

# Intents comunes

Una intent permite iniciar una actividad en otra app mediante la descripción de una acción simple que desees realizar (como "ver un mapa" o "tomar una foto") en un objeto **Intent** (<https://developer.android.com/reference/android/content/Intent?hl=es-419>). Este tipo de intent se conoce como intent *implícita* porque no especifica el componente de la app que se iniciará, sino una *acción*, y brinda algunos *datos* con los que debe realizarse la acción.

Al llamar a **startActivity()**

([https://developer.android.com/reference/android/content/Context?hl=es-419#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/content/Context?hl=es-419#startActivity(android.content.Intent)))

o **startActivityForResult()**

([https://developer.android.com/reference/android/app/Activity?hl=es-419#startActivityForResult\(android.content.Intent,%20int\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#startActivityForResult(android.content.Intent,%20int)))

y pasarle una intent implícita, el sistema **resuelve la intent**

(<https://developer.android.com/guide/components/intents-filters?hl=es-419#Resolution>) en una aplicación que puede manejarla e iniciar su **Activity**

(<https://developer.android.com/reference/android/app/Activity?hl=es-419>) correspondiente. Si existe más de una app que puede manejar la intent, el sistema presenta al usuario un cuadro de diálogo para que elija la app que desea usar.

En esta página, se describen varias intents implícitas que pueden usarse para realizar acciones comunes, organizadas por el tipo de app que maneja la intent. En cada sección, también se muestra la manera en que puedes crear un **filtro de intents**

(<https://developer.android.com/guide/components/intents-filters?hl=es-419#Receiving>) para publicitar la capacidad que tu app tiene de realizar la misma acción.

**Advertencia:** Si no hay apps en el dispositivo que puedan recibir la intent implícita, tu app fallará cuando llame a **startActivity()**.

([https://developer.android.com/reference/android/content/Context?hl=es-419#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/content/Context?hl=es-419#startActivity(android.content.Intent)))

. Para verificar primero que exista una app para recibir la intent, llama a **resolveActivity()**.

([https://developer.android.com/reference/android/content/Intent?hl=es-419#resolveActivity\(android.content.pm.PackageManager\)](https://developer.android.com/reference/android/content/Intent?hl=es-419#resolveActivity(android.content.pm.PackageManager)))

en tu objeto **Intent** (<https://developer.android.com/reference/android/content/Intent?hl=es-419>). Si el resultado no es nulo, hay al menos una aplicación que puede administrar la intent y es seguro llamar a **startActivity()**.

([https://developer.android.com/reference/android/content/Context?hl=es-419#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/content/Context?hl=es-419#startActivity(android.content.Intent)))

. Si el resultado es nulo, no debes usar la intent y, si es posible, debes inhabilitar la función que la invoca.

Si no estás familiarizado con el modo de crear intents o filtros de intents, primero debes leer [Intent y filtros de intents](#)

(<https://developer.android.com/guide/components/intents-filters?hl=es-419>).

Si deseas obtener información sobre cómo desencadenar las intents enumeradas en esta página para el host de desarrollo, consulta [Cómo verificar intents con Android Debug Bridge](#) (#AdbIntents).

## Google Voice Actions

Las [Google Voice Actions](#) (<https://developers.google.com/voice-actions/?hl=es-419>) activan algunas de las intents enumeradas en esta página en respuesta a comandos de voz. Para obtener más información, consulta [Intents activadas por Google Voice Actions](#) ([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference)).

## Alarma

### Crear una alarma

Para crear una alarma nueva, usa la acción

[\*\*ACTION\\_SET\\_ALARM\*\*](#)



([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

#### **Google Voice Actions**

- "establecer una alarma para las 7 a.m."

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SET\\_ALARM](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SET_ALARM))

y especifica detalles como la hora y el mensaje con los extras definidos a continuación.

**Nota:** Los extras de hora, minutos y mensajes solo están disponibles en Android 2.3 (nivel de API 9) y versiones posteriores. Los demás extras se agregaron en versiones posteriores de la plataforma.

## Acción

### ACTION\_SET\_ALARM

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SET\\_ALARM](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SET_ALARM))

## URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Extras

### EXTRA\_HOUR

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_HOUR](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_HOUR))

La hora de la alarma.

### EXTRA\_MINUTES

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_MINUTES](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_MINUTES))

Los minutos de la alarma.

### EXTRA\_MESSAGE

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_MESSAGE](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_MESSAGE))

Un mensaje personalizado para identificar la alarma.

### EXTRA\_DAYS

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_DAYS](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_DAYS))

Un [ArrayList](https://developer.android.com/reference/java/util/ArrayList?hl=es-419) (<https://developer.android.com/reference/java/util/ArrayList?hl=es-419>) que incluye cada día de la semana en el que debe repetirse esta alarma. Cada día debe declararse con un entero de la clase [Calendar](#)

(<https://developer.android.com/reference/java/util/Calendar?hl=es-419>), como **MONDAY** (<https://developer.android.com/reference/java/util/Calendar?hl=es-419#MONDAY>).

Para una alarma por única vez, no especifique este extra.

#### **EXTRA\_RINGTONE**

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_RINGTONE](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_RINGTONE))

Un URI **content:** que especifica un tono para usar con la alarma, o

#### **VALUE\_RINGTONE\_SILENT**

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#VALUE\\_RINGTONE\\_SILENT](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#VALUE_RINGTONE_SILENT))

para ningún tono.

Para usar el tono predeterminado, no especifique este extra.

#### **EXTRA\_VIBRATE**

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_VIBRATE](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_VIBRATE))

Booleano que especifica si debe vibrar para esta alarma.

#### **EXTRA\_SKIP\_UI**

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_SKIP\\_UI](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_SKIP_UI))

Booleano que especifica si la aplicación que responde debe omitir su IU al establecer la alarma. Si el valor es true, la aplicación debe omitir cualquier IU de confirmación y simplemente establecer la alarma especificada.

### **Ejemplo de intent:**

**Kotlin** (#kotlin)**Java**  
(#java)

```
public void createAlarm(String message, int hour, int minutes) {
    Intent intent = new Intent(AlarmClock.ACTION_SET_ALARM)
        .putExtra(AlarmClock.EXTRA_MESSAGE, message)
        .putExtra(AlarmClock.EXTRA_HOUR, hour)
        .putExtra(AlarmClock.EXTRA_MINUTES, minutes);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

```
}  
}
```

**Nota:**

Para invocar la intent **ACTION\_SET\_ALARM**

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SET\\_ALARM](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SET_ALARM))

, tu app debe tener el permiso **SET\_ALARM**

([https://developer.android.com/reference/android/Manifest.permission?hl=es-419#SET\\_ALARM](https://developer.android.com/reference/android/Manifest.permission?hl=es-419#SET_ALARM)):

```
<uses-permission android:name="com.android.alarm.permission.SET_ALARM" />
```

**Filtro de intents de ejemplo:**

```
<activity ...>  
    <intent-filter>  
        <action android:name="android.intent.action.SET_ALARM" />  
        <category android:name="android.intent.category.DEFAULT" />  
    </intent-filter>  
</activity>
```

**Crear un temporizador**

Para crear un temporizador de cuenta regresiva, usa la acción **ACTION\_SET\_TIMER**



([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

**Google Voice Actions**

- "establecer temporizador de 5"

*minutos"*

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SET\\_TIMER](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SET_TIMER))

y especifica detalles del temporizador como la duración con los campos extras definidos a continuación.

**Nota:** Esta intent se agregó en Android 4.4 (nivel de API 19).

## Acción

### ACTION\_SET\_TIMER

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SET\\_TIMER](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SET_TIMER))

## URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Extras

### EXTRA\_LENGTH

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_LENGTH](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_LENGTH))

La duración del temporizador en segundos.

### EXTRA\_MESSAGE

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_MESSAGE](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_MESSAGE))

Un mensaje personalizado para identificar el temporizador.

### EXTRA\_SKIP\_UI

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA\\_SKIP\\_UI](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#EXTRA_SKIP_UI))

Booleano que especifica si la aplicación que responde debe omitir su IU al establecer el temporizador. Si el valor es true, la aplicación debe omitir

cualquier IU de confirmación y simplemente establecer el temporizador especificado.

### Ejemplo de intent:

Kotlin (#kotlin)Java

(#java)

```
public void startTimer(String message, int seconds) {
    Intent intent = new Intent(AlarmClock.ACTION_SET_TIMER)
        .putExtra(AlarmClock.EXTRA_MESSAGE, message)
        .putExtra(AlarmClock.EXTRA_LENGTH, seconds)
        .putExtra(AlarmClock.EXTRA_SKIP_UI, true);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

### Nota:

Para invocar la intent **ACTION\_SET\_TIMER**

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SET\\_TIMER](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SET_TIMER))

, tu app debe tener el permiso **SET\_ALARM**

([https://developer.android.com/reference/android/Manifest.permission?hl=es-419#SET\\_ALARM](https://developer.android.com/reference/android/Manifest.permission?hl=es-419#SET_ALARM)):

```
<uses-permission android:name="com.android.alarm.permission.SET_ALARM" />
```

### Filtro de intents de ejemplo:

```
<activity ...>
    <intent-filter>
        <action android:name="android.intent.action.SET_TIMER" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

## Mostrar todas las alarmas

Para mostrar la lista de alarmas, usa la acción **ACTION\_SHOW\_ALARMS**  
([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SHOW\\_ALARMS](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SHOW_ALARMS))

Aunque no muchas aplicaciones invocarán esta intent (la usan principalmente las del sistema), cualquier aplicación que se comporte como alarma debe implementar este filtro de intents y responder mostrando la lista de alarmas actuales.

**Nota:** Esta intent se agregó en Android 4.4 (nivel de API 19).

### Acción

**ACTION\_SHOW\_ALARMS**

([https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION\\_SHOW\\_ALARMS](https://developer.android.com/reference/android/provider/AlarmClock?hl=es-419#ACTION_SHOW_ALARMS))

### URI de datos

Ninguno

### Tipo de MIME

Ninguno

### Filtro de intents de ejemplo:

```
<activity ...>
  <intent-filter>
    <action android:name="android.intent.action.SHOW_ALARMS" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

## Calendario



## Agregar un evento del calendario

Para agregar un evento nuevo al calendario del usuario, usa la acción `ACTION_INSERT` ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_INSERT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_INSERT)) y especifica el URI de datos con `Events.CONTENT_URI` ([https://developer.android.com/reference/android/provider/CalendarContract.Events?hl=es-419#CONTENT\\_URI](https://developer.android.com/reference/android/provider/CalendarContract.Events?hl=es-419#CONTENT_URI))

. Después, puedes especificar diversos detalles del evento con los extras definidos a continuación.

