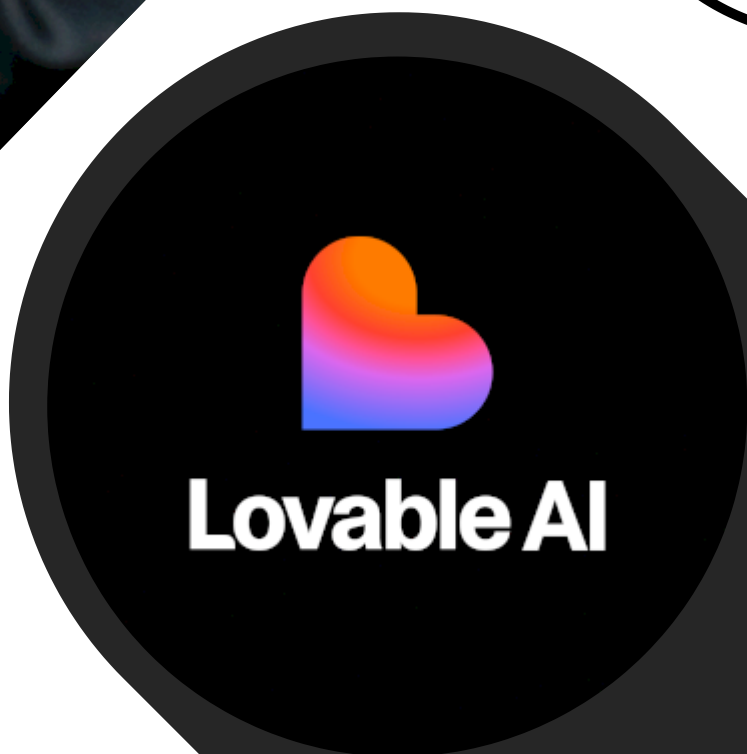
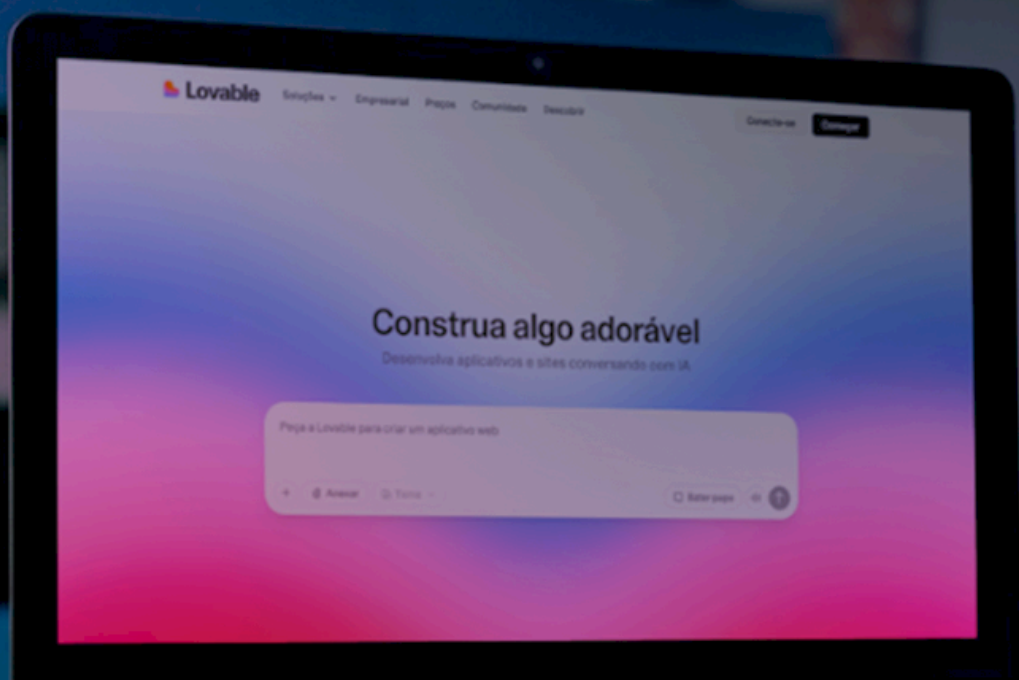


Apostila do Curso

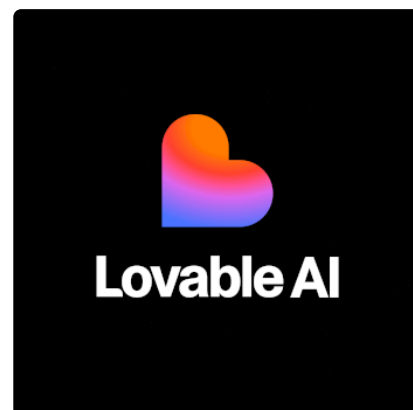
Conteúdo e Atividades



Lovable AI



Nome:



Sobre o curso

Este curso é um guia prático para transformar ideias em software real usando o poder da Lovable AI. Você aprenderá a atuar como um "Arquiteto de Soluções", dominando a Engenharia de Prompt para criar aplicações Full-Stack (Frontend e Backend) sem digitar código manualmente. Do design visual à publicação, você construirá projetos funcionais com banco de dados, segurança e pagamentos integrados.



Quantidade de Aulas
7 aulas



Carga horária
10.5 horas

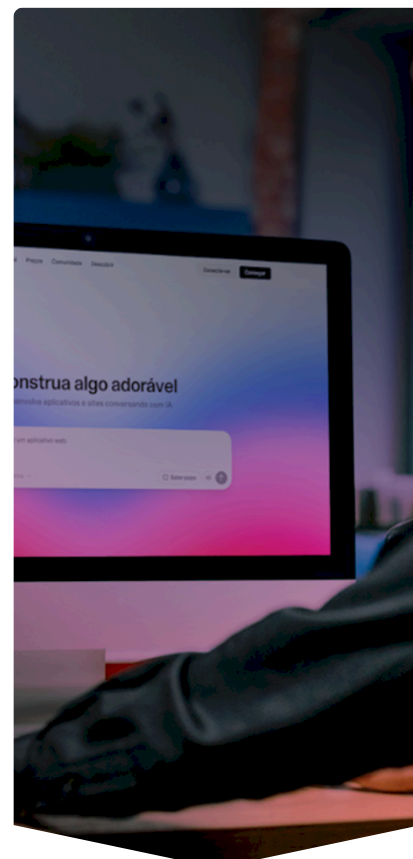


Programas utilizados
Lovable AI

O que aprender com este curso?

Engenharia de Prompt: Como estruturar comandos para gerar interfaces modernas e lógicas complexas.

Integrações de Valor: Conexão com GitHub, monetização real via Stripe e automação de suporte com Chatbots de IA.



