Manual de usuario de la aplicación Filtrando Google sheets v0.5

# Índice

[0. Índice 1](#_Toc437795564)

[1. Autor 2](#_Toc437795565)

[2. Introducción 2](#_Toc437795566)

[3. Inicio rápido 2](#_Toc437795567)

[4. Parámetros de configuración 3](#_Toc437795568)

[4.1. Autenticación 3](#_Toc437795569)

[4.2. Compartir la hoja de datos con el nuevo usuario 12](#_Toc437795570)

[4.3. SpreadsheetId 14](#_Toc437795571)

[4.4. Nombre de la pestaña 16](#_Toc437795572)

[4.5. Rango de datos mostrado 17](#_Toc437795573)

[4.6. Anchura máxima para las columnas 17](#_Toc437795574)

[4.7. Fila de títulos 17](#_Toc437795575)

[5. Filtros 18](#_Toc437795576)

[5.1. La expresión del filtro 18](#_Toc437795577)

[5.2. Cadenas de texto 18](#_Toc437795578)

[5.3. Números 18](#_Toc437795579)

[5.4. Fechas 19](#_Toc437795580)

[5.5. Operadores lógicos 19](#_Toc437795581)

[5.5.1. AND 19](#_Toc437795582)

[5.5.2. OR 19](#_Toc437795583)

[5.5.3. NOT 19](#_Toc437795584)

[5.6. Palabras reservadas 20](#_Toc437795585)

[5.7. Condiciones simples 20](#_Toc437795586)

[5.8. Expresiones regulares 22](#_Toc437795587)

[5.9. Ejemplos 22](#_Toc437795588)

[6. Modo depuración 24](#_Toc437795589)

[7. Agradecimientos 24](#_Toc437795590)

[8. Enlaces 24](#_Toc437795591)

# Autor

La aplicación ha sido programada en Java por Francisco Javier Rojas durante y terminada la presente versión en diciembre de 2015.

Correo electrónico para contactar: frojasg1@hotmail.com

# Introducción

La aplicación permite cargar hojas (sheets) conectándose al entorno de Google y mostrarlas y filtrarlas en local.

De momento sólo se permite trabajar con una configuración (con una hoja de datos).

En esta versión los filtros se introducen en forma de expresión.

Cada configuración (se supone que pertenecientes a hojas diferentes), permite almacenar un número ilimitado de filtros, para evitar tener que escribir el filtro cada vez que se necesite cambiarlo.

# Inicio rápido

Para comenzar a utilizar la aplicación, sigue los siguientes pasos:

1. Bájate la aplicación.
2. Descomprímela en cualquier directorio.
3. En el directorio donde la has descomprimido, busca el subdirectorio \_binary
4. Haz doble click en el fichero: **Filtering\_google\_sheets.jar**
5. Configura los siguientes parámetros:
   1. accountId
   2. fichero p12 para la autenticación
   3. spreadsheetId
   4. nombre de la pestaña que contiene la información
   5. fila de títulos
   6. Expresión para el filtro
6. Haz click en el botón: Recargar
7. Si todo ha ido bien, ahora ya deberías tener los datos filtrados.

Si necesitas información más detallada sobre cómo configurar la información necesaria para conectar con la hoja de datos, lee los siguientes apartados.

# Parámetros de configuración

Antes de empezar a trabajar con una hoja de datos de Google, hay que configurar los parámetros para la autenticación, identificación de la hoja de datos y configuración del comportamiento de la aplicación.

## Autenticación

Es necesario configurar dos parámetros para la autenticación automática al entorno de Google:

1. El accountId.
2. La ubicación del archivo que contiene la clave privada para la conexión.

En los dos siguientes puntos veremos cómo generar un AccountId y una vez hecho eso, cómo bajar el fichero p12 necesario para autenticarse con ese AccountId.

La primera cosa necesaria para conectarse a Google es el accountId, que es una especie de usuario generado especialmente para el acceso con unos permisos muy concretos.

Para generar un nuevo accountId:

1. hay que conectarse a <https://console.developers.google.com/>

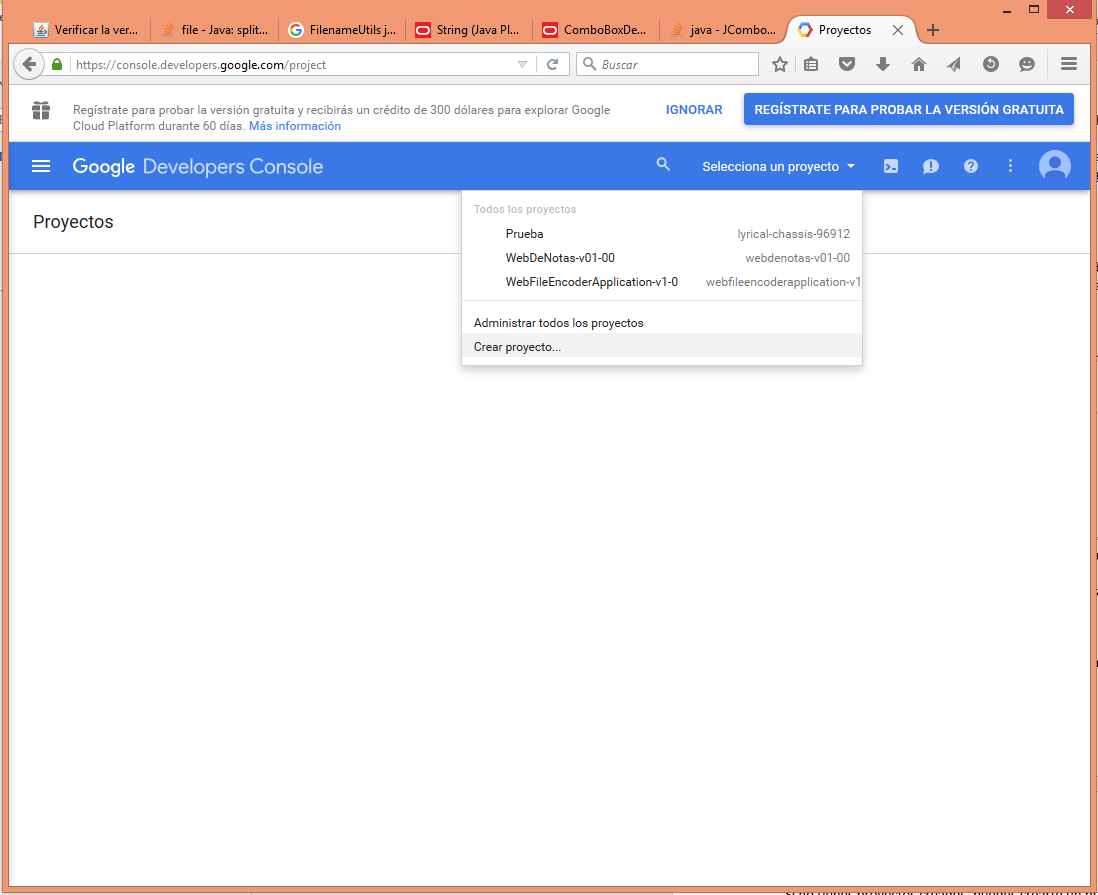
Y entrar con tu usuario habitual de Google.

Tras ello, puedes generar un nuevo accountId en cualquier proyecto que tengas creado.

