# Documentación de la Aplicación Consultas Médicas

## Introducción

La aplicación Consultas Médicas es una plataforma diseñada para facilitar la gestión de consultas médicas. Está desarrollada en Ionic Capacitor y se puede utilizar tanto en dispositivos móviles como en navegadores web (PWA). La app incluye las siguientes funcionalidades:

* Registrarse e iniciar sesión como un usuario autentificado.
* Programación de citas médicas con fecha y hora específicas.
* Gestión de datos básicos de los pacientes.
* Visualización de un panel de control con las citas programadas del día y navegación en general de la página.
* Mapa de la zona con los hospitales y consultorios más cercanos.
* Registro de consultas pasadas con fotografía de receta médica.
* Grabación de videos de consultas médicas, con opciones para compartir o guardar.
* Recepción de notificaciones de emergencia.
* Módulo de emergencia médica, detectando caídas para llamar al 911.
* Chat de pacientes donde el usuario puede comunicarse con otros pacientes.

## Estructura de la Aplicación

A continuación, se describen las páginas de la aplicación y las librerías utilizadas en cada una.

1. Login

firebase: Para autenticar usuarios mediante Firebase Authentication.

@capacitor/preferences: Para guardar configuraciones o información básica del usuario.

@ionic/angular: Para componentes UI como formularios y botones.

2. Registro

firebase: Para registrar nuevos usuarios en Firebase Authentication.

@angular/forms: Para manejar y validar formularios reactivos.

@ionic/angular: Para componentes de UI como entradas de texto y botones.

3. CalendarioCitasPage

firebase: Para almacenar y sincronizar las citas en tiempo real mediante Firebase Realtime Database.

@angular/forms: Para seleccionar fechas y horarios.

@capacitor/preferences: Para manejar configuraciones relacionadas con la visualización del calendario.

@ionic/angular: Para componentes como calendarios y listas.

4. PerfilPaciente

firebase: Para manejar información del perfil en Firebase Firestore.

@capacitor/camera: Para permitir a los usuarios subir o tomar fotos de perfil.

@ionic/angular: Para componentes UI como formularios y botones.

5. Panel de Control

firebase: Para obtener datos en tiempo real desde Firebase Firestore.

@ionic/angular: Para mostrar gráficos y listas de información en la interfaz de usuario.

6. ConsultasPasadas

firebase: Para listar y mostrar consultas previas del usuario mediante Firebase Realtime Database.

@ionic/angular: Para manejar listas y scroll infinito.

7. ConsultoriosCercanos

@capacitor/geolocation: Para obtener la ubicación actual del usuario.

@types/google.maps: Para mostrar un mapa con los consultorios cercanos usando Google Maps API.

@ionic/angular: Para componentes UI como mapas y listas.

8. GrabarConsulta

cordova-plugin-media-capture y @awesome-cordova-plugins/media-capture: Para capturar grabaciones de video o audio.

firebase: Para almacenar grabaciones en Firebase Storage.

@capacitor/filesystem: Para manejar archivos grabados en dispositivos.

@ionic/angular: Para mostrar opciones de grabación y administración de contenido.

9. ChatDePacientes

firebase: Para manejar mensajes en tiempo real usando Firebase Realtime Database.

@ionic/angular: Para mostrar mensajes en la interfaz.

10. Librerías Globales en la Aplicación

@capacitor/push-notifications: Para notificaciones push desde FCM.

@ionic/angular: Para componentes de UI en todas las páginas.

@capacitor/core: Para acceder a funcionalidades nativas del dispositivo.

firebase: Suite completa para autenticación, base de datos en tiempo real, almacenamiento y notificaciones.

rxjs: Para programación reactiva en Angular.

@angular/\*: Framework principal para el desarrollo de la aplicación.

zone.js: Para manejo de zonas en Angular.

@capacitor/app: Para controlar eventos globales de la aplicación.

@capacitor/status-bar: Para personalizar la barra de estado en dispositivos móviles.

@capacitor/motion: Para la funcionalidad de caída de emergencia

@awesome-cordova-plugins/call-number/ngx: para usarse junto a motion para llamar a servicios de emergencia

## Módulos Implementados

**Práctica 1: Despliegue Multiplataforma con Inicio de Sesión y Registro de Usuarios**

Esto se ha cumplido gracias a las paginas de Registro y login con firebase auth y guards para garantizar el manejo de sesiones y de autentificación de usuarios, además de tener la PWA presente en toda la app, la prueba de esto se puede ver en la existencia del documento firebase.json y la liga de la pwa:

<https://consultasmedicas-807b5.firebaseapp.com/login>

**Práctica 2: Uso de la Cámara y Almacenamiento de Imágenes**

Esto se cumple en la página consultas-pasadas bajo el uso de ModalCamaraComponent modificado desde el repositorio de **Rogelio/practicas-idgs08**

