

Plano de Negócio e Arquitetura Técnica

1. Proposta de Valor

Criar uma plataforma web **all-in-one** para profissionais de estética (salões, clínicas de beleza, esteticistas) nos EUA, integrando agendamento, gerenciamento de clientes e pagamentos seguros via Stripe Connect Standard. O diferencial é oferecer **simplicidade e compliance**: o profissional gerencia todos os agendamentos e recebe pagamentos sem burocracia, contando com a infraestrutura do Stripe para tokenização de dados de cartão e conformidade PCI/Dados sensíveis ¹. Por exemplo, a StyleSeat, líder no segmento, integrou Stripe Connect para permitir que clientes “descubram, agendem e paguem” diretamente pelo app, melhorando a experiência e criando nova fonte de receita para esteticistas ². Nossa plataforma seguirá esse modelo: além de facilitarmos o processo de agendamento, cuidaremos do **onboarding dos profissionais** no Stripe (contas Connect Standard) e do processamento de pagamentos, de forma legal e automatizada ³ ¹. Em resumo, a proposta é economizar tempo e atritos para o profissional de beleza, aumentando sua receita e segurança financeira com uma solução especializada.

2. Segmento de Clientes

Inicialmente, o foco será em **profissionais de estética**: salões de beleza, clínicas estéticas, spas e esteticistas individuais. Esse é um mercado robusto nos EUA – estima-se cerca de **979.000 salões de beleza ativos** (set/2022) ⁴, parte de um setor avaliado em ~US\$53 bilhões em 2024 e projetado a crescer ~6,8% ao ano ⁵. Adicionalmente, pesquisas indicam que **70% dos spas** já oferecem agendamento online ⁶, mostrando aceitação de tecnologia no segmento. Como plano de expansão, podemos atingir **barbearias** (apoio ao grooming masculino, segmento em alta ⁷) e **academias/fitness** (classes, personal trainers, tal como fazem apps de wellness). Esses negócios compartilham a mesma necessidade de agendamento e pagamento recorrente. Em resumo, começaremos no nicho de beleza (onde 75% dos profissionais são mulheres ⁷) e, à medida que validamos o produto, ampliamos para segmentos adjacentes (barbearias e estúdios de fitness), todos centrados em serviços baseados em horário e recorrência de clientes.

3. Modelo de Monetização

O sistema será SaaS por assinatura mensal, cobrando o profissional/estabelecimento. Basear-nos-emos nos valores de mercado: por exemplo, o **GlossGenius** cobra cerca de US\$48/mês ⁸, o **Vagaro** US\$120/mês ⁹, e o **Fresha** tem plano grátis (com comissão nas transações) ou planos a partir de US\$9,95 por colaborador ¹⁰. Nosso plano inicial poderia ser algo competitivo entre US\$30–50/mês, com opções para versões avançadas (e sem taxas de transação extras). Para contratos corporativos (cadeias de salões), podem haver preços diferenciados. A receita virá das assinaturas mensais; podemos também oferecer um pequeno markup em transações (como a StyleSeat fez para criar nova fonte de receita ²), mas o foco será plano fixo. O Stripe Billing será usado para gerenciar essas assinaturas recorrentes de clientes (profissionais), simplificando cobranças mensais ¹¹. Isso garante receita previsível para a plataforma e alinhamento com modelos de concorrentes.

4. Estratégia de Aquisição de Clientes

A aquisição usará marketing digital e parcerias. Entre as estratégias: *índice de presença online*: SEO e marketing de conteúdo em nichos de beleza; *mídias sociais*: anúncios e divulgação em Instagram/YouTube (onde tendências de beleza têm grande influência) ¹² ⁶ ; *parcerias locais*: colaborações com sindicatos de beleza, fornecedores de produtos ou escolas de estética para indicar a plataforma. Oferecer trials grátis ou descontos iniciais também estimula experimentação. Como **exemplo de mercado**, 75% dos profissionais de beleza são mulheres e metade são pessoas de cor ⁷ , apontando para campanhas segmentadas. Podemos ainda adotar o modelo marketplace (como o StyleSeat usa) para conectar profissionais a clientes, aumentando nosso alcance viral. O objetivo é crescer via *network effect*: cada novo estúdio traz visibilidade à plataforma.