Sumário

1 - Conhecendo a Lovable AI

- 1.1 - Apostila Aula 1 - Fundamentos do Lovable AI
- 1.2 - Introdução
- 1.3 - O que é a Lovable AI
- 1.4 - A Evolução do Desenvolvimento
- 1.5 - Origem e Credibilidade
- 1.6 - O Novo Profissional: Arquiteto de IA
- 1.7 - Tour pela Interface: Funcionalidades Chave
- 1.8 - Histórico e Publicação

2 - Fundamentos no Lovable

- 2.1 - Apostila Aula 2 - Engenharia de Prompt e Arquitetura Full-Stack
- 2.2 - Introdução
- 2.3 - Engenharia de Prompt: A Arte de Pedir
- 2.4 - Conceitos de Arquitetura "Invisível"
- 2.5 - Componentização e Iteração
- 2.6 - Segurança e Versionamento (GitHub)
- 2.7 - Frontend vs. Backend: O Caso da Tela de Login
- 2.8 - Atividade 1:
- 2.9 - Atividade 2:

3 - Supabase e Lovable Cloud

- 3.1 - Apostila Aula 3 - Backend, Banco de Dados e Autenticação
- 3.2 - Introdução
- 3.3 - O Ecossistema: Supabase e Lovable Cloud
- 3.4 - A Abstração: O Conceito de Controle Remoto
- 3.5 - Arquitetura de Dados: Auth vs. Public Tables
- 3.6 - Integração: Gerenciado vs. Próprio
- 3.7 - Gerenciamento de Autenticação na Prática
- 3.8 - Atividade 1:
- 3.9 - Atividade 2:

4 - Estilização e API

- 4.1 - Apostila Aula 4 - Estilização Avançada e Integração com APIs
- 4.2 - Introdução e Recapitulação
- 4.3 - O "Shopping de Componentes": 21st.dev
- 4.4 - APIs: O Conceito do "Garçom Digital"
- 4.5 - Integração de Pagamentos: Stripe
- 4.6 - O Estado Atual do Projeto Nexus
- 4.7 - Atividade 1:

- 4.8 - Atividade 2:

5 - Chat de suporte automático

- 5.1 - Apostila Aula 5 - Inteligência Artificial e Edge Functions
- 5.2 - Introdução e Recapitulação
- 5.3 - Integração Lovable AI: A Ponte Inteligente
- 5.4 - O Conceito de Edge Function (O "Chef" Especialista)
- 5.5 - Implementação do Chatbot: Frontend e Lógica
- 5.6 - Refinamento: Formatação de Resposta
- 5.7 - Conclusão da Jornada Nexus
- 5.8 - Atividade 1:
- 5.9 - Atividade 2:

6 - Projeto final I

- 6.1 - Apostila Aula 6 - Lógica de Negócios e Manipulação de Dados
- 6.2 - Introdução e Novo Desafio
- 6.3 - Configuração de Ambiente: Novo Projeto Supabase
- 6.4 - Prompt 1: Autenticação e Rotas Protegidas
- 6.5 - Prompt 2: Lógica de Banco de Dados e Funcionalidades
- 6.6 - Validação e Testes
- 6.7 - Próximos Passos
- 6.8 - Atividade 1:
- 6.9 - Atividade 2:

7 - Projeto final II

- 7.1 - Apostila Aula 7 - Monetização, Inteligência e Carreira
- 7.2 - Introdução: O Passo Final
- 7.3 - Integração de Inteligência: O Chatbot
- 7.4 - Monetização: Integração com Stripe
- 7.5 - Carreira: Segurança em Projetos Low-Code
- 7.6 - O Novo Profissional: Arquiteto de Soluções
- 7.7 - Venda de Soluções e Mercado
- 7.8 - Conclusão do Curso
- 7.9 - Atividade 1:
- 7.10 - Atividade 2:

1.1. Apostila Aula 1 - Fundamentos do Lovable AI

1.2. Introdução

S seja muito bem-vindo(a) à sua primeira apostila oficial do curso! Este material serve como um complemento técnico e aprofundado de tudo o que vimos na aula.

Hoje, demos o pontapé inicial em uma jornada que vai transformar a maneira como você enxerga o desenvolvimento de software. Não estamos apenas aprendendo uma ferramenta nova; estamos aprendendo uma nova metodologia de trabalho.

Nas próximas páginas, vamos detalhar os conceitos do Lovable AI, entender a tecnologia por trás da plataforma, conhecer sua origem no Vale do Silício e dissecar as funcionalidades da interface que exploramos no nosso tour inicial. Aproveite o conteúdo!

1.3. O que é a Lovable AI

Para dominar a ferramenta, primeiro precisamos esquecer a comparação com construtores de sites tradicionais (como WordPress ou Wix). A Lovable não te entrega um template pronto para editar.

A melhor definição técnica é que a Lovable atua como um Engenheiro de Software Full-Stack baseado em IA. Você fornece as regras de negócio e o design através de linguagem natural (prompts), e ela escreve o código do zero..

O grande diferencial é a entrega de uma aplicação real. O resultado final não é uma "caixa preta", mas sim um código-fonte proprietário, legível e exportável, utilizando a stack mais moderna do mercado atual:

- **Front-end:** React e Next.js (garantindo velocidade e SEO "Search Engine Optimization").
- **Estilização:** Tailwind CSS (para designs responsivos e customizáveis).
- **Back-end/Banco de Dados:** Integração com Supabase e Prisma (para lógica de servidor e persistência de dados).

1.4. A Evolução do Desenvolvimento

A Lovable representa a quarta geração na história do desenvolvimento de software, resolvendo o dilema entre "facilidade de uso" e "poder de customização".

1. Programação Manual (High Code): O método tradicional. Controle total, mas extremamente lento e caro.

2. Frameworks: O surgimento de facilitadores como Rails e Django, que aceleraram o processo, mas ainda exigiam conhecimento técnico profundo.

3. Low-Code/No-Code: Ferramentas visuais (arrastar e soltar). Trouxeram velocidade, mas muitas vezes prendem o usuário à plataforma e limitam a escala do projeto.

4. AI-Generated Code (Lovable): A união dos mundos. Você tem a velocidade do No-Code com a liberdade do código manual, pois a IA escreve o código que um humano escreveria, só que instantaneamente.

1.5. Origem e Credibilidade

É importante entender que a Lovable não é uma ferramenta genérica.

Ela nasceu no epicentro da tecnologia mundial e foi acelerada pela Y Combinator (a mesma que lançou Airbnb e Dropbox).

Seus fundadores trazem uma bagagem técnica pesada: Sahil Lavingia (fundador do Gumroad) e Adam D'Angelo (fundador do Quora e ex-CTO do Facebook).

A visão deles é clara: capacitar equipes pequenas (ou "eu-quipes") a construir produtos com a mesma complexidade e robustez de grandes empresas de tecnologia.

1.6. O Novo Profissional: Arquiteto de IA

Com essa ferramenta, o seu papel muda drasticamente. Você deixa de ser um "digitador de código" para se tornar um "Arquiteto de Soluções".

O trabalho braçal (boilerplate) é feito pela máquina.

O seu valor profissional passa a se concentrar em três pilares:

Engenharia de Prompt: A capacidade de traduzir uma ideia de negócio em instruções claras e lógicas que a IA possa executar sem erros.

