Intents y filtros de intents

Una [Intent](https://developer.android.com/reference/android/content/Intent.html) es un objeto de mensajería que puedes usar para solicitar una acción de otro [componente de una app](https://developer.android.com/guide/components/fundamentals.html#Components). Si bien las intents facilitan la comunicación entre componentes de varias formas, existen tres casos de uso principales:

* **Iniciar una actividad**

Una [Activity](https://developer.android.com/reference/android/app/Activity.html) representa una única pantalla en una aplicación. Puedes iniciar una nueva instancia de una [Activity](https://developer.android.com/reference/android/app/Activity.html) pasando una [Intent](https://developer.android.com/reference/android/content/Intent.html) a [startActivity()](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)). La [Intent](https://developer.android.com/reference/android/content/Intent.html) describe la actividad que se debe iniciar y contiene los datos necesarios para ello.

Si deseas recibir un resultado de la actividad cuando finalice, llama a [startActivityForResult()](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent,%20int)). La actividad recibe el resultado como un objeto [Intent](https://developer.android.com/reference/android/content/Intent.html) separado en la devolución de llamada de [onActivityResult()](https://developer.android.com/reference/android/app/Activity.html#onActivityResult(int,%20int,%20android.content.Intent)) de la actividad. Para obtener más información, consulta la guía [Actividades](https://developer.android.com/guide/components/activities.html).

* **Iniciar un servicio**

Un [Service](https://developer.android.com/reference/android/app/Service.html) es un componente que realiza operaciones en segundo plano sin una interfaz de usuario. Con Android 5.0 (nivel de API 21) y versiones posteriores, puedes iniciar un servicio con [JobScheduler](https://developer.android.com/reference/android/app/job/JobScheduler.html). Para obtener más información acerca de [JobScheduler](https://developer.android.com/reference/android/app/job/JobScheduler.html), consulta su [API-reference documentation](https://developer.android.com/reference/android/app/job/JobScheduler.html).

En las versiones anteriores a Android 5.0 (nivel de API 21), puedes iniciar un servicio usando métodos de la clase [Service](https://developer.android.com/reference/android/app/Service.html). Puedes iniciar un servicio para realizar una operación única (como descargar un archivo) pasando una [Intent](https://developer.android.com/reference/android/content/Intent.html) a [startService()](https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)). La [Intent](https://developer.android.com/reference/android/content/Intent.html) describe el servicio que se debe iniciar y contiene los datos necesarios para ello.

Si el servicio está diseñado con una interfaz cliente-servidor, puedes establecer un enlace con el servicio de otro componente pasando una [Intent](https://developer.android.com/reference/android/content/Intent.html) a [bindService()](https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent,%20android.content.ServiceConnection,%20int)). Para obtener más información, consulta la guía [Servicios](https://developer.android.com/guide/components/services.html).

* **Transmitir una emisión**

Una emisión es un aviso que cualquier aplicación puede recibir. El sistema transmite varias emisiones de eventos, como cuando se inicia el sistema o comienza a cargarse el dispositivo. Puedes transmitir una emisión a otras apps pasando una [Intent](https://developer.android.com/reference/android/content/Intent.html) a [sendBroadcast()](https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)) o [sendOrderedBroadcast()](https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent,%20java.lang.String)).

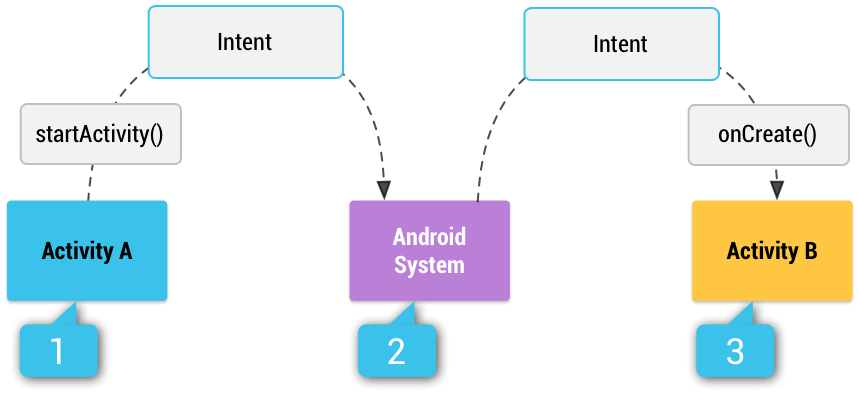
El resto de esta página explica cómo funcionan las intents y cómo debes usarlas. Para obtener información relacionada, consulta [Cómo actualizar con otras apps](https://developer.android.com/training/basics/intents/index.html) y [Cómo compartir contenido](https://developer.android.com/training/sharing/index.html).

Tipos de intents

Existen dos tipos de intents:

* Las **intents explícitas** especifican qué aplicación las administrará, ya sea incluyendo el nombre del paquete de la app de destino o el nombre de clase del componente completamente calificado. Normalmente, el usuario usa una intent explícita para iniciar un componente en su propia aplicación porque conoce el nombre de clase de la actividad o el servicio que desea iniciar. Por ejemplo, puedes utilizarla para iniciar una actividad nueva en respuesta a una acción del usuario o iniciar un servicio para descargar un archivo en segundo plano.
* Las **intents implícitas** no nombran el componente específico, pero, en cambio, declaran una acción general para realizar, lo cual permite que un componente de otra aplicación la maneje. Por ejemplo, si deseas mostrar al usuario una ubicación en un mapa, puedes usar una intent implícita para solicitar que otra aplicación apta muestre una ubicación específica en un mapa.

La figura 1 muestra cómo se usa una intent para iniciar una actividad. Cuando el objeto [Intent](https://developer.android.com/reference/android/content/Intent.html) nombra un componente de actividad específico de forma explícita, el sistema inicia ese componente al instante.



