



Especificação Técnica

App de Marketplace de Serviços



Índice




- | | |
|------------------------|------------------------|
| 1. Visão Geral | 5. Banco de Dados |
| 2. Fluxo do Usuário | 6. Endpoints da API |
| 3. Wireframes | 7. Estrutura de Pastas |
| 4. Páginas do Frontend | 8. Implementação |

1. Visão Geral do Projeto

App marketplace de serviços onde **usuários** solicitam serviços e **prestadores** aceitam e executam. O pagamento é intermediado pelo app e só é liberado após confirmação com fotos do serviço concluído.

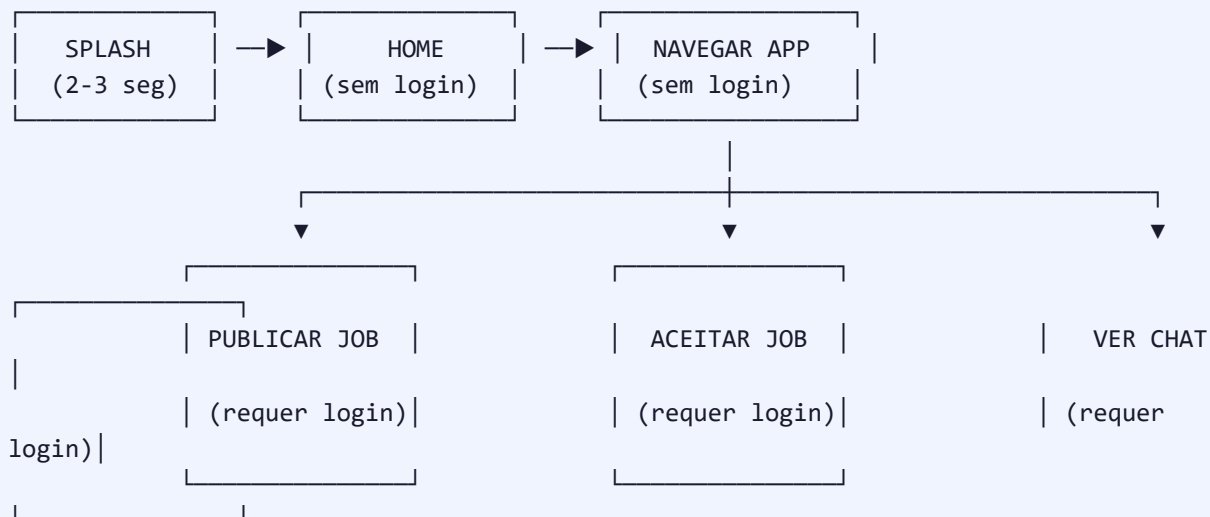
Funcionalidades Principais:

- ✓ Cadastro de clientes e prestadores de serviço
- ✓ Publicação de pedidos de serviço com fotos e requisitos
- ✓ Sistema de propostas e aceite
- ✓ Chat interno com notificações em tempo real
- ✓ Pagamento intermediado pelo app
- ✓ Aprovação de conclusão com fotos
- ✓ Sistema de avaliações mútuas