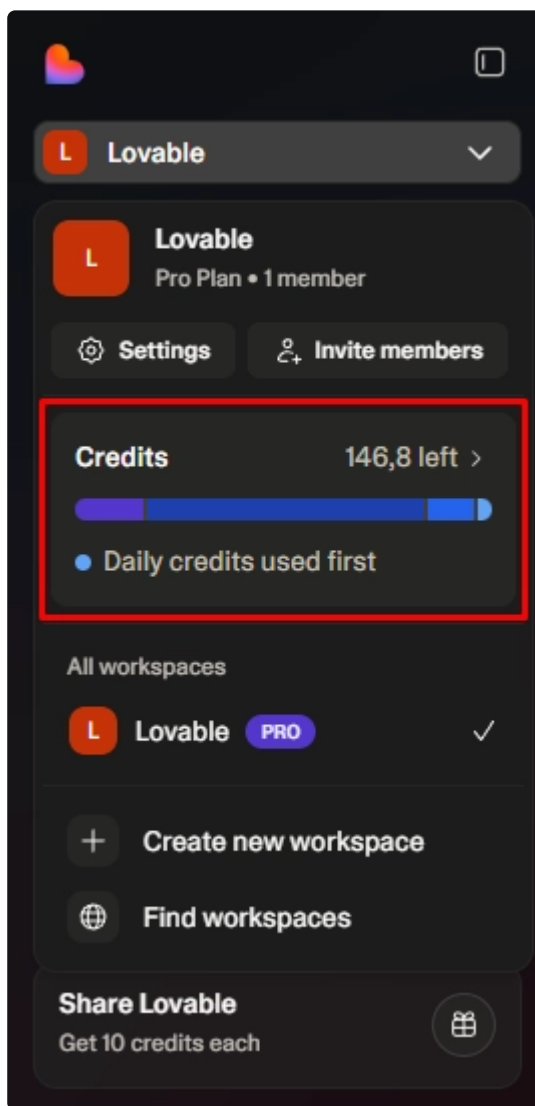
Visão Sistêmica: Entender como conectar as partes (Banco de Dados, Autenticação, Pagamentos) em vez de se preocupar com a sintaxe de uma função específica.

Refinamento (Code Review): A IA gera a base, mas o olhar humano é necessário para ajustes finos de design, experiência do usuário (UX) e regras de negócio complexas.

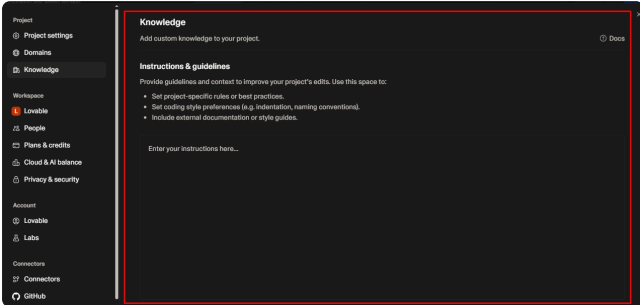
1.7. Tour pela Interface: Funcionalidades Chave

Sistema de Créditos: A plataforma opera com uma moeda interna.

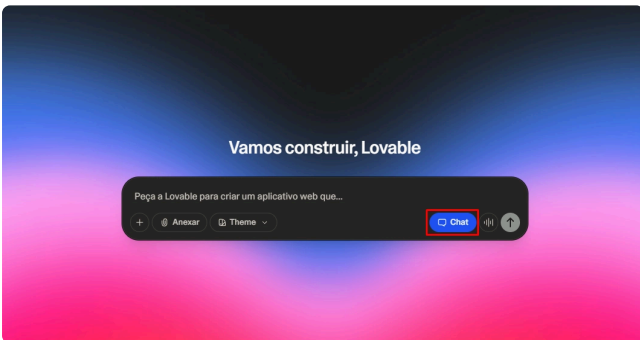
Usuários do plano gratuito recebem uma recarga diária (atualmente 5 créditos) para manter o ciclo de aprendizado e testes constantes.



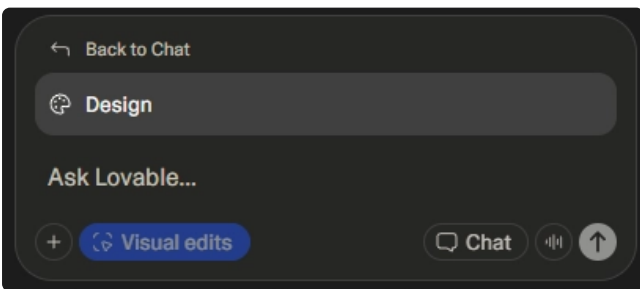
Knowledge (Conhecimento): Esta é uma das áreas mais poderosas. Aqui você define o "cérebro" do projeto. Ao anexar documentos de texto ou diretrizes, você ensina à IA o contexto do seu negócio, garantindo que ela não gere respostas genéricas.



Chat Mode: Uma ferramenta essencial para economia de recursos. Permite discutir o projeto, pedir explicações e planejar funcionalidades com a IA sem gastar créditos de geração de código imediatamente.

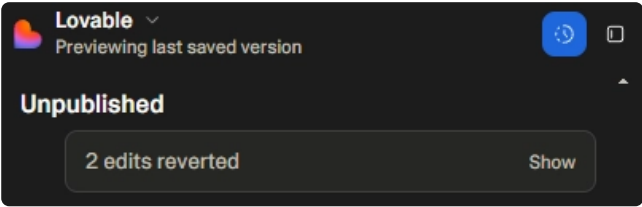


Visual Edits: Permite selecionar um elemento visualmente na tela e pedir uma alteração específica. Isso evita que a IA reescreva arquivos inteiros desnecessariamente, agilizando ajustes de design.



1.8. Histórico e Publicação

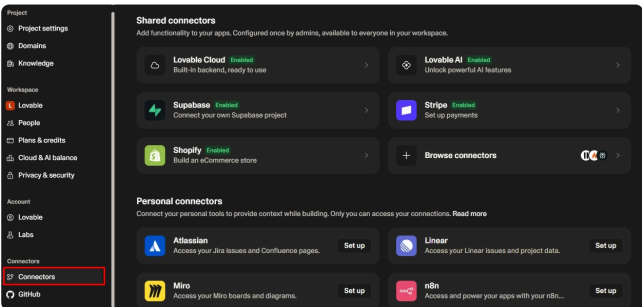
Versionamento: A Lovable possui um controle de versão nativo. Cada alteração cria um "save point". Você pode clicar no histórico para ver exatamente o que mudou no código ou reverter para uma versão anterior caso algo dê errado.



Preview Responsivo: A ferramenta de alternância entre Desktop e Mobile no topo da tela é crucial para garantir que sua aplicação funcione bem em celulares desde o primeiro dia.



Integrações: A interface já prepara o terreno para conexões externas robustas. Vimos os botões de conexão com GitHub (para guardar o código) e Supabase (para o banco de dados), que serão os focos das nossas próximas aulas práticas.



Anotações

2.1. Apostila Aula 2 - Engenharia de Prompt e Arquitetura Full-Stack

2.2. Introdução

S seja bem-vindo(a) à apostila da nossa segunda aula! Se na primeira aula focamos em entender o terreno e a ferramenta, hoje colocamos a mão na massa para construir nossa primeira aplicação real.

O objetivo deste material é aprofundar os conceitos técnicos que utilizamos durante a criação da Landing Page "AudioZen".

Vamos dissecar a lógica por trás dos prompts, entender a arquitetura moderna de software e a importância vital do versionamento de código.

