Pull Request: Clients Module Phase 1 (Core MVP)

🎯 Overview

This PR implements the complete frontend for the Clients Module Phase 1, providing a fully functional CRM interface for managing client profiles and information. The backend was already 100% complete; this PR delivers the UI layer.

📦 What's Included

Components Implemented (6 total)

1. **ClientProfileForm** (`app/components/clients/ClientProfileForm.tsx`)
   - Comprehensive form with validation for creating/editing client profiles
   - All ClientProfile fields organized in sections
   - react-hook-form + Zod validation
   - Responsive design

2. **ClientProfileView** (`app/components/clients/ClientProfileView.tsx`)
   - Read-only display of complete client information
   - Organized sections (Personal, Contact, Professional, Emergency, Notes)
   - Shows profile photo and metadata

3. **ClientHistory** (`app/components/clients/ClientHistory.tsx`)
   - Displays appointment history with status badges
   - Shows service statistics and total spent
   - Fetches from `/api/clients/profiles/[id]/history`

4. **ClientNotesList** (`app/components/clients/ClientNotesList.tsx`)
   - Full CRUD operations for client notes
   - Note type categorization (General, Medical, Preference, Appointment, Feedback)
   - Private note flag support

5. **ClientPreferences** (`app/components/clients/ClientPreferences.tsx`)
   - Configure communication preferences (Email, SMS, WhatsApp, Phone)
   - Set preferred days and time slots
   - Reminder time configuration
   - Language and timezone settings

6. **PhotoUpload** (`app/components/clients/PhotoUpload.tsx`)
   - Upload profile photos with preview
   - File validation (type, size)
   - Remove photo functionality

Pages Implemented (4 total)

1. **Client List Page** (`app/app/dashboard/clients/page.tsx`)
   - Browse all clients with search functionality
   - Stats dashboard (total clients, with email, with phone)

- Client cards with avatar and quick info
- Actions: View, Edit, Delete
- Empty state handling
- Delete confirmation dialog

2. **Client Detail Page** (`app/app/dashboard/clients/[id]/page.tsx`)
   - Tabbed interface:
     - Profile: Complete client information
     - History: Appointment and service history
     - Notes: Client notes management
     - Preferences: Communication settings
     - Edit button for quick navigation
3. **Create Client Page** (`app/app/dashboard/clients/new/page.tsx`)
   - Form to add new clients
   - All fields from ClientProfile schema
   - Success redirect to client detail page
4. **Edit Client Page** (`app/app/dashboard/clients/[id]/edit/page.tsx`)
   - Pre-populated form with existing data
   - Photo upload section
   - Update client information
   - Success redirect to client detail page

Type Definitions

- **New File**: `app/lib/clients/types.ts`
- Complete TypeScript interfaces for all API responses
- Component prop types
- Form data types
- Ensures type safety across the module

✨ Features

Search & Filter

- Real-time client search by name, email, or phone
- Quick stats dashboard

CRUD Operations

- ✅ Create new clients
- ✅ Read/view client profiles
- ✅ Update client information
- ✅ Delete clients with confirmation

Client Management

- Profile photo upload and management
- Comprehensive contact information
- Emergency contact details
- Professional information

- Custom notes with categorization
- Communication preferences

User Experience

- Loading states for all async operations
- Error handling with user-friendly messages
- Success notifications
- Responsive mobile-friendly design
- Empty states with helpful prompts
- Confirmation dialogs for destructive actions

🔌 API Integration

All components integrate with existing backend endpoints:
- `GET /api/clients/profiles` - List clients
- `GET /api/clients/profiles/[id]` - Get client details
- `POST /api/clients/profiles` - Create client
- `PUT /api/clients/profiles/[id]` - Update client
- `DELETE /api/clients/profiles/[id]` - Delete client
- `GET /api/clients/profiles/[id]/history` - Get client history
- `GET/POST/PUT/DELETE /api/clients/notes` - Manage notes
- `GET/POST/PUT /api/clients/preferences` - Manage preferences
- `POST /api/clients/upload-photo` - Upload photos

🛠️ Technical Details

Stack

- **Framework**: Next.js 14 (App Router)
- **Language**: TypeScript
- **Form Management**: react-hook-form
- **Validation**: Zod
- **UI Components**: shadcn/ui
- **Icons**: lucide-react
- **Notifications**: react-hot-toast

Code Quality

- ✅ TypeScript strict mode
- ✅ Proper error handling
- ✅ Loading states
- ✅ Type-safe API calls
- ✅ Consistent component patterns
- ✅ Mobile responsive

Build Verification

```
npm run build
```

- ✅ TypeScript compilation successful
- ✅ No type errors
- ✅ All pages render correctly
- ✅ Build completed without errors

File Changes

- **13 files changed**
- **3,277 insertions**
- **164 deletions**
- All changes are non-breaking and additive

📋 Testing Instructions

1. View Client List

1. Navigate to `/dashboard/clients`
2. Verify stats are displayed
3. Test search functionality
4. Check empty state (if no clients)

2. Create New Client

1. Click "Nuevo Cliente" button
2. Fill out the form with test data
3. Submit and verify redirect to detail page
4. Confirm client appears in list

3. View Client Detail

1. Click on a client from the list
2. Verify all tabs work:
   - Profile: Shows all information
   - History: Shows appointments (if any)
   - Notes: Can add/edit/delete notes
   - Preferences: Can configure settings
3. Test edit button navigation

4. Edit Client

1. From detail page, click "Editar"
2. Modify some fields
3. Upload a profile photo
4. Save and verify changes persist

5. Delete Client

1. From client list, click menu → Delete
2. Confirm deletion dialog

3. Verify client is removed from list

🚀 Deployment Notes

Requirements

- All backend tables already exist (from Phase 2 PR #72)
- No migration needed
- No environment variables required

Post-Deployment

The Clients Module will be immediately functional:
- Users can create and manage client profiles
- Full CRM capabilities available
- History tracking works if appointments exist
- Notes and preferences can be configured

📸 Screenshots

Client List Page

- Clean list view with search
- Client cards with avatars
- Quick stats dashboard

Client Detail Page

- Tabbed interface
- Complete profile view
- History with statistics
- Notes management
- Preferences configuration

Create/Edit Forms

- Organized sections
- Field validation
- Loading states
- Photo upload

✅ Checklist

- [x] All components implemented and functional
- [x] All pages created and tested
- [x] TypeScript types defined
- [x] Error handling implemented
- [x] Loading states added
- [x] Responsive design verified
- [x] Build successful
- [x] No breaking changes
- [x] Documentation complete

- [x] Code follows project patterns

🎓 Related PRs

- **PR #72**: Backend implementation (Client Module Phase 2)
- Added ClientProfile, ClientNote, ClientPreferences tables
- Implemented all API endpoints
- Created service layer

📝 Notes

- This PR completes the Client Module Phase 1 MVP
- Backend was already 100% complete
- All changes are non-breaking and additive
- Ready for production deployment
- No database migrations needed

🔗 References

- Phase 2 Backend PR (https://github.com/qhosting/citaplanner/pull/72)
- Project follows Next.js 13+ App Router conventions
- Uses established patterns from existing modules