

IA III

Inteligência Artificial Aplicada à Saúde

Do Problema ao Aplicativo: criando software com IA

Odontologia e cursos da área da saúde

Sumário

1. De usuário a criador: a nova fronteira do vibe coding
2. Anatomia de um aplicativo (sem código)
3. As ferramentas: um mapa do ecossistema gratuito
4. Especificar bem: do problema clínico ao prompt
5. O ciclo de desenvolvimento: prototipar, testar, refinar, publicar
6. Projetos típicos da disciplina (e como elevá-los)
7. Dados de pacientes: LGPD, segurança e a linha vermelha
8. Limites do vibe coding: quando chamar um profissional
9. Depois da disciplina: usos acadêmicos e comerciais
10. Glossário

1. De usuário a criador: a nova fronteira do vibe coding

Nas disciplinas anteriores, a IA gerou páginas simples (IA I) e análises de dados (IA II). Agora damos o passo final: construir um **aplicativo completo** — com telas, navegação, armazenamento de informações e publicação na internet — sem escrever código. Dez anos atrás, isso exigiria uma equipe e meses de trabalho; hoje, um profissional de saúde com boas ideias e método consegue um protótipo funcional em horas.

O objetivo da disciplina não é formar programadores, e sim **resolvedores de problemas**: pessoas capazes de identificar uma dor real do cotidiano (da clínica, do estudo, da gestão), imaginar uma solução digital e conduzir a IA até um produto utilizável. Essa competência — especificar, testar e iterar — vale para qualquer ferramenta que exista hoje ou venha a existir.

Mentalidade da disciplina

A ferramenta muda todo semestre; o método permanece. Avalie cada plataforma nova com as perguntas: o que ela faz por mim? O que ela exige de mim? Quanto custa quando o projeto cresce?

2. Anatomia de um aplicativo (sem código)

Para conversar bem com a IA — e entender o que ela constrói — ajuda conhecer as três camadas de quase todo aplicativo:

- **Frontend (a face):** tudo que o usuário vê e toca — telas, botões, formulários, cores. É feito com HTML, CSS e JavaScript (frequentemente via bibliotecas como React);
- **Backend (os bastidores):** a lógica que roda nos servidores — regras de negócio, autenticação (login), permissões, processamento;
- **Banco de dados (a memória):** onde as informações ficam guardadas de forma permanente — cadastros, fichas, agendamentos.

Um site de portfólio pode viver só de frontend. Já um cadastro de pacientes exige as três camadas: sem backend não há login seguro; sem banco de dados, tudo se perde ao fechar a página. Saber disso evita frustrações clássicas do iniciante ('meu app esqueceu tudo!') e ajuda a escolher a ferramenta certa: plataformas como o Lovable já entregam as três camadas integradas.

Dois conceitos completam o vocabulário: **deploy** (publicar o aplicativo na internet, com um endereço próprio) e **responsividade** (a interface se adaptar a celulares — essencial, pois é onde seus usuários estarão).

3. As ferramentas: um mapa do ecossistema gratuito

O ecossistema muda rápido; esta é uma fotografia do momento, com foco em opções gratuitas ou com planos gratuitos generosos. Todas seguem a mesma lógica: você descreve, a IA constrói, você testa e pede ajustes.

Plataformas de app completo (frontend + backend + banco + deploy)

- **Lovable:** a preferida das turmas anteriores. Gera aplicativos web completos a partir de conversas, integra banco de dados (Supabase) e publica com um clique. Plano gratuito com limite de mensagens/créditos por período;
- **Bolt.new e Replit:** alternativas no mesmo espírito, com editor visível para quem quiser espiar o código;
- **Base44, Anything e similares:** mesma proposta conversacional; vale experimentar e comparar.

Assistentes de IA generalistas

- **Claude (Anthropic):** gera aplicativos funcionais dentro do próprio chat (artifacts), ótimos para protótipos de página única, calculadoras e quizzes;
- **ChatGPT (OpenAI) e Gemini (Google):** geram código sob demanda e ajudam a planejar, revisar e depurar; o Google AI Studio também monta protótipos de apps;
- **Codex / GitHub Copilot:** agentes de programação mais avançados, úteis para quem quiser ir além do escopo da disciplina.

