**Especificaciones Técnicas del Software (ETS)**

**Proyecto:** *Apaña2App – Plataforma de Apoyo y Bienestar para Personas con TEA*  
**Versión:** 1.0  
**Autor:** Jose Caro   
**Fecha:** Octubre 2025

**1. Introducción**

**1.1 Propósito**

El presente documento describe las especificaciones técnicas del software Apaña2App, definiendo la arquitectura, tecnologías, requisitos funcionales y no funcionales, así como los lineamientos técnicos necesarios para su desarrollo, implementación, mantenimiento y despliegue.

**1.2 Alcance**

Apaña2App es una aplicación móvil multiplataforma que promueve la autonomía, organización y bienestar de personas dentro del espectro autista (TEA).  
Permite gestionar rutinas diarias, registrar notas personales, activar alertas SOS con geolocalización y facilitar la comunicación con tutores responsables mediante sincronización en tiempo real a través de Firebase.

**2. Arquitectura del Sistema**

**2.1 Tipo de Arquitectura**

* Arquitectura basada en servicios distribuidos (Client–Server).
* Estructura modular con componentes reutilizables.
* Comunicación mediante REST API y Firebase SDK.

**2.2 Componentes Principales**

| **Componente** | **Descripción** |
| --- | --- |
| **Frontend (App)** | Desarrollado en *Ionic + Angular 17*, ofrece interfaz adaptable, accesible y responsiva. |
| **Backend (Firebase Functions)** | Gestiona lógica del negocio y eventos automáticos (notificaciones, sincronización). |
| **Base de Datos (Firestore)** | Almacenamiento NoSQL con documentos jerárquicos y estructura flexible. |
| **Autenticación (Firebase Auth)** | Control de acceso mediante email y contraseña. |
| **Firebase Functions** | Gestión de notificaciones push entre usuario y tutor. |
| **Servicio (Angular Capacitor)** | Obtención de coordenadas GPS y visualización de ubicación en tiempo real. |

**3. Requisitos de Hardware y Software**

**3.1 Entorno de Desarrollo**

| **Elemento** | **Especificación** |
| --- | --- |
| Sistema Operativo | Windows 10+ / macOS 12+ |
| IDE | Visual Studio Code |
| Lenguaje principal | TypeScript |
| Framework | Ionic Framework + Angular 17 |
| Node.js | v20+ |
| NPM | v10+ |
| Control de versiones | GitHub |
| Emuladores | Android Studio, Servidor local |

**3.2 Entorno de Ejecución (Cliente)**

| **Requisito** | **Mínimo** | **Recomendado** |
| --- | --- | --- |
| Sistema operativo | Android 8.0 | Android 11 |
| Procesador | Dual-Core 1.5 GHz | Quad-Core 2.0 GHz |
| Memoria RAM | 2 GB | 4 GB |
| Almacenamiento | 200 MB libres | 500 MB libres |
| Conectividad | 3G / Wi-Fi | 4G o superior |

**4. Tecnologías Implementadas**

| **Categoría** | **Tecnología** | **Función Principal** |
| --- | --- | --- |
| **Framework** | Ionic + Angular | Desarrollo multiplataforma (Android/Web) |
| **Lenguaje** | TypeScript, HTML5, SCSS | Estructura y estilos del frontend |
| **Backend** | Firebase Cloud Functions | Lógica de negocio en la nube |
| **Base de Datos** | Firestore (NoSQL) | Persistencia de datos en tiempo real |
| **Autenticación** | Firebase Auth | Registro y control de acceso |
| **Almacenamiento** | Firebase Storage | Gestión de archivos multimedia |
| **Mensajería** | Firebase Functions | Envío de alertas y notificaciones |
| **Mapas y GPS** | Geolocation capacitor | Ubicación y rastreo |
| **Diseño UI/UX** | Figma | Prototipos de interfaces accesibles |
| **Gestión Ágil** | Jira | Control de backlog y tareas de sprint |
| **Repositorio** | GitHub | Control de versiones y despliegue continuo |

**5. Diseño de la Base de Datos (Firestore)**

* Estructura **NoSQL** basada en colecciones y documentos:
  + /users/{uid} → Datos personales y rol (usuario).
  + /tutors/{uid} → Datos personales y rol (tutor).
  + /notes/{noteId} → Título, contenido, fecha y fotos.
  + /routines/{routineId} → Tareas diarias con recordatorios.
  + /alerts/tutors/inbox/{alertId} → Ubicación GPS y estado de emergencia recibida por el tutor.
  + /link/{userUid}/{tutorUid} → Se guarda la relación entre el tutor y el beneficiario.
* Relaciones **1:N** mediante identificadores únicos (uid, docId).
* Reglas de seguridad definidas en **Firestore Rules** con control de acceso por rol.