### Acción

#### `ACTION_INSERT`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_INSERT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_INSERT))

### URI de datos

#### `Events.CONTENT_URI`

([https://developer.android.com/reference/android/provider/CalendarContract.Events?hl=es-419#CONTENT\\_URI](https://developer.android.com/reference/android/provider/CalendarContract.Events?hl=es-419#CONTENT_URI))

### Tipo de MIME

`"vnd.android.cursor.dir/event"`

### Extras

#### `EXTRA_EVENT_ALL_DAY`

([https://developer.android.com/reference/android/provider/CalendarContract?hl=es-419#EXTRA\\_EVENT\\_ALL\\_DAY](https://developer.android.com/reference/android/provider/CalendarContract?hl=es-419#EXTRA_EVENT_ALL_DAY))

Booleano que especifica si este es un evento de día completo.

#### `EXTRA_EVENT_BEGIN_TIME`

([https://developer.android.com/reference/android/provider/CalendarContract?hl=es-419#EXTRA\\_EVENT\\_BEGIN\\_TIME](https://developer.android.com/reference/android/provider/CalendarContract?hl=es-419#EXTRA_EVENT_BEGIN_TIME))

La hora de inicio del evento (milisegundos desde el epoch).

#### `EXTRA_EVENT_END_TIME`

([https://developer.android.com/reference/android/provider/CalendarContract?hl=es-419#EXTRA\\_EVENT\\_END\\_TIME](https://developer.android.com/reference/android/provider/CalendarContract?hl=es-419#EXTRA_EVENT_END_TIME))

La hora de finalización del evento (milisegundos desde el epoch).

**TITLE**

(<https://developer.android.com/reference/android/provider/CalendarContract.EventsColumns?hl=es-419#TITLE>)

El título del evento.

**DESCRIPTION**

(<https://developer.android.com/reference/android/provider/CalendarContract.EventsColumns?hl=es-419#DESCRIPTION>)

La descripción del evento.

**EVENT\_LOCATION**

([https://developer.android.com/reference/android/provider/CalendarContract.EventsColumns?hl=es-419#EVENT\\_LOCATION](https://developer.android.com/reference/android/provider/CalendarContract.EventsColumns?hl=es-419#EVENT_LOCATION))

La ubicación del evento.

**EXTRA\_EMAIL**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_EMAIL](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_EMAIL))

Una lista separada por comas de direcciones de correo electrónico que especifican los invitados.

Se pueden especificar muchos más detalles del evento con las constantes definidas en la clase **CalendarContract.EventsColumns**

(<https://developer.android.com/reference/android/provider/CalendarContract.EventsColumns?hl=es-419>)

.

**Ejemplo de intent:**

**Kotlin** (#kotlin)**Java**  
(#java)

```
public void addEvent(String title, String location, long begin, long end) {
    Intent intent = new Intent(Intent.ACTION_INSERT)
        .setData(Events.CONTENT_URI)
        .putExtra(Events.TITLE, title)
        .putExtra(Events.EVENT_LOCATION, location)
        .putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, begin)
        .putExtra(CalendarContract.EXTRA_EVENT_END_TIME, end);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

```
}
}
```

## Filtro de intents de ejemplo:

```
<activity ...>
  <intent-filter>
    <action android:name="android.intent.action.INSERT" />
    <data android:mimeType="vnd.android.cursor.dir/event" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

# Cámara

## Capturar y mostrar una imagen o video

Para abrir una app de cámara y recibir el video o la foto resultante, usa la acción

### **ACTION\_IMAGE\_CAPTURE**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION\\_IMAGE\\_CAPTURE](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION_IMAGE_CAPTURE))

### o **ACTION\_VIDEO\_CAPTURE**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION\\_VIDEO\\_CAPTURE](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION_VIDEO_CAPTURE))

. También especifica la ubicación del URI donde te gustaría que la cámara guarde la foto o el video en el extra **EXTRA\_OUTPUT**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_OUTPUT](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_OUTPUT)).

## Acción

### **ACTION\_IMAGE\_CAPTURE**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION\\_IMAGE\\_CAPTURE](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION_IMAGE_CAPTURE))

o

### **ACTION\_VIDEO\_CAPTURE**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION\\_VIDEO\\_CAPTURE](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION_VIDEO_CAPTURE))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Extras

### EXTRA\_OUTPUT

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_OUTPUT](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_OUTPUT))

La ubicación del URI donde te gustaría que la aplicación de cámara guarde la foto o el video (como un objeto de Uri (<https://developer.android.com/reference/android/net/Uri?hl=es-419>)).

Cuando la aplicación de cámara muestra correctamente el foco a la actividad (la aplicación recibe la devolución de llamada onActivityResult()

([https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult\(int,%20int,%20android.content.Intent\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult(int,%20int,%20android.content.Intent)))

), puedes obtener acceso a la foto o el video en el URI especificado con el valor

### EXTRA\_OUTPUT

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_OUTPUT](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_OUTPUT)).

**Nota:** Cuando usas ACTION\_IMAGE\_CAPTURE

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION\\_IMAGE\\_CAPTURE](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#ACTION_IMAGE_CAPTURE))

para capturar una foto, la cámara también puede mostrar una copia reducida (miniatura) de la foto en el resultado Intent (<https://developer.android.com/reference/android/content/Intent?hl=es-419>), guardada como Bitmap (<https://developer.android.com/reference/android/graphics/Bitmap?hl=es-419>) en un campo adicional llamado "data".

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
static final int REQUEST_IMAGE_CAPTURE = 1;
static final Uri locationForPhotos;

public void capturePhoto(String targetFilename) {
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

```

        intent.putExtra(MediaStore.EXTRA_OUTPUT,
            Uri.withAppendedPath(locationForPhotos, targetFilename));
        if (intent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(intent, REQUEST_IMAGE_CAPTURE);
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK)
            Bitmap thumbnail = data.getParcelableExtra("data");
            // Do other work with full size photo saved in locationForPhotos
            ...
        }
    }
}

```

Para obtener más información acerca de cómo usar esta intent para capturar una foto, incluido el modo de crear un [Uri](#)

(<https://developer.android.com/reference/android/net/Uri?hl=es-419>) apropiado para la ubicación de salida, lee [Cómo tomar fotos fácilmente](#)

(<https://developer.android.com/training/camera/photobasics?hl=es-419>) o [Cómo grabar videos fácilmente](#) (<https://developer.android.com/training/camera/videobasics?hl=es-419>).

### Filtro de intents de ejemplo:

```

<activity ...>
    <intent-filter>
        <action android:name="android.media.action.IMAGE_CAPTURE" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

```

Al manejar esta intent, tu actividad debe comprobar si se encuentra el extra

#### EXTRA\_OUTPUT

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_OUTPUT](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_OUTPUT))

en la [Intent](#) (<https://developer.android.com/reference/android/content/Intent?hl=es-419>) entrante; luego, debe guardar la imagen o el video capturados en la ubicación especificada por ese extra y llamar a [setResult\(\)](#).

([https://developer.android.com/reference/android/app/Activity?hl=es-419#setResult\(int,%20android.content.Intent\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#setResult(int,%20android.content.Intent)))

con una **Intent** (<https://developer.android.com/reference/android/content/Intent?hl=es-419>) que incluya una miniatura comprimida en un extra llamado **"data"**.

## Iniciar una aplicación de cámara en modo de imagen detenida

Para abrir una aplicación de cámara en el modo de imagen detenida, usa la acción

**INTENT\_ACTION\_STILL\_IMAGE\_CAMERA**



([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

### Google Voice Actions

- "tomar una foto"

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT\\_ACTION\\_STILL\\_IMAGE\\_CAMERA](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT_ACTION_STILL_IMAGE_CAMERA))

### Acción

**INTENT\_ACTION\_STILL\_IMAGE\_CAMERA**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT\\_ACTION\\_STILL\\_IMAGE\\_CAMERA](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT_ACTION_STILL_IMAGE_CAMERA))

### Esquema de URI de datos

Ninguno

### Tipo de MIME

Ninguno

### Extras

Ninguno

### Ejemplo de intent:

**Kotlin** (#kotlin)**Java** (#java)

```
public void capturePhoto() {
    Intent intent = new Intent(MediaStore.INTENT_ACTION_STILL_IMAGE_CAMERA);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, REQUEST_IMAGE_CAPTURE);
    }
}
```

### Filtro de intents de ejemplo:

```
<activity ...>
    <intent-filter>
        <action android:name="android.media.action.STILL_IMAGE_CAMERA" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

## Iniciar una aplicación de cámara en modo de video

Para abrir una app de cámara en el modo de video, usa la acción [INTENT\\_ACTION\\_VIDEO\\_CAMERA](#)



([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

### Google Voice Actions

- "grabar un video"

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT\\_ACTION\\_VIDEO\\_CAMERA](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT_ACTION_VIDEO_CAMERA))

### Acción

#### [INTENT\\_ACTION\\_VIDEO\\_CAMERA](#)

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT\\_ACTION\\_VIDEO\\_CAMERA](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT_ACTION_VIDEO_CAMERA))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Extras

Ninguno

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void capturePhoto() {  
    Intent intent = new Intent(MediaStore.INTENT_ACTION_VIDEO_CAMERA);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivityForResult(intent, REQUEST_IMAGE_CAPTURE);  
    }  
}
```