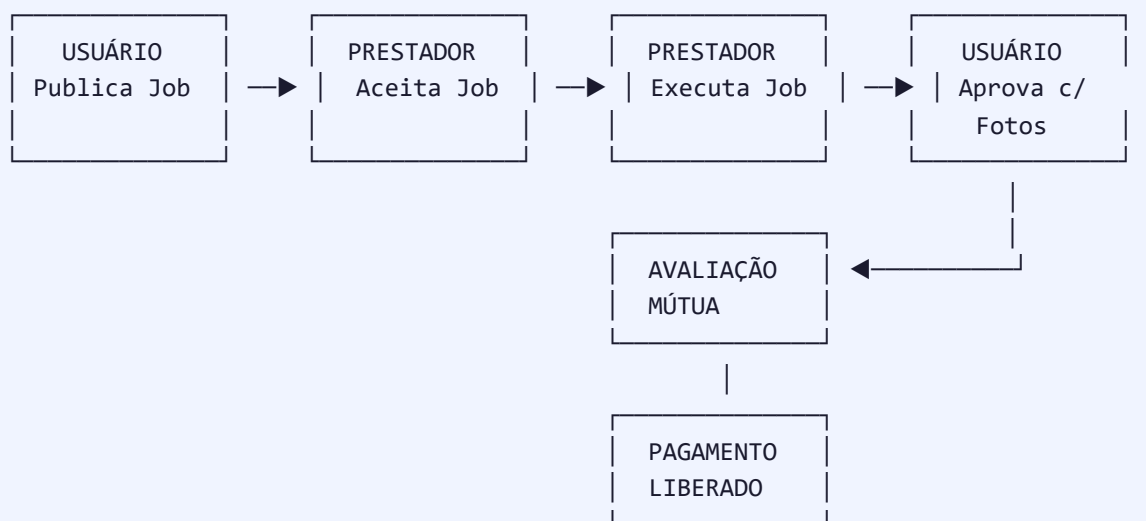
-  Ordenação de jobs por rating (melhores prestadores veem jobs mais caros)
-  Portfólio de trabalhos realizados
-  Carteira digital para prestadores

