

Estructura Actualizada de la Solución LalaHealthCare

LalaHealthCare/

```
├─ LalaHealthCare.sln
|
├─ LalaHealthCare.App/ (antes LalaHealthCare.Mau1)
|   │   └─ LalaHealthCare.App.csproj
|   │   └─ MauiProgram.cs
|   │   └─ App.xaml
|   │   └─ App.xaml.cs
|   │   └─ AppShell.xaml
|   │   └─ AppShell.xaml.cs
|   │   └─ MainPage.xaml
|   │   └─ MainPage.xaml.cs
|   │   └─ _Imports.razor
|   │
|   │   └─ Components/
|   │       │   └─ Routes.razor
|   │       │   └─ Layout/
|   │       │       │   └─ MainLayout.razor
|   │       │       │   └─ AppErrorBoundary.razor ✓ (actualizado)
|   │       │       │   └─ AuthorizedView.razor
|   │       │       │   └─ RedirectToLogin.razor
|   │       │       │   └─ SimpleErrorHandler.razor (opcional)
|   │       │       └─ Pages/
|   │       │           │   └─ Index.razor
|   │       │           │   └─ Login.razor ✓ (usa LoginViewModel)
|   │       │           │   └─ Dashboard.razor ✓ (usa DashboardViewModel)
|   │       │           └─ Logs.razor
|   │
|   └─ ViewModels/
|       │   └─ ViewModelBase.cs
|       │   └─ LoginViewModel.cs
|       │   └─ DashboardViewModel.cs
|
|   └─ Models/
|       │   └─ LogEntry.cs
|
|   └─ Services/
|       │   └─ AppState.cs
|       │   └─ LoggingService.cs
|       │   └─ NavigationService.cs
|
|   └─ Extensions/
|       │   └─ CommandExtensions.cs
```

```
|...|
|...| └─ wwwroot/
|...|   |
|...|   | └─ index.html
|...|   | └─ css/
|...|   |   |
|...|   |   | └─ app.css
|...|
|...| └─ Platforms/
|...|   |
|...|   | └─ Android/
|...|   |   |
|...|   |   | └─ MainActivity.cs
|...|   |   | └─ MainApplication.cs
|...|   |   | └─ AndroidManifest.xml
|...|   |   |
|...|   |   | └─ iOS/
|...|   |   |   |
|...|   |   |   | └─ AppDelegate.cs
|...|   |   |   | └─ Info.plist
|...|   |   |   | └─ Program.cs
|...|   |   | └─ MacCatalyst/
|...|   |   |   |
|...|   |   |   | └─ AppDelegate.cs
|...|   |   |   | └─ Info.plist
|...|   |   |   | └─ Program.cs
|...|   |   | └─ Tizen/
|...|   |   |   |
|...|   |   |   | └─ Main.cs
|...|   |   |   | └─ tizen-manifest.xml
|...|   |   | └─ Windows/
|...|   |   |   |
|...|   |   |   | └─ App.xaml
|...|   |   |   | └─ App.xaml.cs
|...|   |   |   | └─ Package.appxmanifest
|...|
|...| └─ Resources/
|...|   |
|...|   | └─ AppIcon/
|...|   |   |
|...|   |   | └─ appicon.svg
|...|   |   |
|...|   |   | └─ Fonts/
|...|   |   |   |
|...|   |   |   | └─ OpenSans-Regular.ttf
|...|   |   | └─ Images/
|...|   |   | └─ Raw/
|...|   |   | └─ Splash/
|...|   |   |   |
|...|   |   |   | └─ splash.svg
|...|
|...| └─ LalaHealthCare.Business/
|...|   |
|...|   | └─ LalaHealthCare.Business.csproj
|...|   | └─ Services/
|...|   |   |
|...|   |   | └─ IAuthenticationService.cs
|...|   |   | └─ AuthenticationService.cs
|...|   |   | └─ IVisitService.cs
|...|   |   | └─ VisitService.cs
```

```
|
└─ LalaHealthCare.DataAccess/
    ├── LalaHealthCare.DataAccess.csproj
    ├── Models/
    │   ├── User.cs
    │   ├── Visit.cs
    │   └─ AuthDtos.cs
    └─ Repositories/
        ├── IAuthenticationRepository.cs (interfaz incluida en Mock)
        ├── MockAuthenticationRepository.cs
        ├── IVisitRepository.cs (interfaz incluida en Mock)
        └─ MockVisitRepository.cs
```

