

APLICACIONES MÓVILES Y SERVICIOS TELEMÁTICOS

PRÁCTICA DE LABORATORIO 4 – APLICACIÓN MÓVIL AVANZADA USANDO NOTIFICACIONES EN TIEMPO REAL

Profesora: Msig. Adriana Collaguazo

Paralelo: 1

Grupo: 4

- Debbie Donoso
- Guillermo Merizalde
- Jefferson Vega
- Cesar Vera

Enlace del Repositorio:

Fecha de inicio: 15 de julio 2022

Fecha de finalización: 21 de julio 2022

RESULTADOS DE LAS ACTIVIDADES PLANTEADAS

PRIMERA PANTALLA – MAIN ACTIVITY

En el Desarrollo de la pantalla MainActivity se hace uso de un botón para que se pueda pasar a la pantalla Registro mediante la clase Intent.

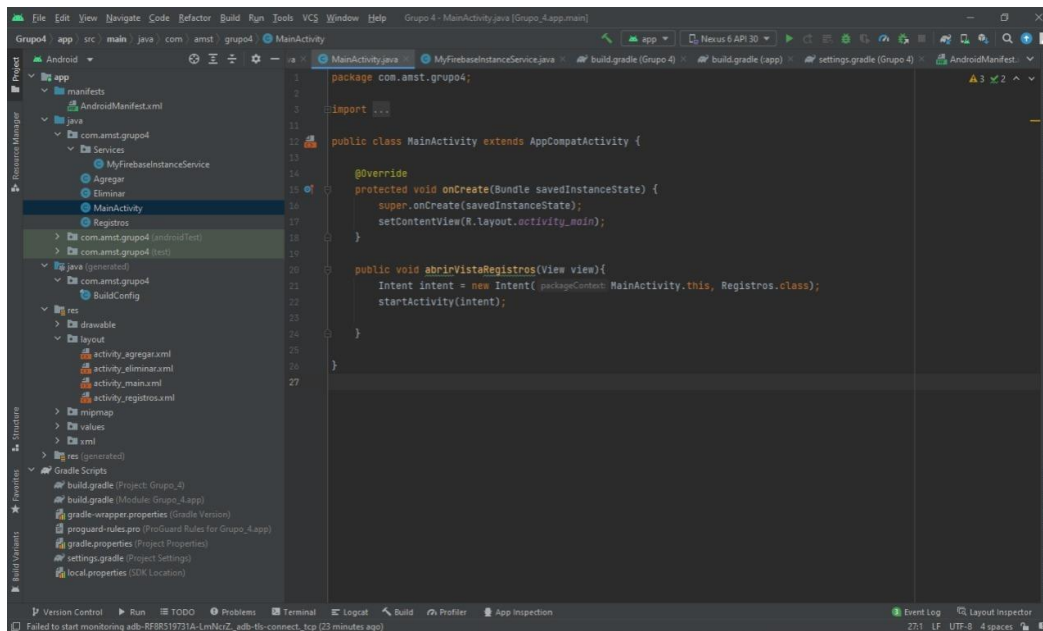


Ilustración 1 Código de MainActivity.java.

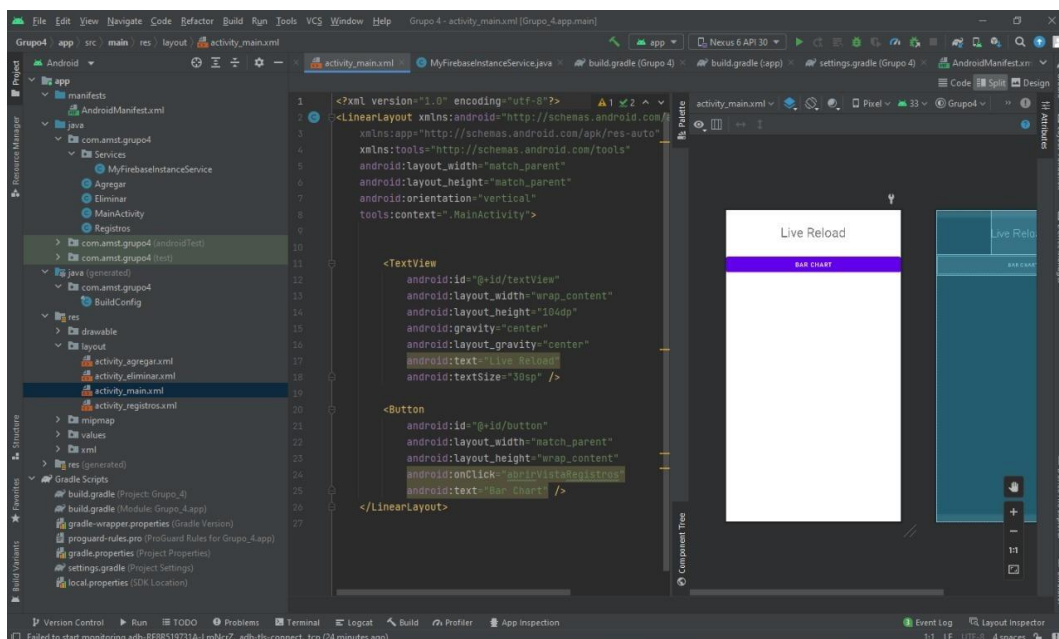


Ilustración 2 Código MainActivity.xml.

SEGUNDA PANTALLA – REGISTRO

En la segunda pantalla Registro.java se realizó la lógica de la pantalla en donde se hace las peticiones de los valores de temperatura y la fecha en que se registró el valor, por medio de la clase JsonRequestArrayRequest, después se hace el llenado de grafico donde se realiza la lógica con la que se representara los valores en el gráfico, posteriormente se realiza la actualización del grafico donde se mandara los datos obtenidos en el api al gráfico.

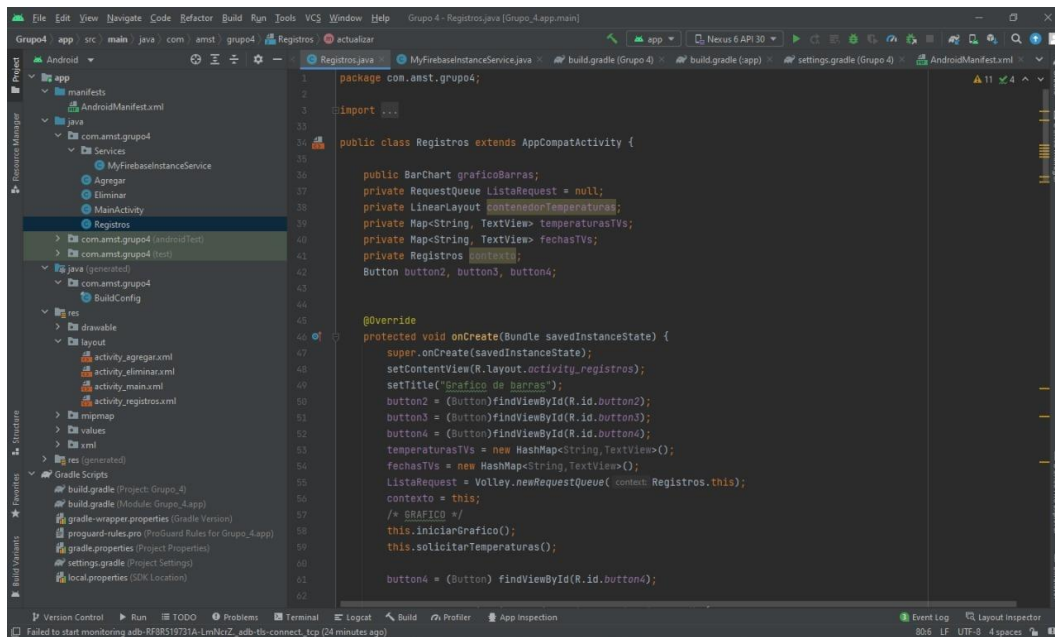


Ilustración 3 Código Registro.java.