**Figura 1:** Ilustración de la forma en que se entrega una intent implícita mediante el sistema para iniciar otra actividad. **[1]** La *actividad A* crea una [Intent](https://developer.android.com/reference/android/content/Intent.html) con una descripción de acción y la pasa a [startActivity()](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)). **[2]** El sistema Android busca en todas las apps un filtro de intents que coincida con ella. Cuando se encuentra una coincidencia, **[3]** el sistema inicia la actividad coincidente (*actividad B*) invocando su método [onCreate()](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) y pasándolo a la [Intent](https://developer.android.com/reference/android/content/Intent.html).

Cuando usas una intent implícita, el sistema Android busca el componente apropiado para iniciar comparando el contenido de la intent con los *filtros de intents* declarados en el [archivo de manifiesto](https://developer.android.com/guide/topics/manifest/manifest-intro.html) de otras aplicaciones en el dispositivo. Si la intent coincide con un filtro de intents, el sistema inicia ese componente y le entrega el objeto [Intent](https://developer.android.com/reference/android/content/Intent.html). Si varios filtros de intents son compatibles, el sistema muestra un cuadro de diálogo para que el usuario pueda elegir la aplicación que se debe usar.

Un filtro de intents es una expresión en el archivo de manifiesto de una aplicación que especifica el tipo de intent que el componente podría recibir. Por ejemplo, declarar un filtro de intents para una actividad permite que otras aplicaciones la inicien directamente con un tipo de intent específico. Asimismo, si *no* declaras ningún filtro de intent para una actividad, esta solo se puede iniciar con una intent explícita.

**Advertencia:** Para garantizar que tu aplicación sea segura, siempre usa una intent explícita cuando inicies un [**Service**](https://developer.android.com/reference/android/app/Service.html) y no declares filtros de intents para los servicios. El uso de una intent explícita para iniciar un servicio es un riesgo de seguridad porque no puedes garantizar qué servicio responderá a la intent, y el usuario no puede ver qué servicio se inicia. A partir de Android 5.0 (nivel de API 21), el sistema arroja una excepción si llamas a [**bindService()**](https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent,%20android.content.ServiceConnection,%20int)) con una intent implícita.

Cómo crear una intent

Un objeto [Intent](https://developer.android.com/reference/android/content/Intent.html) tiene información que el sistema Android usa para determinar qué componente debe iniciar (como el nombre exacto del componente o la categoría que debe recibir la intent), además de información que el componente receptor usa para realizar la acción correctamente (por ejemplo, la acción que debe efectuar y los datos en los que debe actuar).

La información principal que contiene una [Intent](https://developer.android.com/reference/android/content/Intent.html) es la siguiente:

**Nombre del componente**

El nombre del componente que se debe iniciar.

Esto es opcional, pero es la información clave que hace que una intent sea *explícita*, lo que significa que la intent debe enviarse solamente al componente de la aplicación definido en el nombre del componente. Sin un nombre de componente, la intent es *implícita* y el sistema decide qué componente debe recibir la intent conforme la información restante que esta contiene (como la acción, los datos y la categoría, que se describen a continuación). Por lo tanto, si necesitas iniciar un componente específico en tu aplicación, debes especificar el nombre del componente.

**Nota:** Cuando inicies un [**Service**](https://developer.android.com/reference/android/app/Service.html), *especifica siempre el nombre del componente*. De lo contrario, no podrás asegurarte de qué servicio responderá a la intent, y el usuario no podrá ver qué servicio se inicia.

Este campo de la [Intent](https://developer.android.com/reference/android/content/Intent.html) es un objeto [ComponentName](https://developer.android.com/reference/android/content/ComponentName.html) que puedes especificar con un nombre de clase completamente calificado del componente de destino, incluido el nombre del paquete de la aplicación, como com.example.ExampleActivity. Puedes establecer el nombre del componente con [setComponent()](https://developer.android.com/reference/android/content/Intent.html#setComponent(android.content.ComponentName)), [setClass()](https://developer.android.com/reference/android/content/Intent.html#setClass(android.content.Context,%20java.lang.Class%3C?%3E)), [setClassName()](https://developer.android.com/reference/android/content/Intent.html#setClassName(java.lang.String,%20java.lang.String)) o el constructor [Intent](https://developer.android.com/reference/android/content/Intent.html).

**Acción**

Una string que especifica la acción genérica que se debe realizar (como *ver* o *elegir*).

En el caso de la intent de una emisión, es la acción que se produjo y que se está registrando. La acción determina cuál es la estructura del resto de la intent, especialmente la información que se incluye en los datos y extras.

Puedes especificar tus propias acciones para que las usen las intents en tu aplicación (o para que las usen otras aplicaciones a fin de invocar componentes en tu aplicación); pero, usualmente, debes especificar acciones constantes definidas por la clase [Intent](https://developer.android.com/reference/android/content/Intent.html) u otras clases de marcos de trabajo. Estas son algunas acciones comunes para iniciar una actividad:

[ACTION\_VIEW](https://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW)

Usa esta acción en una intent con [startActivity()](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) cuando tengas información que la actividad pueda mostrar al usuario, como una foto para ver en una app de galería o una dirección para ver en una app de mapas.

[ACTION\_SEND](https://developer.android.com/reference/android/content/Intent.html#ACTION_SEND)

También se conoce como la intent de *compartir* y debes usarla en una intent con [startActivity()](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) cuando tengas información que el usuario pueda compartir mediante otra app, como una app de correo electrónico o intercambio social.

Consulta la referencia de la clase [Intent](https://developer.android.com/reference/android/content/Intent.html) para conocer más constantes que definen las acciones genéricas. Otras acciones se definen en otras partes del marco de trabajo de Android, como en [Settings](https://developer.android.com/reference/android/provider/Settings.html) para las acciones que abren pantallas específicas en la aplicación de Configuración del sistema.

Puedes especificar la acción de una intent con [setAction()](https://developer.android.com/reference/android/content/Intent.html#setAction(java.lang.String)) o un constructor [Intent](https://developer.android.com/reference/android/content/Intent.html).

Si defines tus propias acciones, asegúrate de incluir el nombre del paquete de la aplicación como prefijo, tal como se muestra en el siguiente ejemplo.