1. Si no tienes proyectos creados, puedes crearte un proyecto:

En la opción:

* Selecciona un proyecto -> Crear proyecto …

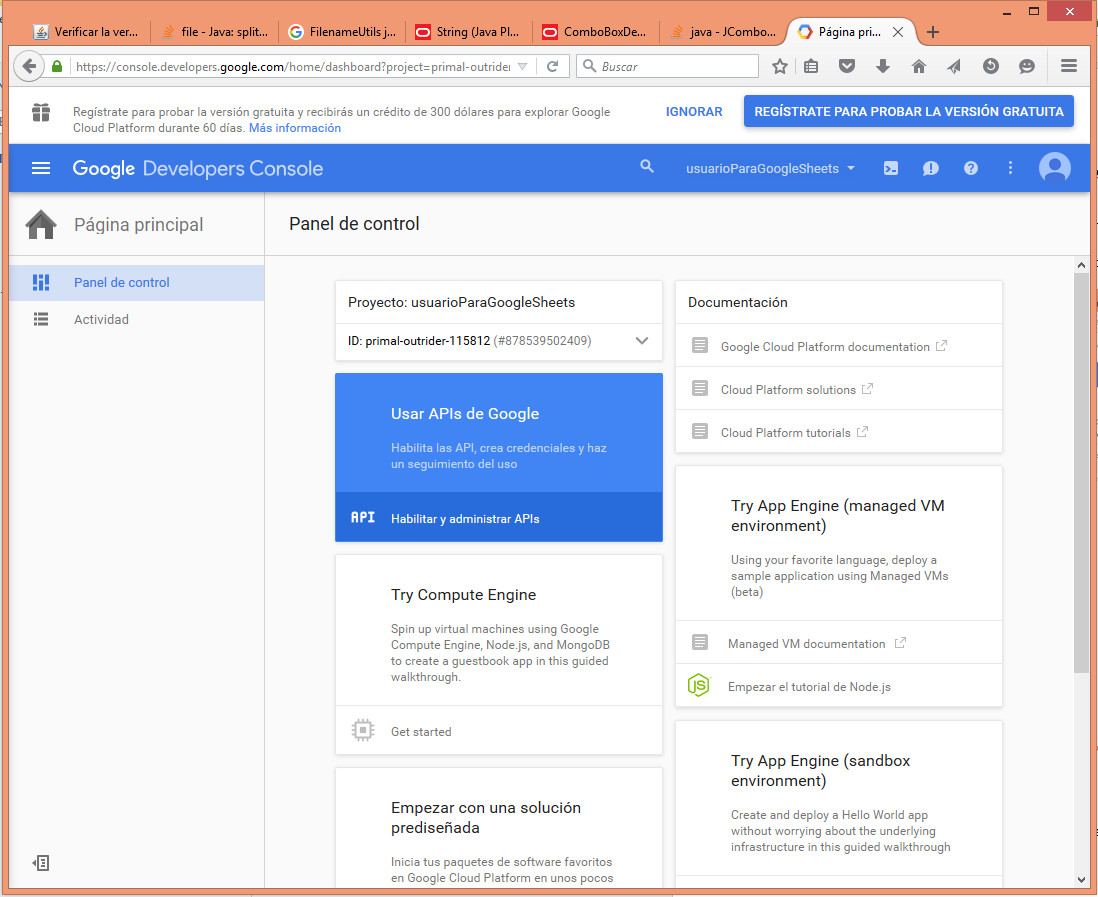


Yo Acabo de crear un proyecto llamado:

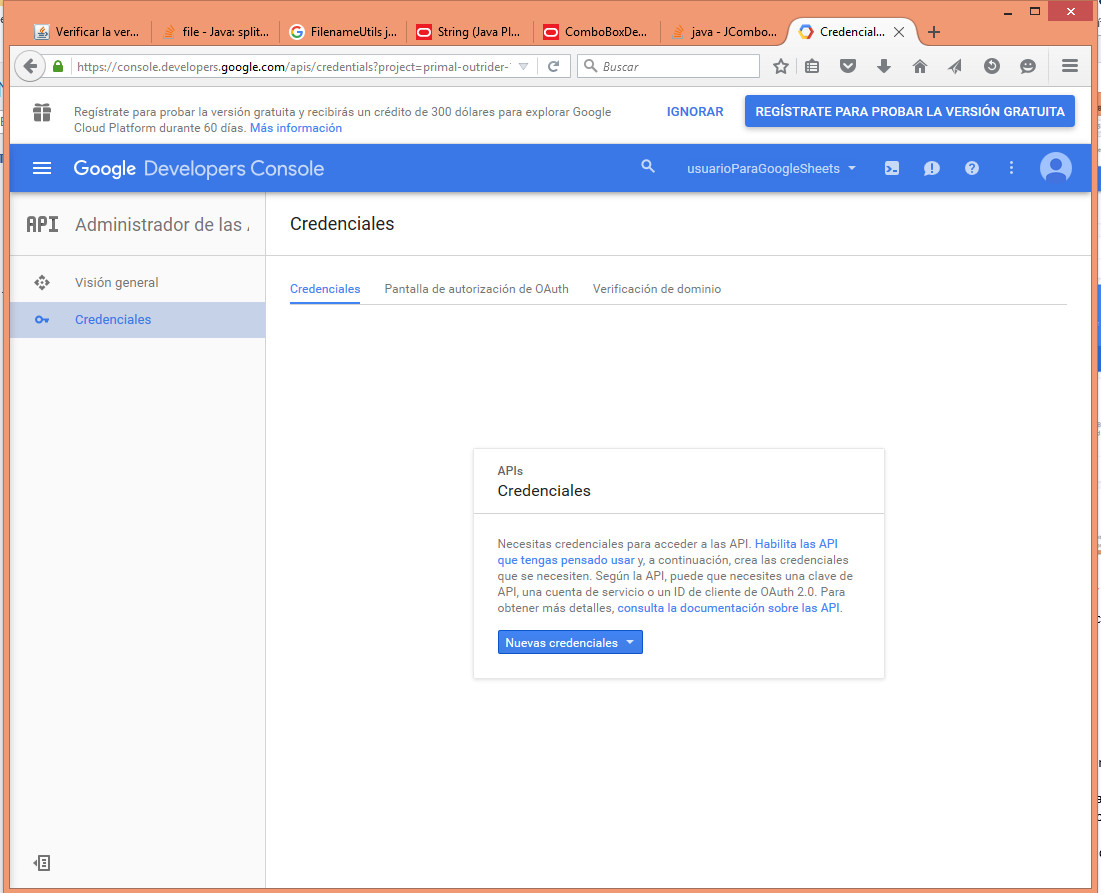
**usuarioParaGoogleSheets**

1. Seleccionas el proyecto en el que quieras crear el accountId (el que acabas de crear u otro ya existente) y Clickas en Página principal (en el menú de la izquierda).

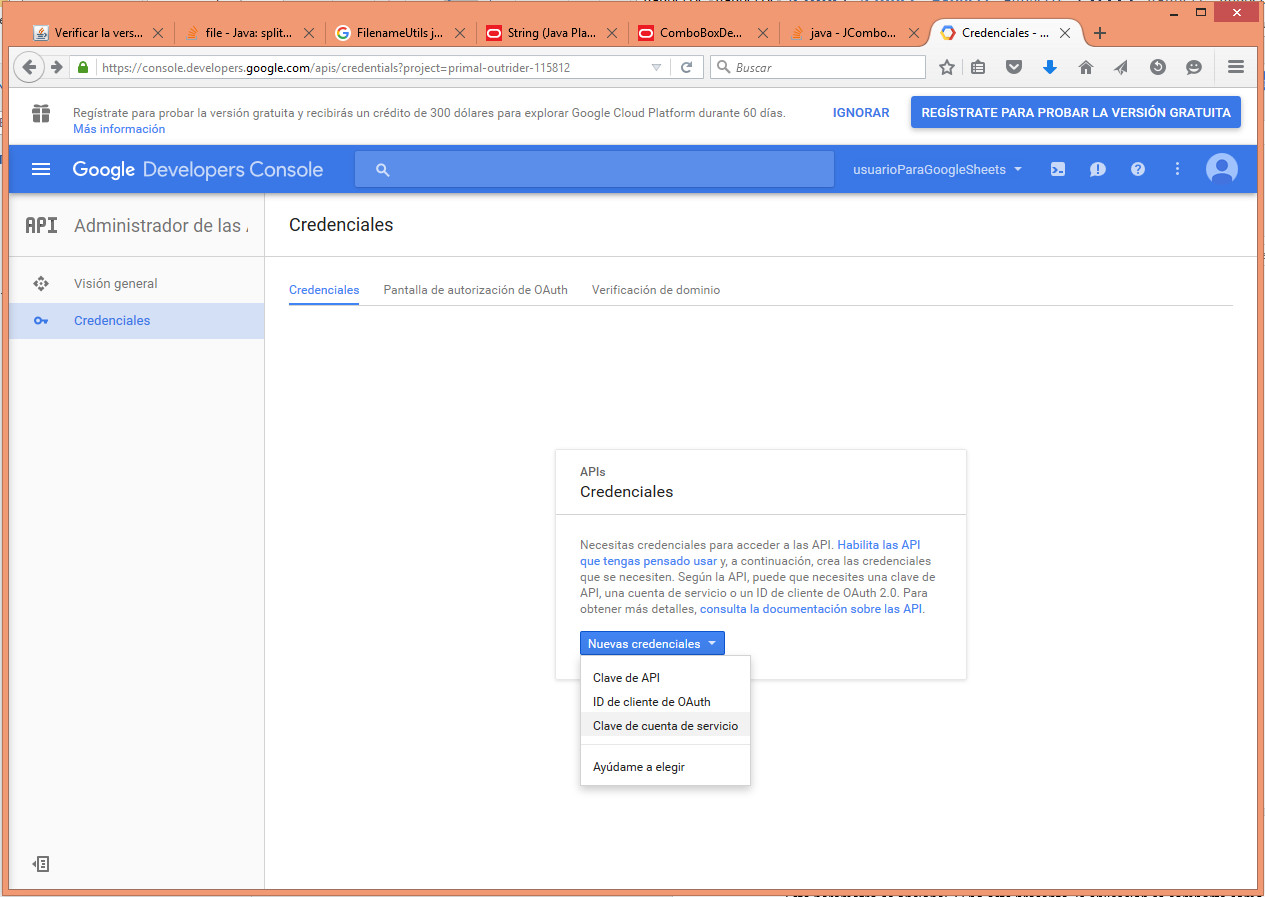
Te aparecerá una pantalla parecida a esta:



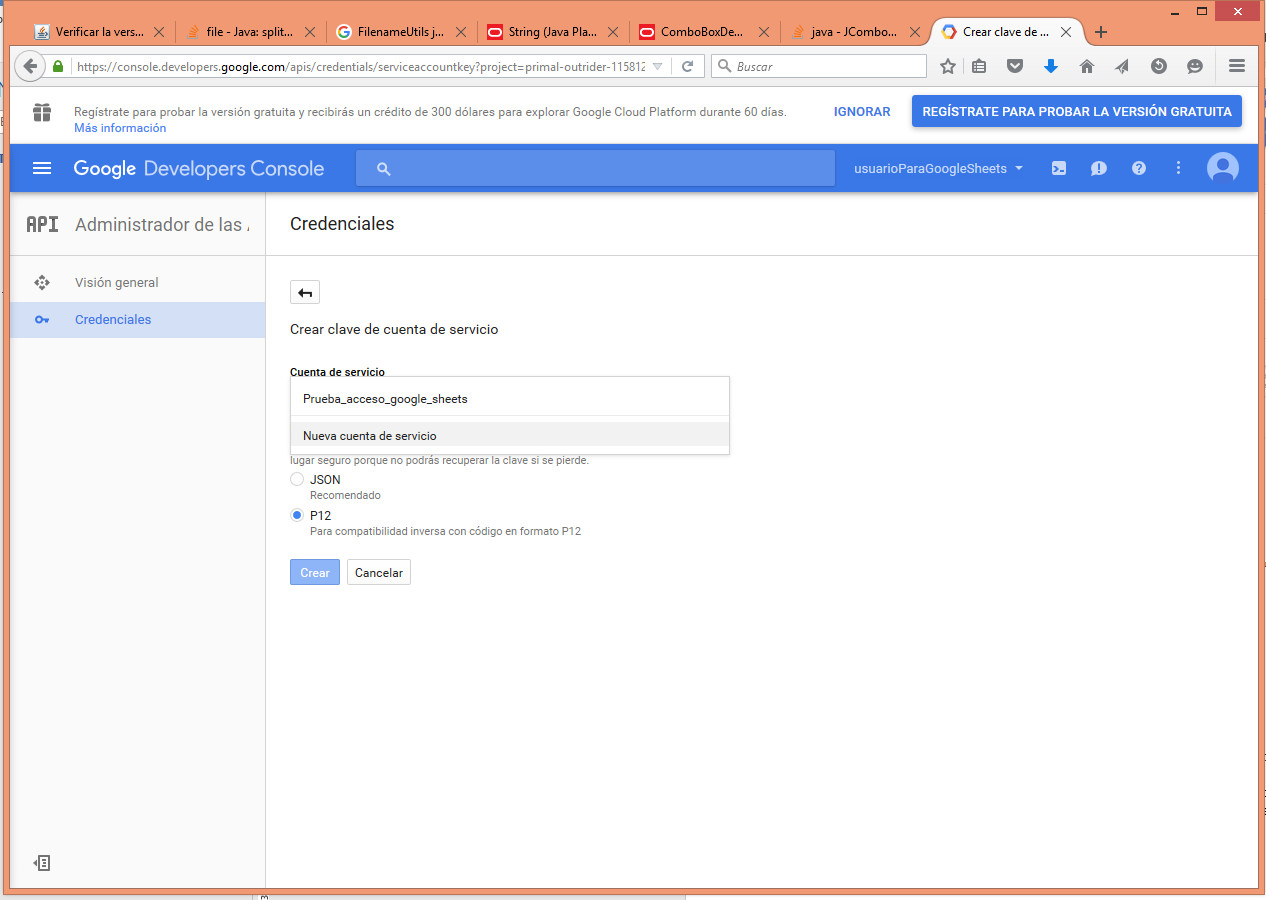
1. En la parte de abajo del recuadro azul, clicka en Habilitar y administrar APIs.



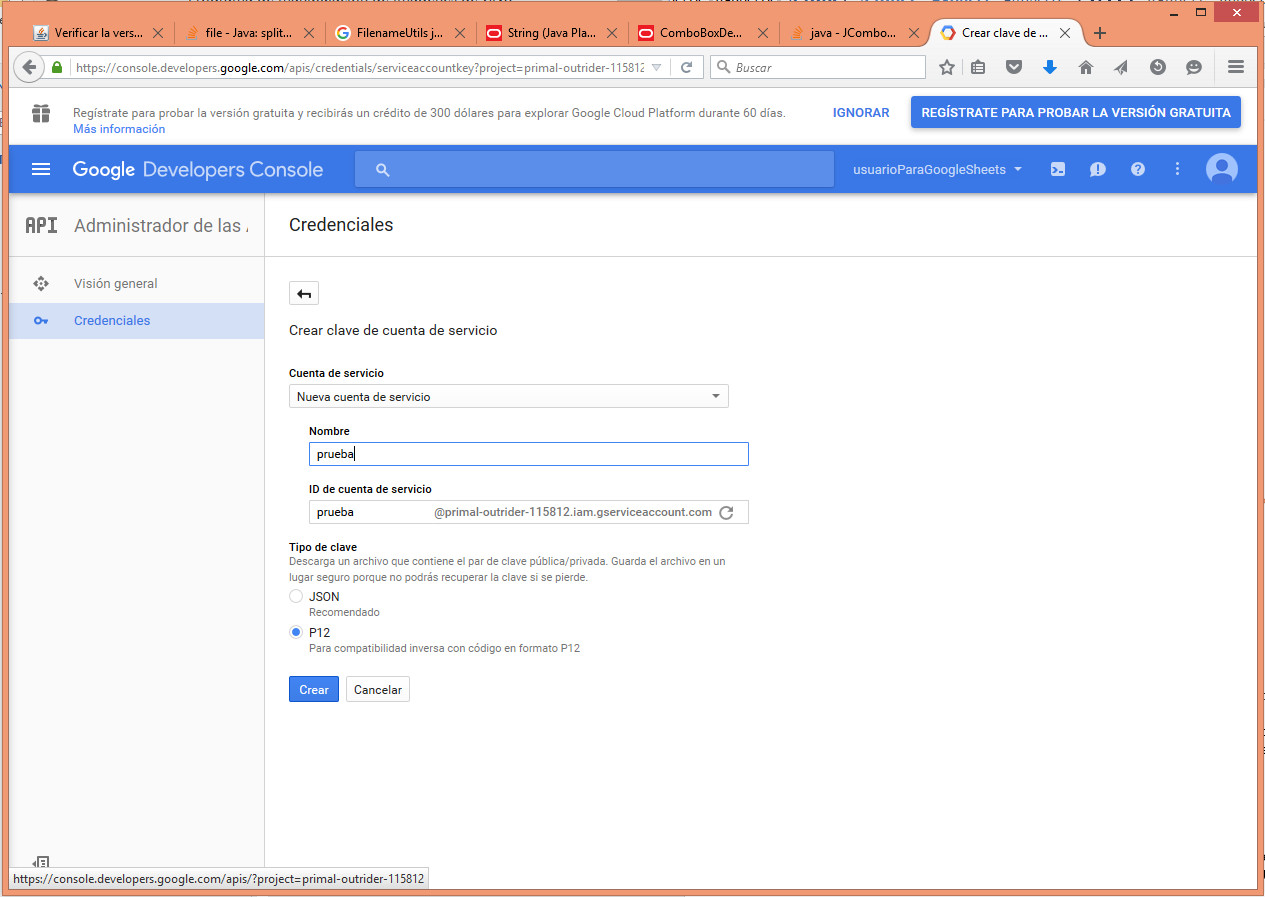
1. En el menú de la izquierda, entra en Credenciales:



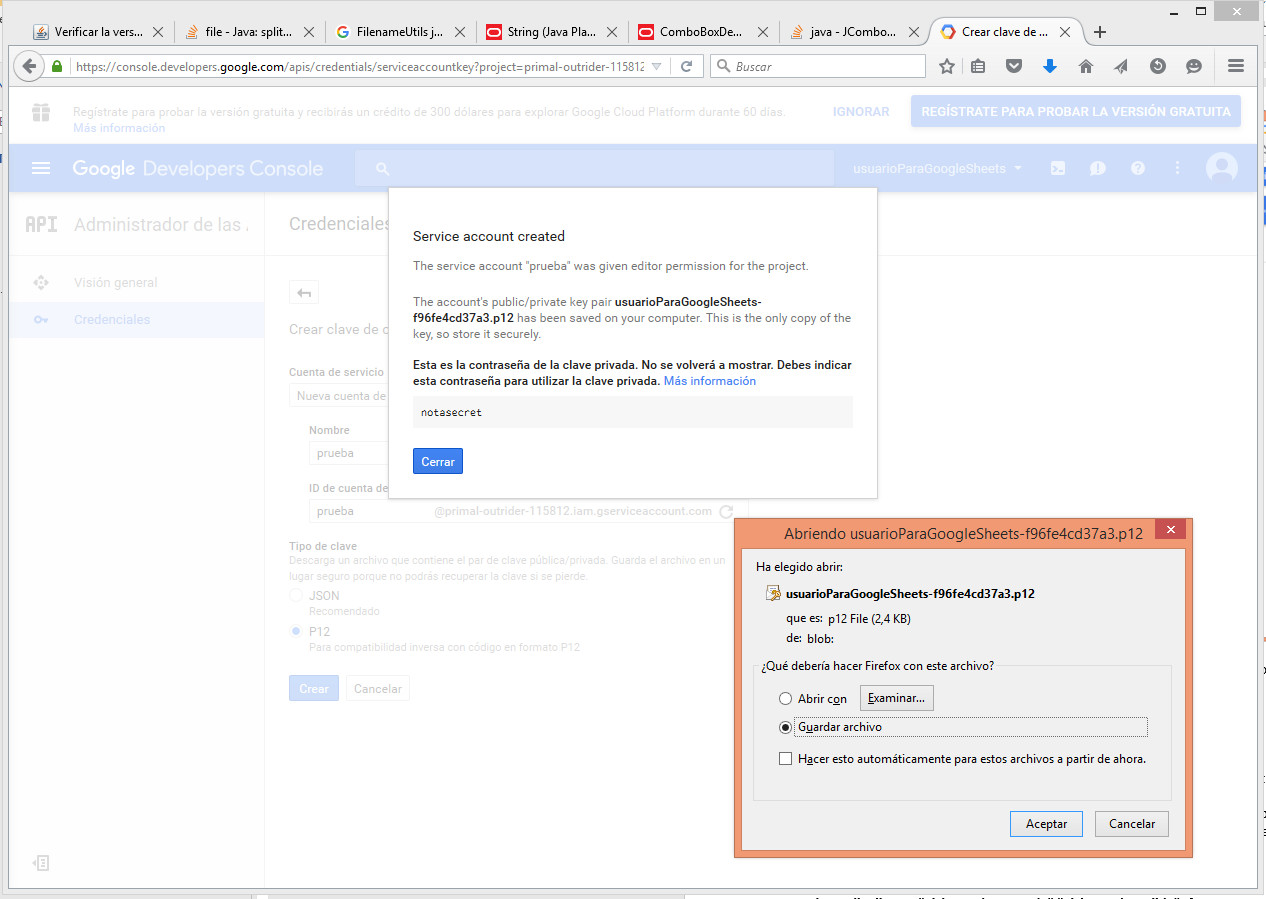
1. Entra en “Nuevas credenciales”->”Crear cuenta de servicio”