5. Concorrência e Diferenciais

Concorrentes diretos incluem softwares de gestão e agendamento para salões, como **Fresha**, **Vagaro**, **GlossGenius**, **Mindbody** etc. Cada um tem modelo próprio:

- **Fresha**: oferece plano básico grátis e cobra 20% em reservas novas ¹⁰ . Custos baixos, mas menos receita previsível e comissão alta.
- **GlossGenius**: foca em freelancers e pequenos salões, plano único cerca de **US\$48/mês** ⁸ , interface premium (formulários inteligentes, marketing integrado).
- **Vagaro**: solução completa, robusta, cobra cerca de **US\$120/mês** por salão ⁹ , com site próprio e marketplace de clientes.
- **Mindbody**: voltada para fitness e bem-estar, muito usada por academias e spas. Tem enorme base (2,8M de usuários e 40k negócios ¹³), mas costuma ser cara e complexa.

Nosso diferencial será **foco nichado e integração completa com Stripe**. Usaremos Stripe Connect Standard para **gestão legal dos pagamentos**: o profissional abre sua conta Stripe e recebe direto, enquanto nossa plataforma só orquestra via API ³ . Isso elimina a necessidade de sermos “money transmitter”: o Stripe tokeniza os dados de cartão (garantindo PCI) e cuida de KYC/AML e licenças nos EUA ¹ . Teremos interface simples e visual amigável, suportando agendamentos recorrentes e relatórios analíticos desde o MVP. A transparência de custo será um atrativo: sem taxas ocultas (Stripe é claro no pricing), diferentemente de concorrentes que cobram comissão. Além disso, podemos incluir funcionalidades exclusivas (por exemplo, lembretes automáticos, integração com Google Calendar, upselling de produtos, e relatórios de performance), aproveitando o know-how específico do mercado de estética. Em resumo, competimos com preços e usabilidade, mas nos destacamos por facilidade de **onboarding (Stripe-hosted)**, **compliance embutido** e soluções de gestão completas.

6. Projeções de Receita

Para estimar receita, consideramos o tamanho do mercado e preços pretendidos. Com ~979.000 salões nos EUA ⁴ , mesmo uma pequena penetração gera bons números. Por exemplo, 0,1% do mercado (~979 salões) pagando US\$40/mês cada resultaria em cerca de **US\$468.000/ano**. Se chegarmos a 0,5% do mercado (~4.900 clientes), a receita anual seria ~US\$2,35 milhões. Em cenários de **crescimento moderado**, esperamos algo como:

- **Ano 1** (MVP e early adopters): 100–300 clientes · US\$30–50/mês ⇒ ~US\$36k–180k no ano.

- **Ano 2:** 1.000–2.000 clientes · US\$30–50/mês ⇒ ~US\$360k–1,2M/ano.
- **Ano 3:** 5.000+ clientes (expansão vertical) · US\$30–50/mês ⇒ US\$1,8–3,6M/ano.

Essas projeções incluem churn de ~5-10% (típico de SaaS), upsells e planos diferenciados. Também podemos faturar com serviços extras (e.g. desenvolvimento de sites, marketing digital integrado) e parcerias. Em comparação, plataformas como Mindbody faturaram bilhões de dólares em volume de transações ¹⁴, o que indica que a demanda é grande. Assim, a receita baseada em assinatura é realista e escalável à medida que a base de usuários cresce.

7. Roadmap de MVP e Expansão

- **MVP (6-9 meses):** construir o mínimo viável com agendamento básico (cadastro de serviços e horários), cadastro de usuário/profissional, painel de controle simples e integração Stripe Connect. As funcionalidades críticas são: criar conta Stripe Standard via API, gerar *Account Link* para onboarding ³, permitir ao cliente agendar e pagar via Stripe (PaymentIntent) e ao profissional ver/gerenciar seus compromissos. O backend .NET e frontend inicial (React ou Blazor) serão lançados em beta para um grupo de salões parceiros.
- **Versão 1.0 (12 meses):** após feedback, aprimorar interface (mobile-friendly), adicionar relatórios financeiros (exportação de receita, lista de clientes), email/SMS automáticos (lembretes de agendamento). Implementar agendamento recorrente (ex.: séries de atendimentos semanais) e assinaturas mensais no Stripe Billing para cobrar o uso do sistema pelos salões ¹⁵ ¹¹. Incluir painel administrativo para monitorar atividade geral da plataforma.
- **Expansões (após 1-2 anos):** lançar funcionalidades avançadas (ex.: integração com Google/Microsoft Calendar, portal do cliente para reagendamento, lembretes por app móvel). Expandir para **barbearias** (customizar tipo de serviço: cortes, barba, grooming masculino) e **academias/estúdios de fitness** (classes em grupo, aulas particulares). Integrar mecanismos de marketing básico (vendas de créditos ou pacotes de serviços, promoções sazonais). Adicionar suporte multilíngue (inglês/português/espanhol) e integração com redes sociais para reservas diretas. Em paralelo, escalar a infraestrutura na nuvem para suportar milhares de usuários e explorar soluções do Stripe (como *Instant Payouts* e análise avançada via Sigma) para enriquecer o serviço.