[KOTLIN](https://developer.android.com/guide/components/intents-filters.html#kotlin)[JAVA](https://developer.android.com/guide/components/intents-filters.html#java)

static final String ACTION\_TIMETRAVEL = "com.example.action.TIMETRAVEL";

**Datos**

El URI (un objeto [Uri](https://developer.android.com/reference/android/net/Uri.html)) que hace referencia a los datos en los que se debe realizar la acción o el tipo de MIME de esos datos. El tipo de datos suministrado está generalmente determinado por la acción de la intent. Por ejemplo, si la acción es [ACTION\_EDIT](https://developer.android.com/reference/android/content/Intent.html#ACTION_EDIT), los datos deben contener el URI del documento que se debe editar.

Cuando creas una intent, es importante especificar el tipo de datos (su tipo de MIME) además de su URI. Por ejemplo, una actividad que puede mostrar imágenes probablemente no sea capaz de reproducir un archivo de audio aunque los formatos de URI sean similares. Especificar el tipo de MIME de tus datos ayuda al sistema Android a encontrar el mejor componente para recibir tu intent. Sin embargo, el tipo de MIME a veces se puede deducir del URI, especialmente cuando los datos son un URI content:. Un URI content: indica que los datos se encuentran en el dispositivo y son controlados por un [ContentProvider](https://developer.android.com/reference/android/content/ContentProvider.html), lo que hace que el sistema pueda ver el tipo de datos de MIME.

Para establecer solamente el URI de datos, llama a [setData()](https://developer.android.com/reference/android/content/Intent.html#setData(android.net.Uri)). Para establecer solo el tipo de MIME, llama a [setType()](https://developer.android.com/reference/android/content/Intent.html#setType(java.lang.String)). Si es necesario, puedes establecer ambos explícitamente con [setDataAndType()](https://developer.android.com/reference/android/content/Intent.html#setDataAndType(android.net.Uri,%20java.lang.String)).

**Advertencia:** Si deseas establecer tanto el URI como el tipo de MIME, *no* llames a [**setData()**](https://developer.android.com/reference/android/content/Intent.html#setData(android.net.Uri)) ni a [**setType()**](https://developer.android.com/reference/android/content/Intent.html#setType(java.lang.String)), ya que cada uno anula el valor del otro. Siempre usa [**setDataAndType()**](https://developer.android.com/reference/android/content/Intent.html#setDataAndType(android.net.Uri,%20java.lang.String)) para establecer el URI y el tipo de MIME.

**Categoría**

Una string que contiene información adicional sobre el tipo de componente que la intent debe manejar. En una intent, se puede incluir la cantidad deseada de descripciones de categorías, pero la mayoría de las intents no requieren una categoría. Estas son algunas categorías comunes:

[CATEGORY\_BROWSABLE](https://developer.android.com/reference/android/content/Intent.html#CATEGORY_BROWSABLE)

La actividad de destino permite que la inicie un navegador web para mostrar datos a los que hace referencia un vínculo, como una imagen o un mensaje de correo electrónico.

[CATEGORY\_LAUNCHER](https://developer.android.com/reference/android/content/Intent.html#CATEGORY_LAUNCHER)

La actividad es la actividad inicial de una tarea y está enumerada en el selector de la aplicación del sistema.

Para ver la lista completa de categorías, consulta la descripción de la clase [Intent](https://developer.android.com/reference/android/content/Intent.html).

También puedes especificar una categoría con [addCategory()](https://developer.android.com/reference/android/content/Intent.html#addCategory(java.lang.String)).

Las propiedades nombradas anteriormente (nombre de componente, acción, datos y categoría) representan las características definitorias de una intent. Mediante la lectura de estas propiedades, el sistema Android puede resolver qué componente de la aplicación debe iniciar. Sin embargo, una intent puede tener información adicional que no afecte cómo se resuelve en un componente de la aplicación. Una intent también puede incluir la siguiente información:

**Extras**

Pares clave-valor que contienen información adicional necesaria para completar la acción solicitada. Al igual que algunas acciones usan tipos particulares de URI de datos, algunas acciones también usan extras específicos.

Puedes agregar datos adicionales con varios métodos [putExtra()](https://developer.android.com/reference/android/content/Intent.html#putExtra(java.lang.String,%20android.os.Bundle)), cada uno de los cuales acepta dos parámetros: el nombre de la clave y el valor. También puedes crear un objeto [Bundle](https://developer.android.com/reference/android/os/Bundle.html) con los datos adicionales y, luego, insertar el [Bundle](https://developer.android.com/reference/android/os/Bundle.html) en la [Intent](https://developer.android.com/reference/android/content/Intent.html) con [putExtras()](https://developer.android.com/reference/android/content/Intent.html#putExtras(android.content.Intent)).

Por ejemplo, al crear una intent para enviar un correo electrónico con [ACTION\_SEND](https://developer.android.com/reference/android/content/Intent.html#ACTION_SEND), puedes especificar el destinatario *para* con la clave [EXTRA\_EMAIL](https://developer.android.com/reference/android/content/Intent.html#EXTRA_EMAIL) y el *asunto* con la clave [EXTRA\_SUBJECT](https://developer.android.com/reference/android/content/Intent.html#EXTRA_SUBJECT).

La clase [Intent](https://developer.android.com/reference/android/content/Intent.html) especifica muchas constantes EXTRA\_\* para tipos de datos estandarizados. Si tienes que declarar tus propias claves extra (para intents que recibe tu aplicación), asegúrate de incluir el nombre del paquete de la aplicación como prefijo, tal como se muestra en el siguiente ejemplo:

[KOTLIN](https://developer.android.com/guide/components/intents-filters.html#kotlin)[JAVA](https://developer.android.com/guide/components/intents-filters.html#java)

static final String EXTRA\_GIGAWATTS = "com.example.EXTRA\_GIGAWATTS";

**Advertencia:** No uses datos [**Parcelable**](https://developer.android.com/reference/android/os/Parcelable.html) ni [**Serializable**](https://developer.android.com/reference/java/io/Serializable.html) para enviar una intent que debería recibir otra app. Si una app intenta acceder a datos de un objeto [**Bundle**](https://developer.android.com/reference/android/os/Bundle.html), pero no tiene acceso a la clase empaquetada o serializada, el sistema arroja un [**RuntimeException**](https://developer.android.com/reference/java/lang/RuntimeException.html).