1. Seleccionar: P12
2. Entrar en “Cuenta de servicio” -> “Nueva cuenta de servicio”

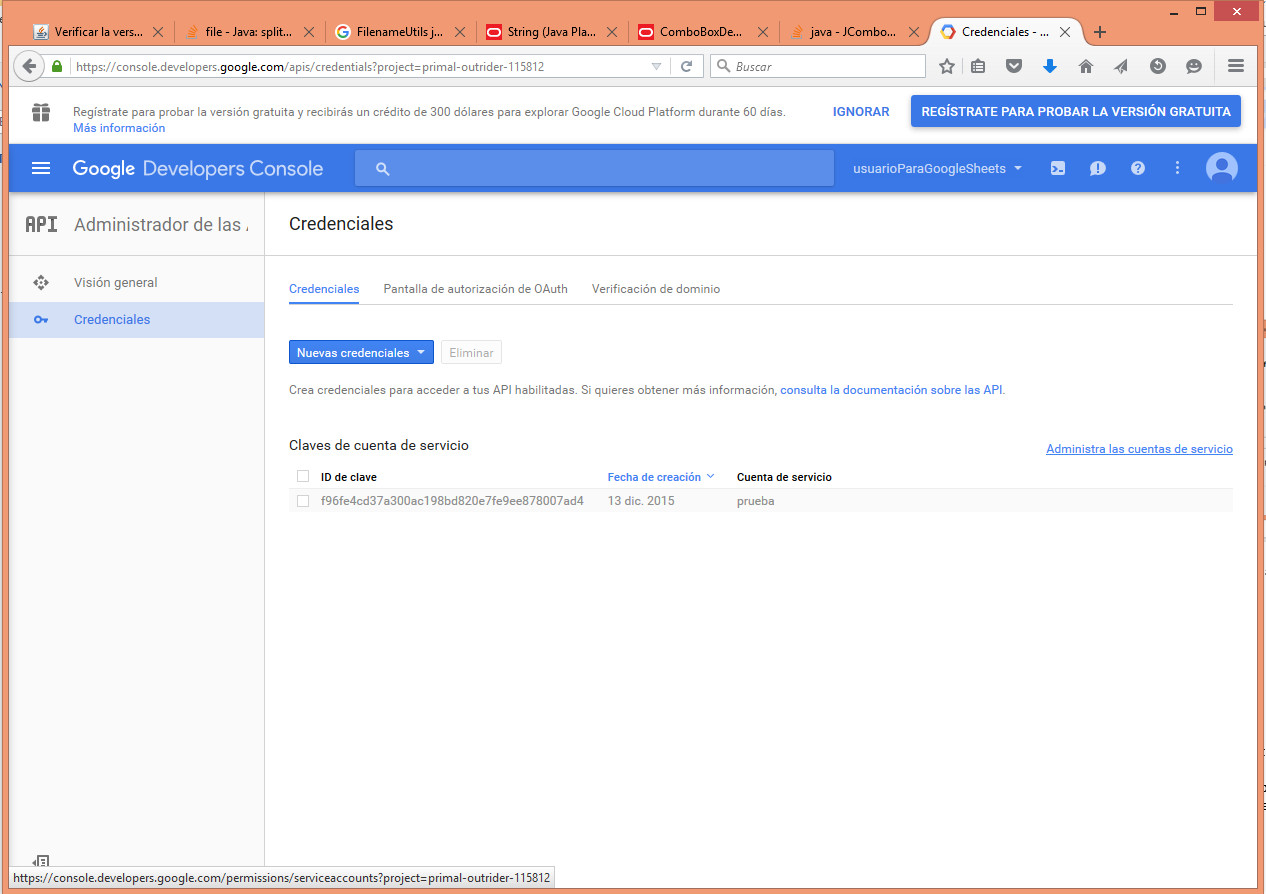


1. Ahora permite poner el nombre de la nueva cuenta. Yo le he puesto como nombre **prueba**.
2. Clickar en “Crear”



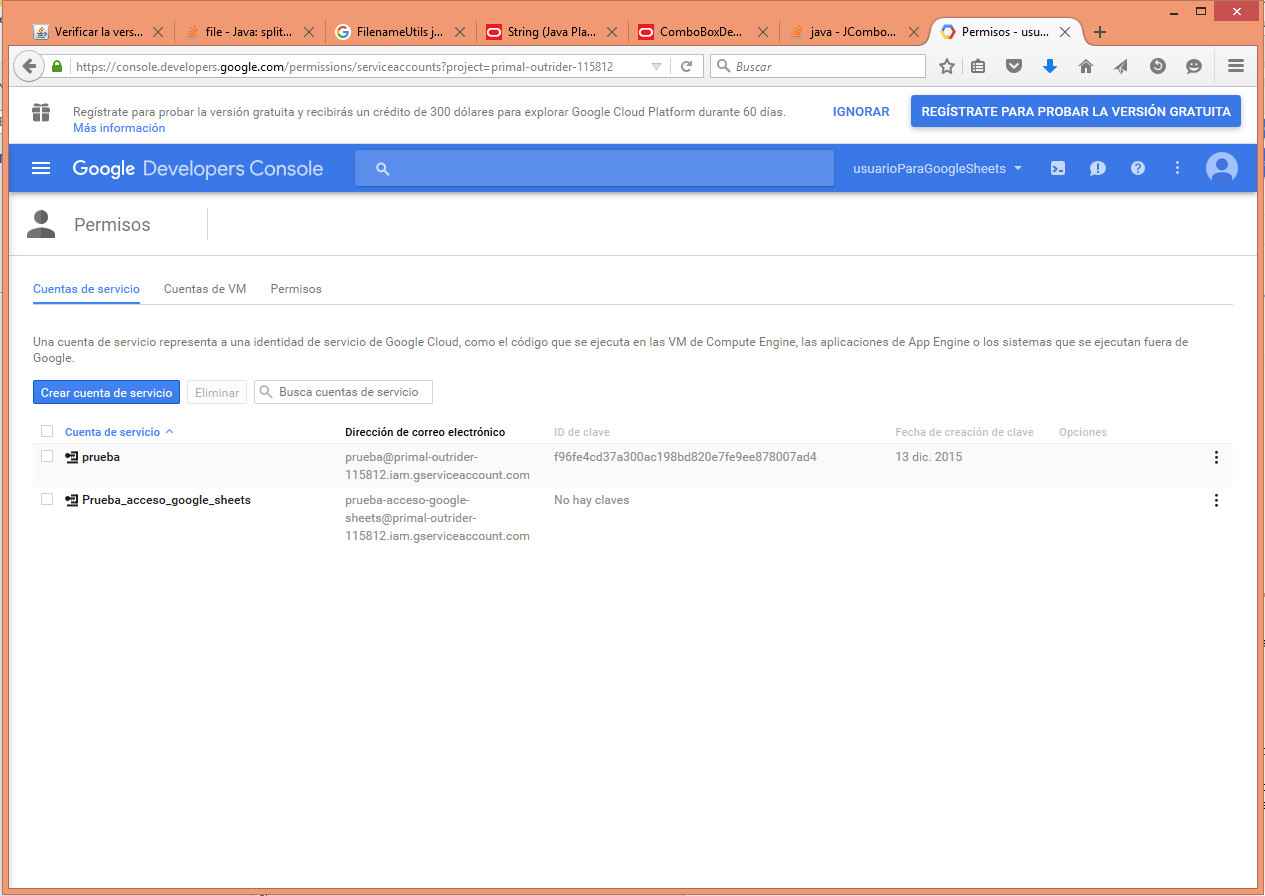
1. Guarda el archivo P12 que se descarga, lo necesitarás para configurar la autenticación en la aplicación.
2. Después clicka en Cerrar.

Si en un futuro crees que alguien no autorizado ha podido acceder a tu fichero P12, podrás eliminar ese fichero P12 en google, y generar uno nuevo.



1. Después clicka en Administra las cuentas de servicio (a la derecha de la pantalla).

Aparecerá la pantalla con la información de los accountIds:



1. De esta pantalla, se puede copiar la dirección de correo electrónico necesaria para la configuración del parámetro AccountId de la aplicación.

Además, hay que utilizar esa dirección de correo electrónico para compartirle la hoja de datos a la que quieres acceder remotamente. Si no compartes la hoja (spreadsheet) con esta dirección de correo, no podrás acceder a los datos.

Asegúrate de que la dirección de correo que coges es la asociada al mismo accountId que tiene el fichero P12 que has generado antes, si no, no funcionará la autenticación.

En mi caso, la dirección de correo electrónico que debo utilizar sería esta:

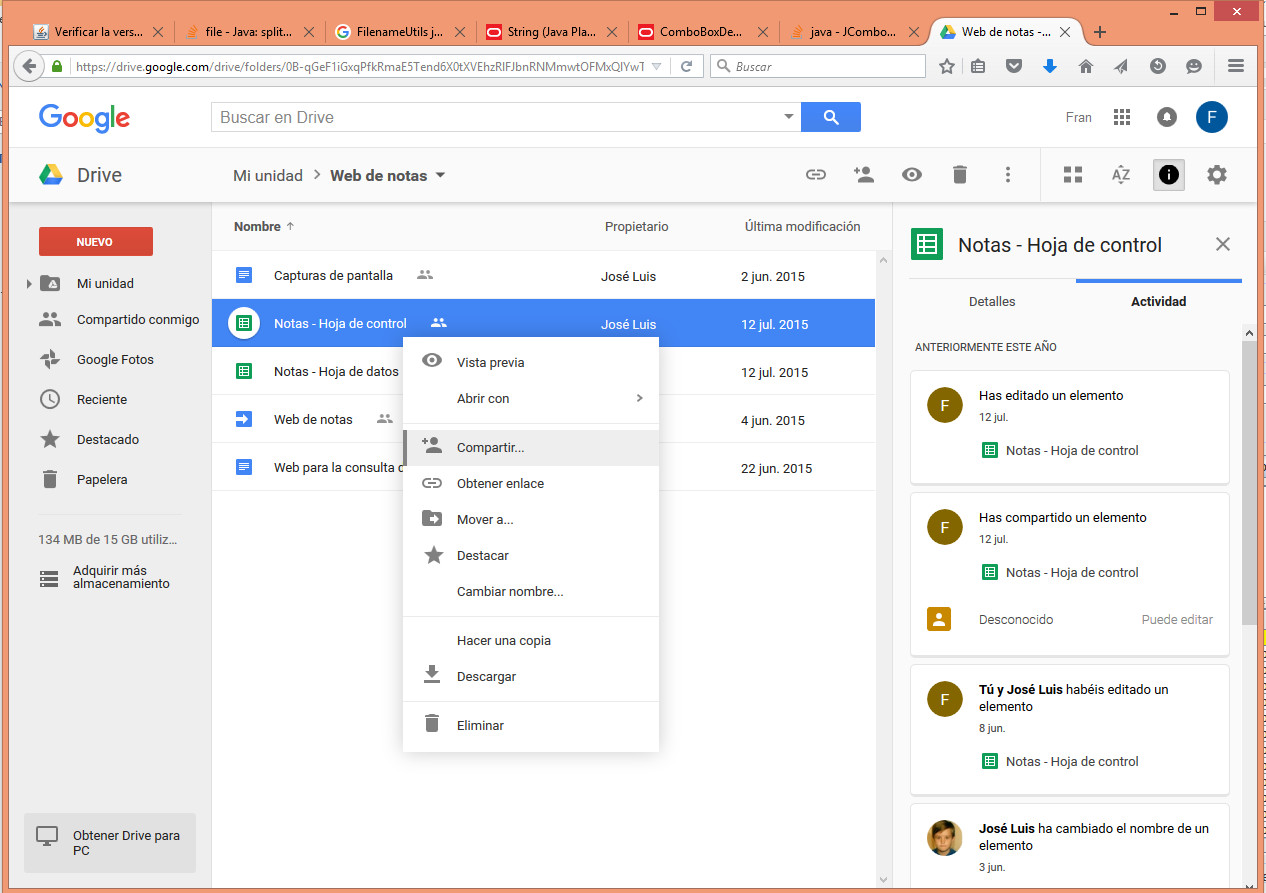
**prueba@primal-outrider-115812.iam.gserviceaccount.com**

