**📋 Resumo do Projeto: Sistema de Agendamento para Salão de Beleza**

**🎯 Objetivo do Projeto**

Desenvolver um sistema completo de agendamento para salão de beleza, permitindo que clientes marquem horários e administradores gerenciem os agendamentos.

**✅ O QUE JÁ FOI IMPLEMENTADO**

**🔧 Backend (Python/FastAPI) - COMPLETO**

* **✅ API REST completa** com endpoints para:
  + CRUD de Clientes (/clientes/)
  + CRUD de Agendamentos (/agendamentos/)
  + Filtros por data e cliente
  + Validações de negócio (não agendar no passado, verificar conflitos)
* **✅ Sistema de Autenticação**:
  + Registro e login de usuários
  + JWT tokens para segurança
  + Diferentes níveis de acesso (user, staff, admin)
  + Hash de senhas com bcrypt
* **✅ Banco de Dados**:
  + Modelos com SQLModel (Cliente, Agendamento, User)
  + SQLite com migrations automáticas
  + Relacionamentos entre tabelas
* **✅ Qualidade de Código**:
  + Testes automatizados com pytest
  + CI/CD com GitHub Actions
  + Formatação com Black e isort
  + Linting com flake8

**🎨 Frontend (React) - PARCIALMENTE COMPLETO**

* **✅ Configuração Inicial**:
  + Projeto React com Vite
  + ESLint, Prettier, React Router configurados
  + Estrutura de pastas organizada
* **✅ Componentes de UI**:
  + Header, Footer, Loading, Modal
  + Sistema de formulários reutilizável
  + Estilização responsiva com CSS
* **✅ Página de Agendamentos**:
  + Listagem completa de agendamentos
  + Filtros por data e serviço
  + Paginação e ordenação
  + Integração com API backend

**🚧 O QUE AINDA PRECISA SER FEITO**

**🔥 PRIORIDADE ALTA (Próximos Passos)**

**Issue #16**: Implementar JWT Authentication

**Descrição:**  
Configurar autenticação JWT no backend.

**Tarefas:**

* Configurar JWT com Python
* Criar middleware de autenticação
* Proteger endpoints existentes
* Implementar refresh tokens

**Critérios de Aceitação:**

* Tokens sendo gerados corretamente
* Middleware validando tokens
* Endpoints protegidos adequadamente
* Sistema de refresh implementado

**Issue #17**: Login e Registro no Frontend

**Descrição:**  
Implementar interfaces de login e registro no frontend.

**Tarefas:**

* Criar página de login
* Criar página de registro
* Implementar gestão de estado de autenticação
* Adicionar proteção de rotas no frontend

**Critérios de Aceitação:**

* Páginas de login e registro funcionando
* Estado de autenticação persistido
* Rotas protegidas corretamente
* Logout implementado
* Interface mais intuitiva

**Issue #18**: Sistema de Notificações

**Descrição:**  
Implementar sistema de notificações por email.

**Tarefas:**

* Configurar serviço de email (SendGrid/Mailgun)
* Criar templates de email
* Implementar notificação de novo agendamento
* Implementar lembretes de agendamento

**Critérios de Aceitação:**

* Emails sendo enviados corretamente
* Templates visualmente adequados
* Sistema de fila para emails implementado
* Testes para notificações implementados

**Issue #19**: Relatórios e Estatísticas

**Descrição:**  
Criar sistema de relatórios e dashboard administrativo.

**Tarefas:**

* Criar endpoints para estatísticas
* Implementar dashboard no frontend
* Adicionar gráficos e métricas
* Criar relatórios exportáveis

**Critérios de Aceitação:**

* Dashboard com métricas principais
* Gráficos funcionando com dados reais
* Exportação de relatórios em PDF/CSV
* Filtros por período funcionando

**Issue #20: Dockerização da Aplicação**

**Descrição:**  
Criar Dockerfiles e docker-compose para a aplicação completa.

**Tarefas:**

* Dockerfile para backend
* Dockerfile para frontend
* docker-compose para desenvolvimento
* docker-compose para produção

**Critérios de Aceitação:**

* Containers buildando sem erros
* Aplicação funcionando em containers
* Variáveis de ambiente configuradas
* Documentação de uso do Docker

**Issue #21: Configuração de Produção**

**Descrição:**  
Configurar ambiente de produção e deploy automatizado.

**Tarefas:**

* Configurar variáveis de ambiente de produção
* Configurar database de produção (PostgreSQL)
* Implementar CI/CD para deploy automático
* Configurar monitoramento (Sentry, Logging)

**Critérios de Aceitação:**

* Aplicação rodando em ambiente de produção
* Deploy automatizado funcionando
* Monitoramento configurado
* Backup de database configurado

**Issue #22: Otimizações de Performance**

**Descrição:**  
Otimizar performance do frontend e backend.

**Tarefas:**

* Implementar cache no backend
* Otimizar queries do banco de dados
* Implementar lazy loading no frontend
* Otimizar bundle do frontend

**Critérios de Aceitação:**

* Tempo de resposta do backend melhorado
* Bundle size do frontend reduzido
* Queries otimizadas e indexadas
* Métricas de performance coletadas