**Indicadores**

Los indicadores se definen en la clase [Intent](https://developer.android.com/reference/android/content/Intent.html) que funciona como metadatos de la intent. Los indicadores pueden indicar al sistema Android cómo iniciar una actividad (por ejemplo, a qué [tarea](https://developer.android.com/guide/components/tasks-and-back-stack.html) debe pertenecer) y cómo tratarla después de que se inicia (por ejemplo, si pertenece a la lista de actividades recientes).

Para obtener más información, consulta el método [setFlags()](https://developer.android.com/reference/android/content/Intent.html#setFlags(int)).

Ejemplo de una intent explícita

Una intent explícita es una intent que se usa para iniciar un componente específico de la aplicación, como una actividad o un servicio particular en la aplicación. Para crear una intent explícita, define el nombre de componente del objeto [Intent](https://developer.android.com/reference/android/content/Intent.html) (todas las otras propiedades de la intent son opcionales).

Por ejemplo, si creaste un servicio en tu app denominado DownloadService, diseñado para descargar un archivo de la Web, puedes iniciarlo con el siguiente código:

[KOTLIN](https://developer.android.com/guide/components/intents-filters.html#kotlin)[JAVA](https://developer.android.com/guide/components/intents-filters.html#java)

// Executed in an Activity, so 'this' is the [**Context**](https://developer.android.com/reference/android/content/Context.html)  
// The fileUrl is a string URL, such as "http://www.example.com/image.png"  
Intent downloadIntent = new Intent(this, DownloadService.class);  
downloadIntent.setData([**Uri.parse**](https://developer.android.com/reference/android/net/Uri.html#parse(java.lang.String))(fileUrl));  
startService(downloadIntent);

El constructor [Intent(Context, Class)](https://developer.android.com/reference/android/content/Intent.html#Intent(android.content.Context,%20java.lang.Class%3C?%3E)) proporciona [Context](https://developer.android.com/reference/android/content/Context.html) a la aplicación; y el componente, un objeto [Class](https://developer.android.com/reference/java/lang/Class.html). De esta manera, la intent inicia explícitamente la clase DownloadService en la app.

Para obtener más información sobre la creación y el inicio de un servicio, consulta la guía [Servicios](https://developer.android.com/guide/components/services.html).

Ejemplo de una intent implícita

Una intent implícita especifica una acción que puede invocar cualquier aplicación en el dispositivo que puede realizar la acción. El uso de una intent implícita es útil cuando la aplicación no puede realizar la acción, pero otras aplicaciones probablemente sí, y quieres que el usuario elija qué aplicación usar.

Por ejemplo, si tienes contenido que quieres que el usuario comparta con otras personas, crea una intent con la acción [ACTION\_SEND](https://developer.android.com/reference/android/content/Intent.html#ACTION_SEND) y agrega extras que especifiquen el contenido para compartir. Cuando llamas a [startActivity()](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) con esa intent, el usuario puede elegir una aplicación mediante la cual compartir el contenido.

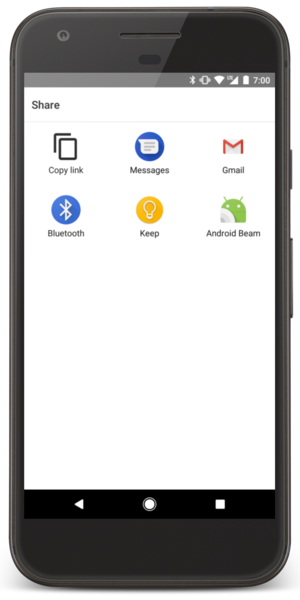
**Advertencia:** Es posible que un usuario no tenga *ninguna* aplicación para manejar la intent implícita que envías a [**startActivity()**](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)). O bien, es posible que no se pueda acceder a una aplicación debido a restricciones de perfil o a la configuración de un administrador. En ese caso, no se produce la llamada y se bloquea la app. Para verificar que una actividad recibirá la intent, llama a [**resolveActivity()**](https://developer.android.com/reference/android/content/Intent.html#resolveActivity(android.content.pm.PackageManager)) en el objeto [**Intent**](https://developer.android.com/reference/android/content/Intent.html). 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.html#startActivity(android.content.Intent)). Si el resultado es nulo, no debes usar la intent y, si es posible, debes inhabilitar la función que emite la intent. El siguiente ejemplo muestra cómo verificar que la intent resuelva una actividad. Este ejemplo no usa un URI, pero se declara el tipo de datos de la intent para especificar el contenido que tienen los extras.

[KOTLIN](https://developer.android.com/guide/components/intents-filters.html#kotlin)[JAVA](https://developer.android.com/guide/components/intents-filters.html#java)

// Create the text message with a string  
Intent sendIntent = new Intent();  
sendIntent.setAction(Intent.ACTION\_SEND);  
sendIntent.putExtra(Intent.EXTRA\_TEXT, textMessage);  
sendIntent.setType("text/plain");  
  
// Verify that the intent will resolve to an activity  
if (sendIntent.resolveActivity(getPackageManager()) != null) {  
    startActivity(sendIntent);  
}

Cuando se llama a [startActivity()](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)), el sistema examina todas las aplicaciones instaladas para determinar cuáles pueden manejar este tipo de intent (una intent con la acción [ACTION\_SEND](https://developer.android.com/reference/android/content/Intent.html#ACTION_SEND) y que tiene datos de texto/sin formato). Si solo hay una aplicación que puede manejarla, esta se abre inmediatamente y se le entrega la intent. Si varias actividades aceptan la intent, el sistema muestra un cuadro de diálogo, como el que se muestra en la figura 2, para que el usuario pueda elegir la aplicación que se debe usar.