## Filtro de intents de ejemplo:

```
<activity ...>  
    <intent-filter>  
        <action android:name="android.media.action.VIDEO_CAMERA" />  
        <category android:name="android.intent.category.DEFAULT" />  
    </intent-filter>  
</activity>
```

# Aplicación Personas/Contactos

## Seleccionar un contacto

Para que el usuario seleccione un contacto y le brinde a tu aplicación acceso a toda la información de contacto, usa la acción [ACTION\\_PICK](#)



([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_PICK](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_PICK)) y especifica el tipo de MIME en [Contacts.CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT_TYPE) ([https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT_TYPE))

La [Intent](https://developer.android.com/reference/android/content/Intent?hl=es-419) (<https://developer.android.com/reference/android/content/Intent?hl=es-419>) resultante entregada a tu devolución de llamada de [onActivityResult\(\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult(int,%20int,%20android.content.Intent)) ([https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult\(int,%20int,%20android.content.Intent\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult(int,%20int,%20android.content.Intent))) contiene el URI `content:` que apunta al contacto seleccionado. La respuesta le otorga permisos temporarios a tu aplicación para que lea ese contacto con la API de [Proveedor de contactos](https://developer.android.com/guide/topics/providers/contacts-provider?hl=es-419) (<https://developer.android.com/guide/topics/providers/contacts-provider?hl=es-419>) incluso si la aplicación no incluye el permiso [READ\\_CONTACTS](https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ_CONTACTS) ([https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ\\_CONTACTS](https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ_CONTACTS))

**Sugerencia:** Si necesitas acceder solamente a una parte específica de la información de contacto, como un número de teléfono o la dirección de correo electrónico, consulta la siguiente sección acerca de cómo [seleccionar datos de contacto específicos](#) (`#PickContactData`).

## Acción

[ACTION\\_PICK](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_PICK)

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_PICK](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_PICK))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

[Contacts.CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT_TYPE)

([https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT_TYPE))

## Ejemplo de intent:

[Kotlin](#) (`#kotlin`) [Java](#) (`#java`)

```

static final int REQUEST_SELECT_CONTACT = 1;

public void selectContact() {
    Intent intent = new Intent(Intent.ACTION_PICK);
    intent.setType(ContactsContract.Contacts.CONTENT_TYPE);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, REQUEST_SELECT_CONTACT);
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
    if (requestCode == REQUEST_SELECT_CONTACT && resultCode == RESULT_OK
        Uri contactUri = data.getData();
        // Do something with the selected contact at contactUri
        ...
    }
}

```

Para obtener información acerca de cómo recuperar detalles de contacto cuando ya tengas el URI de contacto, lee [Cómo recuperar detalles de un contacto](https://developer.android.com/training/contacts-provider/retrieve-details?hl=es-419) (https://developer.android.com/training/contacts-provider/retrieve-details?hl=es-419). Recuerda que, cuando recuperas el URI de contacto con la intent anterior, **no** necesitas el permiso **READ\_CONTACTS** (https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ\_CONTACTS) para leer los detalles de ese contacto.

## Seleccionar datos de un contacto específico

Para que un usuario seleccione una parte específica de la información de un contacto, como un número de teléfono, una dirección de correo electrónico u otro dato, usa la acción **ACTION\_PICK** (https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\_PICK) y especifica el tipo de MIME en uno de los tipos de contenido mencionados a continuación, como **CommonDataKinds.Phone.CONTENT\_TYPE** (https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds.Phone?hl=es-419#CONTENT\_TYPE) para obtener el número de teléfono del contacto.

Si deseas obtener solamente un tipo de datos de un contacto, aplicar esta técnica con un **CONTENT\_TYPE** de las clases **ContactsContract.CommonDataKinds**

(<https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds?hl=es-419>)

es más eficaz que usar `Contacts.CONTENT_TYPE`

([https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT_TYPE))

(como se muestra en la sección anterior) porque el resultado te brinda acceso directo a los datos deseados sin necesidad de que realices una consulta más compleja a

Proveedor de contactos

(<https://developer.android.com/guide/topics/providers/contacts-provider?hl=es-419>).

La `Intent` (<https://developer.android.com/reference/android/content/Intent?hl=es-419>) resultante entregada a tu devolución de llamada de `onActivityResult()`

([https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult\(int,%20int,%20android.content.Intent\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult(int,%20int,%20android.content.Intent)))

contiene el URI `content:` que apunta a los datos del contacto seleccionado. La respuesta le otorga permisos temporarios a tu aplicación para que lea esos datos de contacto incluso si la aplicación no incluye el permiso `READ_CONTACTS`

([https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ\\_CONTACTS](https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ_CONTACTS))

.

## Acción

`ACTION_PICK`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_PICK](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_PICK))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

`CommonDataKinds.Phone.CONTENT_TYPE`

([https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds.Phone?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds.Phone?hl=es-419#CONTENT_TYPE))

Elegir entre los contactos con un número de teléfono.

`CommonDataKinds.Email.CONTENT_TYPE`

([https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds.Email?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds.Email?hl=es-419#CONTENT_TYPE))

Elegir entre los contactos con una dirección de correo electrónico.

**CommonDataKinds.StructuredPostal.CONTENT\_TYPE**

([https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds.StructuredPostal?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.CommonDataKinds.StructuredPostal?hl=es-419#CONTENT_TYPE))

Elegir entre los contactos con una dirección postal.

O uno de los otros valores de `CONTENT_TYPE` en `ContactsContract`

(<https://developer.android.com/reference/android/provider/ContactsContract?hl=es-419>).

**Ejemplo de intent:**

**Kotlin** (#kotlin)**Java**

(#java)

```
static final int REQUEST_SELECT_PHONE_NUMBER = 1;

public void selectContact() {
    // Start an activity for the user to pick a phone number from contac
    Intent intent = new Intent(Intent.ACTION_PICK);
    intent.setType(CommonDataKinds.Phone.CONTENT_TYPE);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, REQUEST_SELECT_PHONE_NUMBER);
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
    if (requestCode == REQUEST_SELECT_PHONE_NUMBER && resultCode == RESL
        // Get the URI and query the content provider for the phone numk
        Uri contactUri = data.getData();
        String[] projection = new String[]{CommonDataKinds.Phone.NUMBER}
        Cursor cursor = getContentResolver().query(contactUri, projectio
            null, null, null);
        // If the cursor returned is valid, get the phone number
        if (cursor != null && cursor.moveToFirst()) {
            int numberIndex = cursor.getColumnIndex(CommonDataKinds.Phor
            String number = cursor.getString(numberIndex);
            // Do something with the phone number
            //...
        }
    }
}
```

**Ver un contacto**

Si deseas mostrar los detalles de un contacto conocido, usa la acción `ACTION_VIEW` ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW)) y especifica el contacto con un URI `content:` como datos de la intent.

Existen principalmente dos formas de recuperar inicialmente el URI de contacto:

- Usar el URI del contacto que muestra `ACTION_PICK` ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_PICK](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_PICK)), según se indica en la sección anterior (este enfoque no requiere ningún permiso de la aplicación).
- Acceder a la lista de todos los contactos directamente, como se describe en [Cómo recuperar una lista de contactos](#) (<https://developer.android.com/training/contacts-provider/retrieve-names?hl=es-419>) (este enfoque requiere el permiso `READ_CONTACTS` ([https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ\\_CONTACTS](https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ_CONTACTS))).

## Acción

### `ACTION_VIEW`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW))

## Esquema de URI de datos

`content:<URI>`

## Tipo de MIME

Ninguno. El tipo se infiere del URI de contacto.