**6. Módulos Funcionales**

1. **Autenticación y Registro**
   * Inicio de sesión y creación de cuentas.
   * Validación de credenciales con Firebase Auth.
   * Recuperación de contraseña.
2. **Rutinas y Recordatorios**
   * Registro y edición de actividades diarias.
   * Notificaciones automáticas con hora programada.
3. **Notas Personales**
   * Agenda tipo diario con texto e imágenes adjuntas.
   * Filtrado por fecha y categorías.
4. **Perfiles**

* Ver perfil
* Editar perfil

1. **Vinculación de tutor**

Vincula a el beneficiario con el tutor a través de rut y correo de tutor.

1. **Módulo SOS**
   * Envío instantáneo de alerta con ubicación GPS.
   * Notificación push al tutor registrado.
2. **Gestión de Tutores**
   * Visualización del estado del usuario.

**7. Requisitos No Funcionales**

| **Categoría** | **Descripción** |
| --- | --- |
| **Rendimiento** | Respuesta < 2 seg en operaciones locales y < 4 seg en sincronización. |
| **Usabilidad** | Interfaz simple, iconografía intuitiva y colores pastel para accesibilidad TEA. |
| **Seguridad** | Datos cifrados (HTTPS/TLS 1.3). Reglas Firestore con validación de roles. |
| **Disponibilidad** | Uptime estimado: 99% (Firebase Hosting). |
| **Escalabilidad** | Estructura modular y adaptable a nuevos módulos. |
| **Mantenibilidad** | Código documentado y separado en servicios, componentes y modelos. |
| **Compatibilidad** | Soporte multiplataforma (Android, Web). |

**8. Integraciones Externas**

* **Capacitor Plugins:** cámara, geolocalización, notificaciones locales.

**9. Control de Versiones y Despliegue**

| **Elemento** | **Descripción** |
| --- | --- |
| **Repositorio** | GitHub (rama main = producción, dev = pruebas) |
| **Versionado** | SemVer (ejemplo: v1.2.1) |
| **CI/CD** | Integración continua con Firebase CLI y GitHub Actions |
| **Entornos** | localhost:8100 (desarrollo) / Android Studio. |
| **Pruebas** | Manuales. |