8. Arquitetura Técnica

- **Backend (.NET/ASP.NET Core):** Usaremos **ASP.NET Core** para garantir alta performance e escalabilidade. O sistema seguirá uma arquitetura em camadas (por exemplo, *UI* → *BLL* → *DAL* ou *Clean/Onion*), separando apresentação, regras de negócio e acesso a dados ¹⁶. Isso facilita manutenção e testes. O backend proverá APIs REST seguras para o frontend. ORM: **Entity Framework Core** com o provedor Npgsql para PostgreSQL (banco relacional) ¹⁷. O .NET tem ótima compatibilidade com PostgreSQL (o banco relacional mais popular segundo pesquisa StackOverflow ¹⁷), permitindo modelar agendamentos, usuários, serviços e transações.
- **Frontend (React ou Blazor):** O front-end será uma SPA em **React** (JavaScript/TypeScript) ou **Blazor** (C#), para uma UI dinâmica e responsiva. Oferecerá painéis para profissionais (agenda, finanças, configurações) e para clientes (agendamento de horário). Poderá usar Stripe.js/Elements para coletar dados de cartão sem expô-los ao servidor (acompanhando práticas PCI).
- **Banco de Dados Relacional (PostgreSQL):** Usaremos PostgreSQL para armazenar informações estruturadas (usuários, agendamentos, serviços, histórico de pagamentos, configurações). PostgreSQL é maduro, confiável e se integra bem com .NET EF Core ¹⁷. Dados sensíveis (e.g.

informações pessoais) serão criptografados em repouso, e senhas jamais serão armazenadas em texto (hash salted). Não armazenaremos dados de pagamento (PAN, CVV) — estes ficarão tokenizados no Stripe.

- **Stripe Connect Standard – Onboarding e Pagamentos:** Cada profissional abre/associa sua conta Stripe Standard via nossa plataforma. Implementaremos o fluxo sugerido pela Stripe: chame a API `/v1/accounts` com `type=standard` para criar a conta, depois use o *Account Links API* para gerar uma URL de onboarding ³. O profissional será redirecionado ao formulário Stripe-hosted para verificar identidade (KYC) e completar o cadastro. Para pagamentos, usaremos o modelo de *PaymentIntents* do Stripe: ao confirmar um agendamento, criaremos um *PaymentIntent* associado ao cliente e usaremos `transfer_data[destination]` ou `on_behalf_of` para direcionar fundos à conta conectada do profissional. Assim, o cliente paga pelo serviço e o montante (menos taxa) vai direto para o profissional. Para faturamento da própria plataforma (assinatura mensal), utilizaremos **Stripe Billing** para criar assinaturas e cobranças recorrentes dos clientes (salões) ¹¹ ¹⁵.
- **API REST e Autenticação (JWT):** Toda comunicação cliente-servidor será via APIs REST em JSON. Para autenticação, usaremos tokens JWT Bearer. Em .NET Core, isso é padrão para APIs seguras ¹⁸. Ao logar, o usuário (profissional ou admin) recebe um JWT assinado que incluirá claims (ID, role). Esse token acompanha todas as requisições. O servidor valida o token em cada requisição (pelo `JwtBearerHandler` ¹⁸). As permissões (roles de administrador x profissional) são controladas com base nos claims do JWT.
- **Controle de Usuários e Admin:** Haverá dois perfis principais: *profissional* (salão/esteticista) e *cliente final* (opcional, se permitirmos reservas com conta). Além disso, o sistema terá contas de *administrador* para gerenciar assinaturas, planos e visualizar métricas globais. Painéis administrativos permitirão supervisionar usuários, resolver problemas e configurar parâmetros do sistema (e.g., taxas, templates de e-mail).
- **Fluxos de Agendamento, Pagamento, Relatórios e Recorrência:**
 - *Agendamento:* o cliente visualiza horários disponíveis (baseados na agenda do profissional) e escolhe serviço e data/hora. O backend registra o agendamento no DB e envia confirmação.
 - *Pagamento:* após o agendamento (ou no momento do agendamento, dependendo da política), o sistema cria um *PaymentIntent* no Stripe e coleta pagamento via cartão. O valor é dividido entre plataforma e profissional conforme definido (possível *application_fee* para comissão). Stripe lida com captura/transferência automática.
 - *Relatórios:* usaremos relatórios nativos do banco ou do Stripe Sigma. É possível gerar dashboards com total faturado por período, quantidade de agendamentos, cancelamentos etc. O próprio Stripe Sigma (consulta SQL) pode ser integrado para análise de receita e fraude, ou extraímos dados do nosso banco.
 - *Recorrência:* para clientes que desejam agendamentos periódicos (e.g., consulta mensal de pele), implementaremos agendamentos recorrentes que se repetem automaticamente. Tecnicamente, usaríamos jobs agendados (ex.: Hangfire no .NET) para criar novos compromissos conforme recorrência. Para recorrência financeira, manteremos assinaturas ativas no Stripe para cobranças mensais automáticas aos profissionais (uso da plataforma) ¹¹ ¹⁵.
- **Segurança e Conformidade (PCI/Dados Sensíveis):** Segurança é prioridade. Todas as transações de cartão usarão tokenização Stripe (via Stripe.js/Checkout) de forma que nenhum dado de cartão seja armazenado em nossos servidores ¹. A Stripe “tokenizes card data to help with PCI compliance” ¹, portanto a maior parte do escopo PCI é transferido para a Stripe. Usaremos HTTPS em todas as conexões, validação de entrada no backend e proteção contra ataques comuns (SQL injection, XSS, CSRF). Os JWT serão assinados com chave secreta forte e expirados periodicamente.