## Ejemplo de intent:

**Kotlin** (#kotlin)**Java** (#java)

```
public void viewContact(Uri contactUri) {  
    Intent intent = new Intent(Intent.ACTION_VIEW, contactUri);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

## Editar un contacto existente

Para editar un contacto conocido, usa la acción **`ACTION_EDIT`**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_EDIT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_EDIT)), especifica el contacto con un URI **`content:`** como datos de la intent e incluye cualquier información del contacto conocido en los campos extras especificados por las constantes en **`ContactsContract.Intents.Insert`** (<https://developer.android.com/reference/android/provider/ContactsContract.Intents.Insert?hl=es-419>)

.

Existen dos formas de recuperar inicialmente el URI de contacto:

- Usar el URI del contacto que muestra **`ACTION_PICK`** ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_PICK](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_PICK)), según se indica en la sección anterior (este enfoque no requiere ningún permiso de la aplicación).
- Acceder a la lista de todos los contactos directamente, como se describe en [Cómo recuperar una lista de contactos](#) (<https://developer.android.com/training/contacts-provider/retrieve-names?hl=es-419>) (este enfoque requiere el permiso **`READ_CONTACTS`** ([https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ\\_CONTACTS](https://developer.android.com/reference/android/Manifest.permission?hl=es-419#READ_CONTACTS))).

### Acción

**`ACTION_EDIT`**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_EDIT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_EDIT))

### Esquema de URI de datos

**`content:<URI>`**

### Tipo de MIME

El tipo se infiere del URI de contacto.

### Extras

Uno o más de los extras definidos en **`ContactsContract.Intents.Insert`** (<https://developer.android.com/reference/android/provider/ContactsContract.Intents.Insert?hl=es-419>) para que puedas completar los campos con los detalles de contacto.

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void editContact(Uri contactUri, String email) {  
    Intent intent = new Intent(Intent.ACTION_EDIT);  
    intent.setData(contactUri);  
    intent.putExtra(Intent.EXTRA_EMAIL, email);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

Para obtener más información acerca de cómo editar un contacto, lee [Cómo modificar contactos con intents](https://developer.android.com/training/contacts-provider/modify-data?hl=es-419)

(<https://developer.android.com/training/contacts-provider/modify-data?hl=es-419>).

## Insertar un contacto

Para insertar un contacto nuevo, usa la acción **ACTION\_INSERT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_INSERT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_INSERT)),

especifica **Contacts.CONTENT\_TYPE**

([https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT_TYPE))

como el tipo de MIME e incluye cualquier información de contacto conocida en los campos extras especificados por las constantes en **ContactsContract.Intents.Insert**

(<https://developer.android.com/reference/android/provider/ContactsContract.Intents.Insert?hl=es-419>)

.

### Acción

**ACTION\_INSERT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_INSERT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_INSERT))

### Esquema de URI de datos

Ninguno

### Tipo de MIME

### Contacts.CONTENT\_TYPE

([https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/ContactsContract.Contacts?hl=es-419#CONTENT_TYPE))

## Extras

Uno o más de los extras definidos en ContactsContract.Intents.Insert

(<https://developer.android.com/reference/android/provider/ContactsContract.Intents.Insert?hl=es-419>)

.

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void insertContact(String name, String email) {  
    Intent intent = new Intent(Intent.ACTION_INSERT);  
    intent.setType(Contacts.CONTENT_TYPE);  
    intent.putExtra(Intent.Insert.NAME, name);  
    intent.putExtra(Intent.Insert.EMAIL, email);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

Para obtener más información acerca de cómo insertar un contacto, lee Cómo modificar contactos con intents

(<https://developer.android.com/training/contacts-provider/modify-data?hl=es-419>).

## Correo electrónico

### Redactar un correo electrónico con archivos adjuntos opcionales

Para redactar un mensaje de correo electrónico, usa una de las acciones que se muestran a continuación sobre la base de si incluirá archivos adjuntos, y agrega detalles del correo electrónico como el destinatario y el asunto con las claves adicionales mencionadas a continuación.

### Acción



**ACTION\_SENDTO**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SENDTO](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SENDTO))  
(sin archivos adjuntos)

**ACTION\_SEND**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SEND](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SEND))  
(para un archivo adjunto)

**ACTION\_SEND\_MULTIPLE**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SEND\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SEND_MULTIPLE))  
(para varios archivos adjuntos)

**Esquema de URI de datos**

Ninguno

**Tipo de MIME**

"text/plain"

"\*/\*"

**Extras****Intent.EXTRA\_EMAIL**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_EMAIL](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_EMAIL))

Una matriz de strings de todas las direcciones de correo electrónico "Para" de los destinatarios.

**Intent.EXTRA\_CC**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_CC](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_CC))

Una matriz de strings de todas las direcciones de correo electrónico "CC" de los destinatarios.

**Intent.EXTRA\_BCC**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_BCC](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_BCC))

Una matriz de strings de todas las direcciones de correo electrónico "CCO" de los destinatarios.

**Intent.EXTRA\_SUBJECT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_SUBJECT](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_SUBJECT))

Una string con el asunto del mensaje de correo electrónico.

**Intent.EXTRA\_TEXT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_TEXT](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_TEXT))

Una string con el cuerpo del mensaje de correo electrónico.

**Intent.EXTRA\_STREAM**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_STREAM](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_STREAM))

Un Uri (<https://developer.android.com/reference/android/net/Uri?hl=es-419>) que apunta a los archivos adjuntos. Si usas la acción **ACTION\_SEND\_MULTIPLE** ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SEND\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SEND_MULTIPLE))

, en su lugar debe ser una **ArrayList**

(<https://developer.android.com/reference/java/util/ArrayList?hl=es-419>) que contenga varios objetos Uri

(<https://developer.android.com/reference/android/net/Uri?hl=es-419>).

**Ejemplo de intent:**

**Kotlin** (#kotlin)**Java**  
(#java)

```
public void composeEmail(String[] addresses, String subject, Uri attachment) {
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("*/*");
    intent.putExtra(Intent.EXTRA_EMAIL, addresses);
    intent.putExtra(Intent.EXTRA_SUBJECT, subject);
    intent.putExtra(Intent.EXTRA_STREAM, attachment);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

Para asegurarte de que solo una app de correo electrónico (y no otra app de redes sociales o mensajería de texto) maneje la intent, usa la acción **ACTION\_SENDTO** ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SENDTO](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SENDTO)) e incluye el esquema de datos **"mailto:"**. Por ejemplo:

**Kotlin** (#kotlin)**Java**  
(#java)

```
public void composeEmail(String[] addresses, String subject) {
    Intent intent = new Intent(Intent.ACTION_SENDTO);
    intent.setData(Uri.parse("mailto:")); // only email apps should handle
    intent.putExtra(Intent.EXTRA_EMAIL, addresses);
    intent.putExtra(Intent.EXTRA_SUBJECT, subject);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

### Filtro de intents de ejemplo:

```
<activity ...>
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <data android:type="*/*" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.SENDTO" />
        <data android:scheme="mailto" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

## Almacenamiento en archivo

### Recuperar un tipo específico de archivo

Para solicitar que el usuario seleccione un archivo como un documento o una foto, y que se muestre una referencia a la aplicación, usa la acción `ACTION_GET_CONTENT` ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_GET\\_CONTENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_GET_CONTENT)) y especifica el tipo de MIME deseado. La referencia del archivo que se muestra a la aplicación es transitoria al ciclo de vida actual de la actividad, por lo que, si deseas acceder a ella más tarde, deberás importar una copia que podrás leer más adelante. Esta intent también permite al usuario crear un nuevo archivo en el proceso (por ejemplo, en lugar de seleccionar una foto existente, el usuario puede capturar una nueva foto con la cámara).

La intent del resultado enviada a tu método `onActivityResult()`.

([https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult\(int,%20int,%20android.content.Intent\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult(int,%20int,%20android.content.Intent)))

incluye datos con una URI que apunta al archivo. El URI puede ser cualquier cosa, como un URI `http:`, `file:` o `content:`. No obstante, si deseas restringir los archivos seleccionables a solo aquellos que sean accesibles desde un proveedor de contenido (un URI `content:`) y estén disponibles como un flujo de archivos con `openFileDescriptor()`.

([https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor\(android.net.Uri,%20java.lang.String\)](https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor(android.net.Uri,%20java.lang.String)))

, debes agregar la categoría `CATEGORY_OPENABLE`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY\\_OPENABLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY_OPENABLE))

a tu intent.

En Android 4.3 (nivel de API 18) y versiones posteriores, también puedes permitir que el usuario seleccione varios archivos agregando `EXTRA_ALLOW_MULTIPLE`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_ALLOW\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_ALLOW_MULTIPLE))

a la intent, establecida en `true`. Luego, puedes acceder a cada uno de los archivos seleccionados en un objeto `ClipData`

(<https://developer.android.com/reference/android/content/ClipData?hl=es-419>) mostrado por `getClipData()`.

([https://developer.android.com/reference/android/content/Intent?hl=es-419#getClipData\(\)](https://developer.android.com/reference/android/content/Intent?hl=es-419#getClipData())).

## Acción

`ACTION_GET_CONTENT`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_GET\\_CONTENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_GET_CONTENT))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

El tipo de MIME correspondiente al tipo de archivo que el usuario debe seleccionar.

## Extras

`EXTRA_ALLOW_MULTIPLE`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_ALLOW\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_ALLOW_MULTIPLE))

Booleano que declara si el usuario puede seleccionar más de un archivo a la vez.

**EXTRA\_LOCAL\_ONLY**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_LOCAL\\_ONLY](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_LOCAL_ONLY))

Booleano que declara si el archivo que se muestra debe estar disponible directamente desde el dispositivo, en lugar de necesitar una descarga desde un servicio remoto.

**Categoría (opcional)****CATEGORY\_OPENABLE**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY\\_OPENABLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY_OPENABLE))

Para mostrar solamente archivos que puedan abrirse y que puedan representarse como un flujo de archivos con `openFileDescriptor()` ([https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor\(android.net.Uri,%20java.lang.String\)\)](https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor(android.net.Uri,%20java.lang.String))))

**Ejemplo de intent para tomar una foto:**

**Kotlin** (#kotlin)**Java** (#java)

```
static final int REQUEST_IMAGE_GET = 1;

public void selectImage() {
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.setType("image/*");
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, REQUEST_IMAGE_GET);
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_GET && resultCode == RESULT_OK) {
        Bitmap thumbnail = data.getParcelable("data");
        Uri fullPhotoUri = data.getData();
        // Do work with photo saved at fullPhotoUri
        ...
    }
}
```

## Ejemplo de filtro de intents para devolver una foto:

```
<activity ...>
  <intent-filter>
    <action android:name="android.intent.action.GET_CONTENT" />
    <data android:type="image/*" />
    <category android:name="android.intent.category.DEFAULT" />
    <!-- The OPENABLE category declares that the returned file is accessible
         from a content provider that supports OpenableColumns (https://developer.android.com/reference/android/content/ContentProvider#openableColumns)
         and ContentResolver.openFileDescriptor() (https://developer.android.com/reference/android/content/ContentResolver#openFileDescriptor\(java.lang.String,%20int,%20int,%20android.content.Intent\))
    <category android:name="android.intent.category.OPENABLE" />
  </intent-filter>
</activity>
```

## Abrir un tipo específico de archivo

En lugar de obtener una copia de un archivo que debes importar a tu app (usando la acción **ACTION\_GET\_CONTENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_GET\\_CONTENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_GET_CONTENT)), cuando se use en Android 4.4 o versiones posteriores, puedes solicitar *abrir* un archivo administrado por otra app con la acción **ACTION\_OPEN\_DOCUMENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_OPEN\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_OPEN_DOCUMENT))

y especificando un tipo de MIME. Para permitir también que el usuario cree un documento nuevo en el cual la app pueda realizar operaciones de escritura, usa en su lugar la acción **ACTION\_CREATE\_DOCUMENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CREATE\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CREATE_DOCUMENT))

. Por ejemplo, en lugar de seleccionar entre documentos PDF existentes, la intent **ACTION\_CREATE\_DOCUMENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CREATE\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CREATE_DOCUMENT))

permite a los usuarios seleccionar dónde querrían crear un documento nuevo (dentro de otra aplicación que administra el almacenamiento del documento). Tu aplicación recibe la ubicación del URI de dónde puede escribir el nuevo documento.