Critérios de escolha

Para os projetos da disciplina: (1) tem plano gratuito suficiente? (2) publica o app com endereço compartilhável? (3) guarda dados de forma persistente, se o projeto exigir? (4) permite exportar o código, para não ficar refém da plataforma? Documente no seu trabalho qual ferramenta usou e por quê — isso também é resultado acadêmico.

4. Especificar bem: do problema clínico ao prompt

A qualidade do aplicativo nasce antes do primeiro prompt: nasce na clareza do problema. Um roteiro que funciona:

- **1. A dor:** que tarefa do cotidiano é lenta, repetitiva ou desorganizada? (ex.: 'anamneses em papel se perdem e são ilegíveis');
- **2. O usuário:** quem vai usar? (o próprio dentista? a recepção? o paciente?);
- **3. As funcionalidades essenciais:** liste de 3 a 6 ações que o app deve permitir — e corte o resto. Menos é mais no protótipo (conceito de MVP, o produto mínimo viável);
- **4. As telas:** descreva as páginas necessárias (ex.: lista de pacientes, formulário de anamnese, tela de visualização);
- **5. Aparência e tom:** cores, estilo (clínico/sóbrio? acolhedor?), idioma.

Exemplo de prompt inicial

Prompt-modelo (adapte!)

Crie um aplicativo web em português para uma clínica odontológica de ensino. Funcionalidades: (1) cadastrar paciente fictício com nome, telefone e data de nascimento; (2) preencher ficha de anamnese com perguntas de saúde geral e bucal, com campos sim/não e observações; (3) listar pacientes com busca por nome; (4) visualizar e editar a anamnese de cada paciente. Interface limpa, responsiva para celular, tons de azul. Use apenas dados fictícios de exemplo.

Depois do primeiro resultado, a regra é **uma mudança por vez**: pedidos acumulados confundem a IA e dificultam localizar o que quebrou. Teste, descreva o que viu ('ao salvar, a lista não atualiza'), peça a correção, teste de novo.

5. O ciclo de desenvolvimento: prototipar, testar, refinar, publicar

- **Prototipar:** gere a primeira versão com o prompt-roteiro. Não busque perfeição — busque algo clicável;
- **Testar como usuário:** use o app de verdade, no computador e no celular. Peça a um colega que o use sem instruções: onde ele trava? O que não entende?;
- **Refinar:** transforme cada problema observado em um pedido claro à IA. Mantenha uma lista do que falta (backlog) e priorize;
- **Publicar (deploy):** gere o endereço público e teste-o em outro dispositivo. Esse link é parte da entrega da disciplina;
- **Documentar:** registre em um texto curto o problema, o público, as funcionalidades, a ferramenta usada, as limitações e os próximos passos. É o embrião de um relatório técnico — e de um portfólio profissional.

Se a plataforma travar em um erro repetido, três saídas na ordem: (1) peça para a própria IA diagnosticar e corrigir; (2) desfaça para uma versão anterior e tente um caminho diferente; (3) simplifique a funcionalidade. Persistência com método faz parte da avaliação — e da vida com tecnologia.

6. Projetos típicos da disciplina (e como elevá-los)

Os projetos mais comuns das turmas — todos válidos — e sugestões para ir além do básico:

- **Currículo / portfólio profissional:** eleve com seção de casos clínicos (fotos autorizadas ou imagens ilustrativas), formulário de contato e versão em inglês;
- **Cadastro de pacientes:** eleve com busca, filtros e exportação da lista;
- **Anamnese digital:** eleve com alertas automáticos (ex.: destacar alergias e uso de anticoagulantes) e resumo imprimível;
- **Prontuário eletrônico simples:** eleve com odontograma clicável e histórico de procedimentos por data;
- **Ideias menos exploradas:** calculadora de risco de cárie, guia de conduta em traumatismo dental, agenda com lembretes, quiz de estudo para provas, orientações pós-operatórias personalizadas para o paciente, triagem de urgência odontológica.

Critério de excelência

Um bom projeto não é o que tem mais telas, e sim o que resolve com clareza um problema real e declarado. A pergunta da banca será: 'quem usaria isso, e por quê?'