**10. Seguridad y Protección de Datos**

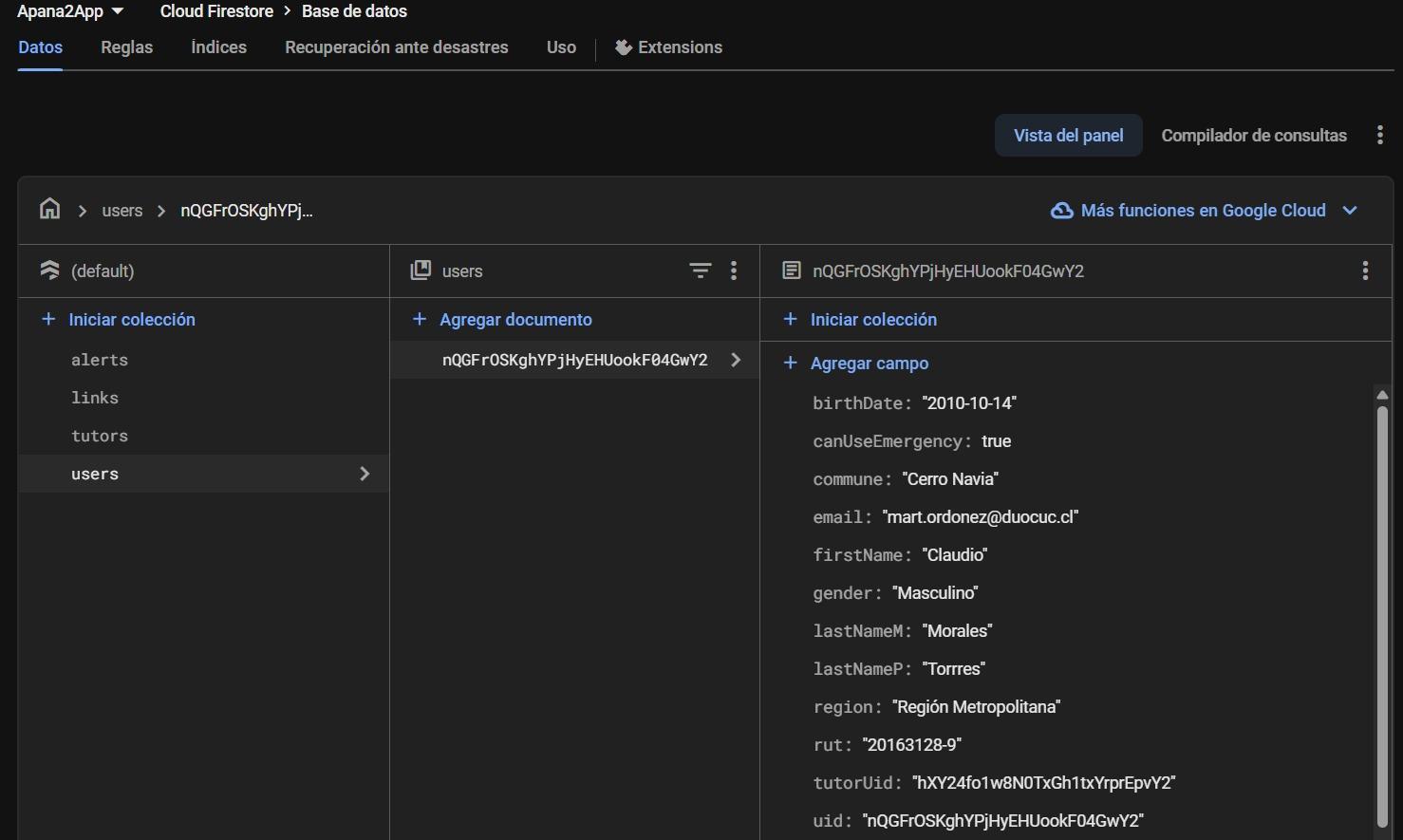
* Cumplimiento de la **Ley 19.628 – Protección de Datos Personales (Chile)**.
* Control de acceso basado en roles (user, tutor, admin).
* Tokens de autenticación JWT y expiración controlada.
* Cifrado de datos en tránsito y almacenamiento seguro.
* Validación de permisos antes de leer o escribir en Firestore.
* Módulo de consentimiento informado y políticas de privacidad en pantalla inicial.