Aqui, você encontrará o detalhamento teórico sobre Engenharia de Prompt, a explicação técnica da "Tech Stack" (React, Tailwind, Supabase) e o passo a passo lógico da integração com o GitHub. Vamos começar!

2.3. Engenharia de Prompt: A Arte de Pedir

A maior lição desta aula é que a Inteligência Artificial não lê pensamentos; ela executa comandos literais. A qualidade do código gerado é diretamente proporcional à qualidade da instrução fornecida.

Encaramos a IA da Lovable como um estagiário sênior: tecnicamente brilhante, mas que precisa de direção clara.

Para evitar resultados genéricos, utilizamos uma estrutura de três pilares fundamentais:

1. Contexto (Persona): Definimos quem a IA deve ser. Exemplo: "Atue como um especialista em UI/UX focado em conversão". Isso ajusta o "tom" do design e as prioridades do layout.

2. Tarefa (Objetivo): O que deve ser executado. Exemplo: "Crie uma Landing Page para fones de ouvido".

3. Detalhes (Constraints): As restrições e especificidades técnicas. Exemplo: "Use Tailwind CSS, tema escuro, botões arredondados e layout minimalista".

Dominar essa tríade é o que separa um usuário amador de um Engenheiro de Produto capaz de extrair o máximo da ferramenta.

2.4. Conceitos de Arquitetura "Invisível"

Durante a aula, mencionamos termos técnicos que formam a espinha dorsal de qualquer aplicação moderna. É crucial entender o que acontece nos bastidores.

Full-Stack (A Analogia do Restaurante):

A Lovable cria aplicações completas. O **Frontend** é o "Salão do Restaurante" (o que o cliente vê, clica e interage). O **Backend** é a "Cozinha" (onde os dados são processados e armazenados). Sem a cozinha, o salão é apenas decorativo.

A Tech Stack (Nossa Caixa de Ferramentas):

- **React:** A biblioteca JavaScript utilizada para construir a interface. Pense nela como blocos de LEGO inteligentes que formam os componentes visuais.
- **Tailwind CSS:** O framework de estilização. Na aula, usamos termos como "scale-105" e "rounded-full"; esses são comandos diretos do Tailwind para controlar animação e forma sem escrever CSS puro.
- **Supabase:** Será o nosso Backend e Banco de Dados (o "armário de arquivos" na nuvem), fundamental para tornar a aplicação funcional, como veremos na próxima aula.

2.5. Componentização e Iteração

Uma das melhores práticas demonstradas foi a **Componentização**. Em vez de tentar descrever o site inteiro em uma frase gigante, quebramos o problema em partes menores (Header, Hero Section, Cards de Produtos, Rodapé).

Isso evita que a IA se perca e permite um controle refinado sobre cada seção.

O processo de desenvolvimento com IA é **Iterativo**. Começamos com uma base, analisamos o resultado e refinamos.

Na aula, vimos como adicionar uma seção de preços ou alterar o destaque de um produto específico ("AudioZen Pro") sem quebrar o layout existente, apenas adicionando novas instruções ao contexto.

2.6. Segurança e Versionamento (GitHub)

Criar código é apenas metade do trabalho; mantê-lo seguro é a outra metade. Introduzimos o GitHub não como uma ferramenta complexa de código, mas como uma "Máquina do Tempo".

Ao conectar o projeto Lovable ao GitHub, garantimos duas coisas essenciais:

1. Backup Off-Platform: O código proprietário sai da plataforma da Lovable e fica salvo em um repositório seu. Isso garante que você é o dono real da aplicação.

2. Pontos de Restauração: Se uma alteração futura quebrar o site, o histórico de "commits" do GitHub permite voltar para a versão anterior que funcionava perfeitamente.

2.7. Frontend vs. Backend: O Caso da Tela de Login

Finalizamos a aula criando uma tela de Login/Cadastro visualmente profissional. No entanto, notamos que ela não realiza a ação de logar de verdade.

Isso ilustra perfeitamente a diferença entre as camadas. Criamos o **Frontend** (a casca, o formulário, o botão), mas falta o **Backend** (o sistema que verifica se o e-mail e senha existem no banco de dados).

Na próxima aula, resolveremos exatamente isso, conectando o Supabase para transformar essa "maquete interativa" em um sistema funcional capaz de armazenar usuários.

2.8. Atividade 1:

Objetivo: Praticar a estrutura dos 3 Pilares da Engenharia de Prompt (Contexto, Tarefa e Detalhes) que vimos na aula.

Cenário: Você foi contratado para criar a Landing Page de uma **Cafeteria Artesanal** chamada "Aroma & Grão". O cliente quer algo aconchegante, em tons pastéis e marrom, focado em mostrar o cardápio e o ambiente.

Desafio A (O Jeito Errado): Escreva abaixo como seria um prompt "fraco" e vago para esse pedido (aquele que a IA faria algo genérico).

Escreva seu prompt detalhado abaixo:

Anotações

Desafio B (O Jeito Lovable): Agora, reescreva esse pedido criando um **Prompt Estruturado**. Lembre-se de definir:

Contexto: Quem a IA deve ser? (Ex: Especialista em UI/UX de alimentos...). **Tarefa:** O que ela deve criar? (Landing Page, Seções...).

Detalhes: Cores, estilo visual (Tailwind), sentimentos que o site deve passar.

Sua resposta aqui:

Anotações

2.9. Atividade 2:

Objetivo: Treinar a capacidade de dar instruções visuais específicas (Tech Stack) para um componente, assim como fizemos com os "Cards de Produto" na aula.

Cenário: O site da cafeteria já tem a estrutura básica. Agora, você precisa adicionar uma seção de "**Depoimentos de Clientes**".

Instrução: Escreva um prompt focado apenas nesta seção. Não peça o site todo, peça apenas este componente.

Requisitos para o seu Prompt:

Use termos técnicos visuais que aprendemos (Ex: "borda arredondada", "sombra suave", "layout em vertical ou horizontal").

Defina que deve haver 3 cards de depoimentos lado a lado.

Peça para incluir estrelas de avaliação (ex: 5 estrelas) e a foto do cliente em formato circular.

Escreva seu prompt detalhado abaixo:

Anotações

3.1. Apostila Aula 3 - Backend, Banco de Dados e Autenticação

3.2. Introdução

Bem-vindo(a) à apostila da terceira aula! Até agora, nossos projetos eram apenas "cascas" bonitas: tinham visual (Frontend), mas não tinham memória (Backend).

Hoje, resolvemos o problema da tela de login "oca" criada na aula anterior.

Mergulhamos no universo do Backend, entendendo onde os dados vivem, como a segurança funciona e como conectar tudo isso ao nosso projeto Lovable.

Vamos detalhar tecnicamente o que é o Supabase, a diferença entre ele e o Lovable Cloud, e a arquitetura de segurança que separa dados de login de dados de perfil.

3.3. O Ecossistema: Supabase e Lovable Cloud

Para criar uma aplicação Full-Stack, precisamos de dois pilares: um lugar para o site "morar" e um lugar para os dados "morarem".