Obtén más información sobre el lanzamiento de otras apps en [Cómo enviar al usuario a otra app](https://developer.android.com/training/basics/intents/sending.html).



**Figura 2:** Diálogo de selección

Cómo forzar un selector de apps

Cuando más de una aplicación responde a la intent implícita, el usuario puede seleccionar qué aplicación se debe usar y establecerla como la opción predeterminada para la acción. Esto es útil cuando se realiza una acción para la cual el usuario probablemente desee utilizar la misma aplicación en lo sucesivo, por ejemplo, al abrir una página web (los usuarios suelen preferir un solo navegador web).

No obstante, si varias aplicaciones responden a la intent y es posible que el usuario quiera usar una diferente cada vez, debes mostrar explícitamente un cuadro de diálogo de selección. El cuadro de diálogo de selección le pregunta al usuario qué aplicación desea utilizar para la acción (no puede seleccionar una aplicación predeterminada para la acción). Por ejemplo, cuando tu app realiza la acción de "compartir" con la acción [ACTION\_SEND](https://developer.android.com/reference/android/content/Intent.html#ACTION_SEND), es posible que los usuarios deseen concretar la compartición usando una app diferente según la situación en la que estén. Por ello, debes usar siempre el cuadro de diálogo de selección, como se muestra en la figura 2.

Para mostrar el diálogo de selección, crea una [Intent](https://developer.android.com/reference/android/content/Intent.html) usando [createChooser()](https://developer.android.com/reference/android/content/Intent.html#createChooser(android.content.Intent,%20java.lang.CharSequence)) y transfiérela a [startActivity()](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)), tal como se muestra en el siguiente ejemplo. Aquí se muestra un diálogo con una lista de apps que responden a la intent transferida al método [createChooser()](https://developer.android.com/reference/android/content/Intent.html#createChooser(android.content.Intent,%20java.lang.CharSequence)), con el texto proporcionado como título del diálogo.

[KOTLIN](https://developer.android.com/guide/components/intents-filters.html#kotlin)[JAVA](https://developer.android.com/guide/components/intents-filters.html#java)

Intent sendIntent = new Intent(Intent.ACTION\_SEND);  
...  
  
// Always use string resources for UI text.  
// This says something like "Share this photo with"  
String title = getResources().getString(R.string.chooser\_title);  
// Create intent to show the chooser dialog  
Intent chooser = Intent.createChooser(sendIntent, title);  
  
// Verify the original intent will resolve to at least one activity  
if (sendIntent.resolveActivity(getPackageManager()) != null) {  
    startActivity(chooser);  
}

Cómo recibir una intent implícita

Para informar las intents implícitas que puede recibir tu app, declara uno o más filtros para cada componente de esta con un elemento [<intent-filter>](https://developer.android.com/guide/topics/manifest/intent-filter-element.html) en el [archivo de manifiesto](https://developer.android.com/guide/topics/manifest/manifest-intro.html). Cada filtro de intents especifica el tipo de intents que acepta según la acción, los datos y la categoría de la intent. El sistema entregará la intent implícita al componente de la aplicación solo si la intent puede pasar por uno de los filtros.

**Nota:** Una intent explícita siempre se entrega al destino, independientemente de los filtros de intents que declare el componente.

Un componente de aplicación debe declarar filtros independientes para cada tarea única que puede hacer. Por ejemplo, una actividad de una aplicación de galería de imágenes puede tener dos filtros: uno para ver una imagen y otro para editarla. Cuando comienza la actividad, inspecciona la [Intent](https://developer.android.com/reference/android/content/Intent.html) y decide cómo comportarse conforme la información de la [Intent](https://developer.android.com/reference/android/content/Intent.html) (como mostrar los controles del editor o no).

Cada filtro de intents está definido por un elemento [<intent-filter>](https://developer.android.com/guide/topics/manifest/intent-filter-element.html) en el archivo de manifiesto de la app, anidado en el componente correspondiente (como un elemento [<activity>](https://developer.android.com/guide/topics/manifest/activity-element.html)). En el [<intent-filter>](https://developer.android.com/guide/topics/manifest/intent-filter-element.html), puedes especificar el tipo de intents que se aceptará con uno o más de estos tres elementos:

[<action>](https://developer.android.com/guide/topics/manifest/action-element.html)

Declara la acción de la intent aceptada, en el atributo name. El valor debe ser el de la string literal de una acción, no la constante de clase.

[<data>](https://developer.android.com/guide/topics/manifest/data-element.html)

Declara el tipo de datos que se acepta, mediante el uso de uno o más atributos que especifican varios aspectos del URI de datos (scheme, host, port, path, etc.) y el tipo de MIME.

[<category>](https://developer.android.com/guide/topics/manifest/category-element.html)

Declara la categoría de la intent aceptada, en el atributo name. El valor debe ser el de la string literal de una acción, no la constante de clase.

**Nota:** Para recibir intents implícitas, *debes incluir* la categoría [**CATEGORY\_DEFAULT**](https://developer.android.com/reference/android/content/Intent.html#CATEGORY_DEFAULT) en el filtro de intents. Los métodos [**startActivity()**](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)) y [**startActivityForResult()**](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent,%20int)) tratan todas las intents como si pertenecieran a la categoría [**CATEGORY\_DEFAULT**](https://developer.android.com/reference/android/content/Intent.html#CATEGORY_DEFAULT). Si no declaras esta categoría en el filtro de intents, no se aplicará ninguna intent implícita a la actividad.

Por ejemplo, a continuación, se muestra una declaración de actividad con un filtro de intents para recibir una intent [ACTION\_SEND](https://developer.android.com/reference/android/content/Intent.html#ACTION_SEND) cuando el tipo de datos es un texto:

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

Puedes crear un filtro que incluya más de una instancia de [<action>](https://developer.android.com/guide/topics/manifest/action-element.html), [<data>](https://developer.android.com/guide/topics/manifest/data-element.html) o [<category>](https://developer.android.com/guide/topics/manifest/category-element.html). Si lo haces, debes asegurarte de que el componente pueda administrar cualquier combinación de esos elementos de filtros.

