|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Materia:** |  | MOVIL II | | |
| **Práctica:** |  | BroadcastReceiver y Telefonía: Respuesta automática. | | |
| **Alumno(s):** |  | Francisco Avalos Murillo | **Fecha:** | 23/02/2020 |

**Objetivo**

Es crear una app con la herramienta que nos proporciona Android ( BroadcastReceiver ) en donde dicha aplicación debe mandar un mensaje de texto en cuanto le hable por teléfono al dispositivo que se le instale dicha app

**Material**

Android Studio

Celular (Samsung Galaxy S6)

saldo

**Marco Teorico**

Las apps de Android pueden enviar o recibir mensajes de emisión desde el sistema de Android y otras apps para Android, de forma similar al patrón de diseño de [publicación y suscripción](https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern). Estas emisiones se envían cuando ocurre un evento de interés. Por ejemplo, el sistema Android envía emisiones cuando ocurren diferentes eventos del sistema, como cuando este se inicia o cuando el dispositivo comienza a cargarse. Las apps también pueden enviar emisiones personalizadas, por ejemplo, para notificar a otras apps sobre algo que podría interesarles (como cuando se descargaron algunos datos nuevos).

Por lo general, las emisiones pueden usarse como un sistema de mensajería entre apps y fuera del flujo de usuarios normal. Sin embargo, debes tener cuidado de no abusar de la oportunidad de responder a las emisiones y ejecutar tareas en segundo plano que puedan contribuir a ralentizar el rendimiento del sistema, como se explica en el siguiente video.

Las apps pueden recibir emisiones de dos maneras: mediante receptores declarados en el manifiesto y mediante receptores registrados en el contexto.

### Receptores declarados en el manifiesto

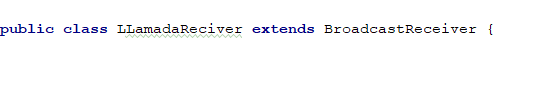
Si declaras un receptor de emisión en tu manifiesto, el sistema inicia la app (si aún no está en ejecución) cuando se envía la emisión.

**Desarrollo**

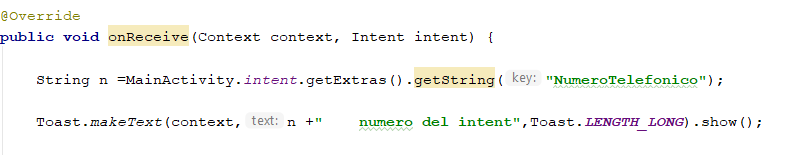
Primero hay que darle los permisos en el manifiesto para que se puedan aplicar las acciones requeridas para la función de la app



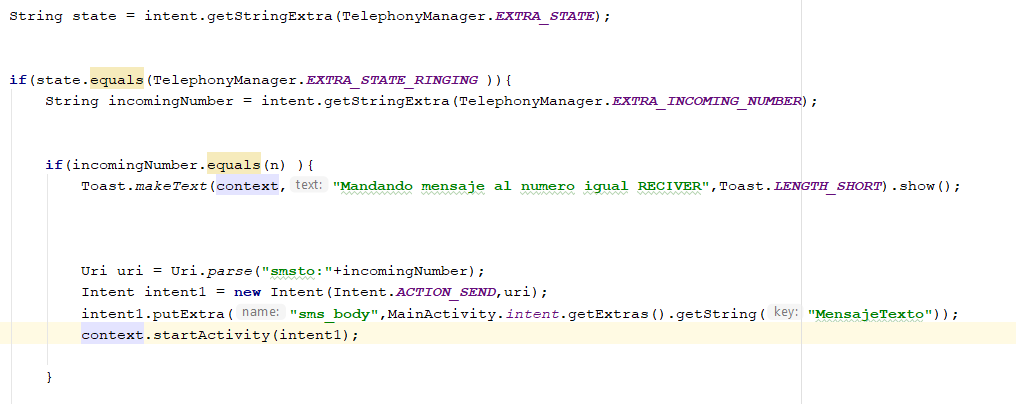
Después creamos una clase en mi caso le puse llamadareciber esta clase extiende hasta broadcastReciber.



Dentro del método onRecive pongo un try catch después creo una variable n simplemente para traerme el numero de la caja de texto que ingresa el usuario en la interfaz, luego lo imprimo con un Toast para ver que si se esta obteniendo bien el numero.



Lo siguiente es un crear otra variable en mi caso le puse state, esta variable lo que hace es guardar una función del método TelephonyManager lego creo un if donde dicha variable la comparo para ver si el celular esta sonando(recibiendo llamada) si es así guardo el numero en una variable llamada incomingnumero, luego esa variable la comparo con un if con la variable que ya habíamos creado antes que es n para ver si son el mismo numero y poder mandarle el mensaje de texto al número correcto.



**Resultado**

**Bibliografía**

[https://drive.google.com/file/d/0B78MaaWLpIezX08yZzU2eTlmR1E/view?usp=sharing](https://drive.google.com/file/d/0B78MaaW#LpIezX08yZzU2eTlmR1E/view?usp=sharing)