Sin bien la intent entregada a tu método **onActivityResult()**

([https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult\(int,%20int,%20android.content.Intent\)](https://developer.android.com/reference/android/app/Activity?hl=es-419#onActivityResult(int,%20int,%20android.content.Intent)))

desde la acción **ACTION\_GET\_CONTENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_GET\\_CONTENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_GET_CONTENT))

puede mostrar un URI de cualquier tipo, la intent resultante de [ACTION\\_OPEN\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_OPEN_DOCUMENT) ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_OPEN\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_OPEN_DOCUMENT))

y [ACTION\\_CREATE\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CREATE_DOCUMENT)

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CREATE\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CREATE_DOCUMENT))

siempre especifica el archivo elegido como un URI `content:` respaldado por un [DocumentsProvider](https://developer.android.com/reference/android/provider/DocumentsProvider?hl=es-419)

(<https://developer.android.com/reference/android/provider/DocumentsProvider?hl=es-419>). Puedes abrir el archivo con `openFileDescriptor()`

([https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor\(android.net.Uri,%20java.lang.String\)](https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor(android.net.Uri,%20java.lang.String)))

y consultar sus detalles con columnas del [DocumentsContract.Document](https://developer.android.com/reference/android/provider/DocumentsContract.Document?hl=es-419)

(<https://developer.android.com/reference/android/provider/DocumentsContract.Document?hl=es-419>)

.

El URI devuelto le otorga a tu aplicación acceso de lectura a largo plazo al archivo (también es posible que tenga acceso de escritura). Por lo tanto, la acción

[ACTION\\_OPEN\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_OPEN_DOCUMENT)

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_OPEN\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_OPEN_DOCUMENT))

es particularmente útil (en lugar de usar [ACTION\\_GET\\_CONTENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_GET_CONTENT)

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_GET\\_CONTENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_GET_CONTENT))

) cuando quieres leer un archivo existente sin realizar una copia en tu aplicación o cuando quieres abrir o editar un archivo en el lugar.

También puedes permitir al usuario que seleccione varios archivos agregando

[EXTRA\\_ALLOW\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_ALLOW_MULTIPLE)

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_ALLOW\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_ALLOW_MULTIPLE))

a la intent, establecido como `true`. Si el usuario selecciona un solo elemento, puedes recuperarlo desde `getData()`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#getData\(\)](https://developer.android.com/reference/android/content/Intent?hl=es-419#getData())). Si el usuario selecciona más de un elemento, `getData()`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#getData\(\)](https://developer.android.com/reference/android/content/Intent?hl=es-419#getData())) muestra un valor nulo y debes obtener cada elemento de un objeto [ClipData](https://developer.android.com/reference/android/content/ClipData?hl=es-419)

(<https://developer.android.com/reference/android/content/ClipData?hl=es-419>) que muestra `getClipData()`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#getClipData\(\)](https://developer.android.com/reference/android/content/Intent?hl=es-419#getClipData())).

**Nota:** La intent **debe** especificar un tipo de MIME y **debe** declarar la categoría **CATEGORY\_OPENABLE** ([https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY\\_OPENABLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY_OPENABLE)). Si corresponde, puedes especificar más de un tipo de MIME agregando una matriz de tipos de MIME con el extra **EXTRA\_MIME\_TYPES** ([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_MIME\\_TYPES](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_MIME_TYPES)). Si haces esto, debes establecer el tipo de MIME principal en **setType()** ([https://developer.android.com/reference/android/content/Intent?hl=es-419#setType\(java.lang.String\)](https://developer.android.com/reference/android/content/Intent?hl=es-419#setType(java.lang.String))) en `"*/*"`.

## Acción

### **ACTION\_OPEN\_DOCUMENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_OPEN\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_OPEN_DOCUMENT))

O

### **ACTION\_CREATE\_DOCUMENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CREATE\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CREATE_DOCUMENT))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

El tipo de MIME correspondiente al tipo de archivo que el usuario debe seleccionar.

## Extras

### **EXTRA\_MIME\_TYPES**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_MIME\\_TYPES](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_MIME_TYPES))

Una matriz de tipos de MIME correspondiente a los tipos de archivos que tu aplicación está solicitando. Cuando uses este extra, debes establecer el tipo de MIME principal en **setType()**.

([https://developer.android.com/reference/android/content/Intent?hl=es-419#setType\(java.lang.String\)](https://developer.android.com/reference/android/content/Intent?hl=es-419#setType(java.lang.String)))

en `"*/*"`.

### **EXTRA\_ALLOW\_MULTIPLE**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_ALLOW\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_ALLOW_MULTIPLE))



Booleano que declara si el usuario puede seleccionar más de un archivo a la vez.

#### EXTRA\_TITLE

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_TITLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_TITLE))

Para usar con ACTION\_CREATE\_DOCUMENT

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CREATE\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CREATE_DOCUMENT))

a fin de especificar un nombre de archivo inicial.

#### EXTRA\_LOCAL\_ONLY

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_LOCAL\\_ONLY](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_LOCAL_ONLY))

Booleano que declara si el archivo que se muestra debe estar disponible directamente desde el dispositivo, en lugar de necesitar una descarga desde un servicio remoto.

## Categoría

#### CATEGORY\_OPENABLE

([https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY\\_OPENABLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#CATEGORY_OPENABLE))

Para mostrar solamente archivos que puedan abrirse y que puedan representarse como un flujo de archivos con openFileDescriptor()

([https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor\(android.net.Uri,%20java.lang.String\)](https://developer.android.com/reference/android/content/ContentResolver?hl=es-419#openFileDescriptor(android.net.Uri,%20java.lang.String)))

.

## Ejemplo de intent para tomar una foto:

Kotlin (#kotlin)Java

(#java)

```
static final int REQUEST_IMAGE_OPEN = 1;

public void selectImage() {
    Intent intent = new Intent(Intent.ACTION_OPEN_DOCUMENT);
    intent.setType("image/*");
    intent.addCategory(Intent.CATEGORY_OPENABLE);
    // Only the system receives the ACTION_OPEN_DOCUMENT, so no need to
    startActivityForResult(intent, REQUEST_IMAGE_OPEN);
}
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
    if (requestCode == REQUEST_IMAGE_OPEN && resultCode == RESULT_OK) {
        Uri fullPhotoUri = data.getData();
        // Do work with full size photo saved at fullPhotoUri
        ...
    }
}
```

Las aplicaciones de terceros en realidad no pueden responder a una intent con la acción **ACTION\_OPEN\_DOCUMENT**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_OPEN\\_DOCUMENT](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_OPEN_DOCUMENT))

. En su lugar, el sistema recibe esta intent y muestra todos los archivos disponibles desde diversas aplicaciones en una interfaz de usuario unificada.

Para proporcionar los archivos de tu app en esta IU y permitir que otras apps los abran, debes implementar un **DocumentsProvider**

(<https://developer.android.com/reference/android/provider/DocumentsProvider?hl=es-419>) e incluir un filtro de intents para **PROVIDER\_INTERFACE**

([https://developer.android.com/reference/android/provider/DocumentsContract?hl=es-419#PROVIDER\\_INTERFACE](https://developer.android.com/reference/android/provider/DocumentsContract?hl=es-419#PROVIDER_INTERFACE))

("android.content.action.DOCUMENTS\_PROVIDER"). Por ejemplo:

```
<provider ...
    android:grantUriPermissions="true"
    android:exported="true"
    android:permission="android.permission.MANAGE_DOCUMENTS">
    <intent-filter>
        <action android:name="android.content.action.DOCUMENTS_PROVIDER" />
    </intent-filter>
</provider>
```

Para obtener más información acerca de cómo hacer que los archivos administrados por tu aplicación puedan abrirse desde otras aplicaciones, lee la guía **Marco de trabajo de acceso a almacenamiento**

(<https://developer.android.com/guide/topics/providers/document-provider?hl=es-419>).

# Acciones locales

## Pedir un auto

Para pedir un taxi, utiliza la acción

**ACTION\_RESERVE\_TAXI\_RESERVATION**



([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

### Google Voice Actions

- "conseguir un taxi"
- "pedir un auto"

(Solo Wear OS)

([https://developer.android.com/reference/com/google/android/gms/actions/ReserveIntents?hl=es-419#ACTION\\_RESERVE\\_TAXI\\_RESERVATION](https://developer.android.com/reference/com/google/android/gms/actions/ReserveIntents?hl=es-419#ACTION_RESERVE_TAXI_RESERVATION))

.

**Nota:** Las apps deben pedir la confirmación del usuario antes de completar la acción.

## Acción

**ACTION\_RESERVE\_TAXI\_RESERVATION**

([https://developer.android.com/reference/com/google/android/gms/actions/ReserveIntents?hl=es-419#ACTION\\_RESERVE\\_TAXI\\_RESERVATION](https://developer.android.com/reference/com/google/android/gms/actions/ReserveIntents?hl=es-419#ACTION_RESERVE_TAXI_RESERVATION))

## URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Extras

Ninguno

## Ejemplo de intent:

Kotlin (#kotlin)

Java  
(#java)

```
public void callCar() {
    Intent intent = new Intent(ReserveIntents.ACTION_RESERVE_TAXI_RESERV
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

## Filtro de intents de ejemplo:

```
<activity ...>
    <intent-filter>
        <action android:name="com.google.android.gms.actions.RESERVE_TAXI_RES
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

# Mapas

## Mostrar una ubicación en un mapa

Para abrir un mapa, usa la acción **ACTION\_VIEW**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW)) y especifica la información de ubicación en los datos de la intent con uno de los esquemas definidos a continuación.