Cuando quieras administrar varios tipos de intents, pero solo en combinaciones específicas de tipos de acciones, datos y categorías, deberás crear varios filtros de intents.

Una intent implícita se prueba con un filtro comparando la intent con cada uno de los tres elementos. Para que se entregue al componente, la intent debe pasar las tres pruebas. Si no pasa tan solo una de ellas, el sistema Android no la entregará al componente. Sin embargo, como un componente puede tener varios filtros de intents, una intent que no pase por el filtro de un componente quizá pase por otro filtro. Se proporciona más información sobre la manera en que el sistema resuelve las intents en la sección sobre [Resolución de intents](https://developer.android.com/guide/components/intents-filters.html#Resolution) que aparece más abajo.

**Advertencia:**El uso de un filtro de intents no es una manera segura de prevenir que otras apps inicien tus componentes. Si bien los filtros de intents restringen un componente para que responda solamente a ciertos tipos de intents implícitas, otra app puede iniciar potencialmente el componente de tu app usando una intent explícita si el desarrollador determina los nombres de los componentes. Si es importante que *solo tu app* pueda iniciar uno de los componentes, no declares filtros de intents en tu manifiesto. En su lugar, establece el atributo [**exported**](https://developer.android.com/guide/topics/manifest/activity-element.html#exported) en **"false"** para ese componente.

Para evitar ejecutar accidentalmente el [**Service**](https://developer.android.com/reference/android/app/Service.html) de otra aplicación, siempre usa una intent explícita cuando inicies tu propio servicio.

**Nota:** Para todas las actividades, debes declarar tus filtros de intents en el archivo de manifiesto. Sin embargo, los filtros de receptores de emisiones pueden registrarse de forma dinámica llamando a [**registerReceiver()**](https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver,%20android.content.IntentFilter,%20java.lang.String,%20android.os.Handler)). Puedes anular el registro del receptor con [**unregisterReceiver()**](https://developer.android.com/reference/android/content/Context.html#unregisterReceiver(android.content.BroadcastReceiver)). Si lo haces, tu aplicación esperará emisiones específicas solo durante un plazo especificado mientras esté ejecutándose.

Ejemplos de filtros

Para demostrar algunos comportamientos de filtros de intents, se incluye un ejemplo del archivo de manifiesto de una app de intercambio social:

<activity android:name="MainActivity">  
    <!-- This activity is the main entry, should appear in app launcher -->  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>  
  
<activity android:name="ShareActivity">  
    <!-- This activity handles "SEND" actions with text data -->  
    <intent-filter>  
        <action android:name="android.intent.action.SEND"/>  
        <category android:name="android.intent.category.DEFAULT"/>  
        <data android:mimeType="text/plain"/>  
    </intent-filter>  
    <!-- This activity also handles "SEND" and "SEND\_MULTIPLE" with media data -->  
    <intent-filter>  
        <action android:name="android.intent.action.SEND"/>  
        <action android:name="android.intent.action.SEND\_MULTIPLE"/>  
        <category android:name="android.intent.category.DEFAULT"/>  
        <data android:mimeType="application/vnd.google.panorama360+jpg"/>  
        <data android:mimeType="image/\*"/>  
        <data android:mimeType="video/\*"/>  
    </intent-filter>  
</activity>

La primera actividad, MainActivity, es el punto de entrada principal (la actividad que se abre cuando el usuario lanza inicialmente la app con el ícono de selector):

* La acción [ACTION\_MAIN](https://developer.android.com/reference/android/content/Intent.html#ACTION_MAIN) indica que este es el punto de entrada principal y que no se esperan datos de intents.
* La categoría [CATEGORY\_LAUNCHER](https://developer.android.com/reference/android/content/Intent.html#CATEGORY_LAUNCHER) indica que el ícono de esta actividad debe colocarse en el selector de la aplicación del sistema. Si el elemento [<activity>](https://developer.android.com/guide/topics/manifest/activity-element.html) no especifica un ícono con icon, el sistema usa el del elemento [<application>](https://developer.android.com/guide/topics/manifest/application-element.html).

Estos dos deben sincronizarse para que la actividad aparezca en el selector de la aplicación.

La segunda actividad, ShareActivity, tiene como objetivo facilitar el intercambio de texto y contenido multimedia. Aunque los usuarios pueden ingresar a esta actividad navegando hasta ella desde MainActivity, también pueden ingresar a ShareActivity directamente desde otra app que emita una intent implícita que coincida con uno de los dos filtros de intents.

**Nota:** El tipo de MIME, [**application/vnd.google.panorama360+jpg**](https://developers.google.com/panorama/android/), es un tipo de datos especial que especifica fotos panorámicas que puedes administrar con las API de [Google panorama](https://developer.android.com/reference/com/google/android/gms/panorama/package-summary.html).

Cómo usar una intent pendiente

Un objeto [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) es un contenedor de un objeto [Intent](https://developer.android.com/reference/android/content/Intent.html). El propósito principal de un objeto [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) es otorgar permiso a una aplicación externa para usar la [Intent](https://developer.android.com/reference/android/content/Intent.html) contenida como si se ejecutara desde un proceso de su propia aplicación.

Estos son algunos de los casos de uso principales de una intent pendiente:

* Declarar una intent que se ejecutará cuando el usuario realice una acción con su [notificación](https://developer.android.com/guide/topics/ui/notifiers/notifications.html) (el objeto [NotificationManager](https://developer.android.com/reference/android/app/NotificationManager.html) del sistema Android ejecuta la [Intent](https://developer.android.com/reference/android/content/Intent.html)).
* Declarar una intent que se ejecutará cuando el usuario realice una acción con el [widget de tu aplicación](https://developer.android.com/guide/topics/appwidgets/index.html) (la app de la pantalla principal ejecuta la [Intent](https://developer.android.com/reference/android/content/Intent.html)).
* Declarar una intent que se ejecutará en un momento especificado en el futuro (el [AlarmManager](https://developer.android.com/reference/android/app/AlarmManager.html) del sistema Android ejecuta la [Intent](https://developer.android.com/reference/android/content/Intent.html)).