## Compartir la hoja de datos con el nuevo usuario

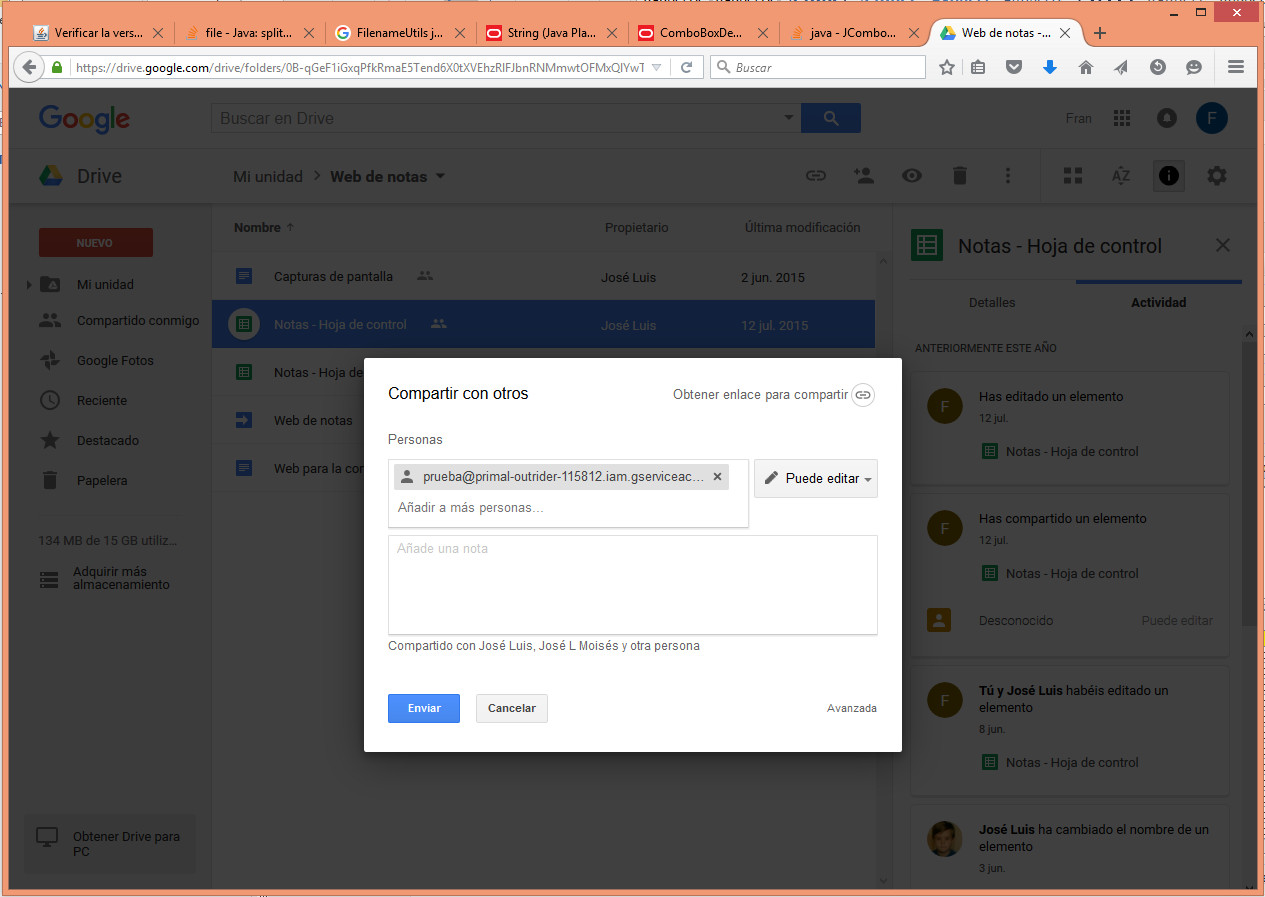
Para poder acceder a la hoja de google que deseas filtrar en local, necesitas compartir esa hoja con el accountId generado en el punto anterior.

Para ello sigue los siguientes pasos:

1. Entrar en drive.google.com y buscar la hoja que deseas compartir.



1. Clickando botón derecho sobre la hoja que deseas compartir, selecciona “Compartir”.



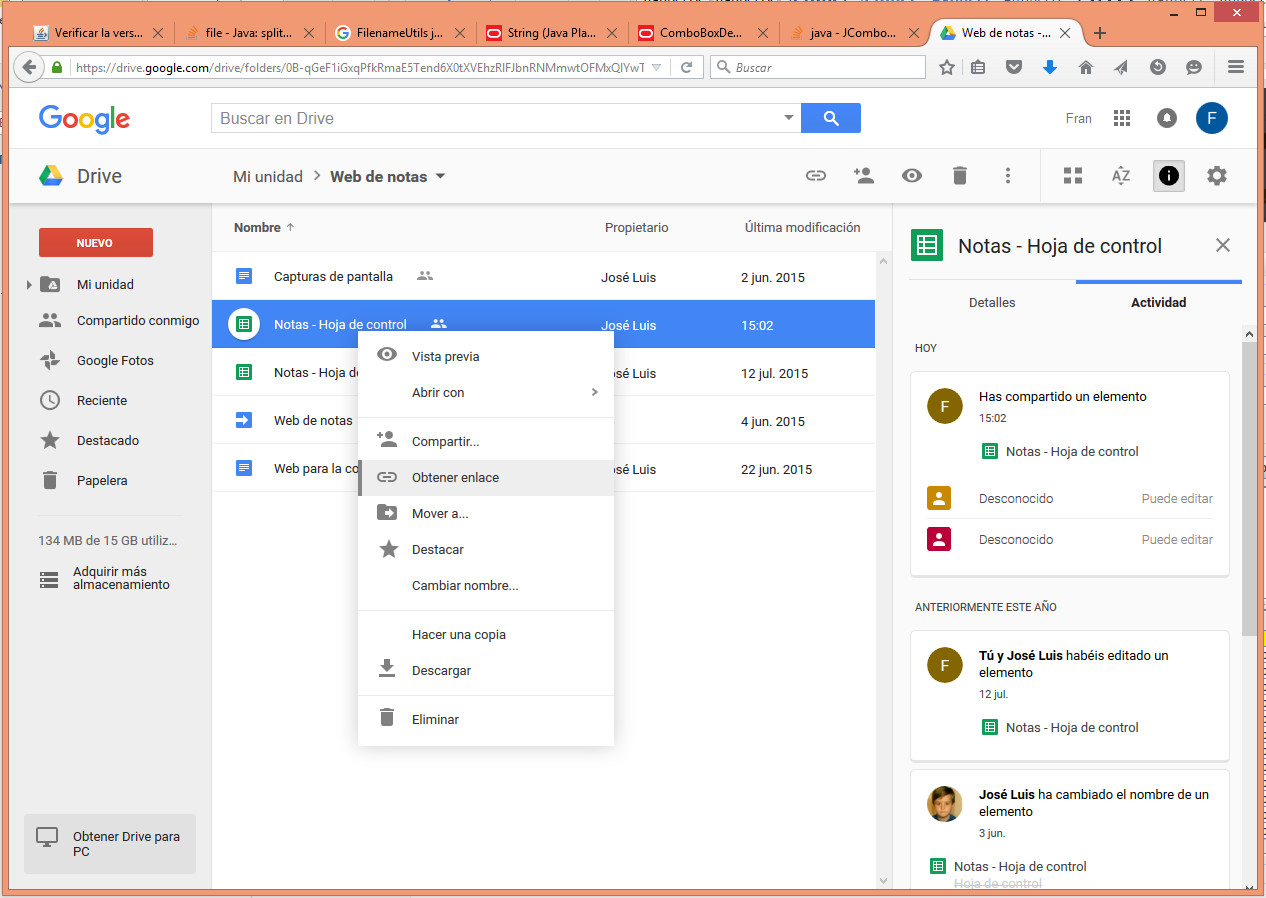
1. Poner la dirección de correo electrónico obtenida en el apartado anterior.

