**Manual Técnico**

* **StatefulWidget**

Clase que se utiliza para crear widgets cuyo estado puede cambiar durante la vida útil del widget, permiten actualizar y redibujar la interfaz de usuario en respuesta a cambios en el estado.

Es responsable de crear una instancia de su clase de estado y administrarla, mientras que el objeto State se encarga de almacenar y actualizar el estado del widget.



* **Void initState**

Se utiliza en clases de widgets StatefulWidget para realizar tareas de inicialización cuando el widget es creado y agregado al árbol de widgets.

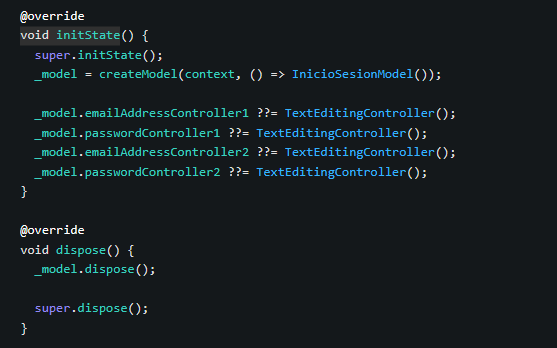
En el método initState() se suelen inicializar variables y estados para el widget, también se puede realizar llamadas a servicios externos o API para obtener datos iniciales necesarios para el widget.

Se configuran controladores y suscripciones a eventos, como escuchadores de cambios o eventos de interacción

* **void dispose**

Se utiliza en clases de widgets StatefulWidget para realizar tareas de limpieza y liberación de recursos cuando el widget se elimina del árbol de widgets; se liberan los recursos asignados, como controladores, suscripciones a eventos, conexiones a servicios o API, o cualquier recurso que pueda estar en uso por el widget.

También se cancelan tareas o temporizadores activos para evitar fugas de memoria o comportamientos inesperados.



* **Widget build(BuildContext context)**

Se utiliza en clases de widgets StatefulWidget o StatelessWidget para construir y retornar la representación visual del widget en la interfaz de usuario, tambien se puede diseñar y configurar la apariencia visual del widget y generar automáticamente el código correspondiente en el método build().



* **createState()**

Se utiliza en una clase StatefulWidget para crear una instancia de su clase de estado asociada. La clase de estado es responsable de almacenar y actualizar el estado del widget y es donde se implementa la lógica y la funcionalidad del widget; está asociada al diseño y las configuraciones realizadas en la interfaz visual de la herramienta.



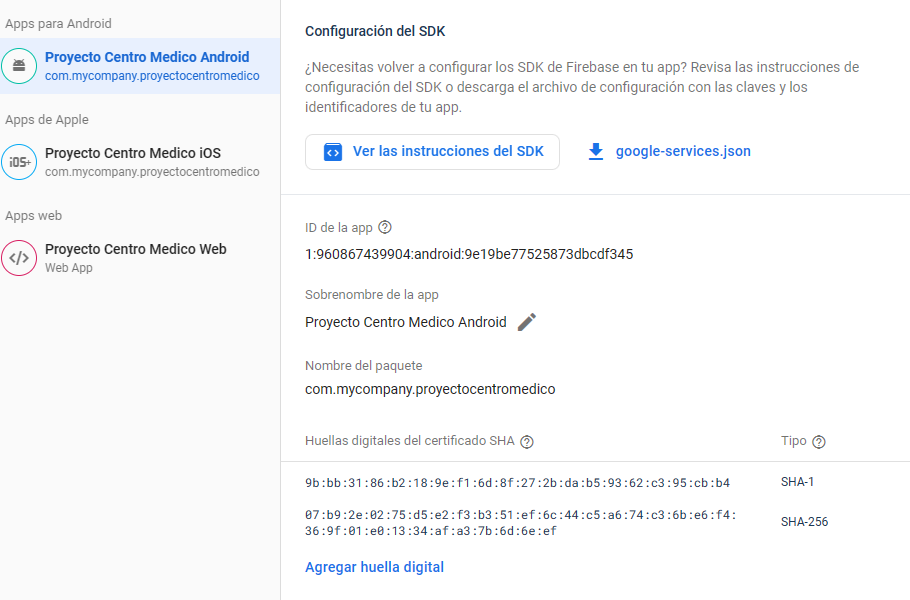
* **Evento de Toque onPressed**

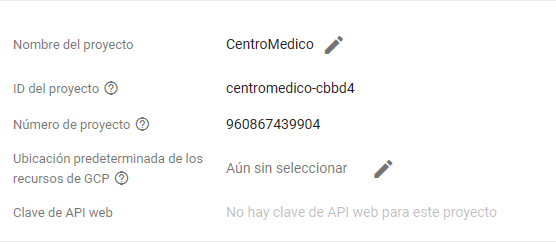
Se utiliza en muchos widgets para definir una acción o una función que se ejecutará cuando el usuario interactúe con el widget presionándolo. Es una forma de capturar eventos de interacción y responder a ellos