2. Fluxo do Usuário

2.1 Fluxo Principal de Navegação

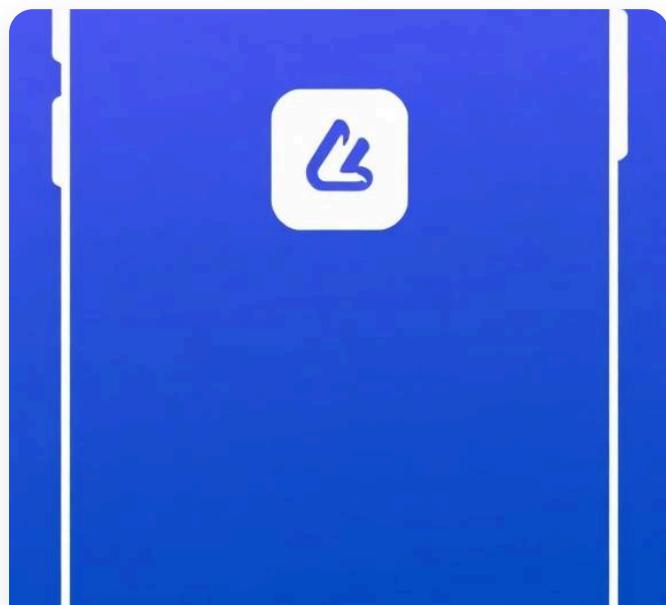


2.2 Fluxo do Serviço

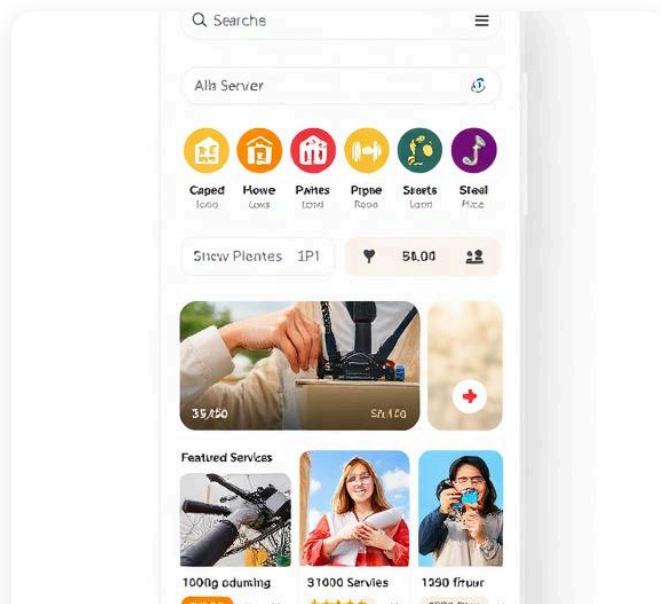


3. Wireframes das Telas

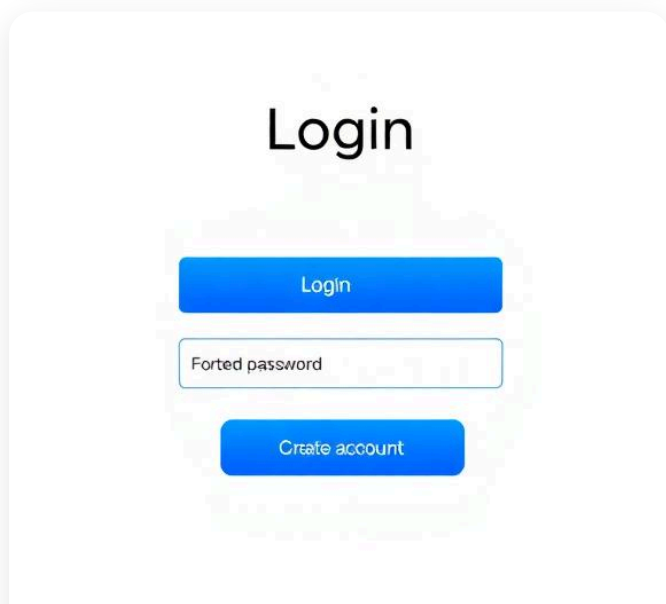
Abaixo estão os wireframes das principais telas do aplicativo:



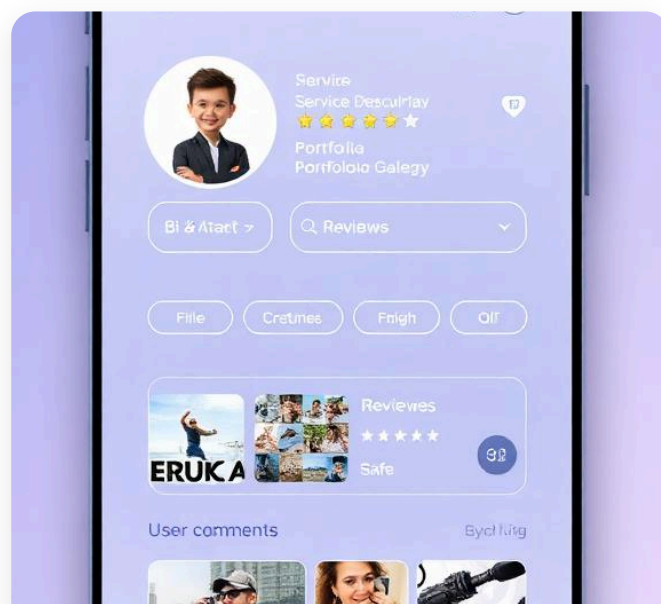
01 - Splash Screen



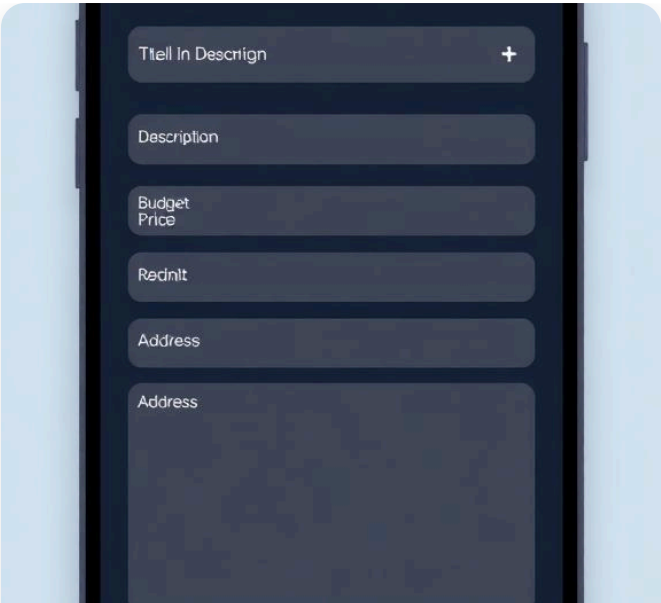
02 - Home



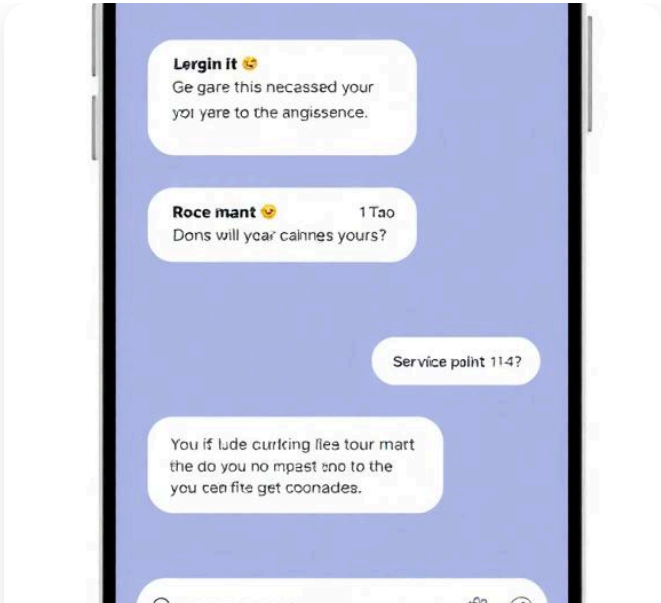
03 - Login



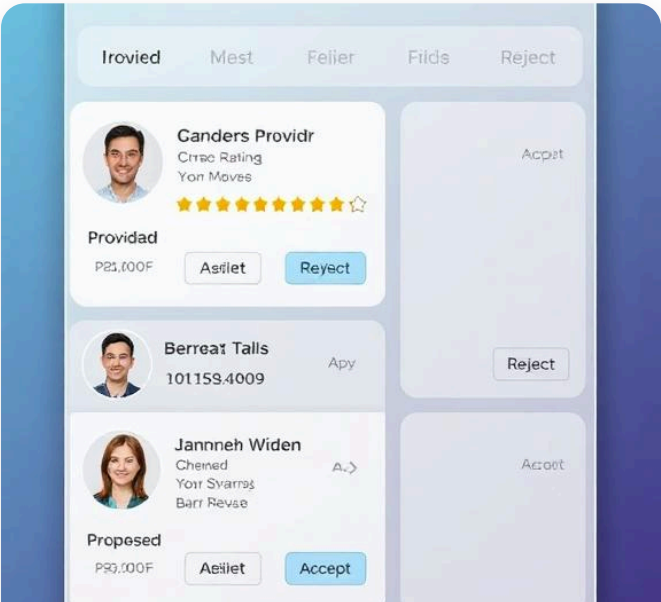
04 - Perfil do Prestador



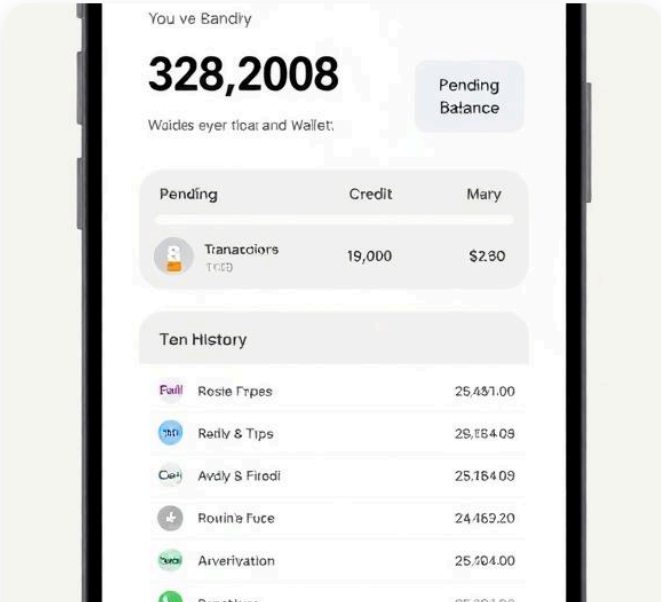
05 - Criar Novo Serviço



06 - Chat



07 - Lista de Propostas



08 - Carteira Digital

4. Páginas do Frontend

4.1 Páginas Públicas (Sem Login)

Página	Rota	Arquivo	Descrição
Splash Screen	/splash	SplashScreen.tsx	Tela inicial com logo e loading
Home	/	Home.tsx	Categorias e serviços em destaque
Busca	/search	Search.tsx	Busca com filtros
Detalhes Prestador	/provider/:id	ProviderDetails.tsx	Perfil público do prestador
Detalhes Job	/job/:id	JobDetails.tsx	Detalhes do serviço
Categorias	/categories	Categories.tsx	Lista de categorias

4.2 Páginas de Autenticação

Página	Rota	Arquivo
Login	/auth/login	auth/Login.tsx
Registro	/auth/register	auth/Register.tsx
Recuperar Senha	/auth/forgot-password	auth/ForgotPassword.tsx

4.3 Páginas do Cliente

Página	Rota	Descrição
Publicar Serviço	/client/new-job	Formulário para criar pedido
Meus Pedidos	/client/my-jobs	Lista de serviços publicados

Propostas Recebidas	/client/job/:id/proposals	Ver e aceitar propostas
Acompanhar Serviço	/client/job/:id/tracking	Status do serviço em andamento
Aprovar Conclusão	/client/job/:id/approve	Verificar fotos e liberar pagamento
Avaliar Prestador	/client/job/:id/review	Formulário de avaliação

4.4 Páginas do Prestador


Página	Rota	Descrição