**Práctica 3: Compartir Archivos desde la Cámara y del Sistema**

Esto se cumple en la misma página que el punto anterior, implementando fotografía, codificación a base64 y compartido de imágenes

**Práctica 4: Acceso a la Ubicación**

Esto se cumple en la pagina consultorios-cercanos usando la Google Maps API con un mapa customizado para poder hacer búsquedas específicas, en este caso todos los lugares con la palabra “salud” en su nombre, tags o descripción, la importación de este API se puede ver en el archivo index.html

**Práctica 5: Notificaciones Push**

Esto se cumple con FCM, un modulo de fiirebase que permite mandar notificaciones push cada cierto tiempo o notificaciones programadas desde firebase cloud funtions, pero al ser este un complemento de paga, lo único que se pudo hacer es crear notificaciones de prueba

**Práctica 6: Consumo de Video en Streaming**

Esto se cumple en la pagina panel-control. En la parte inferior de la página se puede ver un video acerca de salud mental y se implementó con el api de Youtube Data API V3, la importación de este API se puede ver en el archivo index.html

**Práctica 7: Uso del Acelerómetro y Sensores del Dispositivo**

Esto se cumple junto a otra funcionalidad en el componente principal de la aplicación: app.component.ts, esto para asegurar la funcionalidad en toda la app y en todo momento, se implementó la librería de @capacitor/motion y @awesome-cordova-plugins/call-number/ngx para detectar si el usuario tiene un accidente al caer o tropezarse, la app muestra una alerta esperando confirmación de la salud del usuario, de no obtener respuesta después de 10 segundos, automáticamente se llama a un numero de emergencia

**Práctica 8: Grabación y Compartición de Video**

Esto se cumple en la página grabar consulta, donde se implementó funcionalidad dependiendo de la plataforma, para celulares se usa la librería @awesome-cordova-plugins/media-capture/ngx ya que es la única disponible que cumple con lo requerido, la lógica de compartir y guardar video es similar a la de la pagina consultas pasadas. Para la funcionalidad de web, se usó blobs implementados desde ionic para poder grabar y compartir el video

**Práctica 9: Sincronización de Datos en Tiempo Real entre Dispositivos**

Esto se cumple gracias a la implementación de Firebase Real Time Database, donde se guardan los datos de un usuario autentificado como nombre, comentarios y citas, esta función trabaja con Firebase Auth para el manejo de sesiones y la seguridad de datos

**Práctica 10: Manejo de Servicios de Chat o Mensajería**

Esto se cumple parcialmente gracias a la implementación de Firebase Firestore Database, donde se registran los usuarios al entrar a la lista de chats y pueden ver otros usuarios registrados, al hacer esto, los usuarios pueden seleccionar un usuario y crear un chat con el usuario, que se guarda y encripta en Firestore, esta funcionalidad, solo se puede usar en paginas web.

Cambios desde repositorio practicas-idgs08

1. **Cámara**

|  |  |  |
| --- | --- | --- |
| **Aspecto** | **ConsultasMedicas** | **practicas-idgs08** |
| **Propósito del Componente** | Capturar imágenes y procesarlas para ser utilizadas en citas médicas. | Capturar imágenes sin integración o lógica específica. |
| **Modal Interaction** | Retorna la imagen capturada y permite guardarla, descargarla o compartirla. | Solo cierra el modal sin pasar datos. |
| **Uso de Servicios** | Integrado con un servicio (CitasService) para manejar datos en Firebase. | No utiliza servicios. |
| **Interfaz del Modal** | Diseño más interactivo, con soporte para dispositivos móviles y escritorio. | Diseño básico sin validaciones específicas. |
| **Integración con Firebase** | Sí, para asociar imágenes a citas pasadas. | No. |
| **Feedback al Usuario** | Alertas y mensajes en guardar o compartir imágenes. | Sin feedback. |

1. **Autenticación de Usuarios:**
   * *Funcionalidad Común:* Ambos repositorios implementan módulos para el inicio de sesión y registro de usuarios.
   * *Modificaciones en ConsultasMedicas:* Se han mejorado las validaciones de formularios y se ha integrado la autenticación con Firebase Auth
2. **Gestión de Citas:**
   * *Funcionalidad Común:* Ambos sistemas permiten la programación y visualización de datos.
   * *Modificaciones en ConsultasMedicas:* Se ha añadido un calendario interactivo para la gestión de citas y guardado de datos en Firebase RTD automáticos para los pacientes.

**Los demás módulos fueron hechos desde cero sin ninguna referencia desde Practicas-idgs08**