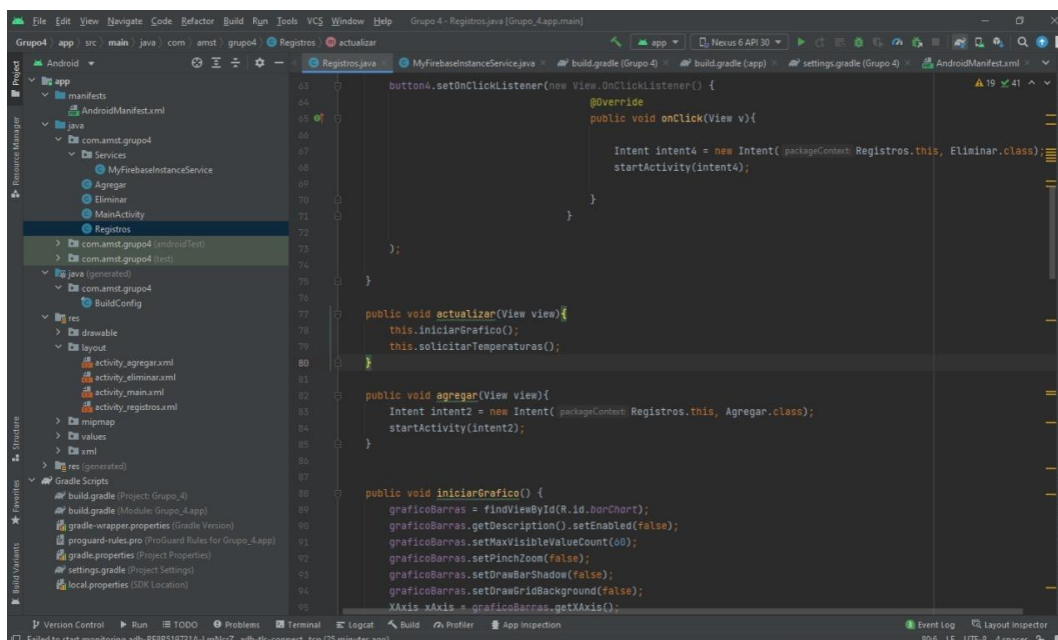


Ilustración 4 Código Registro.java.

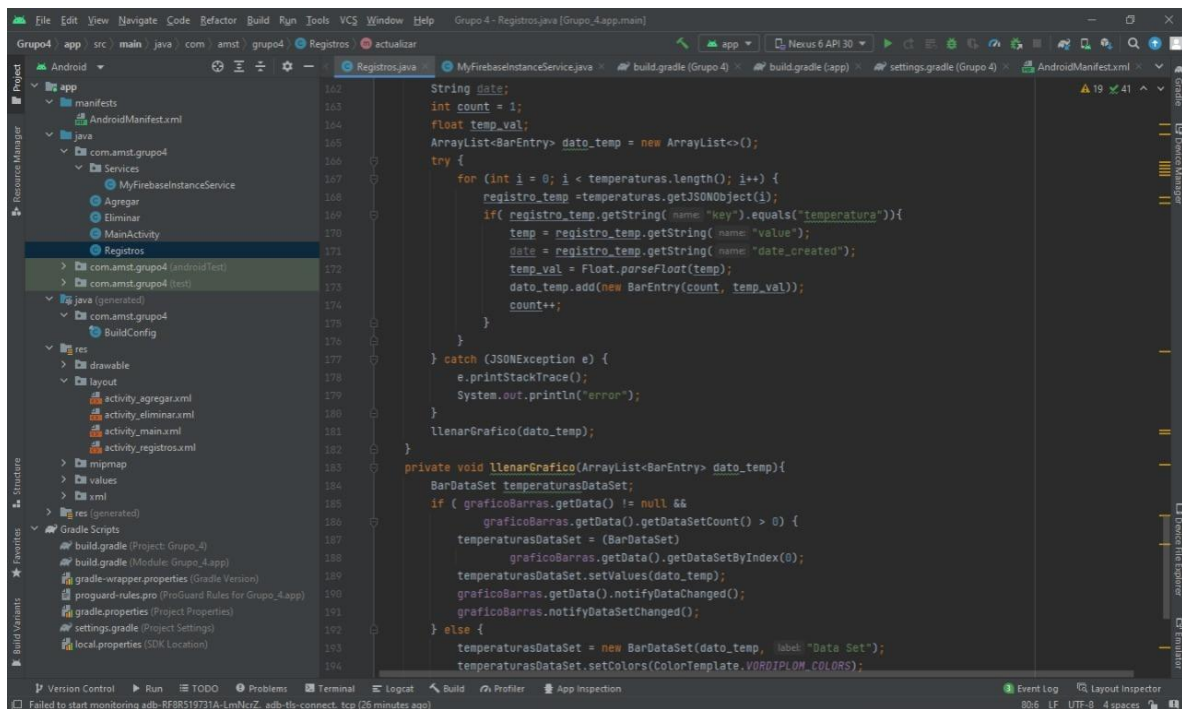


Ilustración 7. Código Registro.java

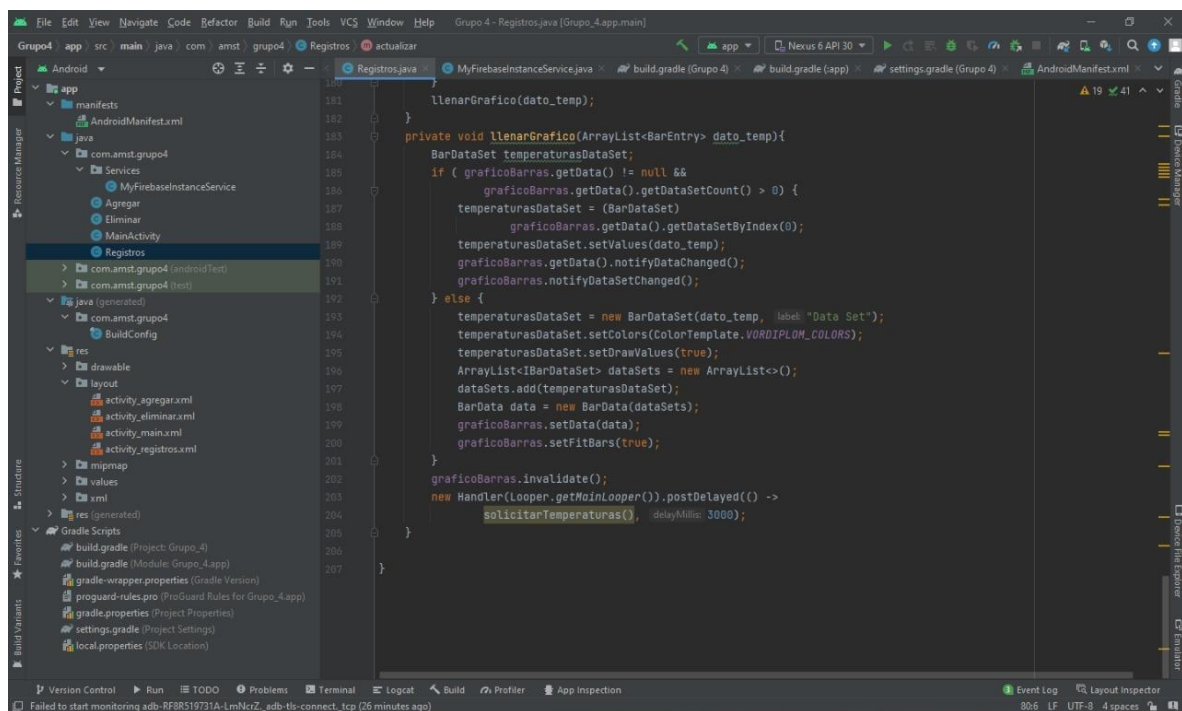


Ilustración 8. Código Registro.java

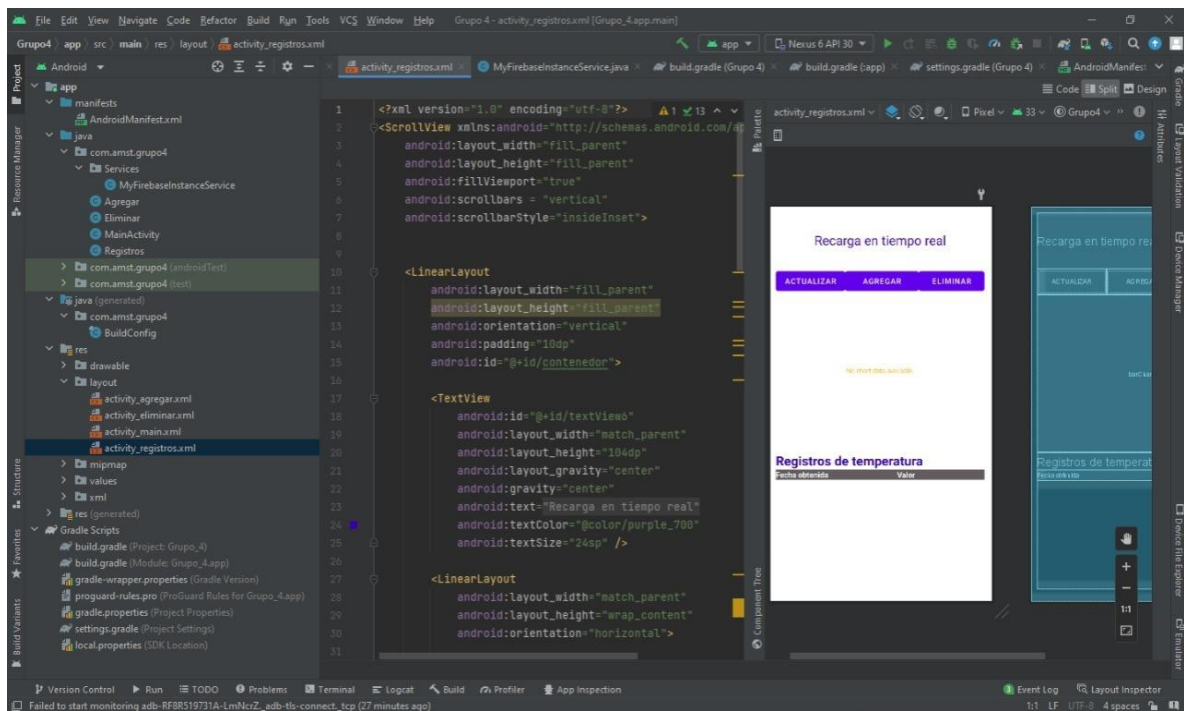


Ilustración 9 Código `activity_registro.xml`

TERCERA PANTALLA – AGREGAR

En la tercera pantalla “Agregar” se realizó el xml correspondiente para que el usuario puede ingresar una temperatura que desee añadir a la base de datos, además de que el usuario también debe ingresar la fecha en la que está añadiendo la nueva temperatura.