Supabase (O Cofre de Dados): Muitas vezes comparado ao Firebase, (outro banco de dados), o Supabase é uma plataforma Open Source que oferece Backend como Serviço.

Ele fornece um banco de dados PostgreSQL real, autenticação completa e armazenamento de arquivos.

Lovable Cloud (A Hospedagem): É o ambiente de Deploy. Quando publicamos o projeto, a Lovable pega todo o código React/Tailwind (Frontend) e o coloca em um servidor acessível via URL pública.

A Relação: O site (hospedado no Lovable Cloud) atua como a interface que solicita informações ao Supabase. Eles trabalham em simbiose, mas são entidades separadas.

3.4. A Abstração: O Conceito de Controle Remoto

Durante a aula, vimos abas de "Database" dentro da interface da Lovable.

É crucial entender que a Lovable não criou um banco de dados próprio do zero.

A interface da Lovable funciona como um "Controle Remoto Simplificado" ou uma interface de abstração.

Quando você pede para a IA "criar uma tabela de usuários", a Lovable envia comandos SQL invisíveis para o Supabase.

O Supabase executa a ação pesada, e a Lovable apenas exibe o resultado de forma amigável.

Isso permite que você utilize o poder de um banco de dados empresarial (PostgreSQL) sem precisar escrever uma linha de código SQL complexo neste momento.

3.5. Arquitetura de Dados: Auth vs. Public Tables

Um dos conceitos mais importantes abordados foi a separação dos dados de usuário. Por segurança, o Supabase não mistura credenciais com informações públicas.

Tabela `auth.users` (O Cofre):

Esta é uma área restrita e gerenciada automaticamente pelo Supabase. Ela armazena apenas dados críticos: ID único do usuário (UUID), e-mail criptografado, senha (hash) e datas de login.

Nem mesmo nós, como desenvolvedores, devemos editar isso manualmente com frequência.

Tabela `public.profiles` (A Pasta de Arquivos):

Como a tabela `auth` é bloqueada, precisamos de um lugar para guardar Nome, Foto, Biografia e Preferências. Para isso, criamos a tabela `profiles`.

A lógica é conectar as duas: A tabela `profiles` possui uma coluna que referencia o ID do usuário na tabela `auth`. Assim, garantimos segurança máxima para o login e flexibilidade total para o perfil.

3.6. Integração: Gerenciado vs. Próprio

Existem duas formas de usar o Supabase com a Lovable, e na aula fizemos a transição de uma para a outra.

Modo Gerenciado (Sandbox): Ao clicar em "Enable Cloud" pela primeira vez, a Lovable cria um banco de dados temporário e gerenciado por ela. É ótimo para testes rápidos, mas você não tem acesso ao painel administrativo completo.

Modo Próprio (Conexão Direta): Para projetos profissionais, criamos nossa própria organização no painel do Supabase e a conectamos via API (Settings > Integrations).

Essa conexão nos dá poderes de administrador ("Super Admin"), permitindo acesso direto às tabelas, logs de segurança, configuração de e-mails e regras de permissão avançadas (RLS).

3.7. Gerenciamento de Autenticação na Prática

Ao assumirmos o controle do Supabase, ganhamos a capacidade de configurar o fluxo de entrada dos usuários.

Confirmação de E-mail: Por padrão, o Supabase exige que o usuário clique em um link no e-mail para ativar a conta.

Para desenvolvimento e testes ágeis, aprendemos a desativar essa opção em *Authentication > Providers > Email > Confirm Email*, permitindo login imediato.

Ciclo de Vida do Usuário: Testamos na prática a criação, o login e a exclusão de usuários.

Validação de Segurança: Ao deletar um usuário diretamente no painel do Supabase e tentar logar novamente no site feito na Lovable, comprovamos que a integração é em tempo real.

Se o usuário não existe no "Cofre" (Supabase), o "Porteiro" (Lovable) não permite a entrada.

3.8. Atividade 1:

Objetivo: Testar o entendimento sobre a separação de segurança no Supabase (Auth vs. Public Tables).

Cenário: Você está desenhando o banco de dados de uma rede social. Você tem as seguintes informações dos usuários para guardar: **Senha, Foto de Perfil, E-mail de Login, Biografia e Data do Último Acesso.**

Desafio: Classifique onde cada informação deve ser guardada para manter o sistema seguro. Escreva os itens na coluna correta abaixo.

Tabela auth.users (O Cofre Blindado):

Anotações

Tabela public.profiles (A Pasta Pública):

Anotações

- *(Dica: Lembre-se que o "Cofre" só guarda o essencial para o sistema reconhecer a identidade, enquanto a "Pasta" guarda o que o site mostra para os amigos).*

3.9. Atividade 2:

Objetivo: Relembrar a configuração crítica de ambiente de desenvolvimento que fizemos na aula para agilizar os testes.

Cenário: Você está criando um protótipo rápido para um cliente e precisa testar o fluxo de cadastro 50 vezes hoje. Porém, toda vez que você cria uma conta de teste no seu site, o sistema trava dizendo "Aguardando Verificação" e te obriga a abrir o Gmail e clicar em um link para liberar o acesso. Isso está matando sua produtividade.

Pergunta: Como Administrador do projeto no Supabase, qual configuração específica você deve alterar para resolver isso e permitir logins imediatos?

Onde você clica no menu lateral do Supabase? (Dica: Ícone de cadeado/segurança).

Anotações

Qual opção você deve **desativar** nas configurações de E-mail Provider?

Anotações

4.1. Apostila Aula 4 - Estilização Avançada e Integração com APIs

4.2. Introdução e Recapitulação

Sejam bem-vindos à apostila da quarta aula! Estamos avançando rápido. Na aula passada, demos "memória" ao site com o Supabase. Hoje, focamos em dar "beleza" e "poder financeiro".

Esta apostila vai detalhar o uso de bibliotecas de componentes de IA (21st.dev), desmistificar o conceito técnico de APIs e guiar você pela integração segura de pagamentos com a Stripe.

4.3. O "Shopping de Componentes": 21st.dev

Para criar interfaces modernas sem gastar horas de desenvolvimento ou créditos excessivos na Lovable, utilizamos o **21st.dev**.

Ele não é apenas um site de design; é uma biblioteca de componentes construídos com a mesma "Tech Stack" da Lovable: **React**, **Tailwind CSS** e **Shadcn UI**.

Isso significa compatibilidade nativa. Ao copiar um prompt ou código de lá, a Lovable entende perfeitamente como renderizar o componente (botões neon, cards animados, bordas brilhantes) sem precisar de adaptações complexas.

Estratégia de Desenvolvimento: Em vez de tentar descrever visualmente um efeito complexo ("faça um brilho que rode na borda..."), copiamos a instrução técnica exata do 21st.dev. Isso economiza tokens da IA e garante um resultado visual de alto nível.

4.4. APIs: O Conceito do "Garçom Digital"

A sigla **API** (Application Programming Interface) assusta, mas o conceito é simples. APIs são mensageiros seguros que permitem que dois softwares diferentes conversem.

Analogia do Restaurante:

O Cliente: Seu site na Lovable.

A Cozinha: O serviço externo (Google, Stripe, Maps) que tem o recurso que você quer.