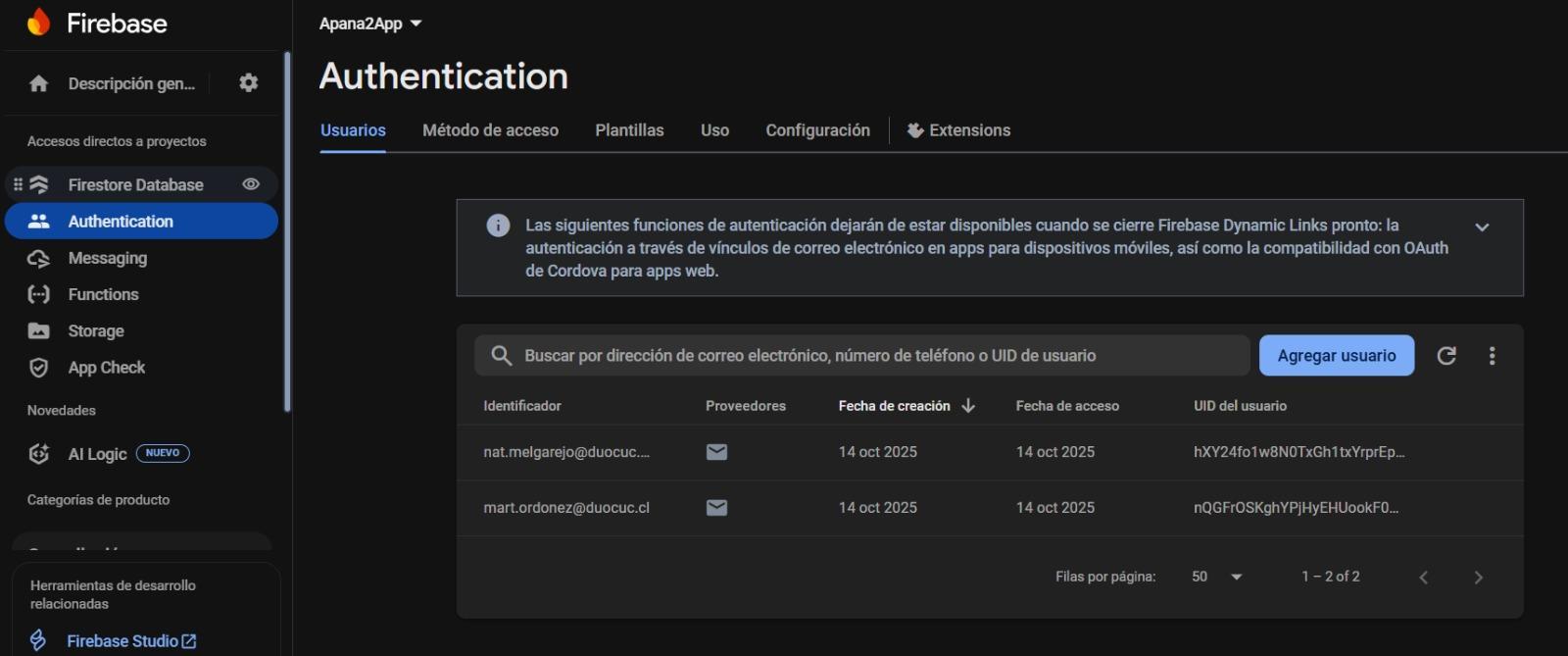
**11. Futuras Mejoras**

* Módulo de seguimiento emocional mediante IA básica.
* Dashboards para tutores con analítica de rutinas y estados.
* Sistema de mensajería directa en tiempo real (Firebase Chat).
* Integración con wearables para monitoreo biométrico.