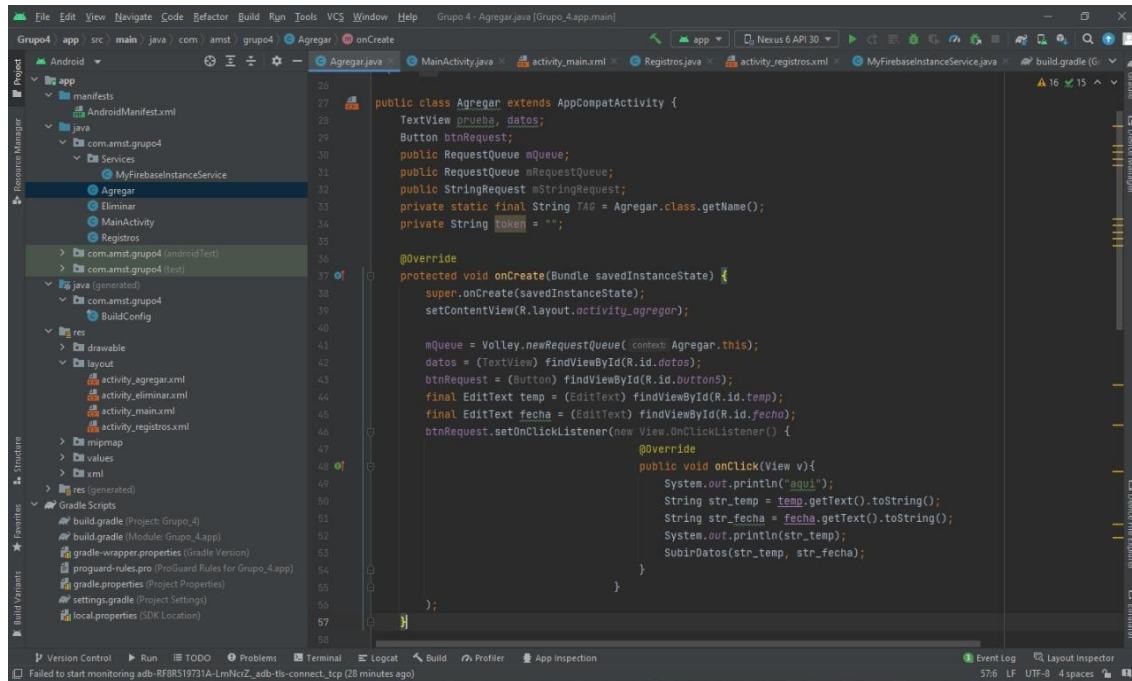


Ilustración 10. Código Agregar.java, se observa que se inicializan los TextView, botones y EditText, además dentro del OnClick del botón “Registrar datos” se obtienen los valores ingresados por el usuario (valor de temperatura y fecha), estos valores son pasados a la función “subirDatos”.

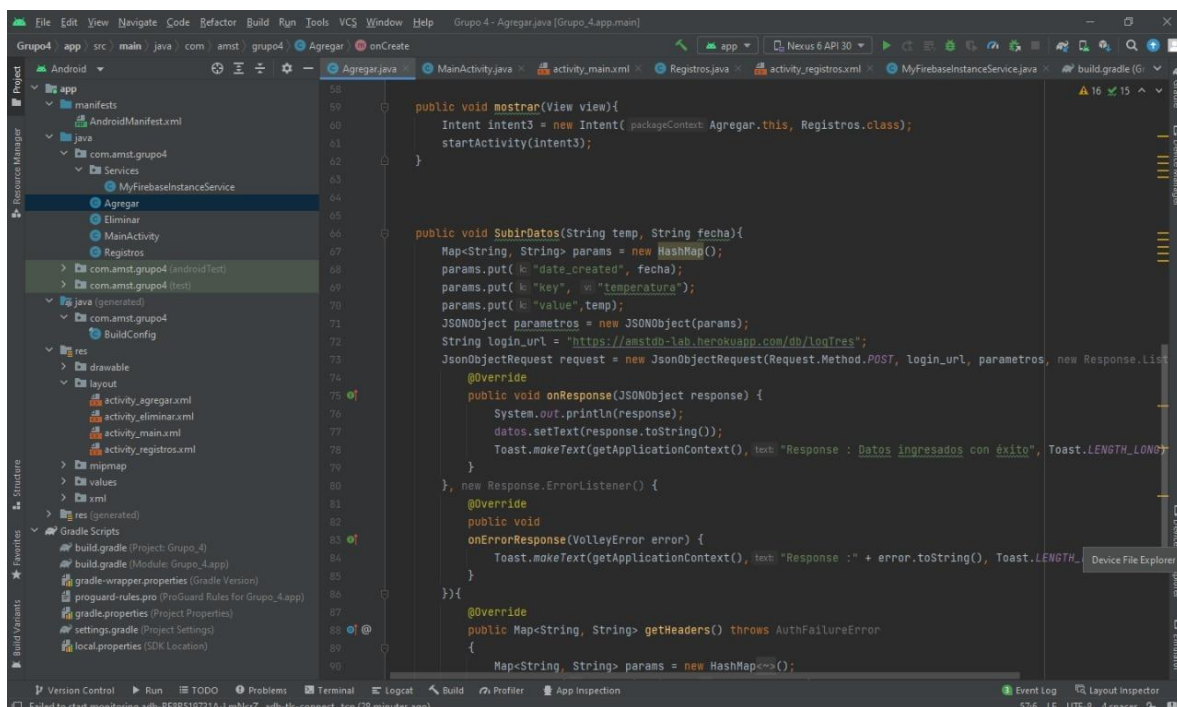


Ilustración 11. Código Agregar.java, se observa que se inicia un nuevo intent cuando se presione el botón “Mostrar actualización”, también se observa la función subirDatos, la cual recibe los datos ingresados por el usuario y los envía a la base de datos a través del método POST, se muestra un mensaje (toast) cada que el dato haya sido ingresado con éxito.