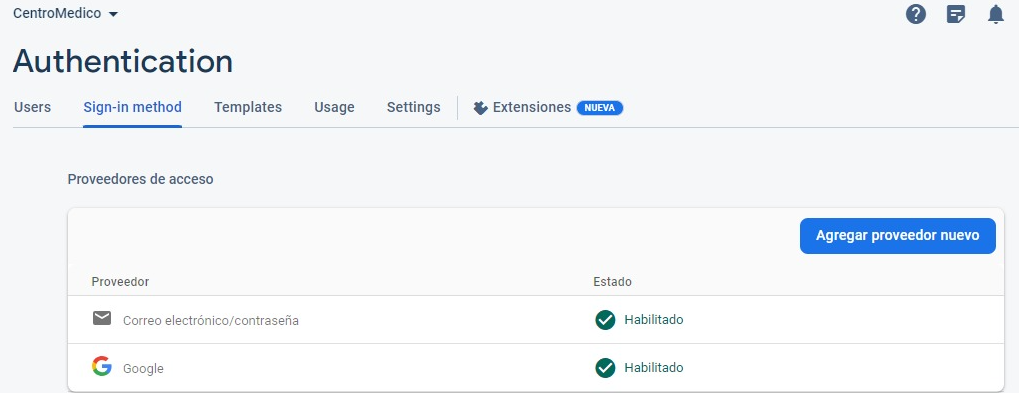
Se asigna una acción o función que se ejecutará cuando el usuario interactúe con un widget presionándolo.



**Configuración con Firebase**

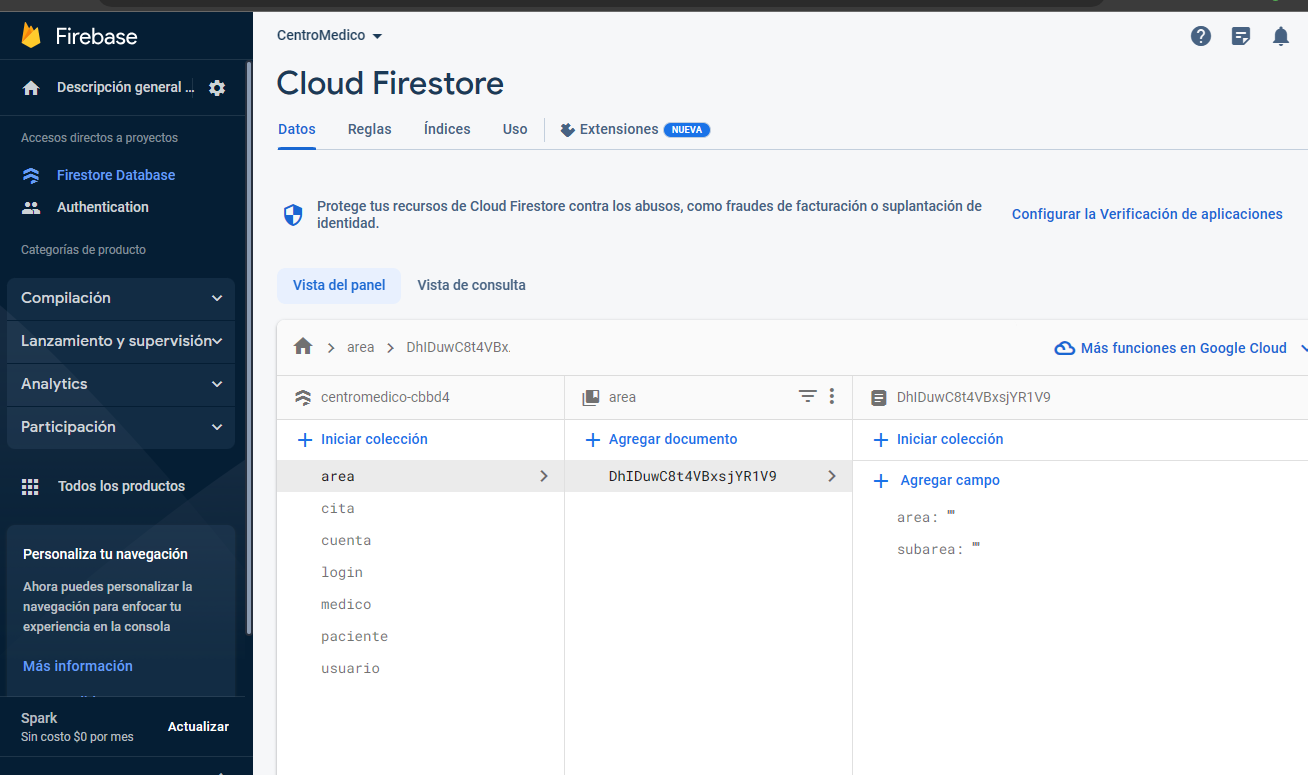






Se crea BD utilizando Firebase con la opción de Firestores que permite realizar consultas más complejas y rápidas, y el escalamiento se ajusta a un nivel más alto que Realtime Database.

La Base de Datos se llama:CentroMedico



Se crearon diferentes colecciones que son lo equivalente a crear una tabla

* Area
* Cita
* Cuenta
* Login
* Medico
* Paciente
* Usuario

También ser agregaron los campos a cada colección, que se establecen diferente tipo de datos que son similares a los de otros gestores de BD