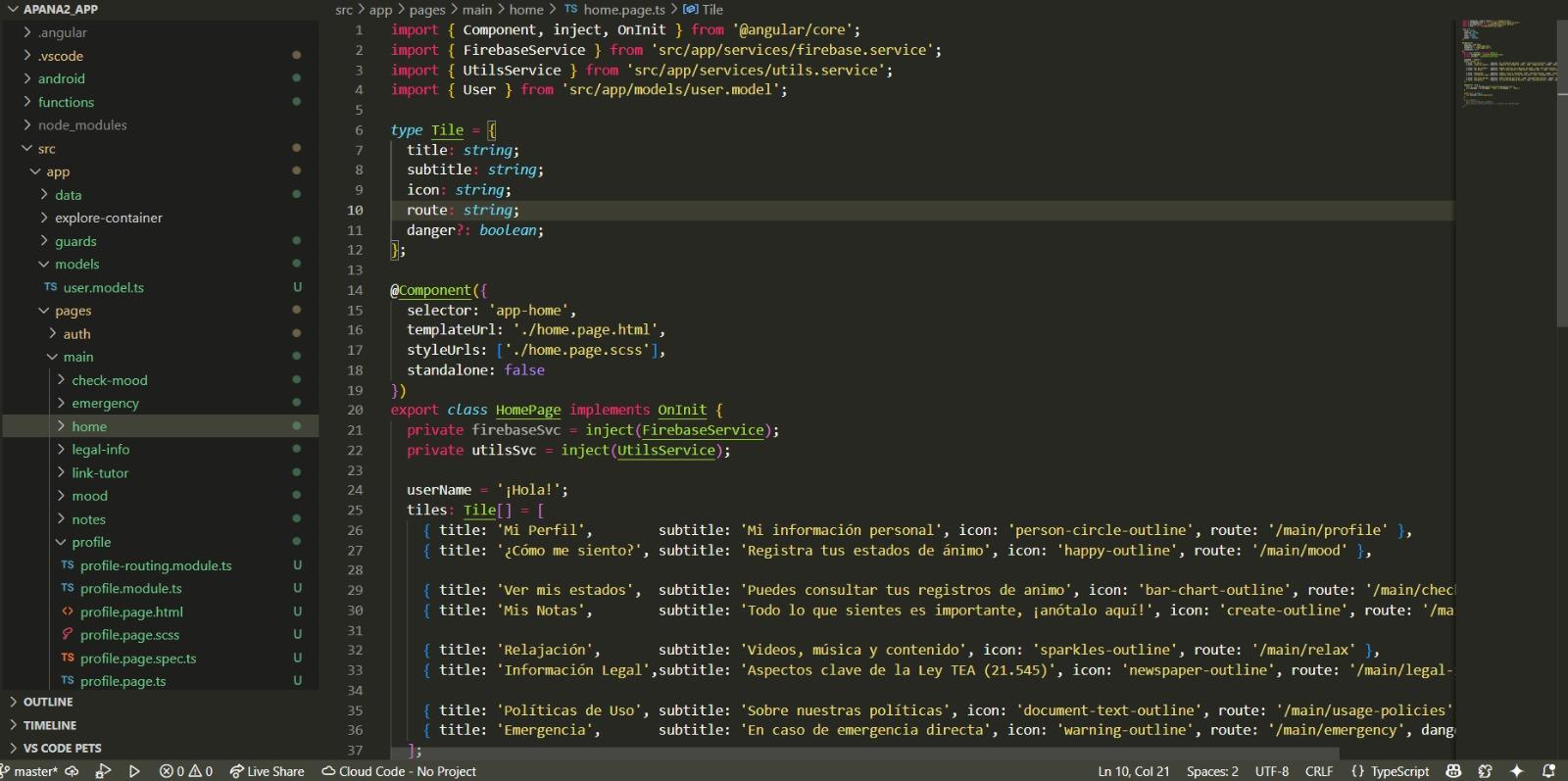
**12. Capturas**

* **FireBase:**

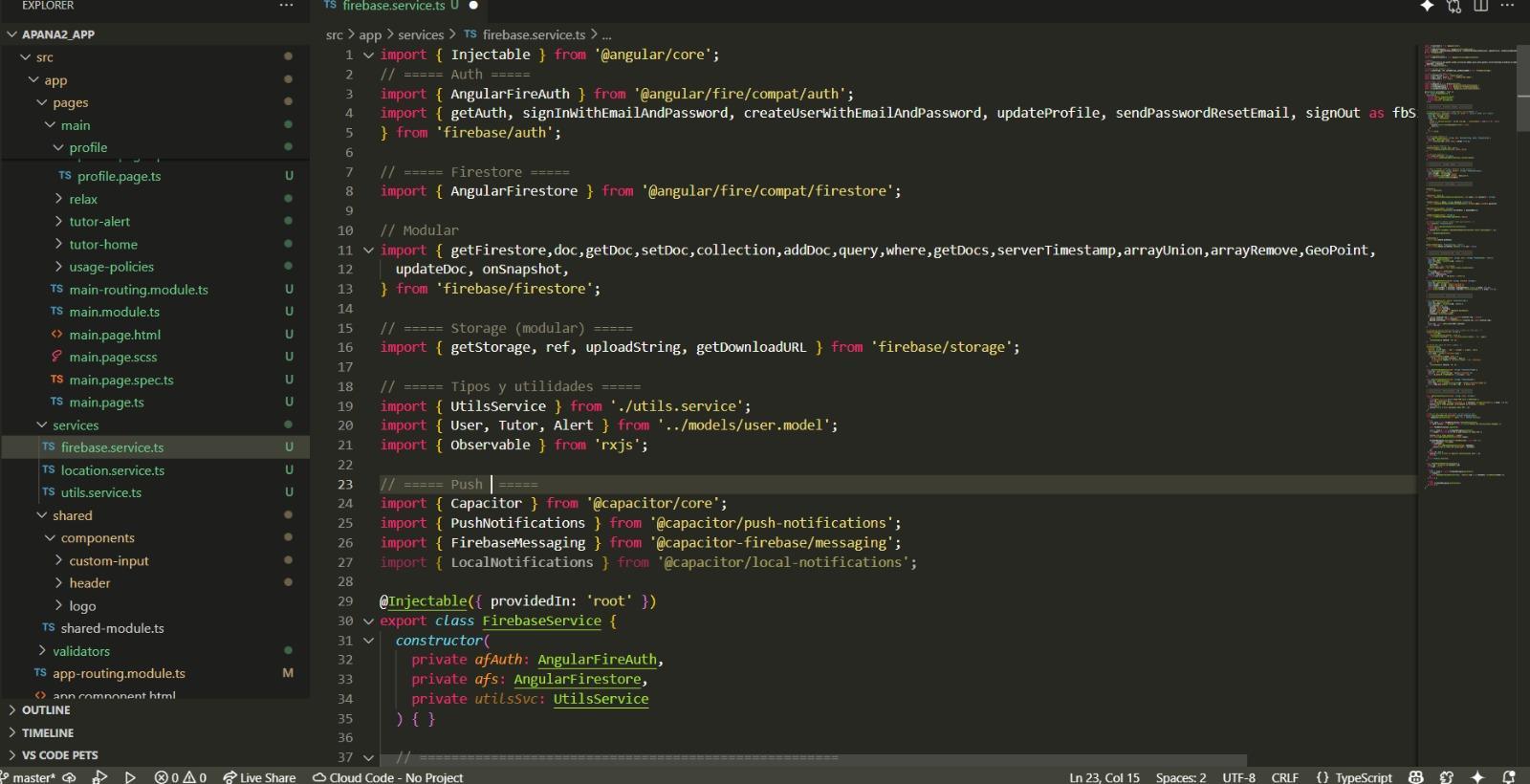
