

Método de Desenvolvimento e Trabalho

Como transformo ideias em produtos digitais robustos, seguros e confiáveis com IA, engenharia, arquitetura, segurança e auditoria.

Decleones Andrade

Portfólio Profissional | Versão 2026

IA acelera. Engenharia organiza. Arquitetura sustenta. Segurança protege. Auditoria garante confiança.

DOCUMENTO OFICIAL PARA PÁGINA MÉTODO, CURRÍCULO E PORTFÓLIO

Método de Desenvolvimento e Trabalho - Decleones Andrade

Este documento apresenta, de forma profissional e objetiva, o método utilizado por Decleones Andrade para planejar, desenvolver, testar, auditar e evoluir produtos digitais. Ele foi criado para ser usado no app de Portfólio Profissional, junto ao vídeo de apresentação do método, currículo e materiais de divulgação.

Posicionamento profissional

Sou desenvolvedor em formação, estudante de Agrocomputação e criador de soluções digitais voltadas para agro, gestão, finanças, igrejas, saúde, produtividade e pequenos negócios. Meu trabalho combina visão de produto, engenharia de software, organização por fases, segurança, documentação, testes e uso estratégico de inteligência artificial.

Princípio central

A IA é usada como copiloto técnico para acelerar raciocínio, análise, documentação e código. A qualidade vem do método, da validação humana, da arquitetura, dos testes e da auditoria.

Objetivo do método

- Transformar ideias em sistemas reais, com clareza de problema, público, dados e resultado esperado.
- Reduzir retrabalho, dividindo o desenvolvimento em fases testáveis e auditáveis.
- Garantir confiabilidade, considerando segurança, permissões, regras de negócio e validação de dados.
- Construir portfólio técnico sólido, mostrando não apenas telas, mas processo, arquitetura, disciplina e entrega.

Resumo executivo

Diagnóstico	Entender problema, usuário, dor, escopo e prioridades.
Arquitetura	Definir estrutura técnica, dados, integrações, segurança e escalabilidade.
Desenvolvimento	Implementar por fases, com controle de arquivos, branches e validação contínua.
Auditoria	Testar, revisar, documentar, corrigir e evitar regressões antes de avançar.

Os 6 pilares do método

O método é sustentado por seis pilares que guiam o projeto desde a primeira ideia até a publicação e evolução do produto.

1. Diagnóstico e escopo

Antes de codar, entendo a dor, o público-alvo, os dados, as regras e o resultado que o sistema precisa entregar.

2. Arquitetura e modelagem

Defino frontend, backend, banco de dados, autenticação, permissões, entidades, relações e padrões de organização.

3. Desenvolvimento por fases

A implementação é feita em etapas menores, com metas claras, testes locais e validação antes de avançar.

4. Segurança e confiabilidade

Considero autenticação, controle de acesso, validação, privacidade, LGPD quando aplicável e tratamento de erros.

5. Auditoria e testes

Reviso comportamento, build, erros, fluxos críticos e risco de quebrar funcionalidades já prontas.

6. Documentação e evolução

Registro decisões, arquivos alterados, próximos passos, comandos, testes realizados e pontos pendentes.

Esses pilares tornam o desenvolvimento mais previsível, profissional e seguro. A proposta não é simplesmente gerar código, mas construir um produto com fundamento técnico e utilidade real.

Ciclo prático de trabalho

O fluxo abaixo representa como uma ideia passa a ser organizada, transformada em sistema e preparada para uso real.

1	Entendimento do problema	Identificar a dor, o usuário, o contexto de uso e o valor que o app precisa entregar.
2	Escopo e prioridades	Separar essencial, desejável e futuro. Evitar excesso de funcionalidades no início.
3	Mapa de telas e fluxos	Desenhar jornadas de usuário, cadastros, dashboards, relatórios e ações principais.
4	Modelagem de dados	Definir entidades, relacionamentos, campos, validações e regras de negócio.
5	Arquitetura técnica	Escolher stack, pastas, serviços, integrações, autenticação, permissões e deploy.
6	Implementação por fase	Criar branches, alterar arquivos com controle e testar cada etapa antes de avançar.
7	Testes reais	Validar login, CRUDs, filtros, relatórios, responsividade, erros e fluxos críticos.
8	Auditoria e revisão	Revisar o que foi alterado, corrigir regressões e confirmar estabilidade do build.
9	Documentação	Registrar decisões, comandos, arquivos, banco de dados, regras e próximos passos.
10	Publicação e evolução	Preparar deploy, apresentação, portfólio, melhoria visual e novas funcionalidades.