**Issue #23: Sistema de Avaliações**

**Descrição:**  
Implementar sistema de avaliações para os serviços.

**Tarefas:**

* Criar modelo de Avaliação
* Implementar endpoints para avaliações
* Criar interface de avaliação no frontend
* Adicionar ratings aos serviços

**Critérios de Aceitação:**

* Clientes podem avaliar serviços
* Ratings sendo calculados e exibidos
* Interface intuitiva para avaliações
* Moderação de avaliações implementada

**Issue #24: Integração com Calendar APIs**

**Descrição:**  
Integrar com APIs de calendário (Google Calendar, Outlook).

**Tarefas:**

* Implementar OAuth para Google Calendar
* Implementar sincronização com calendários
* Criar interface de configuração
* Implementar webhooks para sincronização

**Critérios de Aceitação:**

* Sincronização com Google Calendar funcionando
* Interface de configuração intuitiva
* Sincronização bidirecional implementada
* Tratamento de conflitos de agendamento

**📋 PRIORIDADE MÉDIA**

**4. Funcionalidades Avançadas**

* Dashboard com estatísticas
* Relatórios de agendamentos
* Sistema de notificações por email
* Calendário visual

**5. Sistema de Permissões**

* Diferentes views para admin/staff/user
* Controle de acesso granular
* Auditoria de ações

**🔮 PRIORIDADE BAIXA (Futuro)**

**6. Features Avançadas**

* Integração com Google Calendar
* Sistema de avaliações
* Múltiplos serviços e funcionários
* Pagamentos online

**7. Deploy e Produção**

* Dockerização
* Deploy em cloud (Railway, Render)
* Domínio próprio e SSL
* Backup automático

**🏗️ ARQUITETURA ATUAL**

Tela de computador com letras brancas em fundo preto

O conteúdo gerado por IA pode estar incorreto. Tela de computador com letras e números em fundo preto

O conteúdo gerado por IA pode estar incorreto. Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

O conteúdo gerado por IA pode estar incorreto.

**🔄 ESTADO ATUAL DO SISTEMA**

**✅ Funcionalidades Prontas para Uso**

1. **API Backend**: Totalmente funcional com autenticação
2. **Banco de Dados**: Modelos e relações implementados
3. **Listagem de Agendamentos**: Frontend consome API corretamente
4. **CI/CD**: Testes automatizados no GitHub

**🚧 Funcionalidades em Desenvolvimento**

1. **Interface de Usuário**: Básica funcionando, precisa de melhorias
2. **Autenticação Frontend**: Backend pronto, frontend pendente

**📊 Progresso Geral**

* **Backend**: 95% completo (produção ready)
* **Frontend**: 60% completo (necessita autenticação e CRUD)
* **Integração**: 80% completa (API comunicação ok)

**🎯 PRÓXIMOS PASSOS IMEDIATOS (Sugestão)**

**Issue #16**: Implementar JWT Authentication

**Issue #17**: [**Login e Registro no Frontend**](https://github.com/FlavioBuenoDev/projeto-salao/issues/18)

**Issue #18**[**Issue #18: Sistema de Notificações**](https://github.com/FlavioBuenoDev/projeto-salao/issues/19)   
**Issue #19**: [**Relatórios e Estatísticas**](https://github.com/FlavioBuenoDev/projeto-salao/issues/20)

[**Issue #20: Dockerização da Aplicaçã**](https://github.com/FlavioBuenoDev/projeto-salao/issues/21)o  
[**Issue #21: Configuração de Produção**](https://github.com/FlavioBuenoDev/projeto-salao/issues/22)  
[**Issue #22: Otimizações de Performance**](https://github.com/FlavioBuenoDev/projeto-salao/issues/23)

[**Issue #23: Sistema de Avaliações**](https://github.com/FlavioBuenoDev/projeto-salao/issues/24)  
[**Issue #24: Integração com Calendar APIs**](https://github.com/FlavioBuenoDev/projeto-salao/issues/25)

**💡 PONTOS FORTES DO PROJETO**

* ✅ **Código bem estruturado** e organizado
* ✅ **Testes automatizados** garantindo qualidade
* ✅ **Documentação completa** das APIs
* ✅ **Interface responsiva** e moderna
* ✅ **Sistema de autenticação** seguro

**⚠️ PONTOS DE ATENÇÃO**

* 🔄 Frontend precisa de autenticação para ser utilizável
* 🔄 Formulários de criação/edição ainda não implementados
* 🔄 Sistema em desenvolvimento, não pronto para produção

**🎉 CONCLUSÃO**

O projeto está **bem encaminhado** com uma base sólida no backend e uma estrutura promissora no frontend. As funcionalidades core estão implementadas, faltando principalmente a parte de interação do usuário final (formulários e autenticação frontend).

**Próxima meta crítica**: Ter um MVP (Minimum Viable Product) funcional onde usuários possam fazer login e gerenciar agendamentos através da interface web.

O projeto demonstra boas práticas de desenvolvimento e está pronto para evoluir para as próximas fases!