Dashboard	/provider/dashboard	Visão geral e estatísticas
Jobs Disponíveis	/provider/available-jobs	Serviços para aceitar (ordenados por rating)
Meus Serviços	/provider/my-jobs	Serviços aceitos
Executar Serviço	/provider/job/:id/execute	Chat e controle do serviço
Concluir Serviço	/provider/job/:id/complete	Upload de fotos de conclusão
Portfólio	/provider/portfolio	Galeria de trabalhos
Minhas Avaliações	/provider/reviews	Reviews recebidos

4.5 Páginas Compartilhadas

Página	Rota	Descrição
Chat	/chat	Lista de conversas
Conversa	/chat/:id	Chat individual
Notificações	/notifications	Central de notificações
Perfil	/profile	Dados pessoais
Carteira	/wallet	Saldo e transações

Configurações	/settings	Preferências do app
---------------	-----------	---------------------

5. Banco de Dados - Tabelas

 **Dica:** Todas as tabelas usam UUID como chave primária e incluem timestamps de criação.

5.1 profiles

Coluna	Tipo	Descrição
id	UUID (PK)	ID único
user_id	UUID (FK → auth.users)	Referência ao usuário
name	VARCHAR(255)	Nome completo
phone	VARCHAR(20)	Telefone
cpf	VARCHAR(14)	CPF único
avatar_url	TEXT	URL da foto
bio	TEXT	Descrição
user_type	ENUM('client', 'provider')	Tipo de usuário
is_verified	BOOLEAN	Verificado?
created_at	TIMESTAMPTZ	Data de criação

5.2 categories

Coluna	Tipo	Descrição
id	UUID (PK)	ID único
name	VARCHAR(100)	Nome da categoria

slug	VARCHAR(100)	Slug para URL
description	TEXT	Descrição
icon	VARCHAR(50)	Nome do ícone
image_url	TEXT	Imagem da categoria
parent_id	UUID (FK)	Categoria pai
is_active	BOOLEAN	Ativa?