7. Dados de pacientes: LGPD, segurança e a linha vermelha

Aqui está a regra mais importante da disciplina: **os protótipos usam apenas dados fictícios**. Nada de nomes, telefones, fotos ou históricos de pacientes reais — nem 'só para testar'. Motivos:

- A **LGPD** classifica dados de saúde como sensíveis e exige base legal, consentimento e medidas de segurança que um protótipo de graduação não oferece;
- Aplicativos gerados por IA frequentemente nascem com falhas de segurança (permissões abertas, dados expostos) que só uma revisão profissional detecta;
- Plataformas gratuitas podem armazenar dados em servidores fora do Brasil, sem os contratos exigidos para dados de saúde;
- O sigilo profissional (Código de Ética Odontológica) se aplica ao estudante e ao profissional independentemente da tecnologia.

Se um projeto evoluir para uso real com pacientes, o caminho responsável inclui: revisão de segurança por profissional de software, política de privacidade, termos de uso, hospedagem adequada, e verificação de exigências regulatórias — softwares com finalidade clínica podem ser enquadrados como dispositivos médicos pela Anvisa. Nada disso cabe na disciplina; tudo isso deve estar no radar de quem quiser transformar o protótipo em produto.

8. Limites do vibe coding: quando chamar um profissional

Reconhecer limites é sinal de maturidade técnica. O vibe coding atual é excelente para protótipos, ferramentas pessoais, materiais educativos e demonstrações de conceito. Ele se torna insuficiente — e arriscado — quando o software:

- Guarda dados reais de pessoas (privacidade e segurança);
- Apoia decisões clínicas de fato (responsabilidade profissional e regulação);
- Movimenta pagamentos ou integra-se a outros sistemas críticos;
- Precisa funcionar com garantias (backup, suporte, disponibilidade) para muitos usuários.

Além disso, aplicativos gerados por IA acumulam com facilidade o chamado 'débito técnico': código funcional, porém desorganizado, difícil de manter e evoluir. Para um protótipo, irrelevante; para um produto, um custo real. A parceria ideal do futuro é o profissional de saúde que sabe especificar e prototipar trabalhando com o profissional de software que sabe consolidar e proteger.

9. Depois da disciplina: usos acadêmicos e comerciais

No caminho acadêmico

O protótipo pode virar produto técnico de TCC (várias instituições aceitam 'desenvolvimento de tecnologia' como modalidade), resumo em congresso, ou base para pesquisa de usabilidade com questionários validados (ex.: escala SUS — System Usability Scale, coincidência divertida de sigla). Relatar o processo de desenvolvimento com IA é, em si, contribuição relevante na literatura de informática em saúde.

No caminho profissional

Ferramentas internas para o próprio consultório (orientações pós-operatórias, modelos de documentos, calculadoras clínicas), presença digital profissional e, para os mais ambiciosos, produtos para o mercado odontológico — sempre observando as exigências do capítulo 7. O maior ativo, porém, é a competência: profissionais que entendem o que a IA pode e não pode fazer serão os tradutores entre a clínica e a tecnologia — papel raro e valioso.

Fecho das três disciplinas

IA I mostrou o que é a inteligência artificial; IA II a colocou a serviço da pesquisa; IA III a colocou a serviço da criação. O que vem depois depende das perguntas que você fizer — às máquinas e a si mesmo.

10. Glossário

Backend	Parte do aplicativo que roda nos servidores: lógica, autenticação, permissões.
Backlog	Lista priorizada de melhorias e correções pendentes de um software.
Banco de dados	Componente que armazena informações de forma permanente e organizada.
Débito técnico	Custo futuro de manter código feito às pressas ou sem organização.
Deploy	Publicação do aplicativo na internet, tornando-o acessível por um endereço (URL).
Frontend	Parte visível do aplicativo: telas, botões e interações no navegador.
Login / autenticação	Mecanismo que identifica o usuário e controla o acesso às informações.
MVP	Minimum Viable Product — versão mínima do produto que já resolve o problema central.
No-code / low-code	Plataformas que permitem criar software sem escrever código (ou escrevendo muito pouco).
Prompt	Descrição em linguagem natural do que se deseja que a IA construa ou modifique.
Protótipo	Versão preliminar e funcional de um produto, feita para testar a ideia.
Responsividade	Capacidade da interface de se adaptar a diferentes telas, especialmente celulares.
SaaS	Software as a Service — software oferecido pela internet por assinatura, sem instalação.
Supabase	Serviço de backend e banco de dados usado por plataformas como o Lovable.
Usabilidade	Facilidade e eficiência com que usuários reais conseguem usar um sistema.