O Garçom (API): O intermediário. Seu site não entra na "cozinha" da Stripe (seria inseguro). Ele manda o pedido pelo "garçom", que traz a resposta ("Pagamento Aprovado").

Chaves de API (API Keys): São as credenciais de segurança (como um crachá VIP). Elas identificam quem está fazendo o pedido.

Public Key: Pode ser vista no navegador (usada para exibir elementos na tela).

Secret Key: Jamais deve ser compartilhada. Ela dá poder total sobre a conta (fazer estornos, criar assinaturas). No nosso projeto, a Lovable guarda essa chave em um ambiente seguro (Backend/Edge Function), nunca a expando no código do site.

4.5. Integração de Pagamentos: Stripe

A regra de ouro da programação financeira é: **Nunca construa seu próprio sistema de armazenamento de cartões de crédito.**

A segurança necessária (PCI Compliance) é extremamente complexa e cara. Ao usar a Stripe, terceirizamos essa responsabilidade. O cartão do cliente nunca toca no nosso banco de dados; ele vai direto para os cofres blindados da Stripe.

Fluxo de Integração Realizado:

1. Criação do Produto: Definimos o "Plano Pro" e seu preço diretamente no painel da Stripe.

2. Modo de Teste (Sandbox): Trabalhamos em um ambiente seguro onde dinheiro real não é movimentado, permitindo erros e ajustes sem prejuízo.

3. Conexão via Lovable: Usamos a Secret Key para autorizar a Lovable a listar nossos produtos e criar sessões de checkout.

4. Teste de Checkout: Utilizamos os dados de teste fornecidos pela Stripe (Cartão 4242 4242 4242 4242) para simular uma compra bem-sucedida de ponta a ponta.

4.6. O Estado Atual do Projeto Nexus

Ao final desta aula, o Nexus evoluiu de um protótipo visual para um **MVP (Produto Mínimo Viável)** funcional.

Ele possui:

Frontend: Interface moderna com componentes do 21st.dev.

Backend: Banco de dados e Autenticação (Supabase).

Funcionalidade: Sistema de vendas real (Stripe).

O próximo passo lógico, que veremos na Aula 5, é adicionar inteligência e interatividade por Chatbots e, finalmente, publicar nossa criação para o mundo.

4.7. Atividade 1:

Objetivo: Fixar a analogia do "Garçom" utilizada na aula para entender como o site se comunica com serviços externos.

Cenário: Você está explicando para um cliente como o sistema de login com o Google funciona no site dele. Use a analogia do restaurante para preencher as lacunas abaixo corretamente.

Colunas para relacionar:

- (A) O Cliente na Mesa
- (B) A Cozinha do Restaurante
- (C) O Garçom
- (D) O Crachá VIP

Preencha com a letra correspondente:

É a **API**, que leva o pedido e traz a resposta.

É o **Seu Site (Frontend)**, que precisa da informação, mas não pode entrar na cozinha.

É a **API Key (Chave de Segurança)**, que prova que o pedido é confiável e autorizado.

É o **Serviço Externo (Google/Stripe)**, onde os dados e recursos pesados ficam guardados.

4.8. Atividade 2:

Objetivo: Testar o conhecimento sobre segurança financeira e uso correto das chaves da Stripe.

Cenário: Você está configurando a monetização do "Nexus" e chegou a hora de conectar a Stripe na Lovable. Você abriu o painel de desenvolvedor da Stripe e viu duas chaves diferentes.

Pergunta 1: Qual das chaves abaixo você deve copiar e colar no prompt da Lovable para permitir que ela crie sessões de checkout seguras? (Marque a correta).

Chave Publicável (pk_test_...)

Chave Secreta (sk_test_...)

Pergunta 2 (Situação de Risco): Um amigo sugeriu: *"Para economizar, vamos criar uma tabela no nosso banco de dados chamada 'cartoes_credito' e guardar o número do cartão dos clientes lá"*.

Com base no que aprendemos na aula, por que você **NUNCA** deve fazer isso?

Anotações

- (Dica: Lembre-se do nível de segurança militar exigido e do papel da Stripe como intermediário).

Anotações

5.1. Apostila Aula 5 - Inteligência Artificial e Edge Functions

5.2. Introdução e Recapitulação

Sejam bem-vindos à apostila da nossa quinta aula! Hoje, celebramos um marco importante. O Nexus deixou de ser um projeto estático e ganhou "vida" através da integração de Inteligência Artificial.

Recapitulando nossa trajetória técnica até aqui:

Aula 1 & 2 (Frontend): Construimos a interface visual e entendemos a Engenharia de Prompt.

Aula 3 (Backend): Implementamos "memória" e segurança com Supabase (Autenticação e Banco de Dados).

Aula 4 (Financeiro): Conectamos o projeto ao mundo real com a API de pagamentos da Stripe.

Aula 5 (Inteligência): Integramos um Chatbot autônomo e entendemos a infraestrutura de nuvem que torna isso possível.

5.3. Integração Lovable AI: A Ponte Inteligente

Para criar o Chat de Suporte, não construímos uma IA do zero. Utilizamos a integração nativa da Lovable AI, que atua como um "Gateway" (Portal).

Por que usar o Gateway da Lovable?

Simplificação: Evitamos a configuração manual complexa de servidores e chaves de API individuais para cada modelo de IA.

Variabilidade de Modelos: A Lovable permite trocar o "cérebro" do chat (ex: GPT-4, Gemini, Claude) com facilidade, sem reescrever o código.

Gestão de Limites (Rate Limiting): A plataforma gerencia o fluxo de perguntas, evitando bloqueios comuns em APIs públicas que ocorrem quando muitos usuários perguntam ao mesmo tempo.

5.4. O Conceito de Edge Function (O "Chef" Especialista)

Este é o conceito técnico mais importante da aula. Para conectar nosso chat à IA de forma segura e rápida, utilizamos **Edge Functions**.

Analogia do Restaurante:

Se o Site é o Cliente e a API é o Garçom, a **Edge Function** é um Chef Especialista que trabalha sob demanda.

Diferente de um servidor tradicional (que fica ligado 24h gastando energia), a Edge Function "acorda" apenas quando recebe um pedido, processa a resposta em milissegundos e volta a "dormir".

Isso reduz drasticamente os custos de infraestrutura (Serverless).

Por que "Edge" (Borda)? O código não roda em um único computador na Califórnia. Ele é replicado globalmente.

- Se o usuário está no Brasil, a função roda num servidor em São Paulo.
- Se está no Japão, roda em Tóquio.

Isso garante latência (atraso) quase zero.

Segurança: É na Edge Function que guardamos as chaves secretas.

O usuário final nunca tem acesso direto à API da IA, apenas à resposta processada pela função.

5.5. Implementação do Chatbot: Frontend e Lógica

A construção do chat foi dividida em duas etapas estratégicas via prompt:

1. Frontend (A Casca): Primeiro, definimos o visual. Usamos o estilo **Glassmorphism** (efeito de vidro fosco) para manter a coerência com o design moderno do Nexus.

Criamos os componentes visuais: botão flutuante, janela de chat, balões de mensagem e input de texto.

2. Integração Lógica (O Cérebro): Configuramos a Edge Function para usar o modelo **Gemini 2.5-flash** (rápido e eficiente).