## SpreadsheetId

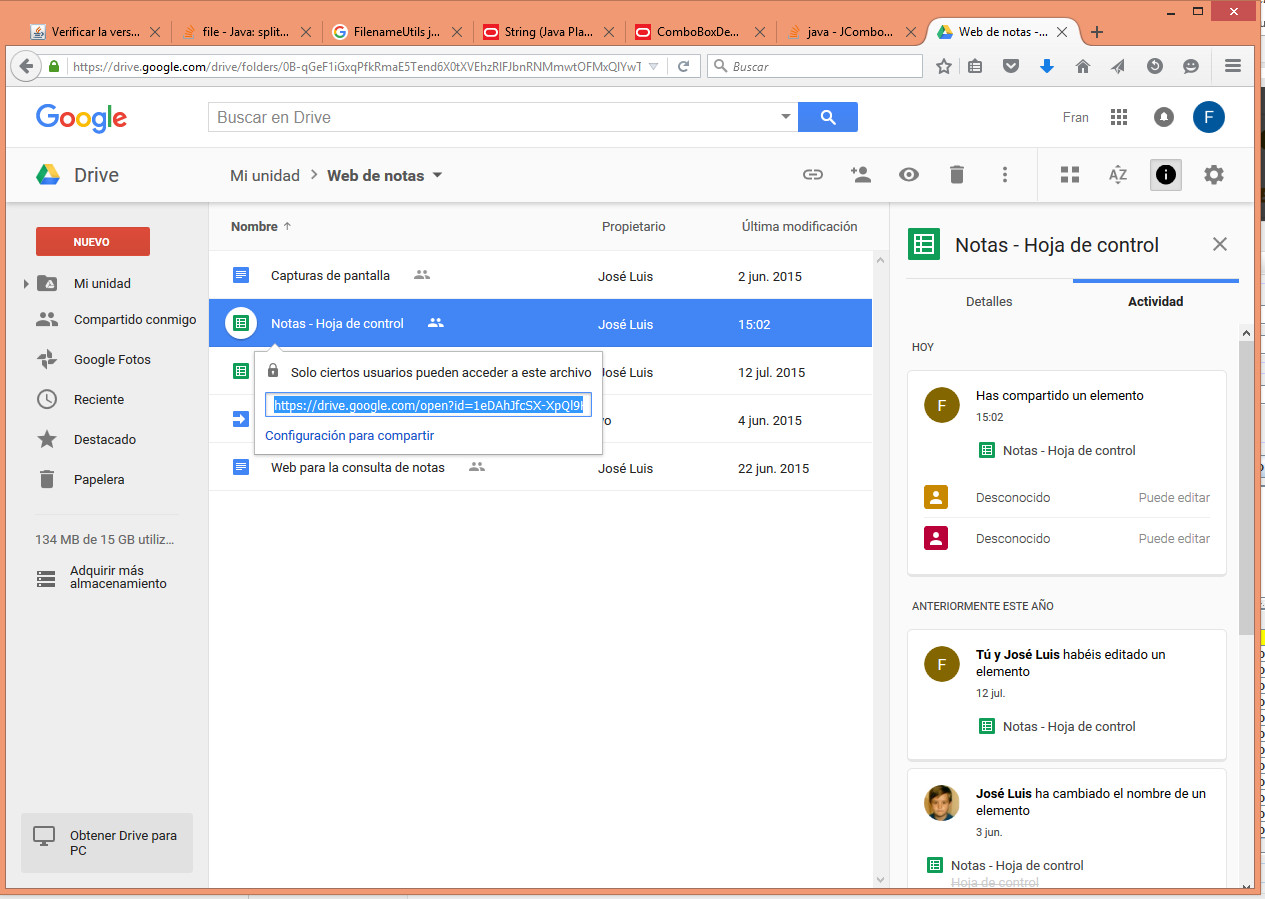
El spreadsheetId es el identificador del fichero que contiene la hoja de google.

Se puede obtener así:

1. Entrar en drive.google.com y buscar la hoja que deseas compartir.



1. Clickando botón derecho sobre la hoja que deseas compartir, selecciona “Obtener enlace”.



1. Copia el enlace con CTRL+C y cópialo en un editor de texto.



1. El SpreadsheetId es el valor del parámetro id de la URL. Está seleccionado (en gris) para poder distinguirlo del resto de la URL.
2. La parte seleccionada, hay que copiarla en el parámetro de configuración SpreadsheetId de la aplicación.

## Nombre de la pestaña

El último parámetro de configuración necesario para identificar los datos que deseamos filtrar es el nombre de la pestaña dentro de la hoja de google.

## Rango de datos mostrado

Se puede configurar el rango de datos mostrado en la tabla.

Hay dos opciones:

* Autodetectar. Si se selecciona esta opción, una vez que se carguen los datos, se calculará el rango mínimo que incluya todas las celdas no vacías.
* Forzar rango. Si se selecciona esta opción, deberá indicarse el rango en el edit de su derecha.

El formato del rango se indica de la manera tomada de Excel de Microsoft.

El formato sería así:

ID\_COLUMNA NUMERO\_DE\_FILA ID\_COLUMNA NUMERO\_DE\_FILA

Por ejemplo, para seleccionar el rango desde A1 hasta D3, la cadena de rango resultante sería:

A1D3

Si la cadena no tiene un formato correcto el filtrado fallará.

## Anchura máxima para las columnas

La anchura de las columnas de la tabla es calculada automáticamente cada vez que se filtra.

Si el parámetro “Anchura máxima para las columnas” no está configurado, la anchura automática calculada será la suficiente para que se muestren todas las cadenas completas de cada columna.

Esto tiene el inconveniente de que es posible que haya cadenas muy largas en alguna columna, siendo la anchura muy grande.

Se puede limitar el tamaño de anchura máxima con este parámetro.

## Fila de títulos

Este parámetro de configuración indica a la aplicación qué fila tiene que ser tomada para obtener el título de la columna.

# Filtros

Se puede tener un número ilimitado de filtros, cada uno identificado por un nombre.

Los nombres de los filtros están en el combobox “Nombre del filtro”.

Cuando se selecciona un filtro, la expresión del filtro se coloca en el panel de texto que hay debajo.

Si modificas un filtro y deseas guardar los cambios para el futuro, deberás pulsar el botón “Guardar filtro”.

Esto también es válido para filtros nuevos, en cuyo caso, deberás escribir el nombre de un nuevo filtro y pulsar “Guardar filtro”.

Si deseas eliminar un filtro, selecciónalo en el combobox y pulsa el botón “Eliminar filtro”.

## La expresión del filtro

La expresión de un filtro puede estar compuesta por operadores lógicos, palabras reservadas, cadenas de texto, números y paréntesis.

Las expresiones incluyen condiciones simples que pueden ser combinadas mediante los operadores lógicos.

Los operadores lógicos y las palabras reservadas no distinguen entre mayúsculas y minúsculas. Las cadenas de texto (encerradas entre comillas dobles), sí que distinguen entre mayúsculas y minúsculas.

Un filtro con expresión vacía, no realiza ningún filtrado y puede ser útil para mostrar la información de la hoja de datos sin filtrar.

## Cadenas de texto

Las cadenas de texto pueden contener cadenas o fechas.

Van encerradas entre comillas dobles.

Para incluir una comilla doble dentro de una cadena de texto, habrá que poner dos comillas dobles seguidas.

Así, por ejemplo, una cadena que sea igual a una comilla doble, se representaría así:

“””” Cuatro comillas dobles, las comillas dobles de los extremos como delimitadores de la cadena, y las dos comillas dobles internas, representarían a la comilla doble representada por esta cadena de texto.

## Números

Los números de las expresiones son representados en formato anglosajón, es decir, con el “.” como separador decimal.

Es posible que los números tengan formato del país (español) en la hoja de datos, y ponerlo en formato anglosajón en la expresión. La condición debería evaluar bien de todas formas.

De momento el país no se puede seleccionar y está puesto como España directamente en el código.

Si una condición numérica aplica sobre una celta de texto que no contiene un número, el número resultante para realizar al condición es 0.

## Fechas

Las fechas se convierten a timestamp por eso las condiciones con fechas pueden tener sus peculiaridades.

Por ejemplo si hay una fecha con tiempo (“13/12/2015 17:25”), la condición es un “=” y el parámetro a comparar es ( “13/12/2015” ) ,aunque el día de la fecha sea igual, la comparación dará como resultado Falso.

