



TatasApp



Creadores

- Alexander Aguilera
- Andrea Pino
- David Guentelican



Arquitectura General

TatasApp es una aplicación móvil híbrida desarrollada con **Ionic + Angular** y utiliza una arquitectura MVC basada en servicios (controlador), páginas (pages) e interfaces (modelo). La app está orientada a la gestión y acompañamiento de adultos mayores y sus familiares, permitiendo la administración de usuarios, familiares, eventos y alertas.

- **Pages:** Vistas principales de la app (usuarios, familiares, eventos, configuración, etc.)
- **Services:** Lógica de negocio, acceso a APIs, almacenamiento local y notificaciones.
- **Interfaces:** Tipado TypeScript para datos y modelos.
- **Routing:** Navegación entre páginas y protección de rutas.
- **Assets:** Recursos estáticos (imágenes, íconos, etc.)
- **Base de datos local SQLite:** Persistencia de sesión y almacenamiento de datos básicos del usuario logueado, utilizados en distintas funcionalidades de la app.



Flujo de Trabajo

1. El usuario interactúa con la interfaz (páginas Ionic/Angular).
2. Los componentes llaman a **services** para lógica de negocio y acceso a datos.
3. Los servicios gestionan peticiones HTTP a la API, almacenamiento local (SQLite) y lógica auxiliar.
4. La respuesta se muestra en la UI y se gestionan errores/notificaciones.



Funcionalidades Principales



Usuarios

- Registro y login de usuarios (adulto mayor o familiar).
- Edición de datos personales, correo, contraseña y foto de perfil.
- Visualización de información personal.



Familiares

- Búsqueda y registro de familiares a través de contactos del teléfono.
- Visualización y eliminación de familiares asociados.
- Sincronización con la API para validar y gestionar familiares.



Eventos

- Creación, edición y eliminación de eventos (citas médicas, eventos familiares, personales, etc.).
- Visualización de eventos en un calendario interactivo (FullCalendar).
- Notificaciones automáticas de eventos próximos.
- Visualización de eventos asociados a familiares.



Alertas

- Envío de alertas SOS con ubicación geográfica.
- Recepción y gestión de alertas pendientes e historial de alertas.
- Notificaciones locales y navegación a Google Maps desde la alerta.



Características Adicionales

- **Notificaciones locales** para eventos y alertas.
- **Soporte multiplataforma** (Android/iOS/Web).
- **Integración con contactos del teléfono** (para agregar familiares).
- **Splash screen** y personalización visual.
- **Almacenamiento local con SQLite** para persistencia offline y gestión de sesión.
- **Control de sesión y navegación protegida.**



Dependencias principales

- **@ionic/angular**: Framework UI para apps móviles híbridas.
- **@angular/core, forms, router**: Framework base Angular.
- **@capacitor/core, local-notifications, splash-screen, sqlite**: Plugins nativos para funcionalidades móviles y base de datos local.
- **@awesome-cordova-plugins/contacts**: Acceso a contactos del dispositivo.
- **@fullcalendar/angular**: Calendario interactivo para eventos.
- **rxjs**: Programación reactiva.
- **Otros**: Plugins para geolocalización, almacenamiento, etc.







Ejemplo de Uso de la App

- Un adulto mayor se registra y agrega a sus familiares desde los contactos del teléfono.
- Puede crear eventos (citas médicas, recordatorios) y recibir notificaciones automáticas.
- En caso de emergencia, puede enviar una alerta SOS que notifica a sus familiares con la ubicación.
- Los familiares pueden ver los eventos y alertas asociadas al adulto mayor, y recibir notificaciones en tiempo real.



Ventajas de la Arquitectura

-  Modularidad y fácil mantenimiento.
-  Escalabilidad para agregar nuevas funcionalidades.
-  Reutilización de servicios y componentes.
-  Seguridad en el manejo de datos y sesiones.



Notas

- La app se conecta a una API REST para la gestión de usuarios, familiares, eventos y alertas.
- Utiliza una base de datos local SQLite para persistencia de sesión y datos básicos del usuario logueado.

- Para más detalles sobre endpoints y estructura de la API, consultar la documentación del backend.