Definimos o "System Prompt" (instrução de sistema) para que a IA atue como um suporte técnico do Nexus, conhecendo os planos e preços.

5.6. Refinamento: Formatação de Resposta

Um desafio comum ao lidar com IAs de texto é a formatação. As IAs enviam respostas em **Markdown** (usando símbolos como ****** para negrito ou **-** para listas).

Se o Frontend não souber interpretar isso, o usuário vê os códigos feios (ex: ****01á****).

Resolvemos isso com um prompt de refinamento, instruindo o sistema a renderizar o Markdown corretamente. O resultado foi um chat profissional que exibe negritos reais, listas organizadas e quebra de linhas, proporcionando uma experiência de leitura fluida.

5.7. Conclusão da Jornada Nexus

Ao finalizar esta aula, você tem em mãos um **Software** completo.

O Nexus agora possui os quatro pilares de um software moderno:

Identidade: Sistema de Login seguro.

Memória: Banco de Dados persistente.

Poder Financeiro: Processamento de pagamentos.

Inteligência: Suporte autônomo via IA.

Este projeto serve como a base perfeita (boilerplate) para qualquer ideia futura que você queira desenvolver. A arquitetura que construímos aqui é escalável e utiliza as mesmas tecnologias das grandes startups do mercado.

5.8. Atividade 1:

Objetivo: Praticar a criação de um "System Prompt" (Instrução de Sistema) eficiente, evitando que o Chatbot dê respostas genéricas ou incorretas.

Cenário: Você integrou o Chatbot no projeto "Nexus", mas esqueceu de configurar o contexto. Um usuário perguntou: "Qual o melhor prato do dia?" e a IA respondeu com uma receita de lasanha, pois ela não sabe que é um suporte técnico de software.

Desafio: Escreva abaixo as instruções que você daria no prompt para "consertar" a personalidade dessa IA.

Preencha os 3 mandamentos do seu Bot:

Identidade (Quem você é?): (Ex: Você é o assistente virtual sênior do Nexus...)

Anotações

Restrição (O que você NÃO faz?): (Ex: Não responda sobre assuntos fora de tecnologia/vendas...)

Anotações

Conhecimento (O que você vende?): (Ex: O Plano Pro custa R\$ 29,90 e inclui...)

Anotações

5.9. Atividade 2:

Objetivo: Verificar se você entendeu por que usamos a Lovable como "Gateway" e Edge Functions em vez de conectar a API direto no site.

Cenário: Um colega desenvolvedor iniciante disse: "Para ser mais rápido, vou colocar a minha API Key do Gemini direto no código do botão de Chat do site."

Pergunta: Com base na analogia do "Chef de Cozinha" (Edge Function) que vimos na aula, explique por que a ideia do seu colega é perigosa e incorreta.

Marque as opções que explicam a função da Edge Function:

Ela serve apenas para deixar o site mais bonito visualmente.

Ela atua como um servidor seguro que esconde a **Chave Secreta** (API Key) do navegador do usuário, evitando roubos.

Ela roda "na borda" (em vários lugares do mundo), garantindo que a resposta chegue rápido ao usuário, como um Chef que cozinha na hora.

Ela permite que o usuário acesse o banco de dados da NASA.

Anotações

6.1. Apostila Aula 6 - Lógica de Negócios e Manipulação de Dados

6.2. Introdução e Novo Desafio

Sejam bem-vindos à apostila da sexta aula! Após finalizarmos o Nexus com recursos avançados de IA, hoje iniciamos um novo ciclo com um projeto diferente: o **FinanceFlow**.

O objetivo pedagógico desta mudança é simular o dia a dia de um desenvolvedor: pegar um projeto que possui apenas a interface visual (Frontend) pronta e torná-lo funcional (Backend e Lógica).

Encontramos o FinanceFlow no estado que chamamos de "Cenário de Hollywood": fachadas bonitas, mas sem nada por trás. Se você clicasse em "Login", nada acontecia; os gráficos eram estáticos e os números eram fictícios (Dummy Data).

A missão desta aula foi conectar os fios, transformando esse protótipo visual em uma aplicação capaz de gerenciar dados reais.

6.3. Configuração de Ambiente: Novo Projeto Supabase

Como este é um aplicativo diferente do Nexus, precisamos de um novo Banco de Dados. Não misturamos dados de projetos diferentes na mesma instância para garantir segurança e organização.

Criação do Projeto: No painel do Supabase, criamos uma nova instância dentro da nossa organização. Isso gera novas chaves de API e URL de conexão.

Otimização de Desenvolvimento (DevX): Realizamos uma configuração crucial em *Authentication > Providers > Email*: desativamos a opção "Confirm email".

Por que fazer isso? Em ambiente de produção (mundo real), queremos confirmar o e-mail. Mas em ambiente de desenvolvimento (testes), essa etapa cria atrito. Desativá-la nos permite criar contas de teste instantaneamente e validar o fluxo do sistema sem precisar abrir a caixa de entrada a cada teste.

6.4. Prompt 1: Autenticação e Rotas Protegidas

O primeiro comando para a IA teve como foco a segurança, metaforicamente descrita como "instalar as portas e fechaduras".

Route Protection (Proteção de Rotas): Instruímos o Lovable a bloquear o acesso às páginas internas (Dashboard, Transações) para qualquer pessoa que não esteja autenticada. Se alguém tentar acessar a URL direta sem logar, o sistema deve redirecionar para o Login.

Integração de Formulário: Conectamos os inputs visuais (e-mail e senha) criados no design diretamente à API de Autenticação do Supabase.

O resultado é que o sistema deixa de aceitar qualquer texto e passa a validar credenciais reais contra a tabela `auth.users` do banco de dados.

6.8. Atividade 1:

Objetivo: Praticar a criação de prompts que transformam elementos visuais estáticos ("Dummy Data") em funcionalidades reais conectadas ao Supabase.

Cenário: No painel do **FinanceFlow**, existe um botão lindo chamado "Adicionar Nova Transação". Atualmente, quando você clica nele, ele apenas fecha a janela e não salva nada. É puro "efeito especial".

Desafio: Escreva um prompt para a Lovable que faça esse botão funcionar de verdade.

Requisitos do seu Prompt:

- **Ação:** Ordene que, ao clicar em "Salvar", os dados do formulário sejam enviados para uma tabela específica no Supabase.
- **Estrutura de Dados:** Especifique quais colunas devem ser preenchidas (Ex: valor, descrição, tipo: receita/despesa).
- **Feedback Visual:** Peça para atualizar a lista da tela (reatividade) assim que o salvamento for concluído.

Sua resposta aqui:

Anotações

6.9. Atividade 2:

Objetivo: Fixar o conceito de "Route Protection" e segurança de dados, garantindo que apenas usuários logados vejam informações sensíveis.

Cenário: Você enviou o link `financeflow.app/dashboard` para um amigo testar. Ele clicou e, mesmo sem ter criado conta ou feito login, conseguiu ver os gráficos financeiros. Isso é uma falha grave de segurança!

Desafio: Escreva um comando (prompt) para corrigir essa falha, instruindo a IA a agir como um "segurança".