## Operadores lógicos

Los operadores lógicos implementados son los típicos de los filtros:

* AND
* OR
* NOT

El orden de precedencia entre ellos es el habitual:

AND tiene mayor precedencia que OR

El operador unario NOT no se incluye en el orden de precedencia ya que debe ir seguido obligatoriamente por un paréntesis que incluya la expresión negada.

## AND

Es un operador lógico binario.

El formato es este:

Condicion1 AND Condicion2

El resultado evalúa como cierto si y sólo si ambas Condiciones son ciertas.

Las condiciones pueden ser condiciones simples o expresiones completas. Pueden estar encerradas entre paréntesis y dentro del paréntesis incluir cualquier tipo de expresión.

Si no están encerradas entre paréntesis, aplicará el orden de precedencia habitual.

## OR

Es un operador lógico binario.

El formato es este:

Condicion1 OR Condicion2

El resultado evalúa como falso si y sólo si ambas Condiciones son falsas.

Las condiciones pueden ser condiciones simples o expresiones completas. Pueden estar encerradas entre paréntesis y dentro del paréntesis incluir cualquier tipo de expresión.

Si no están encerradas entre paréntesis, aplicará el orden de precedencia habitual.

## NOT

Es un operador lógico unario.

El formato es este:

NOT( Condición )

El resultado evalúa como falso si la Condición era cierta y como cierto si la Condición era falsa.

Las condiciones pueden ser condiciones simples o expresiones completas.

## Palabras reservadas

Hay varias palabras reservadas que tienen un significado especial:

* CONDITION
* COLUMN
* String
* Number
* Date

## Condiciones simples

Las condiciones simples deben incluir un identificador de columna, sobre la que se realizará la condición, un tipo de dato para la condición (que no tiene por qué coincidir con el tipo de dato de la columna), un operador para la condición ( por ejemplo “=” ) y un parámetro para la condición.

El formato para una condición simple es este:

**CONDITION(** \_\_COLUMNA\_\_, \_\_TIPO\_DATO\_\_, \_\_OPERADOR\_\_, \_\_PARAMETRO\_\_ **)**

Donde los caracteres en negrita son textuales, y los parámetros internos tienen su propio formato.

Vayamos con cada uno de ellos:

* \_\_COLUMNA\_\_

Identifica a una columna.

El formato es uno de estos dos:

* **COLUMN( “**\_\_TITULO\_\_**” )**

Donde \_\_TITULO\_\_ es una cadena de texto (entre comillas), y tiene que ser igual al título de una de las columnas.

* **COLUMN(** \_\_ID\_COLUMNA\_\_ **)**

Donde \_\_ID\_COLUMNA\_\_ es un identificador (sin comillas) y debe ser un identificador válido de columna.

Por ejemplo, el identificador para la primera columna es **A**

* \_\_TIPO\_DATO\_\_

Identifica a un tipo de dato.

Las posibilidades son estas:

* **String**

Indica que los valores para la condición serán tratados como cadenas de texto.

* **Number**

Indica que los valores para la condición serán tratados como números.

* **Date**

Indica que los valores para la condición serán tratados como fechas.

* \_\_OPERADOR\_\_

Identifica al operador de la condición.

Se representan con una cadena de texto, por lo que deben ir delimitados por comillas dobles.

Operadores que son válidos para todos los tipos de datos:

* **“=”**

Operador de igualdad. Evalúa a cierto si el dato de la columna es igual que el dato del \_\_PARAMETRO\_\_ (último parámetro de la condición simple).

* **“>”**

Operador de igualdad. Evalúa a cierto si el dato de la columna es mayorque el dato del \_\_PARAMETRO\_\_ (último parámetro de la condición simple).

* **“<”**

Operador de igualdad. Evalúa a cierto si el dato de la columna es menor que el dato del \_\_PARAMETRO\_\_ (último parámetro de la condición simple).

* **“>=”**

Operador de igualdad. Evalúa a cierto si el dato de la columna es mayor o igual que el dato del \_\_PARAMETRO\_\_ (último parámetro de la condición simple).

* **“<=”**

Operador de igualdad. Evalúa a cierto si el dato de la columna es menor o igual que el dato del \_\_PARAMETRO\_\_ (último parámetro de la condición simple).

Operador válido únicamente para cadenas:

* **“~”**

Operador de expresión regular. Evalúa a cierto si el dato de la columna cumple la expresión regular del dato del \_\_PARAMETRO\_\_ (último parámetro de la condición simple).

* \_\_PARAMETRO\_\_

Identifica al parámetro de la derecha de la condición.

Cumple con uno de los siguientes formatos:

* \_\_COLUMNA\_\_

Se puede especificar como parámetro otra columna, en cuyo caso la condición se realizará sobre el valor de la columna del parámetro 1 de la condición, y el valor de la columna especificada aquí.

El formato en este caso es idéntico al explicado en el formato de \_\_COLUMNA\_\_ (como el del primer parámetro de la condición simple).

* “Cadena de texto”
* Con \_\_TIPO\_DATO\_\_ fuera String (cualquier cadena, para los \_\_OPERADOR\_\_ normal, o expresión regular para el \_\_OPERADOR\_\_ “~”)
* Con \_\_TIPO\_DATO\_\_ fuera Date (cualquier cadena que represente una fecha válida)
* Número

Para \_\_TIPO\_DATO\_\_ Number.

## Expresiones regulares

Las expresiones regulares son una herramienta muy potente, por lo que estar familiarizado con ellas puede ayudar bastante a construir condiciones.

Para que una expresión regular evalúe como cierta, debe coincidir en toda la cadena.

Por ejemplo:

* “Valor” “~” “V”

No evaluaría como cierto, porque V es sólo una parte de la cadena de la izquierda.

* “Valor” “~” “V.\*”

Sí evaluaría como cierto, porque .\* indica cualquier cadena, y esa Expresión Regular sí que coincidiría con la totalidad de la cadena de la izquierda.

La versión de expresiones regulares que utiliza el operador “~”, es la que ofrece Java.

En la siguiente URL podéis encontrar una guía completa de las expresiones regulares aceptadas:

<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>

## Ejemplos

* Ejemplo 1: Filtra las filas que cumplan la condición de que el dato de la columna con título “NOMBRE” comienza por “Fer”.

CONDITION( COLUMN( “NOMBRE” ), String, “~”, “^Fer.\*$” )

* Ejemplo 2: Filtra las filas que no cumplan la condición de que el dato de la columna con título “NOMBRE” comienza por “Fer”.

NOT( CONDITION( COLUMN( “NOMBRE” ), String, “~”, “^Fer.\*$” ) )

* Ejemplo 3: Filtra las filas que cumplan la condición de que el dato de la columna con título “NOMBRE” tiene como valor “Fernando” y la columna “APELLIDO” termina en vocal.

CONDITION( COLUMN( “NOMBRE” ), String, “=”, “Fernando” ) AND

CONDITION( COLUMN( “APELLIDO” ), String, “~”, “^.\*[aeiou]$” )

* Ejemplo 4: Filtra las filas que cumplan la condición de que el dato de la columna con título “NOMBRE” tiene como valor “Fernando” y la columna “SALDO” sea mayor o igual que 1000,5 o la fecha de la columna A sea menor que 13/12/2015 y el número de la columna B sea mayor que el de la columna C.

CONDITION( COLUMN( “NOMBRE” ), String, “=”, “Fernando” ) AND

CONDITION( COLUMN( “SALDO” ), Number, “>=”, 1000.5 ) OR

CONDITION( COLUMN( A ), Date, “<”, “13/12/2015” ) AND

CONDITION( COLUMN( B ), Number, “>”, COLUMN( C ) )

# Modo depuración

De momento no está implementado el modo de depuración.

En una futura versión se programará una forma más ágil de realizar comprobaciones simples sobre filas concretas de la hoja de datos, para poder detectar posibles errores en el motor de condiciones de la aplicación.

# Agradecimientos

A Google por su dedicación al usuario final con la creación de productos en gran parte gratuitos.

A Google por haber puesto a la disposición de la comunidad de programadores las herramientas para poder interactuar con su infraestructura.

A las páginas dedicadas a resolver dudas de programación, como puede ser por ejemplo stackoverflow. Sin ellas sería mucho más difícil el poder avanzar en un desarrollo.

# Enlaces

http://frojasg1.com