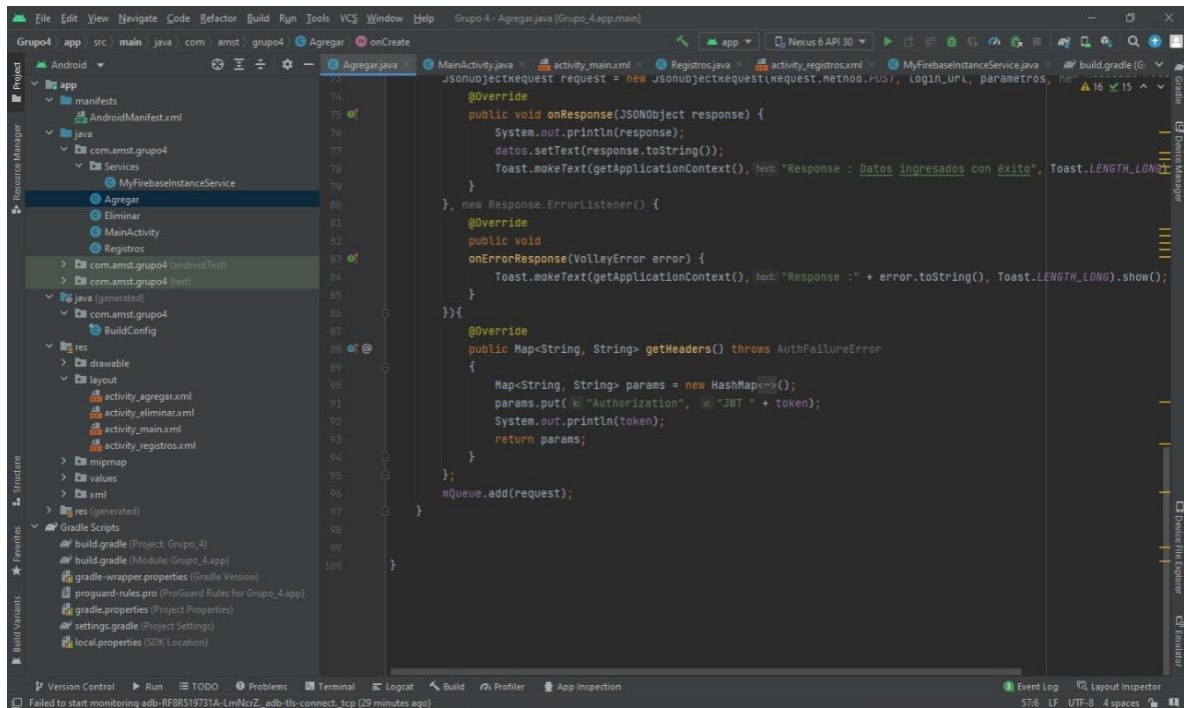


Ilustración 12. Código Agregar.java

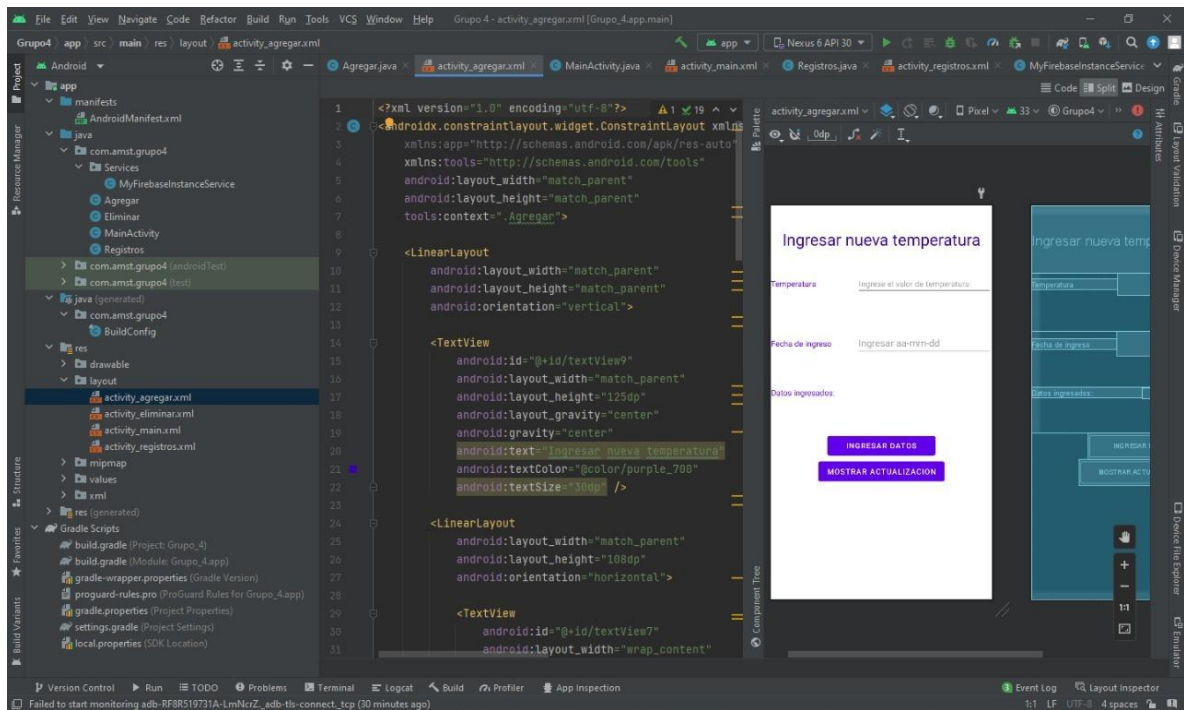


Ilustración 13. Código activity_agregar.xml, se agregó un LinearLayout vertical, dentro de éste se agregó un LinearLayout horizontal, el cual contiene un TextView y EditText para que el usuario pueda ingresar el valor correspondiente, esto mismo se agregó para que el usuario pueda ingresar la fecha, también se agregó otro LinearLayout horizontal, el cual contiene dos TextView para poder mostrar los datos agregados, finalmente se agregaron dos botones para poder agregar los datos y el otro para poder visualizar en el gráfico y la lista de la ventana Registros los nuevos datos.

CUARTA PANTALLA – ELIMINAR

En la cuarta pantalla “Eliminar” se realizó un Activity para poder mostrar en dos listas los id con sus respectivas temperaturas, de esta manera el usuario podría seleccionar el id de la temperatura que desea eliminar de la base, ya que el url para el método Delete requiere del id del objeto que se desea eliminar.

