: Retorna detalhes completos de um ponto específico.

Response (200 OK):

```

1 {
2   "id": 7,
3   "code": "LU-7 ",
4   "namePt": "Lique",
5   "namePinyin": "Lique",
6   "meridian": {
7     "code": "LU",
8     "namePt": "Pulmao",
9     "element": "Metal"
10  },
11  "location": "Na face anterolateral do antebraço...",
12  "depth": "0,3-0,5 cun obliquamente",

```

```

13     "indications": ["cefaleia", "rigidez cervical", "tosse"],
14     "contraindications": [],
15     "functions": "Dispersa Vento, libera o Exterior...",
16     "specialCharacteristics": "Ponto Luo, Ponto de Abertura...",
17     "coordinates": {
18         "x": 48.28,
19         "y": 25.97
20     },
21     "bodyMapCoords": [
22         {"view": "arm-anterior", "x": 48.28, "y": 25.97}
23     ]
24 }

```

F.3 SINCRONIZAÇÃO

POST /api/sync

Descrição: Sincroniza dados do dispositivo com o servidor.

Request Body:

```

1 {
2     "lastSyncTimestamp": "2025-11-15T10:30:00Z",
3     "favorites": [
4         {"pointId": 7, "action": "add", "timestamp": "..."},
5         {"pointId": 12, "action": "remove", "timestamp": "..."}
6     ],
7     "notes": [
8         {
9             "pointId": 7,
10            "content": "Observacao clinica...",
11            "action": "add",
12            "timestamp": "... "
13        }
14    ]
15 }

```

Response (200 OK):

```

1 {
2     "syncTimestamp": "2025-11-15T10:35:00Z",
3     "favorites": [...],
4     "notes": [...],
5     "conflicts": []
6 }

```

F.4 ASSISTENTE DE IA

POST /api/assistant/query

Descrição: Envia pergunta ao assistente de IA com RAG.

Request Body:

```
1 {  
2   "question": "Quais pontos sao indicados para ansiedade?"  
3 }
```

Response (200 OK):

```
1 {  
2   "answer": "Para auxiliar no tratamento da ansiedade...",  
3   "referencedPoints": ["HT-7", "PC-6", "SP-6", "GV-20"],  
4   "disclaimer": "Esta informacao e apenas para fins educacionais..."  
5 }
```

APÊNDICE G – ARQUITETURA RAG DO ASSISTENTE DE IA

CONFIGURAÇÃO DO SPRING AI COM GOOGLE GEMINI

Listing G.1 – Configuração do Spring AI

```

1 @Configuration
2 public class AIConfig {
3
4     @Value("${spring.ai.vertex.ai.gemini.project-id}")
5     private String projectId;
6
7     @Value("${spring.ai.vertex.ai.gemini.location}")
8     private String location;
9
10    @Bean
11    public ChatClient chatClient(VertexAiGeminiChatModel model) {
12        return ChatClient.builder(model)
13            .defaultSystem(ACUPUNCTURE_SYSTEM_PROMPT)
14            .build();
15    }
16 }

```

PIPELINE RAG (RETRIEVAL-AUGMENTED GENERATION)

1. **Recepção:** Usuário envia pergunta em linguagem natural
2. **Retrieval:** *Backend* busca pontos relevantes no banco de dados
3. **Augmentation:** Monta *prompt* enriquecido com contexto dos pontos
4. **Generation:** Google Gemini 1.5 Flash gera resposta contextualizada
5. **Pós-processamento:** Adiciona *disclaimer* educacional e referências

EXEMPLO DE FUNCIONAMENTO

Entrada do usuário: “dor de cabeça forte na testa”

Processamento RAG:

1. Recepção: Pergunta enviada ao *backend* via API REST
2. Retrieval: Busca pontos com indicações relacionadas a cefaleia

3. Augmentation: Contexto montado com GV-20, GB-14, ST-8, EX-HN3
4. Generation: Gemini gera explicação sobre os pontos e suas indicações
5. Resposta: “Para dor de cabeça frontal, os pontos mais indicados são...”

SYSTEM PROMPT DO ASSISTENTE

Listing G.2 – System prompt para contexto de acupuntura

```
1 private static final String ACUPUNCTURE_SYSTEM_PROMPT = ""  
2     Voce e um assistente educacional especializado em acupuntura.  
3     Seu objetivo e auxiliar estudantes e profissionais a compreender  
4     os pontos de acupuntura e suas aplicacoes clinicas.  
5  
6     IMPORTANTE: Suas respostas sao apenas para fins educacionais  
7     e NAO substituem a formacao profissional ou consulta medica.  
8  
9     Sempre mencione os codigos dos pontos (ex: LU-7, ST-36)  
10    e suas principais indicacoes quando relevante.  
11    "";
```