### Acción

**ACTION\_VIEW**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW))

### Esquema de URI de datos

*geo:latitude, longitude*

Mostrar el mapa en la longitud y latitud dadas.

Ejemplo: `"geo:47.6, -122.3"`

`geo:latitude, longitude?z=zoom`

Mostrar el mapa en la longitud y la latitud dadas en un nivel de zoom determinado. Un nivel de zoom de 1 muestra toda la Tierra, centrada en la *lat, long* dadas. El nivel de zoom más alto (más cercano) es 23.

Ejemplo: `"geo:47.6, -122.3?z=11"`

`geo:0, 0?q=lat, lng(label)`

Mostrar el mapa en la longitud y latitud dadas con una etiqueta de string.

Ejemplo: `"geo:0, 0?q=34.99, -106.61(Treasure)"`

`geo:0, 0?q=my+street+address`

Mostrar la ubicación para "mi dirección postal" (puede ser una consulta de ubicación o dirección específica).

Ejemplo: `"geo:0, 0?q=1600+Amphitheatre+Parkway%2C+CA"`

- ★ **Nota:** Todas las strings pasadas en el URI **geo** deben estar codificadas. Por ejemplo, la string **1st & Pike, Seattle** debe convertirse en **1st%20%26%20Pike%2C%20Seattle**. Los espacios de la string pueden codificarse con **%20** o reemplazarse por el signo más (+).

## Tipo de MIME

Ninguno

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void showMap(Uri geoLocation) {
    Intent intent = new Intent(Intent.ACTION_VIEW);
    intent.setData(geoLocation);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

## Filtro de intents de ejemplo:

```
<activity ...>
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <data android:scheme="geo" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

## Música o videos

### Reproducir un archivo multimedia

Para reproducir un archivo de música, usa la acción `ACTION_VIEW` ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW)) y especifica la ubicación del URI del archivo en los datos de la intent.

#### Acción

`ACTION_VIEW`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW))

#### Esquema de URI de datos

`file:<URI>`

`content:<URI>`

`http:<URL>`

#### Tipo de MIME

`"audio/*"`

`"application/ogg"`

`"application/x-ogg"`

`"application/itunes"`

**O cualquier otro que pueda requerir la aplicación.**

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void playMedia(Uri file) {  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setData(file);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

## Filtro de intents de ejemplo:

```
<activity ...>  
    <intent-filter>  
        <action android:name="android.intent.action.VIEW" />  
        <data android:type="audio/*" />  
        <data android:type="application/ogg" />  
        <category android:name="android.intent.category.DEFAULT" />  
    </intent-filter>  
</activity>
```

## Reproducir música según una consulta de búsqueda

Para reproducir música según una consulta de búsqueda, usa la intent [INTENT\\_ACTION\\_MEDIA\\_PLAY\\_FROM\\_SEARCH](#)



([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

### Google Voice Actions

- "reproducir billie jean de michael jackson"

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT\\_ACTION\\_MEDIA\\_PLAY\\_FROM\\_SEARCH](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT_ACTION_MEDIA_PLAY_FROM_SEARCH))

. Una aplicación puede desencadenar esta intent en respuesta al comando de voz del usuario para reproducir música. La aplicación receptora de esta intent realiza una búsqueda dentro de su inventario para hacer coincidir el contenido existente con la consulta dada y comenzar a reproducir ese contenido.

Esta intent debe incluir el extra de la string **EXTRA\_MEDIA\_FOCUS**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_FOCUS](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_FOCUS))

, que especifica el modo de búsqueda pretendido. Por ejemplo, el modo de búsqueda puede especificar si la búsqueda es según el nombre del artista o de la canción.

## Acción

**INTENT\_ACTION\_MEDIA\_PLAY\_FROM\_SEARCH**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT\\_ACTION\\_MEDIA\\_PLAY\\_FROM\\_SEARCH](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#INTENT_ACTION_MEDIA_PLAY_FROM_SEARCH))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Extras

**MediaStore.EXTRA\_MEDIA\_FOCUS**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_FOCUS](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_FOCUS))

**(obligatorio)**

Indica el modo de búsqueda (si el usuario busca un artista, un álbum, una canción o una lista de reproducción en particular). La mayoría de los modos de búsqueda requieren extras adicionales. Por ejemplo, si el usuario tiene interés por escuchar una canción en particular, la intent puede tener tres extras adicionales: el título de la canción, el artista y el álbum. Esta intent admite los siguientes modos de búsqueda para cada valor de

**EXTRA\_MEDIA\_FOCUS**

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_FOCUS](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_FOCUS))

:



**Cualquiera:** `"vnd.android.cursor.item/*"`

Reproducir cualquier canción. La aplicación receptora debe reproducir música sobre la base de una elección inteligente, como la última lista de reproducción que el usuario escuchó.

Extras adicionales:

- **QUERY**

(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)

(obligatorio): string vacía. Este extra siempre se proporciona para fines de compatibilidad con versiones anteriores: las aplicaciones existentes que no conocen los modos de búsqueda pueden procesar esta intent como una búsqueda no estructurada.

**No estructurada:** `"vnd.android.cursor.item/*"`

Reproducir una canción, un álbum o un género en particular desde una consulta de búsqueda no estructurada. Las apps pueden generar una intent con este modo de búsqueda cuando no pueden identificar el tipo de contenido que el usuario quiere escuchar. Cuando sea posible, las apps deberán usar modos de búsqueda más específicos.

Extras adicionales:

- **QUERY**

(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)

(obligatorio): string que contiene cualquier combinación del artista, el álbum, el nombre de la canción o el género.

**Género:** `Audio.Genres.ENTRY_CONTENT_TYPE`

([https://developer.android.com/reference/android/provider/MediaStore.Audio.Genres?hl=es-419#ENTRY\\_CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/MediaStore.Audio.Genres?hl=es-419#ENTRY_CONTENT_TYPE))

Reproducir música de un género en particular.

Extras adicionales:

- `"android.intent.extra.genre"` (obligatorio): el género.

- **QUERY**

(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)

(obligatorio): el género. Este extra siempre se proporciona para fines de compatibilidad con versiones anteriores: las aplicaciones existentes que no conocen los modos de búsqueda pueden procesar esta intent como una búsqueda no estructurada.

**Artista:** Audio.Artists.ENTRY\_CONTENT\_TYPE

([https://developer.android.com/reference/android/provider/MediaStore.Audio.Artists?hl=es-419#ENTRY\\_CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/MediaStore.Audio.Artists?hl=es-419#ENTRY_CONTENT_TYPE))

Reproducir música de un artista en particular.

Extras adicionales:

- EXTRA\_MEDIA\_ARTIST

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_ARTIST](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_ARTIST))

(obligatorio): el artista.

- "android.intent.extra.genre": el género.

- QUERY

(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)

(obligatorio): string que contiene cualquier combinación del artista o el género. Este extra siempre se proporciona para fines de compatibilidad con versiones anteriores: las aplicaciones existentes que no conocen los modos de búsqueda pueden procesar esta intent como una búsqueda no estructurada.

**Álbum:** Audio.Albums.ENTRY\_CONTENT\_TYPE

([https://developer.android.com/reference/android/provider/MediaStore.Audio.Albums?hl=es-419#ENTRY\\_CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/MediaStore.Audio.Albums?hl=es-419#ENTRY_CONTENT_TYPE))

Reproducir música de un álbum en particular.

Extras adicionales:

- EXTRA\_MEDIA\_ALBUM

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_ALBUM](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_ALBUM))

(obligatorio): el álbum.

- EXTRA\_MEDIA\_ARTIST

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_ARTIST](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_ARTIST))

: el artista.

- `"android.intent.extra.genre"`: el género.
- **QUERY**  
(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)  
(obligatorio): string que contiene cualquier combinación del álbum o el artista. Este extra siempre se proporciona para fines de compatibilidad con versiones anteriores: las aplicaciones existentes que no conocen los modos de búsqueda pueden procesar esta intent como una búsqueda no estructurada.

**Canción:** `"vnd.android.cursor.item/audio"`

Reproducir una canción en particular.

Extras adicionales:

- **EXTRA\_MEDIA\_ALBUM**  
([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_ALBUM](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_ALBUM))  
: el álbum.
- **EXTRA\_MEDIA\_ARTIST**  
([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_ARTIST](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_ARTIST))  
: el artista.
- `"android.intent.extra.genre"`: el género.
- **EXTRA\_MEDIA\_TITLE**  
([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_TITLE](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_TITLE))  
(obligatorio): el nombre de la canción.
- **QUERY**  
(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)  
(obligatorio): string que contiene cualquier combinación de: el álbum, el artista, el género o el título. Este extra siempre se proporciona para fines de compatibilidad con versiones anteriores: las aplicaciones existentes que no conocen los modos de búsqueda pueden procesar esta intent como una búsqueda no estructurada.

***Lista de reproducción:*** `Audio.Playlists.ENTRY_CONTENT_TYPE`

([https://developer.android.com/reference/android/provider/MediaStore.Audio.Playlists?hl=es-419#ENTRY\\_CONTENT\\_TYPE](https://developer.android.com/reference/android/provider/MediaStore.Audio.Playlists?hl=es-419#ENTRY_CONTENT_TYPE))

Reproducir una lista de reproducción en particular o que coincida con algún criterio especificado por los extras adicionales.

Extras adicionales:

- `EXTRA_MEDIA_ALBUM`  
([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_ALBUM](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_ALBUM))  
: el álbum.
- `EXTRA_MEDIA_ARTIST`  
([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_ARTIST](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_ARTIST))  
: el artista.
- `"android.intent.extra.genre"`: el género.
- `"android.intent.extra.playlist"`: la lista de reproducción.
- `EXTRA_MEDIA_TITLE`  
([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_TITLE](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_TITLE))  
: el nombre de la canción en el que se basa la lista de reproducción.
- `QUERY`  
(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)  
(obligatorio): string que contiene cualquier combinación del álbum, el artista, el género, la lista de reproducción o el título. Este extra siempre se proporciona para fines de compatibilidad con versiones anteriores: las aplicaciones existentes que no conocen los modos de búsqueda pueden procesar esta intent como una búsqueda no estructurada.

**Ejemplo de intent:**

Si el usuario quiere escuchar música de un artista en particular, una aplicación de búsqueda puede generar la siguiente intent:

