Glamo Client Management - Phase 1 Implementation Summary

✅ Completed Components

1. Database Schema (schema.prisma)

- ✅ Extended Client model with 30+ new fields
- ✅ Added ClientTag model for segmentation
- ✅ Added ClientNote model for observations
- ✅ Added ClientDocument model for file management
- ✅ Added ClientHistory model for audit trail
- ✅ Added required enums (Gender, ContactMethod, ClientStatus, ClientType, etc.)
- ✅ Added proper indexes for performance
- ✅ Added User relations for client management

2. Backend Operations (src/clients/operations.ts)

- ✅ Enhanced listClients with advanced filters (status, clientType, tags, search)
- ✅ Enhanced getClient with full includes (tags, notes, documents, history, appointments, sales)
- ✅ Implemented createClient with validation and history tracking
- ✅ Implemented updateClient with field-level history tracking
- ✅ Implemented deleteClient with soft-delete
- ✅ Implemented getClientNotes query
- ✅ Implemented addClientNote action
- ✅ Implemented updateClientNote action
- ✅ Implemented deleteClientNote action
- ✅ Implemented addClientTag action
- ✅ Implemented removeClientTag action
- ✅ Implemented getClientDocuments query
- ✅ Implemented uploadClientDocument action
- ✅ Implemented deleteClientDocument action
- ✅ Implemented getClientHistory query with pagination

3. Wasp Configuration (main.wasp)

- ✅ Added all new queries to main.wasp
- ✅ Added all new actions to main.wasp
- ✅ Added ClientDetailRoute for detail page
- ✅ Configured proper entities for all operations

4. Frontend Components

List Page (ClientsListPage.tsx)

- ✅ Enhanced with filters and advanced search
- ✅ Added stats cards (Total, Active, Inactive, VIP)
- ✅ Improved table with better columns and formatting

- ✅ Added pagination with page numbers
- ✅ Added export button placeholder

Supporting Components

- ✅ ClientFilters - Advanced filter panel with status, type, tags
- ✅ ClientStatsCards - Dashboard stats cards
- ✅ ClientTableRow - Enhanced row with avatar, tags, actions menu
- ✅ Formatters library (date, currency, phone, CPF, CNPJ)

📋 Remaining Tasks for Phase 1 Completion

1. Client Detail Page (HIGH PRIORITY)

Need to create:
- ClientDetailPage.tsx - Main detail page with tabs
- ClientOverviewTab - Key metrics and quick info
- ClientHistoryTab - Timeline of interactions
- ClientNotesTab - Notes management interface
- ClientDocumentsTab - Document gallery
- ClientProfileTab - Full profile information

2. Client Forms (HIGH PRIORITY)

Need to create:
- ClientFormDialog - Create/Edit dialog component
- Multi-step form wizard:
- Step 1: Basic Information (name, email, phone, CPF)
- Step 2: Address & Contact Preferences
- Step 3: Marketing Consent & Additional Info
- Form validation with Zod
- Photo upload component

3. Supporting UI Components

Need to create:
- NoteEditor - Rich text note editor
- TagSelector - Tag selection and creation
- DocumentUploader - File upload with preview
- ClientDeleteDialog - Confirmation dialog
- Loading states and error boundaries

4. Database Migration

Need to run:

```
cd app
wasp db migrate-dev --name add_client_management_phase1
```

5. Testing

- Manual testing of all CRUD operations
- Test filters and search
- Test notes, tags, documents

- Test history tracking
- Verify permissions

🚀 Next Steps to Complete Phase 1

1. **Create ClientDetailPage** - 2-3 hours
   - Tabbed interface
   - Overview tab with quick stats
   - History timeline
   - Notes management
   - Documents gallery

2. **Create Client Forms** - 2-3 hours
   - Multi-step wizard
   - Form validation
   - Photo upload
   - Address autocomplete (optional)

3. **Run Database Migration** - 30 minutes
   - Create migration
   - Test in development
   - Verify data integrity

4. **Testing & Bug Fixes** - 1-2 hours
   - Manual testing
   - Fix any issues
   - Polish UI/UX

5. **Documentation** - 30 minutes
   - Update README
   - Add usage examples
   - Document API endpoints

📊 Phase 1 Completion Status

Overall Progress: ~75%

- ✅ Database Schema: 100%
- ✅ Backend Operations: 100%
- ✅ Wasp Configuration: 100%
- ✅ List Page: 100%
- ⏳ Detail Page: 0% (To be created)
- ⏳ Forms: 0% (To be created)
- ⏳ Migration: 0% (To be run)
- ⏳ Testing: 0% (To be done)

🎯 Critical Files Still Needed

```
app/src/client/modules/clients/
├── ClientDetailPage.tsx (NEEDED)
├── components/
│   ├── ClientFormDialog.tsx (NEEDED)
│   ├── ClientOverviewTab.tsx (NEEDED)
│   ├── ClientHistoryTab.tsx (NEEDED)
│   ├── ClientNotesTab.tsx (NEEDED)
│   ├── ClientDocumentsTab.tsx (NEEDED)
│   ├── NoteEditor.tsx (NEEDED)
│   ├── TagSelector.tsx (NEEDED)
│   └── DocumentUploader.tsx (NEEDED)
```

💡 Implementation Notes

Design Patterns Used

- Modular component architecture
- Proper separation of concerns
- Reusable UI components
- Type-safe operations
- Permission-based access control

Best Practices Followed

- TypeScript for type safety
- Proper error handling
- Loading and empty states
- Responsive design
- Audit trail for all changes
- LGPD compliance (consent tracking)

Performance Optimizations

- Indexed database queries
- Pagination for large lists
- Lazy loading for detail page
- Optimistic UI updates (when implemented)

🔒 Security Considerations

- ✅ Permission checks on all operations
- ✅ Salon-scoped data isolation
- ✅ Soft deletes for data retention
- ✅ Audit logging for compliance
- ✅ Input validation and sanitization

📖 API Documentation

All operations are properly typed and documented in `operations.ts`.
Key features:
- Advanced filtering and search
- Pagination support
- Full CRUD operations
- History tracking
- Tag management
- Document management
- Note management