Arquitectura MVVM Implementada:

1. ViewModels (LalaHealthCare.App/ViewModels/)

- `ViewModelBase.cs`: Clase base con:
 - Implementación de `INotifyPropertyChanged`
 - Métodos `ExecuteAsync` para manejo consistente de errores
 - Logging automático de operaciones
 - Estados como `IsBusy` y `Title`
- `LoginViewModel.cs`: Maneja la lógica de login
 - Comandos: `LoginCommand`, `BiometricLoginCommand`
 - Validación de entrada
 - Manejo de estado de autenticación
- `DashboardViewModel.cs`: Maneja la lógica del dashboard
 - Comandos: `CheckInCommand`, `CheckOutCommand`, etc.
 - `ObservableCollection<Visit>` para las visitas
 - Búsqueda y filtrado de visitas

2. Models (LalaHealthCare.App/Models/)

- `LogEntry.cs`: Modelo para el sistema de logging

3. Services (LalaHealthCare.App/Services/)

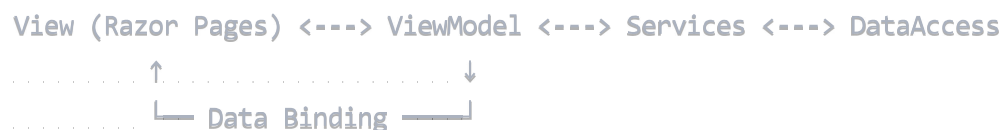
- `AppState.cs`: Estado global de la aplicación

- `LoggingService.cs`: Sistema completo de logging con:
- Guardado local en memoria y archivo
 - Simulación de envío a API
 - Reintentos en caso de fallo

4. Views (LalaHealthCare.App/Components/Pages/)

- Páginas Razor que se enlazan con ViewModels mediante inyección de dependencias
- Binding de datos bidireccional
- Comandos ejecutados desde la UI

Flujo de Datos MVVM:



Manejo de Errores Implementado:

1. **Global:** `AppErrorBoundary.razor` captura errores no manejados
2. **ViewModel:** `ViewModelBase.ExecuteAsync` maneja errores en operaciones
3. **Logging:** Todos los errores se registran automáticamente
4. **UI:** Notificaciones al usuario mediante Snackbar

Características Implementadas:

 Completado:

1. Arquitectura MVVM completa
2. Sistema de logging robusto
3. Manejo global de errores
4. ViewModels con comandos
5. Separación de responsabilidades
6. Login con credenciales y biométrico (mock)
7. Dashboard con lista de visitas
8. Clock In/Clock Out funcional

9. Página de visualización de logs
10. Navegación segura entre páginas

Pendiente:

1. Biometría real (usando APIs del dispositivo)
2. Página de detalles de visita
3. Notificaciones push
4. Sincronización offline/online
5. API REST real (reemplazar mocks)
6. Más páginas (Schedules, Notifications, etc.)
7. Tests unitarios para ViewModels
8. Localización completa (multi-idioma)

Paquetes NuGet Instalados:

LalaHealthCare.App:

- MudBlazor 8.7.0
- Microsoft.AspNetCore.Components.WebView.Maui
- Microsoft.Extensions.Logging.Debug

LalaHealthCare.DataAccess:

- System.Text.Json 9.0.0

LalaHealthCare.Business:

- Referencias a DataAccess

Comandos para ejecutar:

powershell

Compilar La solución

dotnet build

Ejecutar en Android

cd LalaHealthCare.App

dotnet build -t:Run -f net9.0-android

Ejecutar en iOS

dotnet build -t:Run -f net9.0-ios

Ejecutar en Windows

dotnet build -t:Run -f net9.0-windows10.0.19041.0

Notas importantes:

1. **Namespace:** Aunque el proyecto cambió de nombre, los namespaces internos siguen siendo `LalaHealthCare.App`
2. **Referencias:** Si hay problemas de compilación, verifica que las referencias entre proyectos usen el nuevo nombre
3. **_Imports.razor:** Asegúrate de que los imports reflejen el namespace correcto