```
Kotlin (#kotlin)Java  
      (#java)
```

```
public void playSearchArtist(String artist) {
    Intent intent = new Intent(MediaStore.INTENT_ACTION_MEDIA_PLAY_FROM_
        intent.putExtra(MediaStore.EXTRA_MEDIA_FOCUS,
            MediaStore.Audio.Artists.ENTRY_CONTENT_TYPE);
    intent.putExtra(MediaStore.EXTRA_MEDIA_ARTIST, artist);
    intent.putExtra(SearchManager.QUERY, artist);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

### Filtro de intents de ejemplo:

```
<activity ...>
    <intent-filter>
        <action android:name="android.media.action.MEDIA_PLAY_FROM_SEARCH" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

Al manejar esta intent, la actividad debe comprobar el valor del extra

#### EXTRA\_MEDIA\_FOCUS

([https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA\\_MEDIA\\_FOCUS](https://developer.android.com/reference/android/provider/MediaStore?hl=es-419#EXTRA_MEDIA_FOCUS))

en la Intent (<https://developer.android.com/reference/android/content/Intent?hl=es-419>) entrante para determinar el modo de búsqueda. Una vez que tu actividad haya identificado el modo de búsqueda, deberías leer los valores de los extras adicionales para ese modo de búsqueda en particular. Con esta información, tu aplicación puede realizar la búsqueda dentro de su inventario para reproducir el contenido que coincida con la consulta de búsqueda. Por ejemplo:

#### Kotlin (#kotlin)Java (#java)

```
protected void onCreate(Bundle savedInstanceState) {
    //...
    Intent intent = this.getIntent();
    if (intent.getAction().compareTo(MediaStore.INTENT_ACTION_MEDIA_PLAY
```

```

String mediaFocus = intent.getStringExtra(MediaStore.EXTRA_MEDIA
String query = intent.getStringExtra(SearchManager.QUERY);

// Some of these extras may not be available depending on the se
String album = intent.getStringExtra(MediaStore.EXTRA_MEDIA_ALBU
String artist = intent.getStringExtra(MediaStore.EXTRA_MEDIA_ART
String genre = intent.getStringExtra("android.intent.extra.genre
String playlist = intent.getStringExtra("android.intent.extra.pl
String title = intent.getStringExtra(MediaStore.EXTRA_MEDIA_TITL

// Determine the search mode and use the corresponding extras
if (mediaFocus == null) {
    // 'Unstructured' search mode (backward compatible)
    playUnstructuredSearch(query);

} else if (mediaFocus.compareTo("vnd.android.cursor.item/*") ==
    if (query.isEmpty()) {
        // 'Any' search mode
        playResumeLastPlaylist();
    } else {
        // 'Unstructured' search mode
        playUnstructuredSearch(query);
    }

} else if (mediaFocus.compareTo(MediaStore.Audio.Genres.ENTRY_CC
    // 'Genre' search mode
    playGenre(genre);

} else if (mediaFocus.compareTo(MediaStore.Audio.Artists.ENTRY_C
    // 'Artist' search mode
    playArtist(artist, genre);

} else if (mediaFocus.compareTo(MediaStore.Audio.Albums.ENTRY_CC
    // 'Album' search mode
    playAlbum(album, artist);

} else if (mediaFocus.compareTo("vnd.android.cursor.item/audio")
    // 'Song' search mode
    playSong(album, artist, genre, title);

} else if (mediaFocus.compareTo(MediaStore.Audio.Playlists.ENTRY
    // 'Playlist' search mode
    playPlaylist(album, artist, genre, playlist, title);
}
}
}

```

# Nota nueva

## Crear una nota

Para crear una nota nueva, usa la acción **ACTION\_CREATE\_NOTE**

([https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#ACTION\\_CREATE\\_NOTE](https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#ACTION_CREATE_NOTE))

y especifica detalles de la nota como el tema y el texto usando los campos extras definidos a continuación.

**Nota:** Las apps deben pedir la confirmación del usuario antes de completar la acción.

### Acción

**ACTION\_CREATE\_NOTE**

([https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#ACTION\\_CREATE\\_NOTE](https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#ACTION_CREATE_NOTE))

### Esquema de URI de datos

Ninguno

### Tipo de MIME

**PLAIN\_TEXT\_TYPE**

([https://developer.android.com/reference/org/apache/http/protocol/HTTP.html?hl=es-419#PLAIN\\_TEXT\\_TYPE](https://developer.android.com/reference/org/apache/http/protocol/HTTP.html?hl=es-419#PLAIN_TEXT_TYPE))

"\*/\*"

### Extras

**EXTRA\_NAME**

([https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#EXTRA\\_NAME](https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#EXTRA_NAME))

String que indica el título o tema de la nota.

**EXTRA\_TEXT**

([https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#EXTRA\\_TEXT](https://developers.google.com/android/reference/com/google/android/gms/actions/NoteIntents?hl=es-419#EXTRA_TEXT))

String que indica el texto de la nota.

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void createNote(String subject, String text) {
    Intent intent = new Intent(NoteIntents.ACTION_CREATE_NOTE)
        .putExtra(NoteIntents.EXTRA_NAME, subject)
        .putExtra(NoteIntents.EXTRA_TEXT, text);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

## Filtro de intents de ejemplo:

```
<activity ...>
    <intent-filter>
        <action android:name="com.google.android.gms.actions.CREATE_NOTE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="*/*" />
    </intent-filter>
</activity>
```

# Teléfono

## Iniciar una llamada telefónica

Para abrir la app de teléfono y marcar un número, usa la acción [ACTION\\_DIAL](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_DIAL) ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_DIAL](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_DIAL)) y especifica un número de teléfono usando el esquema de URI definido a continuación. Cuando se abre la aplicación de teléfono, se muestra el número de teléfono, pero el usuario debe presionar el botón *Llamar* para comenzar la llamada.

Para realizar una llamada telefónica directamente, usa la acción [ACTION\\_CALL](#)





([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

### Google Voice Actions

- "llamar al 555-5555"
- "llamar a Bob"
- "llamar al buzón de voz"

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CALL](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CALL)) y especifica un número de teléfono usando el esquema de URI definido a continuación. Cuando se abre la aplicación de teléfono, comienza la llamada; el usuario no necesita presionar el botón *Llamar*.

La acción **ACTION\_CALL**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CALL](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CALL)) requiere que agregues el permiso **CALL\_PHONE** al archivo del manifiesto:

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

## Acción

- **ACTION\_DIAL**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_DIAL](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_DIAL))

: abre la aplicación de teléfono o marcador.

- **ACTION\_CALL**

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_CALL](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_CALL))

: inicia una llamada telefónica (requiere el permiso **CALL\_PHONE**).

## Esquema de URI de datos

- **tel:**<phone-number>
- **voicemail:**<phone-number>

## Tipo de MIME

Ninguno

Los números de teléfono válidos son aquellos definidos en [IETF RFC 3966](http://tools.ietf.org/html/rfc3966) (<http://tools.ietf.org/html/rfc3966>). Algunos ejemplos son los siguientes:

- `tel:2125551212`
- `tel:(212) 555 1212`

El marcador del teléfono es bueno para normalizar esquemas, como los números de teléfono. Por ello, el esquema descrito no es estrictamente obligatorio en el método `Uri.parse()`.

([https://developer.android.com/reference/android/net/Uri?hl=es-419#parse\(java.lang.String\)](https://developer.android.com/reference/android/net/Uri?hl=es-419#parse(java.lang.String))). No obstante, si no probaste un esquema o no estás seguro de que pueda manejarse, usa el método `Uri.fromParts()`.

([https://developer.android.com/reference/android/net/Uri?hl=es-419#fromParts\(java.lang.String,%20java.lang.String,%20java.lang.String\)](https://developer.android.com/reference/android/net/Uri?hl=es-419#fromParts(java.lang.String,%20java.lang.String,%20java.lang.String)))

.

## Ejemplo de intent:

[Kotlin](#) (#kotlin)[Java](#) (#java)

```
public void dialPhoneNumber(String phoneNumber) {
    Intent intent = new Intent(Intent.ACTION_DIAL);
    intent.setData(Uri.parse("tel:" + phoneNumber));
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

## Buscar

### Buscar usando una aplicación específica

Para admitir la búsqueda dentro del contexto de tu aplicación, declara un filtro de intents en ella con la acción



`SEARCH_ACTION`, como se muestra en el filtro de intents de ejemplo a continuación.

### Voice search in your app



#### Acción

`"com.google.android.gms.actions.SEARCH_ACTION"`

([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

#### Google Voice Actions

- *"buscar videos de gatos en myvideoapp"*

Admisión de consultas de búsqueda a partir de Google Voice Actions.

#### Extras

##### QUERY

(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)

Una string que contiene la consulta de búsqueda.