Para dados pessoais, aplicaremos criptografia em repouso (ex.: TDE no PostgreSQL) e seguiremos padrões GDPR/CCPA quanto a privacidade. Além disso, como plataforma de pagamentos, será necessário observar regulações (usamos o Connect Standard justamente para isso – a Stripe já detém licenças como MTL nos EUA ¹, aliviando nossa responsabilidade). Em suma, nossa arquitetura é **PCI-aware**: usamos as bibliotecas oficiais Stripe e boas práticas de segurança recomendadas ¹ ¹⁸, sem armazenar informações sensíveis localmente.

Todas essas camadas (frontend, backend, Stripe e BD) serão integradas de forma modular, permitindo futuras extensões (novos serviços, internacionalização, etc.) sem ruptura. Com essa arquitetura, atendemos às demandas do mercado de estética americano, garantindo estabilidade, segurança e escalabilidade na plataforma.

Fontes: Dados de mercado e estatísticas ⁵ ⁶ ⁴; exemplos de concorrentes e preços ⁸ ⁹ ¹⁰; casos de uso do Stripe Connect e Stripe Billing ² ¹⁵ ³ ¹¹; melhores práticas técnicas ASP.NET e Stripe ¹⁸ ¹⁷ ¹.

¹ Stripe Connect | Platform and Marketplace Payment Solutions

<https://stripe.com/gb/connect>

² ⁷ ¹⁴ ¹⁵ StyleSeat partners with Stripe to double revenue for beauty professionals

<https://stripe.com/gb/customers/styleseat>

³ docs.stripe.com

<https://docs.stripe.com/connect/standard-accounts>

⁴ 82 Beauty Industry Statistics + Trend Forecasts [2023 Data]

<https://www.styleseat.com/blog/beauty-industry-statistics/>

⁵ ¹² United States \$95+ Bn Beauty Salon Markets, 2025-2033 by

<https://www.globenewswire.com/news-release/2025/12/03/3199123/0/en/United-States-95-Bn-Beauty-Salon-Markets-2025-2033-by-Service-Type-End-user-States-and-Company-Analysis.html>

⁶ SaaS for Spa Management Market Growth & Forecast [2030]

<https://www.fortunebusinessinsights.com/saas-for-spa-management-market-108584>

⁸ ⁹ ¹⁰ 9 Best Salon Software 2025: The Ultimate Guide

<https://thesalonbusiness.com/best-salon-software/>

¹¹ Stripe Billing | Recurring Payments & Subscription Solutions

<https://stripe.com/ie/billing>

¹³ Best Fitness & Wellness Management Software | Mindbody

<https://www.mindbodyonline.com/>

¹⁶ Common web application architectures - .NET | Microsoft Learn

<https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>

¹⁷ Integrating PostgreSQL with .NET 9 Using EF Core: A Step-by-Step Guide | by Egor Tarasov | Medium

<https://medium.com/@vosarat1995/integrating-postgresql-with-net-9-using-ef-core-a-step-by-step-guide-a773768777f2>

¹⁸ Configure JWT bearer authentication in ASP.NET Core | Microsoft Learn

<https://learn.microsoft.com/en-us/aspnet/core/security/authentication/configure-jwt-bearer-authentication?view=aspnetcore-10.0>