Elementos obrigatórios no Prompt:

Diga explicitamente qual tecnologia de autenticação deve ser verificada (Supabase Auth).

- Defina a regra: "Se o usuário NÃO estiver logado..."
- Defina a ação: "...redirecione ele imediatamente para a página de _____".

Sua resposta aqui:

Anotações

7.1. Apostila Aula 7 - Monetização, Inteligência e Carreira

7.2. Introdução: O Passo Final

Chegamos à etapa final do nosso curso. Até o momento, construímos um software funcional: ele tem interface (Frontend), memória (Banco de Dados) e capacidade de exportação. Mas, para ser um negócio digital, faltam dois pilares: Inteligência e Dinheiro.

Nesta apostila, vamos detalhar como integramos o "cérebro" do suporte (Chatbot) e o "bolso" do projeto (Stripe).

Além da parte técnica, dedicaremos uma seção especial à postura profissional necessária para atuar nesse novo mercado de IA e Low-Code, abordando segurança, arquitetura e vendas.

7.3. Integração de Inteligência: O Chatbot

Para adicionar o Chatbot ao FinanceFlow, utilizamos a integração nativa da Lovable, atuando como um Gateway. Isso elimina a necessidade de configurar servidores intermediários complexos manualmente.

O segredo do sucesso nesta etapa foi o refinamento do Prompt. Não pedimos apenas "um chat". Estruturamos o pedido em três camadas:

1. **Integração:** Solicitamos que a Lovable conecte o componente visual à API de IA (Gemini).
2. **Contexto (System Prompt):** Ensinamos à IA quem ela é ("Suporte do FinanceFlow") e sobre o que ela pode falar (Planos Premium), garantindo que ela não responda sobre assuntos aleatórios.
3. **Refinamento Visual:** Instruímos explicitamente a IA a formatar as respostas corretamente, evitando que códigos Markdown (como asteriscos * ou cerquilhas #) apareçam "quebrados" para o usuário final.

7.4. Monetização: Integração com Stripe

A implementação de pagamentos seguiu um fluxo de segurança rigoroso. O objetivo não é processar o cartão dentro do nosso site, mas sim criar uma "Sessão de Checkout" segura.

O Fluxo Técnico: O usuário clica em "Assinar". O nosso sistema (Backend) usa a **Chave Secreta** (API Key) para conversar com a Stripe e dizer: "Crie uma cobrança de R\$ 29,90". A Stripe devolve uma URL única e segura. O sistema redireciona o usuário para essa URL. Após o pagamento, a Stripe devolve o usuário para o nosso site (URL de Sucesso).

Segurança de Chaves: A Chave Secreta (Secret Key) é o item mais sensível do projeto. Ela nunca deve ser exposta no código frontend (visível no navegador). A Lovable gerencia isso guardando-a em variáveis de ambiente seguras no servidor.

7.5. Carreira: Segurança em Projetos Low-Code

Construir rápido não significa ignorar a segurança. Existe uma diferença vital entre "Segurança da Ferramenta" e "Segurança da Aplicação".

Segurança de Infraestrutura: O Supabase e a Lovable garantem que o servidor não vai cair e que a criptografia básica funciona. Isso é responsabilidade deles.

Segurança Lógica (Sua Responsabilidade): A IA pode escrever um código que "esquece" de checar se o usuário pagou antes de liberar o produto. Isso é uma falha de lógica, não de ferramenta.

O Papel do Pentester: Em projetos comerciais que lidam com dados sensíveis ou dinheiro, é recomendável contratar um Pentester (Hacker Ético) para auditar o sistema, tentando encontrar brechas no RLS (Row Level Security) ou injeções de código que a IA deixou passar.

7.6. O Novo Profissional: Arquiteto de Soluções

A IA comoditizou a escrita de código. O valor do profissional hoje não está em digitar sintaxe, mas em orquestrar sistemas.

Visão Sistêmica: O Arquiteto não foca apenas no botão; ele foca no fluxo de dados. Ele entende como o dado sai do formulário, passa pela validação, é gravado no banco e aciona uma automação de pagamento.

Documentação: A facilidade de criação pode gerar "monstros" complexos. Um bom profissional documenta seus prompts, desenha os fluxos e anota as decisões de arquitetura para garantir a manutenção futura.

7.7. Venda de Soluções e Mercado

O cliente final não compra "React", "Supabase" ou "Lovable". Ele compra a resolução de um problema.

Posicionamento: Ao vender seus serviços, foque na dor do cliente ("Parar de cobrar manualmente no WhatsApp") e não na tecnologia ("Integração de API").

A Vantagem Competitiva: Use o Low-Code/IA como argumento de venda. Ofereça prazos de entrega imbatíveis (semanas em vez de meses) e custos de manutenção reduzidos.

Transparência sobre o uso de tecnologias modernas gera confiança e justifica o valor agregado.

7.8. Conclusão do Curso

Encerramos aqui nossa jornada. Você quebrou a barreira técnica que separa a ideia da execução.

O FinanceFlow é a prova de que você domina a criação de softwares completos: com identidade, memória, inteligência e capacidade financeira.

O próximo passo é com você: use este projeto como laboratório, quebre-o, reconstrua-o ou inicie algo totalmente novo. As ferramentas estão em suas mãos. Sucesso!

7.9. Atividade 1:

Objetivo: Praticar a integração de pagamentos reais, transformando um botão estático em um checkout funcional seguro via API.

Cenário: Você tem um produto incrível ("FinanceFlow Premium"), um preço definido (R\$ 29,90) e um botão "Assinar" na tela. Porém, ao clicar nele, nada acontece. O usuário quer te pagar, mas o sistema não deixa. Você já criou o produto na Stripe e tem a **Chave Secreta (Secret Key)** em mãos.

Desafio: Escreva um prompt para a Lovable que conecte essas pontas.

Requisitos do seu Prompt:

Ação: Instrua a IA a conectar o projeto à Stripe usando a chave que você vai fornecer.

Lógica de Negócio: Ordene que, ao clicar no botão "Assinar", o usuário seja redirecionado para a página de checkout oficial da Stripe (o link de pagamento).

Segurança: Deixe claro que a chave secreta deve ser usada para autenticar essa conexão nos bastidores (Backend/Edge Function).

Sua resposta aqui:

Anotações

7.10. Atividade 2:

Objetivo: Consolidar a integração de Inteligência Artificial para automação de suporte, usando a Lovable como "Gateway".

Cenário: O botão de chat no canto da tela é bonito, mas mudo. Se um usuário perguntar "Como cancelo minha assinatura?", ele fica sem resposta. Você quer que o Gemini (via Lovable AI) responda a essas dúvidas automaticamente, 24 horas por dia.

Desafio: Crie um prompt para dar vida e contexto a esse atendente virtual.

Requisitos do seu Prompt:

Integração: Peça explicitamente para conectar o componente de chat visual à IA da Lovable.

Contexto (Roleplay): Defina quem é a IA. (Ex: "Você é o suporte técnico do FinanceFlow...").

Instrução de Formatação: Peça para que a IA formate as respostas de jeito limpo, evitando códigos estranhos (Markdown quebrado) e usando listas quando necessário.

Sua resposta aqui:

Anotações