

Curso: Programación web 2

Asunto: Informe de la actividad de la solución de un problema usando google app script.

Alumno: Auccacusi Conde Brayan Carlos

1. Problema

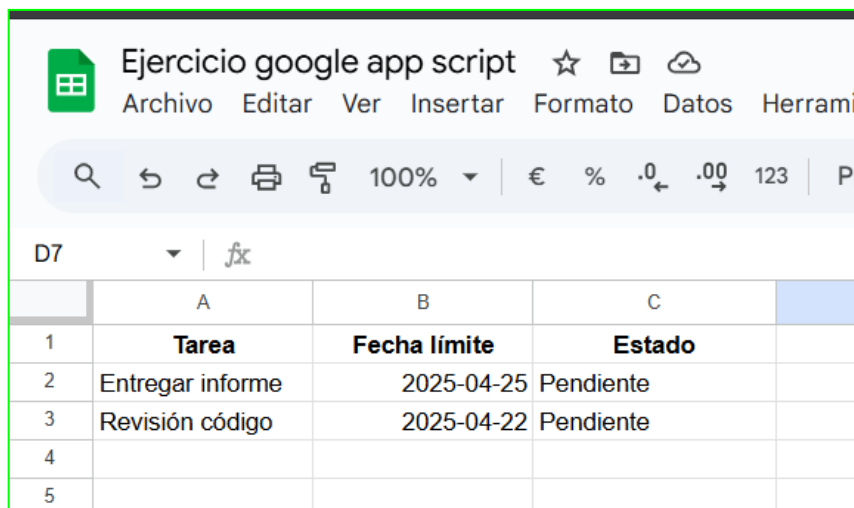
Usar excel o google sheets es una buena herramienta para organizarnos, en este software podemos colocar nuestro horario, una tabla con nuestras actividades pendientes, llevar el control de nuestras actividades realizadas o por realizadas. Sin embargo tiene una desventaja el cual es que debemos estar abriendo nuestra hoja de cálculo constantemente y recordar que actividad creíamos que nos faltaba para ir a comprobarlo y luego ver hasta que fecha tenemos plazo.

2. Solución

Hacer un script usando la plataforma que nos brinda google, *Google app script*, el cual estará sincronizado con nuestra hoja de cálculo, este script nos enviará notificaciones a nuestro correo si hubiera alguna actividad que se pasa de la fecha límite (el cual nosotros podemos establecer como un periodo antes de la fecha límite real).

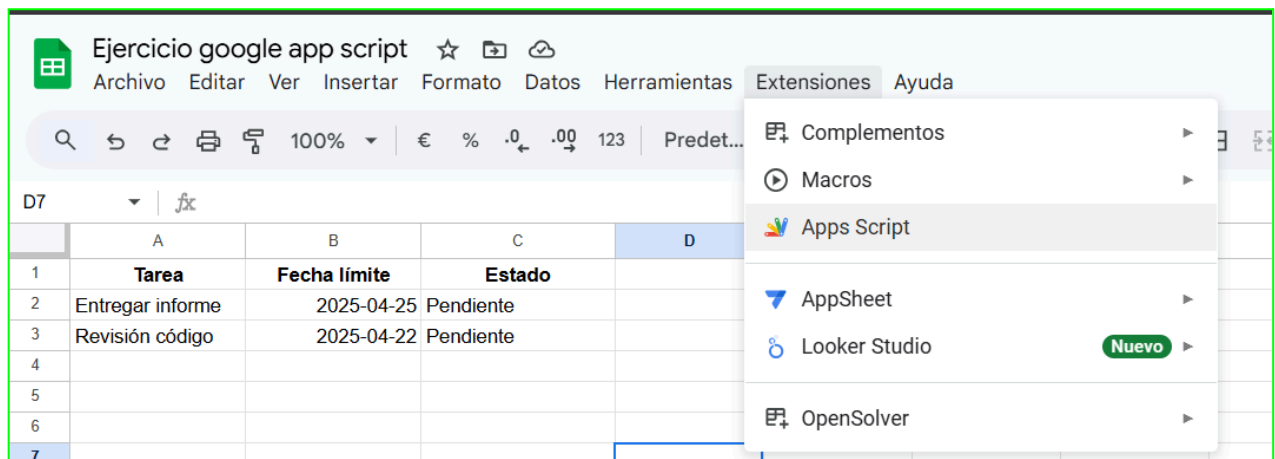
3. Pasos para la creación del script

Para la demostración se creó una tabla de actividades en donde se coloca: La tarea, la fecha de plazo y el estado de la actividad.