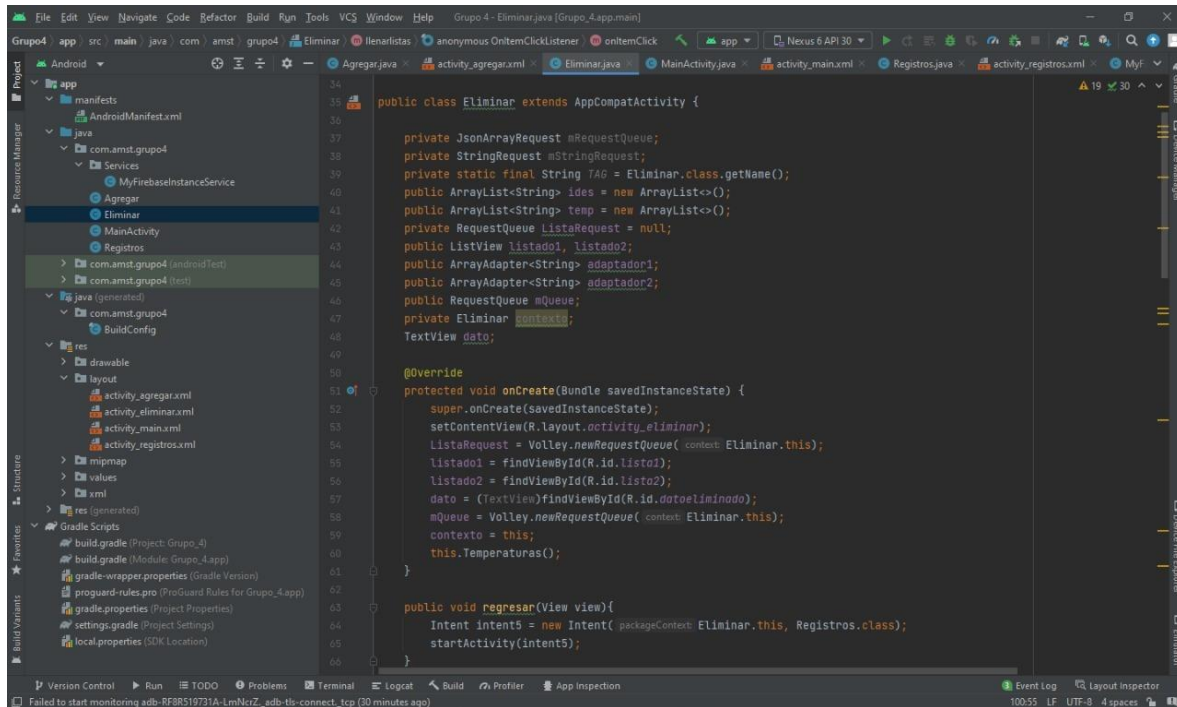


Ilustración 14. Código Eliminar.java

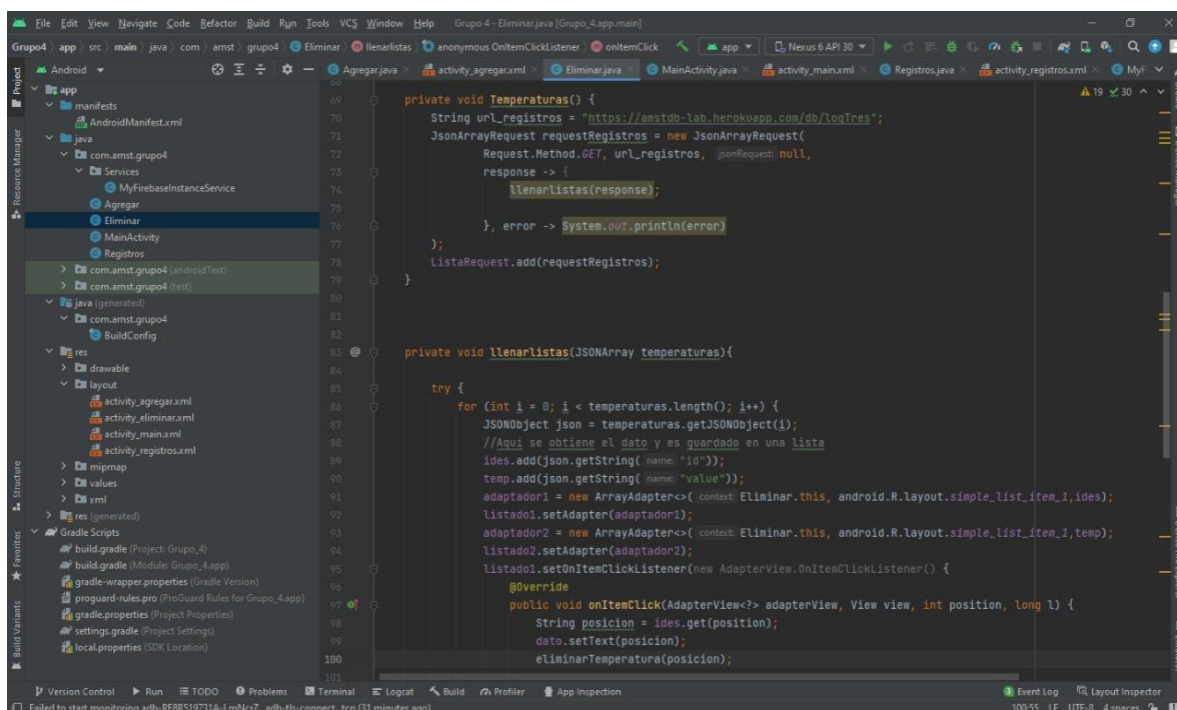


Ilustración 15. Se observa la función llenarListas, la cual recibe un jsonarray y con la ayuda de un for se extraen los datos de temperatura y fecha, éstos son agregados en las respectivas listas creadas en el xml.

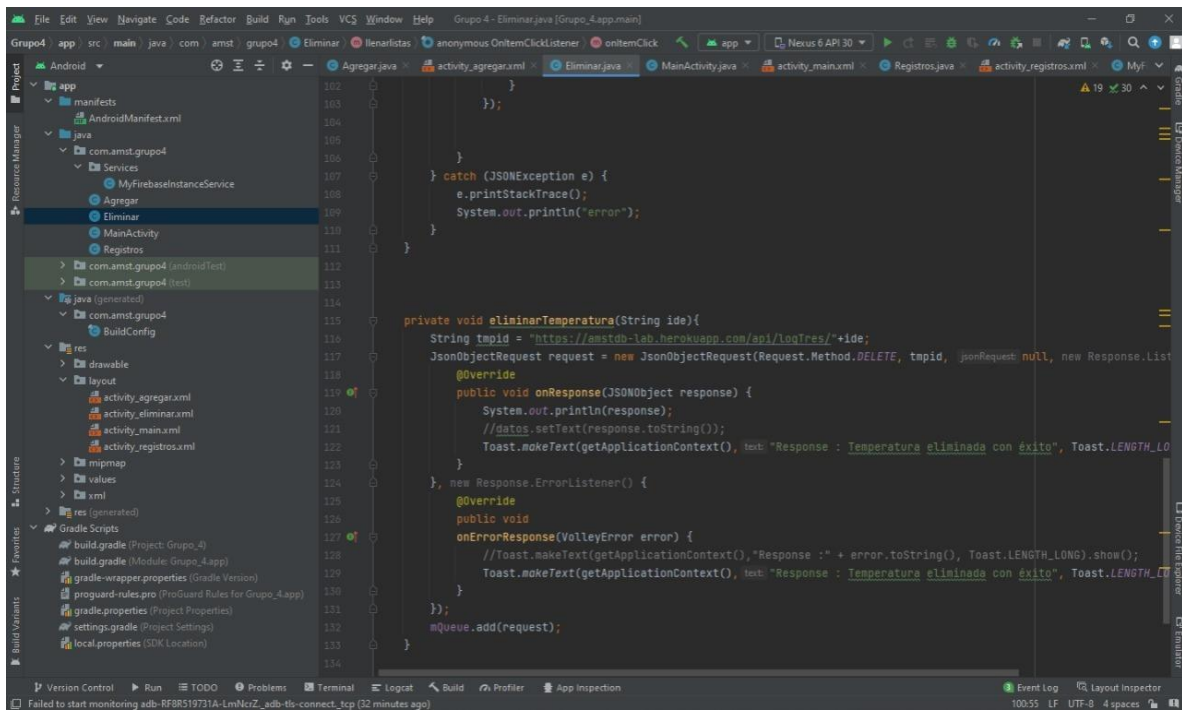


Ilustración 16. Se observa la función eliminarTemperatura, el cual recibe el id de la temperatura seleccionada y a través del método Delete elimina el dato, además cuando se haya eliminado de manera adecuada se muestra un mensaje (Toast).