Como cada objeto [Intent](https://developer.android.com/reference/android/content/Intent.html) está diseñado para ser manejado por un tipo específico de componente de la aplicación ([Activity](https://developer.android.com/reference/android/app/Activity.html), [Service](https://developer.android.com/reference/android/app/Service.html) o [BroadcastReceiver](https://developer.android.com/reference/android/content/BroadcastReceiver.html)), se debe tener la misma consideración al crear una [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html). Cuando use una intent pendiente, la aplicación no la ejecutará con una llamada como [startActivity()](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)). En cambio, debes declarar el tipo de componente deseado cuando crees la [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) llamando al método creador respectivo:

* [PendingIntent.getActivity()](https://developer.android.com/reference/android/app/PendingIntent.html#getActivity(android.content.Context,%20int,%20android.content.Intent,%20int)) para una [Intent](https://developer.android.com/reference/android/content/Intent.html) que inicia una [Activity](https://developer.android.com/reference/android/app/Activity.html).
* [PendingIntent.getService()](https://developer.android.com/reference/android/app/PendingIntent.html#getService(android.content.Context,%20int,%20android.content.Intent,%20int)) para una [Intent](https://developer.android.com/reference/android/content/Intent.html) que inicia un [Service](https://developer.android.com/reference/android/app/Service.html).
* [PendingIntent.getBroadcast()](https://developer.android.com/reference/android/app/PendingIntent.html#getBroadcast(android.content.Context,%20int,%20android.content.Intent,%20int)) para una [Intent](https://developer.android.com/reference/android/content/Intent.html) que inicia un [BroadcastReceiver](https://developer.android.com/reference/android/content/BroadcastReceiver.html).

A menos que la aplicación esté *recibiendo* intents pendientes de otras aplicaciones, los métodos indicados arriba para crear una [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) probablemente sean los únicos métodos [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent.html) que necesitarás.

Cada método toma el [Context](https://developer.android.com/reference/android/content/Context.html) actual de la aplicación, la [Intent](https://developer.android.com/reference/android/content/Intent.html) que desea encapsular y uno o más indicadores que especifican cómo se debe usar la intent (por ejemplo, si se puede usar más de una vez).

Se proporciona más información sobre el uso de intents pendientes con la documentación para cada caso de uso, como en las guías de [Notifications](https://developer.android.com/guide/topics/ui/notifiers/notifications.html) y API de [App Widgets](https://developer.android.com/guide/topics/appwidgets/index.html).

Resolución de intents

Cuando el sistema recibe una intent implícita para iniciar una actividad, busca la mejor actividad para la intent mediante la comparación de la intent con los filtros de intents basándose en tres aspectos:

* Acción
* Datos (tanto URI como tipo de datos)
* Categoría

En las siguientes secciones, se describe la manera de hacer coincidir las intents con los componentes apropiados según la forma de declarar el filtro de intents en un archivo de manifiesto de una app.

Prueba de acción

Para especificar las acciones de intents aceptadas, un filtro de intents puede declarar un número de elementos [<action>](https://developer.android.com/guide/topics/manifest/action-element.html) de cero en adelante, tal como se muestra en el siguiente ejemplo.

<intent-filter>  
    <action android:name="android.intent.action.EDIT" />  
    <action android:name="android.intent.action.VIEW" />  
    ...  
</intent-filter>

Para pasar el filtro, la acción especificada en la [Intent](https://developer.android.com/reference/android/content/Intent.html) debe coincidir con una de las acciones indicadas en el filtro.

Si el filtro no enumera ninguna acción, no hay nada con qué establecer la coincidencia, por lo que ninguna de las intents pasa la prueba. Sin embargo, si una [Intent](https://developer.android.com/reference/android/content/Intent.html) no especifica una acción, pasará la prueba (si el filtro contiene, al menos, una acción).

Prueba de categoría

Para especificar las categorías de intents aceptadas, un filtro de intents puede declarar un número de elementos [<category>](https://developer.android.com/guide/topics/manifest/category-element.html) de cero en adelante, tal como se muestra en el siguiente ejemplo.

<intent-filter>  
    <category android:name="android.intent.category.DEFAULT" />  
    <category android:name="android.intent.category.BROWSABLE" />  
    ...  
</intent-filter>

Para que una intent pase una prueba de categoría, cada categoría en la [Intent](https://developer.android.com/reference/android/content/Intent.html) debe coincidir con la categoría que figura en el filtro. No es necesario que suceda la situación inversa (el filtro de intents puede declarar más categorías de las que se especifican en la [Intent](https://developer.android.com/reference/android/content/Intent.html), y la [Intent](https://developer.android.com/reference/android/content/Intent.html) pasará la prueba). Por lo tanto, una intent sin categorías siempre debería pasar esta prueba, independientemente de qué categorías estén declaradas en el filtro.

**Nota:** Android aplica automáticamente la categoría [**CATEGORY\_DEFAULT**](https://developer.android.com/reference/android/content/Intent.html#CATEGORY_DEFAULT) a todas las intents implícitas que se pasan a [**startActivity()**](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) y a [**startActivityForResult()**](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent,%20int)). Por ello, si deseas que tu actividad reciba intents implícitas, debes incluir una categoría para **"android.intent.category.DEFAULT"** en los filtros de intents, tal como se muestra en el ejemplo de **<intent-filter>** anterior.

Prueba de datos

Para especificar los datos de intents que se aceptan, un filtro de intents puede declarar un número de elementos [<data>](https://developer.android.com/guide/topics/manifest/data-element.html) de cero en adelante, tal como se muestra en el siguiente ejemplo:

<intent-filter>  
    <data android:mimeType="video/mpeg" android:scheme="http" ... />  
    <data android:mimeType="audio/mpeg" android:scheme="http" ... />  
    ...  
</intent-filter>

Cada elemento [<data>](https://developer.android.com/guide/topics/manifest/data-element.html) puede especificar una estructura de URI y un tipo de datos (tipo de contenido multimedia de MIME). Cada parte del URI es un atributo independiente: scheme, host, port y path.