Uso estratégico de inteligência artificial

A inteligência artificial faz parte do processo, mas não substitui a responsabilidade técnica. Ela acelera análises, alternativas e documentação, enquanto o controle do produto permanece no método humano de validação.

Como a IA entra no processo

A IA funciona como copiloto técnico: ajuda a pensar, revisar, comparar caminhos e gerar versões iniciais. O que transforma isso em desenvolvimento profissional é a checagem, a adaptação ao contexto, os testes reais e a auditoria.

Análise e planejamento	Transformar ideias soltas em escopo, fases, módulos e prioridades.
Arquitetura e alternativas	Comparar stacks, padrões, fluxos, banco de dados e possibilidades de integração.
Geração assistida de código	Criar trechos e arquivos base, sempre revisados e testados no contexto real.
Correção de erros	Interpretar logs, mensagens de build, falhas de tipagem, SQL, API ou autenticação.
Documentação técnica	Gerar READMEs, registros de fase, instruções de instalação e explicação de funcionalidades.
Auditoria de qualidade	Revisar riscos, inconsistências, retrabalho, quebras de fluxo e pontos de segurança.

Regra de ouro

Nenhuma sugestão da IA deve ser tratada como verdade final. Todo código, decisão de banco, regra de segurança ou fluxo de usuário precisa ser revisado, testado e validado.

Engenharia, arquitetura e segurança

Projetos robustos precisam de estrutura. Por isso, o método considera o sistema como um conjunto integrado: interface, dados, backend, autenticação, regras de negócio, segurança, performance e experiência do usuário.

Produto	Problema, público, proposta de valor, telas essenciais, jornada do usuário e evolução futura.
Frontend	Interface moderna, responsiva, clara, com foco em smartphone, PWA e experiência real do usuário.
Backend/API	Regras de negócio, endpoints, serviços, integração com banco, validações e tratamento de erros.
Banco de dados	Entidades, relacionamentos, constraints, permissões, consultas, views e rastreabilidade de dados.
Segurança	Autenticação, autorização, separação por usuário/cliente/igreja/empresa, validações e proteção de dados.
Deploy e operação	Ambiente, variáveis, build, publicação, logs, documentação e possibilidade de manutenção.

Checklist de segurança aplicado ao raciocínio

- Quem pode acessar esta informação?
- O dado pertence a qual usuário, cliente, empresa, igreja ou propriedade?
- Existe validação antes de salvar, editar ou excluir?
- O sistema protege rotas, telas e endpoints críticos?
- O banco impede acesso indevido por política, filtro ou regra?
- Há tratamento de erro claro para o usuário e para o desenvolvedor?

Auditoria, testes e confiabilidade

A etapa de auditoria é o que transforma desenvolvimento rápido em desenvolvimento confiável. Ela reduz risco, evita retrabalho e protege funcionalidades que já estão funcionando.

Revisão de arquivos

Antes de alterar, conferir a versão atual e evitar sobrescrever código funcional sem necessidade.

Teste local real

Rodar o app, validar fluxos críticos, abrir telas, cadastrar dados, editar, excluir e simular uso real.

Build e tipagem

Executar build, verificar TypeScript, dependências, imports, rotas, variáveis de ambiente e warnings relevantes.

Banco de dados

Testar inserts, updates, selects, constraints, RLS/policies, chaves estrangeiras e consistência de views.

Responsividade

Priorizar experiência em smartphone, telas limpas, textos essenciais e navegação simples.

Registro de fase

Documentar o que foi feito, arquivos alterados, testes realizados, problemas e próximos passos.

Critério para avançar de fase

Uma fase só é considerada concluída quando o recurso foi implementado, testado, validado visualmente, documentado e não quebrou o que já funcionava. Depois disso, o ideal é salvar a fase com commit e abrir a próxima etapa com controle.

Aplicações práticas do método

O método é usado para construir produtos digitais em diferentes áreas, mantendo uma base comum: diagnóstico, arquitetura, desenvolvimento por fases, segurança, testes e documentação.