5.3 jobs

Coluna	Tipo	Descrição
id	UUID (PK)	ID único
client_id	UUID (FK → profiles)	Cliente que publicou
provider_id	UUID (FK → profiles)	Prestador aceito
category_id	UUID (FK → categories)	Categoria
title	VARCHAR(255)	Título do serviço
description	TEXT	Descrição detalhada
budget	DECIMAL(10,2)	Orçamento oferecido
status	ENUM	open, in_progress, completed, etc.
images	TEXT[]	Array de URLs de imagens
requirements	TEXT[]	Array de requisitos
address_*	VARCHAR	Campos de endereço
latitude, longitude	DECIMAL	Coordenadas
desired_date	DATE	Data desejada

completion_photos	TEXT[]	Fotos de conclusão
-------------------	--------	--------------------

5.4 proposals

Coluna	Tipo	Descrição
id	UUID (PK)	ID único
job_id	UUID (FK → jobs)	Job relacionado
provider_id	UUID (FK → profiles)	Prestador
message	TEXT	Mensagem da proposta
proposed_price	DECIMAL(10,2)	Preço proposto
status	ENUM	pending, accepted, rejected

5.5 reviews

Coluna	Tipo	Descrição
id	UUID (PK)	ID único
job_id	UUID (FK → jobs)	Job avaliado
reviewer_id	UUID (FK → profiles)	Quem avaliou
reviewed_id	UUID (FK → profiles)	Quem foi avaliado
rating	INTEGER (1-5)	Nota geral
punctuality	INTEGER (1-5)	Pontualidade
quality	INTEGER (1-5)	Qualidade
communication	INTEGER (1-5)	Comunicação
comment	TEXT	Comentário

would_hire_again	BOOLEAN	Contrataria novamente?
------------------	---------	------------------------

5.6 Outras Tabelas

- **addresses** - Endereços dos usuários
- **provider_categories** - Categorias que o prestador atende (N:N)
- **provider_settings** - Configurações do prestador
- **provider_stats** - Estatísticas calculadas
- **conversations** - Conversas do chat
- **messages** - Mensagens individuais
- **notifications** - Notificações do sistema
- **payments** - Pagamentos dos serviços
- **wallets** - Carteira digital
- **wallet_transactions** - Histórico de transações
- **portfolios** - Fotos do portfólio
- **push_tokens** - Tokens para push notification

6. Endpoints da API

6.1 Autenticação

Método	Endpoint	Descrição
POST	/auth/register	Cadastro de usuário
POST	/auth/login	Login
POST	/auth/logout	Logout
POST	/auth/forgot-password	Recuperar senha
POST	/auth/reset-password	Redefinir senha

6.2 Perfil

Método	Endpoint	Descrição
GET	/profile	Obter perfil do usuário logado
PUT	/profile	Atualizar perfil
POST	/profile/avatar	Upload de foto

6.3 Jobs (Serviços)

Método	Endpoint	Descrição
GET	/jobs	Listar jobs (público)
GET	/jobs/:id	Detalhes do job

POST	/jobs	Criar novo job (cliente)
PUT	/jobs/:id	Atualizar job
DELETE	/jobs/:id	Cancelar job

6.4 Cliente

Método	Endpoint	Descrição
GET	/client/jobs	Meus jobs
GET	/client/jobs/:id/proposals	Propostas recebidas
POST	/client/jobs/:id/accept-proposal	Aceitar proposta
POST	/client/jobs/:id/approve	Aprovar conclusão

6.5 Prestador

Método	Endpoint	Descrição
GET	/provider/available-jobs	Jobs disponíveis (ordenados por rating)
GET	/provider/jobs	Meus serviços
POST	/provider/jobs/:id/propose	Enviar proposta
POST	/provider/jobs/:id/start	Iniciar serviço
POST	/provider/jobs/:id/complete	Marcar como concluído

6.6 Chat

Método	Endpoint	Descrição
--------	----------	-----------

GET	/conversations	Listar conversas
GET	/conversations/:id/messages	Mensagens da conversa
POST	/conversations/:id/messages	Enviar mensagem
POST	/conversations/:id/read	Marcar como lido

6.7 Avaliações

Método	Endpoint	Descrição
POST	/reviews	Criar avaliação
GET	/reviews/provider/:id	Avaliações do prestador
GET	/reviews/client/:id	Avaliações do cliente