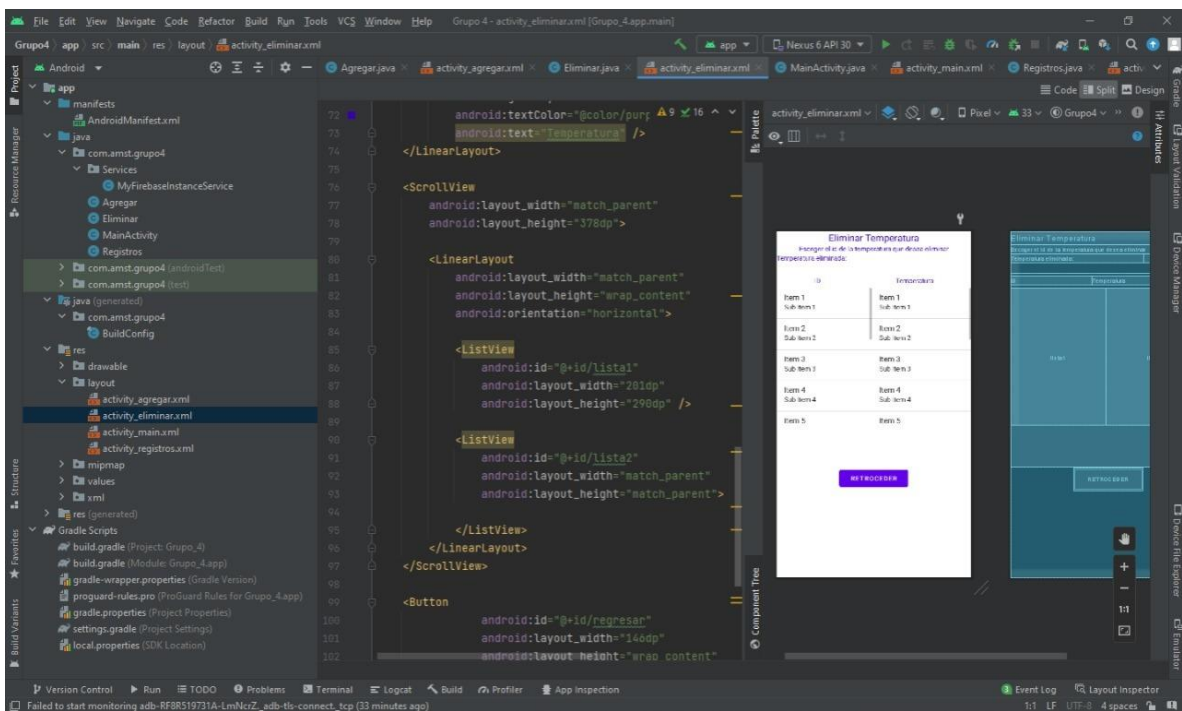


Ilustración 17. Código activity_eliminar.xml

ANDROID MANIFEST

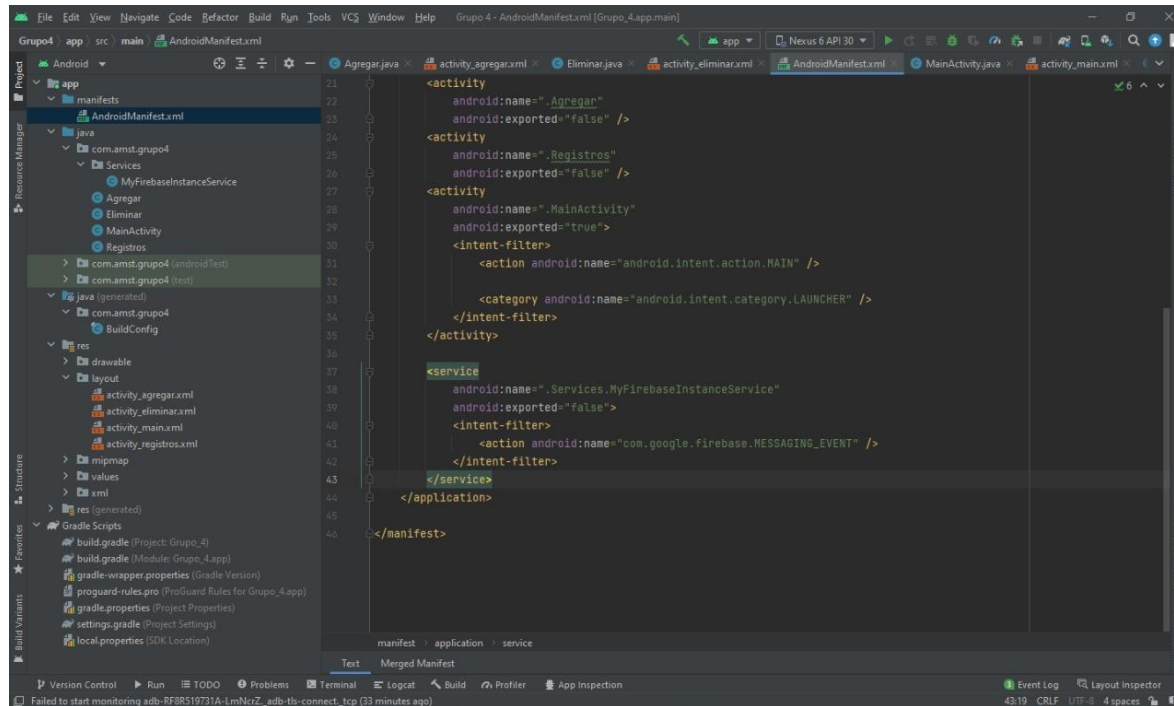


Ilustración 18. Se agrega el servicio creado para poder recibir notificaciones

MyFirebaseInstanceService

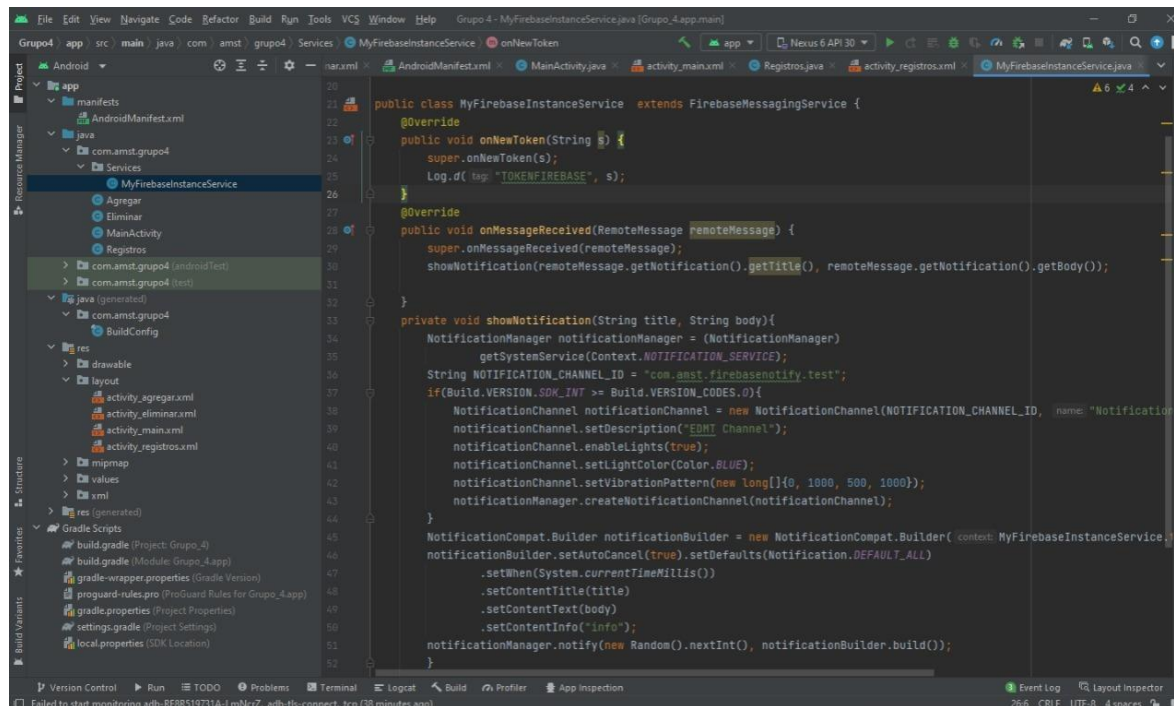


Ilustración 19. Código de la clase de java creada para poder recibir notificaciones desde firebase.

The screenshot shows the Firebase Cloud Messaging console for the project 'lab04-amst'. The left sidebar contains navigation links for 'Descripción general de...', 'Accesos directos a proyectos', 'Cloud Messaging', 'Categorías de producto', 'Compilación', 'Lanzamiento y supervisión', 'Analytics', 'Participación', 'Todos los productos', 'Personaliza tu navegación', and 'Spark'. The main content area is titled 'Cloud Messaging' and has tabs for 'Notificaciones' and 'Informes'. A search bar at the top allows filtering by campaign name, message content, or campaign status. Below the search bar is a table of notifications.

Notifications	Estado	Plataforma	Inicio/Envío	Finalizar	Enviados	Abiertos
lab4_amst Esto es una notificación	✓ Finalizada		21 jul 2022 21:12	—	<1,000	0 %
lab4_amst Esto es una notificación	✓ Finalizada		21 jul 2022 20:59	—	<1,000	0 %
lab4_amst esto es una prueba	✓ Finalizada		21 jul 2022 20:34	—	<1,000	0 %
prueba2 hola	✓ Finalizada		21 jul 2022 19:53	—	<1,000	0 %
prueba alo	✓ Finalizada		20 jul 2022 20:25	—	<1,000	0 %
Notificación grupo 4 Esto es una notificación.	✓ Finalizada		20 jul 2022 19:22	—	<1,000	0 %

At the bottom of the table, it says '0/10 Notificaciones recurrentes'.

Ilustración 20. Se observan los mensajes enviados desde firebase.

The screenshot shows the 'Configuración de proyecto' (Project Configuration) page for the Android app 'com.amst.grupo4'. The left sidebar is the same as in the previous image. The main content area is titled 'Configuración de proyecto' and has a tab for 'Agregar una app'. Below the tab is a list of apps for Android, with 'com.amst.grupo4' selected. To the right of the list is the 'Configuración del SDK' (SDK Configuration) section.

Configuración del SDK

¿Necesitas volver a configurar los SDK de Firebase en tu app? Revisa las instrucciones de configuración del SDK o descarga el archivo de configuración con las claves y los identificadores de tu app.

[Ver las instrucciones del SDK](#) [google-services.json](#)

ID de la app: 1:48080251962:android:7cb669f52b282b045a9903

Sobrenombre de la app: Agregar un sobrenombre

Nombre del paquete: com.amst.grupo4

Huellas digitales del certificado SHA: ca:ff:f3:68:fd:96:d4:ce:8d:2b:f2:db:d2:86:fb:94:02:74:d9:91 Tipo: SHA-1

[Agregar huella digital](#)

[Quitar esta app](#)

Ilustración 21. Se agregó la Huella digital del certificado Sha en el proyecto creado en firebase.