<scheme>://<host>:<port>/<path>

El siguiente ejemplo muestra valores posibles para estos atributos:

content://com.example.project:200/folder/subfolder/etc

En este URI, el esquema es content; el host, com.example.project; el puerto, 200; y la ruta, folder/subfolder/etc.

Cada uno de los atributos es opcional en un elemento [<data>](https://developer.android.com/guide/topics/manifest/data-element.html), pero existen dependencias lineales:

* Si no se especifica el esquema, se ignora el host.
* Si el host no se especifica, se ignora el puerto.
* Si no se especifican el esquema ni el host, se ignora la ruta de acceso.

Cuando se compara el URI de una intent con una especificación de URI en un filtro, solo se compara con las partes del URI que se incluyen en el filtro. Por ejemplo:

* Si un filtro especifica únicamente un esquema, todos los URI con ese esquema coinciden con el filtro.
* Si un filtro especifica un esquema y una autoridad, pero no una ruta de acceso, todos los URI con el mismo esquema y autoridad pasan el filtro, independientemente de la ruta que tengan.
* Si un filtro especifica un esquema, una autoridad y una ruta de acceso, solo los URI con el mismo esquema, autoridad y ruta pasan el filtro.

**Nota:** La especificación de una ruta de acceso puede contener un asterisco (\*) comodín para requerir solamente una coincidencia parcial del nombre de la ruta.

La prueba de datos compara tanto el URI como el tipo de MIME en la intent con el URI y el tipo de MIME especificados en el filtro. Las reglas son las siguientes:

1. Una intent que no contiene ni un URI ni un tipo de MIME pasa la prueba solo si el filtro no especifica ningún URI ni tipo de MIME.
2. Si una intent contiene un URI, pero no un tipo de MIME (ni explícitamente ni que se pueda inferir del URI), pasa la prueba solo si el URI coincide con el formato de URI del filtro y si el filtro tampoco especifica un tipo de MIME.
3. Si una intent contiene un tipo de MIME, pero no un URI, pasa la prueba solo si el filtro enumera el mismo tipo de MIME y no especifica un formato de URI.
4. Una intent que contiene tanto un URI como un tipo de MIME (explícitamente o que se pueda inferir del URI) pasa la parte de tipo de MIME de la prueba solo si ese tipo coincide con el enumerado en el filtro. Pasa la parte del URI de la prueba si este coincide con un URI del filtro o si tiene un URI content: o file:, y el filtro no especifica uno. Es decir, se supone que un componente admite datos content: y file: si el filtro *solo* indica un tipo de MIME.

**Nota:** Si una intent especifica un tipo de URI o MIME, la prueba de datos fallará si no hay elementos **<data>** en **<intent-filter>**.

Esta última regla (d), refleja la expectativa de que los componentes pueden obtener datos locales de un archivo o un proveedor de contenido. Por lo tanto, sus filtros pueden enumerar solamente un tipo de datos sin nombrar explícitamente los esquemas content: o file:. El siguiente ejemplo muestra un caso típico en que un elemento [<data>](https://developer.android.com/guide/topics/manifest/data-element.html) indica a Android que el componente puede obtener datos de imagen de un proveedor de contenido y mostrarlos:

<intent-filter>  
    <data android:mimeType="image/\*" />  
    ...  
</intent-filter>

Como la mayoría de los datos son proporcionados por proveedores de contenido, los filtros que especifican un tipo de datos y no especifican un URI son, tal vez, los más comunes.

Otra configuración común es aquella en la que los filtros tienen un esquema y un tipo de datos. Por ejemplo, un elemento [<data>](https://developer.android.com/guide/topics/manifest/data-element.html) como el siguiente indica a Android que el componente puede recuperar datos de video de la red para realizar la acción:

<intent-filter>  
    <data android:scheme="http" android:mimeType="video/\*" />  
    ...  
</intent-filter>

Coincidencias de intents

Se establecen coincidencias de las intents con filtros de intents no solo para detectar un componente de destino que se debe activar, sino también para detectar algo sobre el conjunto de componentes del dispositivo. Por ejemplo, la app de página principal completa el selector de la aplicación encontrando todas las actividades con filtros de intents que especifiquen la acción [ACTION\_MAIN](https://developer.android.com/reference/android/content/Intent.html#ACTION_MAIN) y la categoría [CATEGORY\_LAUNCHER](https://developer.android.com/reference/android/content/Intent.html#CATEGORY_LAUNCHER). Una coincidencia solo es exitosa si las acciones y categorías de la intent coinciden con el filtro, tal como se describe en la documentación de la clase [IntentFilter](https://developer.android.com/reference/android/content/IntentFilter.html).

Tu aplicación puede usar la coincidencia de intents de forma similar a como lo hace la app de pantalla principal. [PackageManager](https://developer.android.com/reference/android/content/pm/PackageManager.html) tiene un conjunto de métodos query...() que muestran todos los componentes que pueden aceptar una intent específica, al igual que una serie similar de métodos resolve...() a fin de determinar el mejor componente para responder a la intent. Por ejemplo, [queryIntentActivities()](https://developer.android.com/reference/android/content/pm/PackageManager.html#queryIntentActivities(android.content.Intent,%20int)) muestra una lista de todas las actividades que pueden utilizar la intent pasada como un argumento, y [queryIntentServices()](https://developer.android.com/reference/android/content/pm/PackageManager.html#queryIntentServices(android.content.Intent,%20int)) muestra una lista similar de servicios. Ninguno de los métodos activa los componentes; solo enumeran cuáles pueden responder. Existe un método similar, [queryBroadcastReceivers()](https://developer.android.com/reference/android/content/pm/PackageManager.html#queryBroadcastReceivers(android.content.Intent,%20int)), para receptores de emisiones.