****

****

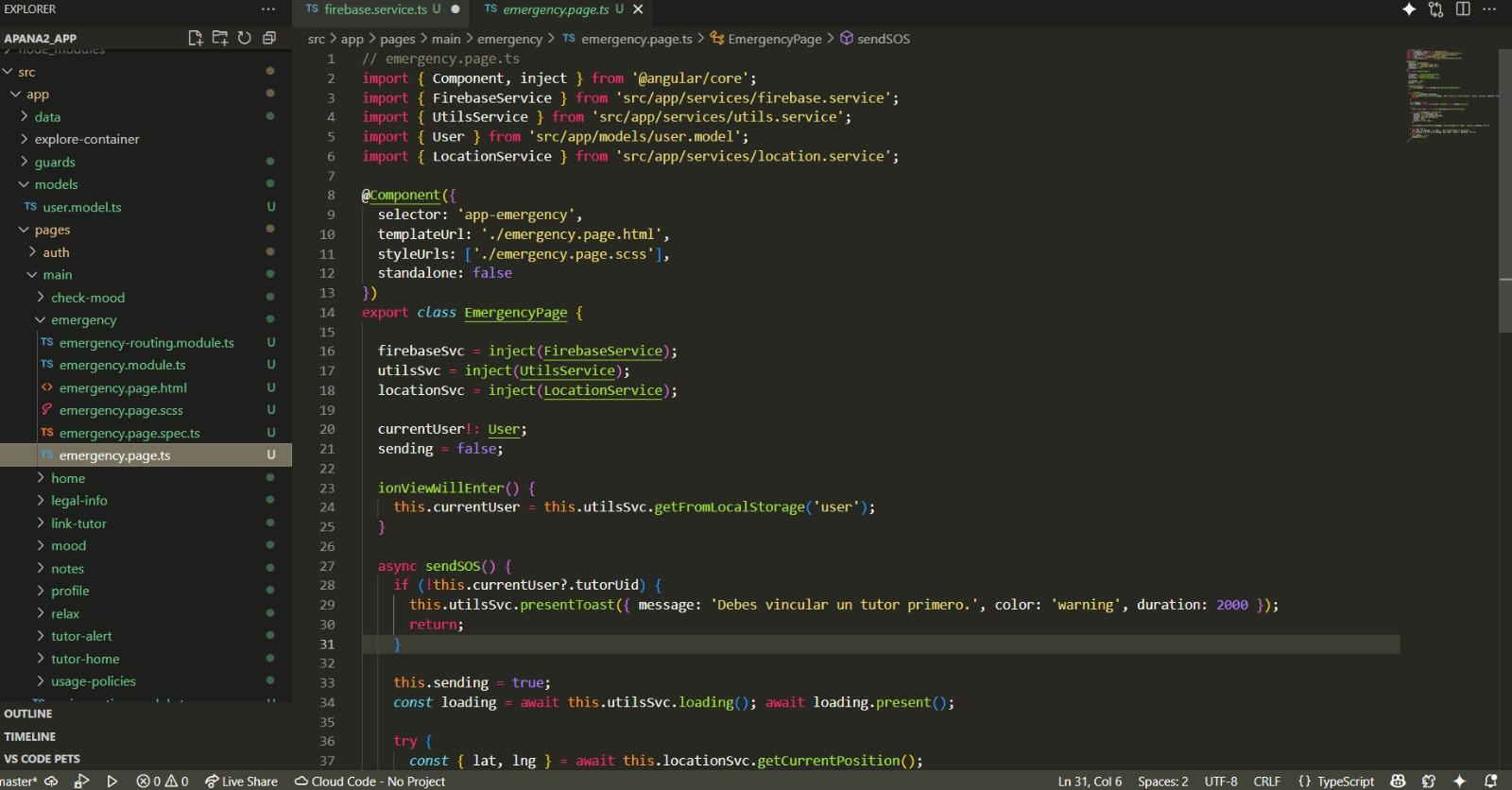
* **Visual Studio Code:**
* **Home**

****

* **Firebase Service**

****

* **Emergency Page**

****