#### Filtro de intents de ejemplo:

```
<activity android:name=".SearchActivity">
  <intent-filter>
    <action android:name="com.google.android.gms.actions.SEARCH_ACTION" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

## Realizar una búsqueda web

Para iniciar una búsqueda web, usa la acción `ACTION_WEB_SEARCH`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_WEB\\_SEARCH](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_WEB_SEARCH))

y especifica la string de búsqueda en el extra `SearchManager.QUERY`

(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>).

#### Acción

##### ACTION\_WEB\_SEARCH

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_WEB\\_SEARCH](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_WEB_SEARCH))

## Esquema de URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Extras

### SearchManager.QUERY

(<https://developer.android.com/reference/android/app/SearchManager?hl=es-419#QUERY>)

La string de búsqueda.

## Ejemplo de intent:

### Kotlin (#kotlin)

### Java (#java)

```
public void searchWeb(String query) {  
    Intent intent = new Intent(Intent.ACTION_WEB_SEARCH);  
    intent.putExtra(SearchManager.QUERY, query);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

## Configuración

### Abrir una sección específica de Configuración

Para abrir una pantalla en la configuración del sistema cuando la aplicación requiere que el usuario cambie algo, usa una de las siguientes acciones de intent a fin de abrir la pantalla de configuración correspondiente al nombre de la acción.

## Acción

### ACTION\_SETTINGS

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_SETTINGS))

### ACTION\_WIRELESS\_SETTINGS

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_WIRELESS\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_WIRELESS_SETTINGS))

#### **ACTION\_AIRPLANE\_MODE\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_AIRPLANE\\_MODE\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_AIRPLANE_MODE_SETTINGS))

#### **ACTION\_WIFI\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_WIFI\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_WIFI_SETTINGS))

#### **ACTION\_APN\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_APN\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_APN_SETTINGS))

#### **ACTION\_BLUETOOTH\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_BLUETOOTH\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_BLUETOOTH_SETTINGS))

#### **ACTION\_DATE\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_DATE\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_DATE_SETTINGS))

#### **ACTION\_LOCALE\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_LOCALE\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_LOCALE_SETTINGS))

#### **ACTION\_INPUT\_METHOD\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_INPUT\\_METHOD\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_INPUT_METHOD_SETTINGS))

#### **ACTION\_DISPLAY\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_DISPLAY\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_DISPLAY_SETTINGS))

#### **ACTION\_SECURITY\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_SECURITY\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_SECURITY_SETTINGS))

#### **ACTION\_LOCATION\_SOURCE\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_LOCATION\\_SOURCE\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_LOCATION_SOURCE_SETTINGS))

#### **ACTION\_INTERNAL\_STORAGE\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_INTERNAL\\_STORAGE\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_INTERNAL_STORAGE_SETTINGS))

#### **ACTION\_MEMORY\_CARD\_SETTINGS**

([https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION\\_MEMORY\\_CARD\\_SETTINGS](https://developer.android.com/reference/android/provider/Settings?hl=es-419#ACTION_MEMORY_CARD_SETTINGS))

Consulta la documentación de [Settings](#)

(<https://developer.android.com/reference/android/provider/Settings?hl=es-419>) para ver las pantallas de configuración adicionales que estén disponibles.

## Esquema de URI de datos

Ninguno

## Tipo de MIME

Ninguno

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void openWifiSettings() {
    Intent intent = new Intent(Settings.ACTION_WIFI_SETTINGS);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

# Mensajería de texto

## Redactar un mensaje SMS o MMS con archivos adjuntos

Para iniciar un mensaje de texto SMS o MMS, usa una de las acciones de intent que se indican a continuación y especifica detalles del mensaje como el número de teléfono, el asunto y el cuerpo del mensaje usando las claves adicionales enumeradas a continuación.

### Acción

#### ACTION\_SENDTO

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SENDTO](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SENDTO))

o

#### ACTION\_SEND

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SEND](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SEND)) o

#### ACTION\_SEND\_MULTIPLE

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SEND\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SEND_MULTIPLE))

## Esquema de URI de datos

sms: <phone\_number>

`smsto: <phone_number>``mms: <phone_number>``mmsto: <phone_number>`

Cada uno de estos esquemas se maneja de la misma manera.

## Tipo de MIME

`"text/plain"``"image/*"``"video/*"`

## Extras

`"subject"`

Una string para el asunto del mensaje (generalmente para MMS solo).

`"sms_body"`

Una string para el texto del mensaje.

### EXTRA\_STREAM

([https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA\\_STREAM](https://developer.android.com/reference/android/content/Intent?hl=es-419#EXTRA_STREAM))

Un Uri (<https://developer.android.com/reference/android/net/Uri?hl=es-419>) que apunta a la imagen o el video por adjuntar. Si usas la acción

### ACTION\_SEND\_MULTIPLE

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SEND\\_MULTIPLE](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SEND_MULTIPLE))

, este campo extra debe ser un ArrayList

(<https://developer.android.com/reference/java/util/ArrayList?hl=es-419>) de Uri

(<https://developer.android.com/reference/android/net/Uri?hl=es-419>) que apunte a las imágenes o los videos que se adjuntarán.

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void composeMmsMessage(String message, Uri attachment) {  
    Intent intent = new Intent(Intent.ACTION_SENDTO);
```

```

        intent.setType(HTTP.PLAIN_TEXT_TYPE);
        intent.putExtra("sms_body", message);
        intent.putExtra(Intent.EXTRA_STREAM, attachment);
        if (intent.resolveActivity(getPackageManager()) != null) {
            startActivity(intent);
        }
    }
}

```

Si deseas asegurarte de que solo una app de mensajería de texto (y no otra app de redes sociales o correo electrónico) maneje la intent, usa la acción **`ACTION_SENDTO`** ([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_SENDTO](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_SENDTO)) e incluye el esquema de datos **`"smsto:"`**. Por ejemplo:

**Kotlin** (#kotlin)**Java**  
(#java)

```

public void composeMmsMessage(String message, Uri attachment) {
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setData(Uri.parse("smsto:")); // This ensures only SMS apps
    intent.putExtra("sms_body", message);
    intent.putExtra(Intent.EXTRA_STREAM, attachment);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
}

```

### Filtro de intents de ejemplo:

```

<activity ...>
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <data android:type="text/plain" />
        <data android:type="image/*" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>

```



**Nota:** Si estás desarrollando una aplicación de mensajería SMS/MMS, debes implementar filtros de intents para varias acciones adicionales a fin de que esté disponible como la *aplicación de SMS predeterminada* en Android 4.4 y versiones posteriores. Para obtener más información, consulta la documentación en [Telephony](https://developer.android.com/reference/android/provider/Telephony?hl=es-419) (<https://developer.android.com/reference/android/provider/Telephony?hl=es-419>).

## Navegador web

### Cargar una URL web

Para abrir una página web, usa la acción `ACTION_VIEW`



([https://developers.google.com/voice-actions/system/?hl=es-419#system\\_actions\\_reference](https://developers.google.com/voice-actions/system/?hl=es-419#system_actions_reference))

#### Google Voice Actions

- "abrir ejemplo.com"

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW)) y especifica la URL web en los datos de la intent.

### Acción

#### `ACTION_VIEW`

([https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION\\_VIEW](https://developer.android.com/reference/android/content/Intent?hl=es-419#ACTION_VIEW))

### Esquema de URI de datos

`http: <URL>`

`https: <URL>`

### Tipo de MIME

`"text/plain"`

`"text/html"`

```
"application/xhtml+xml"
```

```
"application/vnd.wap.xhtml+xml"
```

## Ejemplo de intent:

Kotlin (#kotlin)Java  
(#java)

```
public void openWebPage(String url) {  
    Uri webpage = Uri.parse(url);  
    Intent intent = new Intent(Intent.ACTION_VIEW, webpage);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

## Filtro de intents de ejemplo:

```
<activity ...>  
    <intent-filter>  
        <action android:name="android.intent.action.VIEW" />  
        <!-- Include the host attribute if you want your app to respond  
             only to URLs with your app's domain. -->  
        <data android:scheme="http" android:host="www.example.com" />  
        <category android:name="android.intent.category.DEFAULT" />  
        <!-- The BROWSABLE category is required to get links from web pages.  
             <category android:name="android.intent.category.BROWSABLE" />  
    </intent-filter>  
</activity>
```

**Sugerencia:** Si la aplicación de Android brinda una funcionalidad similar a la de tu sitio web, incluye un filtro de intent para las URL que apunten a él. Luego, si los usuarios tienen instalada tu aplicación, los vínculos de correos electrónicos u otras páginas web que apunten a tu sitio web abrirán tu aplicación Android en lugar de tu página web.

# Verificar intents con Android Debug Bridge

Para verificar que tu aplicación responda a las intents que quieres admitir, puedes usar la herramienta `adb` (<https://developer.android.com/tools/help/adb?hl=es-419>) para desencadenar intents específicas:

1. Configura un dispositivo Android para el desarrollo (<https://developer.android.com/tools/device?hl=es-419#setting-up>) o usa un dispositivo virtual (<https://developer.android.com/tools/devices/emulator?hl=es-419#avds>).
2. Instala una versión de tu app que maneje las intents que quieres admitir.
3. Desencadena una intent usando `adb`:

```
adb shell am start -a <ACTION> -t <MIME_TYPE> -d <DATA> \
  -e <EXTRA_NAME> <EXTRA_VALUE> -n <ACTIVITY>
```

Por ejemplo:

```
adb shell am start -a android.intent.action.DIAL \
  -d tel:555-5555 -n org.example.MyApp/.MyActivity
```

4. Si defines los filtros de intents requeridos, tu aplicación debe administrar la intent.

Para obtener más información, consulta Comandos Shell de ADB

(<https://developer.android.com/tools/help/shell?hl=es-419#am>).

El contenido y las muestras de código que aparecen en esta página están sujetas a las licencias que se describen en la Licencia de Contenido (<https://developer.android.com/license?hl=es-419>). Java y OpenJDK son marcas registradas de Oracle o sus afiliados.

Última actualización: 2021-10-27 (UTC)