6.8 Carteira

Método	Endpoint	Descrição
GET	/wallet	Saldo e transações
POST	/wallet/withdraw	Solicitar saque
POST	/payments/create	Criar pagamento

7. Estrutura de Pastas

```
src/ ├── assets/ | ├── images/ | └── icons/ | ├── components/ | ├── ui/ #
Componentes shadcn | ├── common/ # Header, BottomNav, Avatar, Rating... | ├──
forms/ # AddressForm, JobForm, ReviewForm... | ├── cards/ # JobCard,
ProviderCard, ProposalCard... | ├── modals/ # ConfirmModal, FilterModal... |
└── chat/ # MessageBubble, MessageInput... | ├── pages/ | ├── SplashScreen.tsx
| ├── Home.tsx | ├── Search.tsx | ├── auth/ # Login, Register, ForgotPassword
| ├── client/ # NewJob, MyJobs, Proposals, Approve... | ├── provider/ #
Dashboard, AvailableJobs, Portfolio... | ├── Chat.tsx | ├── Notifications.tsx
| ├── Profile.tsx | └── Wallet.tsx | ├── hooks/ # useAuth, useJobs, useChat...
| ├── contexts/ # AuthContext, ChatContext... | ├── services/ # API calls | ├── lib/
# Utils, formatters, validators | ├── types/ # TypeScript types | └──
integrations/ └── supabase/ supabase/ | ├── config.toml | ├── migrations/ # SQL
migrations | └── functions/ # Edge functions
```


8. Implementação Passo a Passo



Fase 1: Setup Inicial (Semana 1)

- ☐ Criar projeto no Lovable
- ☐ Ativar Lovable Cloud (Supabase)
- ☐ Configurar design system (cores, fontes)
- ☐ Criar tabelas: profiles, categories, addresses
- ☐ Implementar autenticação (Login, Registro)



Fase 2: Estrutura Base (Semana 2)

- ☐ Criar Header e BottomNav
- ☐ Implementar SplashScreen
- ☐ Implementar Home com categorias
- ☐ Criar componentes de cards
- ☐ Implementar estados vazios e loading



Fase 3: Fluxo do Cliente (Semana 3)

- ☐ Criar tabela jobs
- ☐ Implementar formulário de novo job

- ☐ Implementar upload de imagens
- ☐ Criar tabela proposals
- ☐ Implementar visualização de propostas
- ☐ Implementar aceitar proposta



Fase 4: Fluxo do Prestador (Semana 4)

- ☐ Implementar seleção de categorias
- ☐ Implementar configurações de área
- ☐ Implementar listagem de jobs (ordenação por rating)
- ☐ Implementar envio de proposta
- ☐ Implementar marcar início/conclusão
- ☐ Implementar portfólio



Fase 5: Chat e Notificações (Semana 5)

- ☐ Criar tabelas conversations e messages
- ☐ Implementar lista de conversas
- ☐ Implementar chat em tempo real
- ☐ Criar tabela notifications
- ☐ Implementar central de notificações

- ☐ Configurar push notifications



Fase 6: Pagamentos (Semana 6)

- ☐ Criar tabelas payments, wallets, transactions
- ☐ Integrar gateway (Stripe/Mercado Pago)
- ☐ Implementar fluxo de pagamento
- ☐ Implementar carteira do prestador
- ☐ Implementar liberação após aprovação



Fase 7: Avaliações (Semana 7)

- ☐ Criar tabela reviews e provider_stats
- ☐ Implementar formulário de avaliação
- ☐ Criar triggers para atualizar stats
- ☐ Implementar ordenação de jobs por rating



Fase 8: Refinamentos (Semana 8)

- ☐ Revisar todos os fluxos
- ☐ Adicionar animações
- ☐ Testar responsividade

- ☐ Otimizar queries
- ☐ Revisar RLS policies
- ☐ Testes de segurança

Documento de Especificação Técnica

App de Marketplace de Serviços

Gerado automaticamente