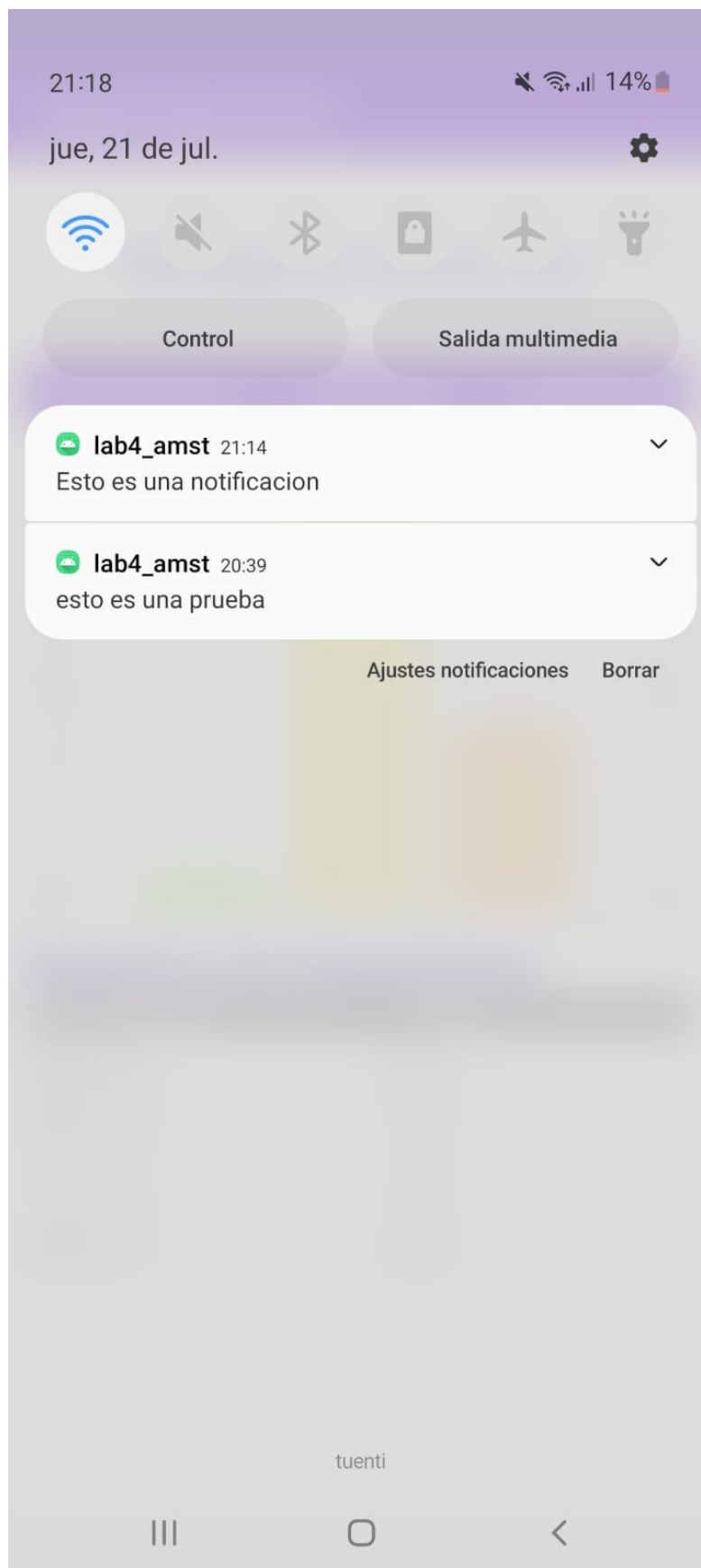


Ilustración 22. Notificación recibida.

build.gradle:app

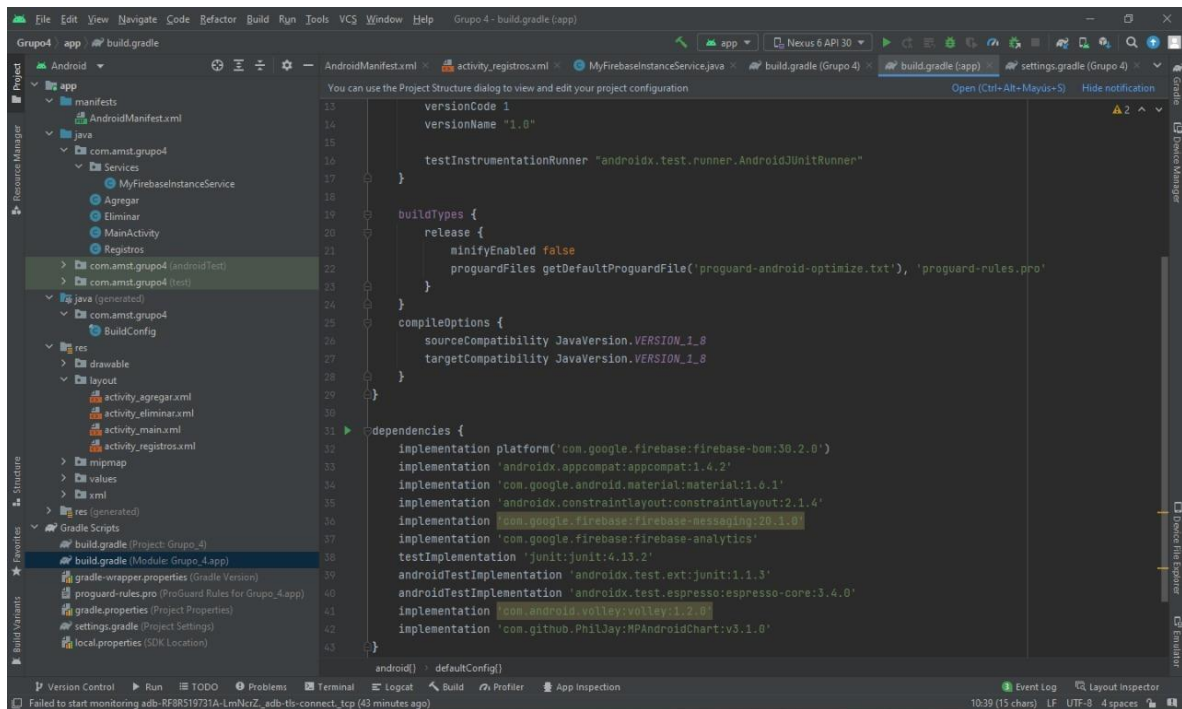


Ilustración 23. Se agregaron las dependencias necesarias para poder recibir notificaciones y poder implementar el gráfico de barras.

RESULTADOS



Live Reload



BAR CHART

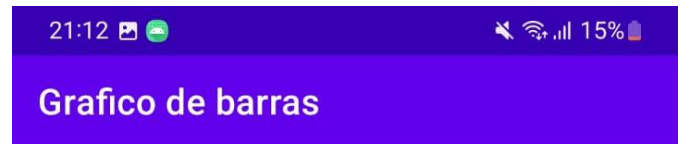


Grafico de barras

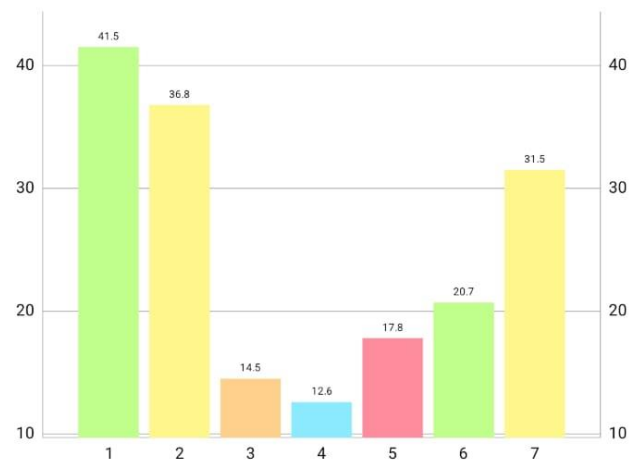
Recarga en tiempo real



ACTUALIZAR

AGREGAR

ELIMINAR



Registros de temperatura

Fecha obtenida	Valor
2022-07-22	41.5 C
2022-07-22	36.8 C
2022-07-22	14.5 C
2022-07-22	12.6 C
2022-07-22	17.8 C
2022-07-22	20.7 C
2022-07-22	31.5 C



Ilustración 24. Ventana de inicio



Ilustración 25. Se muestran las temperaturas obtenidas de la base y se muestran en un gráfico.



Temperatura

Fecha de ingreso

Datos ingresados:

INGRESAR DATOS

MOSTRAR ACTUALIZACION



Ilustración 27. Ventana para agregar una nueva temperatura.



Temperatura

Fecha de ingreso

```
{ "id": 57, "date_created": "2022-07-21", "key": "temperatura", "value": 25 }
```

INGRESAR DATOS

MOSTRAR ACTUALIZACION

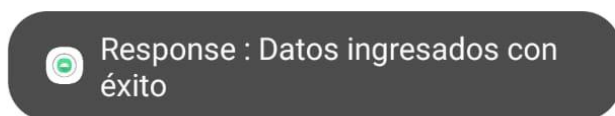
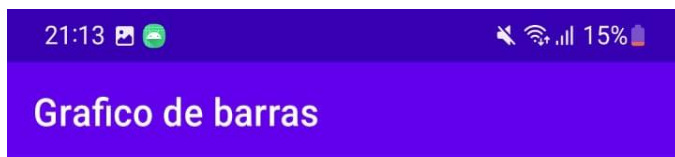
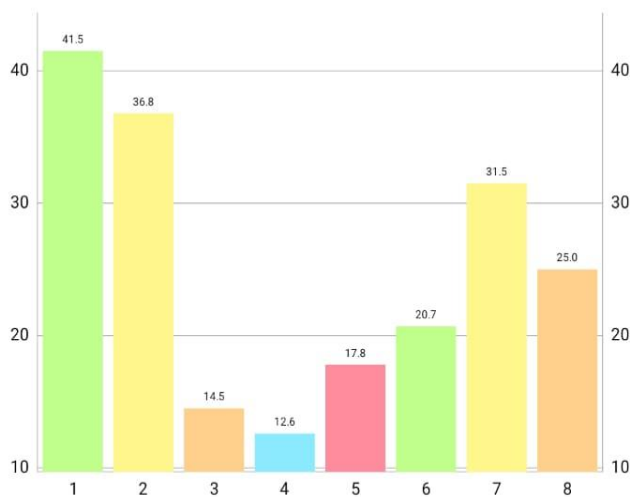


Ilustración 26. Se agrega una nueva temperatura.