Agro e campo	EsplAgro, AgroScore360, Agro Simples, BovMonitor, ClasseG e soluções de monitoramento, diagnóstico, validação rural e gestão produtiva.
Gestão e negócios	LeanMaturity, CalhasAndradePró, AndradeFinance, GrãoValor e plataformas para diagnóstico, controle financeiro, serviços e tomada de decisão.
Igrejas e comunidade	Koinonia, ComuMembro, DireitosEDeveres e sistemas voltados a membros, tesouraria, eventos, discipulado, estatutos e comunicação.
Saúde e cuidado	HistóricoDeAmor e soluções para acompanhamento contínuo, rotinas, histórico, consultas, lembretes e relatórios.
Aprendizado e propósito	Jornada Bíblica, FéEmCrescimento e apps de leitura, constância, acompanhamento espiritual e estudo organizado.

Tecnologias presentes no ecossistema de projetos

- Frontend: React, Next.js, React Native, Expo, Tailwind CSS, PWA.
- Backend: Django REST Framework, FastAPI, NestJS e APIs REST.
- Banco de dados: PostgreSQL, Supabase, Firebase, PostGIS quando necessário.
- Produto: dashboards, relatórios, gráficos, PDF, autenticação, responsividade e deploy.
- Processo: Git, branches por fase, testes locais, documentação e revisão contínua.

Resultado esperado

O resultado é um portfólio formado por soluções reais, com visão de produto e base técnica evolutiva. Cada app representa uma oportunidade de aplicar tecnologia para resolver problemas concretos com organização, clareza e confiabilidade.

Texto pronto para inserir no portfólio

Os blocos abaixo podem ser usados diretamente na página Método do app Portfólio Profissional - Decleones Andrade.

Título da seção	Meu método de desenvolvimento
Subtítulo	Como transformo ideias em produtos digitais robustos, seguros e confiáveis.
Texto principal	Trabalho com um processo estruturado que combina diagnóstico do problema, arquitetura de sistemas, engenharia de software, segurança, testes, documentação e auditoria. Utilizo inteligência artificial como copiloto técnico para acelerar análise, prototipação, revisão e documentação, mantendo validação humana e controle técnico sobre cada etapa do processo.
Frase curta	IA acelera. Engenharia organiza. Arquitetura sustenta. Segurança protege. Auditoria garante confiança.
Chamada para ação	Conheça meus projetos, veja meu currículo e acompanhe como aplico esse método na construção de soluções digitais reais.

Versão curta para currículo

Método próprio de desenvolvimento assistido por IA	Processo estruturado com foco em diagnóstico, arquitetura, engenharia, segurança, testes, documentação e auditoria para criação de produtos digitais robustos e confiáveis.
---	---

Versão para LinkedIn

Desenvolvimento com IA, mas com método	Uso inteligência artificial como ferramenta de produtividade para acelerar raciocínio, prototipação e documentação, mas mantenho validação humana, testes reais, revisão técnica e auditoria em cada etapa do desenvolvimento.
---	--

Checklist de qualidade antes de publicar

Este checklist resume o padrão de trabalho usado para considerar uma entrega pronta para demonstração, portfólio, apresentação acadêmica ou uso inicial.

1	Escopo claro	O objetivo da fase está definido e o recurso resolve uma necessidade real.
2	Arquitetura coerente	O código segue uma estrutura compreensível e evolutiva.
3	Dados consistentes	Tabelas, campos, relacionamentos e regras estão alinhados com o fluxo do produto.
4	Segurança considerada	Acesso, autenticação, permissões, validações e privacidade foram avaliados.
5	Interface utilizável	A tela é clara, responsiva, amigável e não está poluída com textos técnicos desnecessários.
6	Teste real executado	O recurso foi aberto, usado e validado com dados reais ou simulação próxima do uso real.
7	Build validado	Quando aplicável, a aplicação compila sem erros críticos.
8	Documentação atualizada	O que foi feito, alterado e testado está registrado.
9	Sem regressão conhecida	O que já funcionava não foi quebrado pela nova alteração.
10	Próximo passo definido	A próxima fase está clara antes de continuar o desenvolvimento.

Mensagem final

Meu método une aprendizado contínuo, IA responsável, raciocínio técnico, organização por fases, engenharia, arquitetura, segurança e auditoria. O objetivo é criar sistemas mais confiáveis, úteis e preparados para uso real.