Recarga en tiempo real

ACTUALIZAR AGREGAR ELIMINAR



Registros de temperatura

Fecha obtenida	Valor
2022-07-22	41.5 C
2022-07-22	36.8 C
2022-07-22	14.5 C
2022-07-22	12.6 C
2022-07-22	17.8 C
2022-07-22	20.7 C
2022-07-22	31.5 C
2022-07-21	25.0 C



Ilustración 28. Se observa la nueva temperatura agregada, la cual se muestra al final de la fila.



Eliminar Temperatura

Escoger el id de la temperatura que desea eliminar
Temperatura eliminada: 52

ID	Temperatura
50	41.5
51	36.8
52	14.5
53	12.6
54	17.8
55	20.7

RETROCEDER


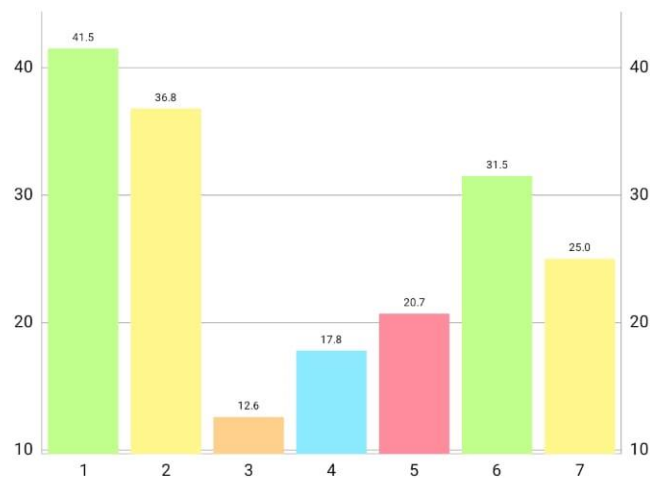
 Response : Temperatura eliminada con éxito



Ilustración 29. Se muestra que se eliminó la temperatura 14.5.

Recarga en tiempo real

ACTUALIZAR AGREGAR ELIMINAR



Registros de temperatura

Fecha obtenida	Valor
2022-07-22	41.5 C
2022-07-22	36.8 C
2022-07-22	12.6 C
2022-07-22	17.8 C
2022-07-22	20.7 C
2022-07-22	31.5 C
2022-07-21	25.0 C

PREGUNTAS DE INVESTIGACION

¿Cuál es la diferencia entre Soft Real time y Hard Real time?

Hard Real Time

En el sistema de tiempo real duro, el requisito de tiempo es una restricción crítica. El sistema debe funcionar dentro del plazo. Si el sistema no funcionó dentro del plazo, se considera una falla de la tarea. Este tipo de sistemas no deben perder la fecha límite. No cumplir con la fecha límite puede ser catastrófico. Los sistemas de control de tráfico aéreo, misiles y sistemas de control de reactores nucleares son algunos ejemplos de sistemas de tiempo real duro. Si el sistema de control de la aeronave no dio las instrucciones a la aeronave dentro del plazo, puede causar que la aeronave se estrelle. Por lo tanto, en un sistema de tiempo real duro, cumplir con la fecha límite es extremadamente importante. Estos sistemas se implementan principalmente en sistemas críticos de seguridad

Soft Real Time

En un sistema de tiempo real suave, el requisito de tiempo no es muy crucial. El sistema debe realizar la tarea o dar el resultado dentro del plazo, pero ocasionalmente puede haber una pequeña tolerancia. Si el sistema no realizó la tarea dentro del plazo, no se considera una falla siempre que proporcione el resultado requerido. Pero se considera que el rendimiento está degradado. El incumplimiento de la fecha límite no provocará un evento catastrófico como en un sistema de tiempo real estricto. Estos sistemas son menos restrictivos. Algunos ejemplos de sistemas de software en tiempo real son la transmisión multimedia, los proyectos científicos avanzados y la realidad virtual.

En el dashboard, actualizamos una lista y un gráfico donde presentamos las temperaturas. ¿Qué otros componentes podemos actualizar (como tablas, edittexts, etc)?

Uno de los componentes en donde es posible realizar actualizaciones son, ListView, ImageSwitcher, TextSwitcher, RelativeLayout, FrameLayout

Las apps de juegos utilizan hilos para el manejo de componentes (personaje, escenario, música), ¿Que otro tipo de aplicaciones (redes sociales, filtros de fotos, etc) utiliza hilos y con qué objetivo?

De acuerdo con el sistema operativo y los requisitos de la aplicación, el proceso se puede asignar a un único subproceso o a varios subprocesos. Cuando al proceso de aplicación se le asignan múltiples subprocesos, entonces se necesita ejecutar estos múltiples subprocesos simultáneamente. Esta técnica de ejecutar o ejecutar varios subprocesos de forma simultánea o concurrente se conoce como subprocesos múltiples.

Multithreading simplemente significa que se tiene más de un hilo ejecutándose dentro de la misma aplicación. Por ejemplo, una aplicación de escritorio que proporciona funciones como edición, impresión, etc. es una aplicación multiproceso. En esta aplicación, como la impresión es un proceso

en segundo plano, se puede editar documentos e imprimir documentos al mismo tiempo asignando estas funciones a dos subprocesos diferentes.

Los subprocesos en las aplicaciones de subprocesos múltiples se ejecutan en paralelo entre sí de manera concurrente. Por lo tanto, los subprocesos múltiples también forman parte de la concurrencia en Java. Tenga en cuenta que, aunque hay varios subprocesos, comparten el área de memoria, por lo que ahorran memoria. Además, los hilos pueden cambiar fácilmente de contexto en poco tiempo. El subprocesamiento múltiple es principalmente útil ya que proporciona la ejecución simultánea de dos o más partes de una aplicación. Esto permite que la aplicación utilice el tiempo de la CPU al máximo y que el tiempo de inactividad se mantenga al mínimo.

¿Qué otros servicios de Firebase podrían ser útil para el desarrollo de dispositivos IOT y como los usaría?

- **Firebase Remote Config:** es un servicio de nube que te permite cambiar el comportamiento y el aspecto de la app sin que los usuarios tengan que descargar una actualización. Se podría ser utilizado para una respuesta en tiempo real de los valores que se obtengan de sensores.

Durante el taller creamos un package Servicios, donde alojamos el servicio de could messaging. ¿Que otro servicio (no de Firebase) podemos utilizar en las aplicaciones móviles?

La reproducción de música de fondo es un ejemplo muy común de servicios en Android. Desde el momento en que un usuario inicia el servicio, la música se reproduce continuamente en segundo plano incluso si el usuario cambia a otra aplicación. El usuario debe detener el servicio explícitamente para pausar la música.

Conclusiones

- Mediante la utilización de la clase Handler se hizo posible ejecutar un subproceso donde realizaría una consulta al api para la actualización del gráfico, de esta manera podemos demostrar que al usar hilos en las aplicaciones móviles podremos realizar actualizaciones en tiempo real.
- Firebase nos ofrece distintos servicios que pueden ser implementados en nuestro proyecto móvil, tales como Remote Cofig o Cloud Messaging. En la práctica se hizo uso de Cloud Messaging en donde se pudo recibir notificaciones directamente al teléfono cuando un mensaje llegaba al Cloud Messaging.
- Los métodos POST, DELETE Y GET permiten trabajar de manera óptima en la creación de una aplicación que utilice base de datos y requiera actualizaciones en tiempo real, pues al combinarlos y mediante las diferentes funciones que se pueden crear en los .java y los elementos que se puedan agregar en el .xml se puede obtener una aplicación óptima que cumpla con los requerimientos del usuario.

Recomendaciones

- Realizar la transcripción del código de manera manual para que de esta manera poder entender la lógica con la del proyecto y poder realizar un buen informe.
- No olvidar instanciar las variables que se requieren en el método onCreate.
- Revisar la documentación de todas las clases que se usaron en la realización del proyecto para que de esta manera se tenga el conocimiento del porqué se la utiliza.

References

- [1] Lithmee, "Difference Between Hard and Soft Real Time System," 18 April 2018. [Online]. Available: <https://www.differencebetween.com/difference-between-hard-and-vs-soft-real-time-system/>. [Accessed 07 July 2022].
- [2] "firebase," 15 06 2022. [Online]. Available: <https://firebase.google.com/docs/remote-config>. [Accessed 21 07 2022].
- [3] "developer," 03 05 2021. [Online]. Available: <https://developer.android.com/guide/topics/ui/custom-components?hl=es-419>. [Accessed 21 07 2022].
- [4] acervolima, "SERVICIOS EN ANDROID CON EJEMPLO," 2022. [Online]. Available: <https://es.acervolima.com/servicios-en-android-con-ejemplo/>.
- [5] S. Testing, "Multithreading In Java – Tutorial With Examples," 16 July 2022. [Online]. Available: <https://www.softwaretestinghelp.com/multithreading-in-java/>. [Accessed 21 July 2022].

Link repositorio:

En el repositorio se encuentra el proyecto y el apk de la aplicación.

<https://github.com/Daniella252018/amst-lab4-g4>