	A	B	C
1	Tarea	Fecha límite	Estado
2	Entregar informe	2025-04-25	Pendiente
3	Revisión código	2025-04-22	Pendiente
4			
5			

Consecuentemente en la sección de extensiones nos redirecciona a “google app script” en donde usando código .gs, que es muy similar sino igual a .js se procedió a crear el script.

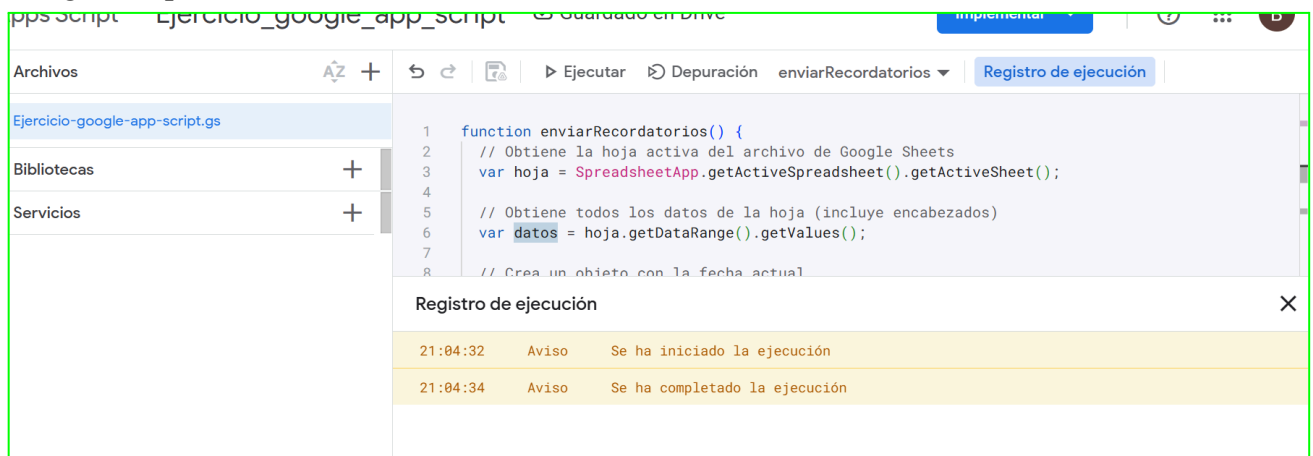


	A	B	C	D
1	Tarea	Fecha límite	Estado	
2	Entregar informe	2025-04-25	Pendiente	
3	Revisión código	2025-04-22	Pendiente	
4				
5				
6				
7				

Aquí se creó el script `Ejercicio_google_app_script.gs` con el siguiente código:

```
1 function enviarRecordatorios() {
2   // Obtiene la hoja activa del archivo de Google Sheets
3   var hoja = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
4
5   // Obtiene todos los datos de la hoja (incluye encabezados)
6   var datos = hoja.getDataRange().getValues();
7
8   // Crea un objeto con la fecha actual
9   var hoy = new Date();
10  hoy.setHours(0, 0, 0, 0); // Resetea la hora para comparar solo por día
11
12  // Arreglo para guardar las tareas pendientes o vencidas
13  var tareasPendientes = [];
14
15  // Recorre cada fila (comienza desde 1 para saltarse el encabezado)
16  for (var i = 1; i < datos.length; i++) {
17    var tarea = datos[i][0];           // Columna A: nombre de la tarea
18    var fechaLimite = new Date(datos[i][1]); // Columna B: fecha límite
19    var estado = datos[i][2];          // Columna C: estado (Hecho, Pendiente)
20
21    fechaLimite.setHours(0, 0, 0, 0); // Igual que 'hoy', resetea la hora
22
23    // Si la tarea NO está hecha y su fecha es hoy o anterior
24    if (estado.toLowerCase() !== 'hecho' && fechaLimite <= hoy) {
25      // Agrega la tarea a la lista de pendientes con su fecha
26      tareasPendientes.push("🔥 " + tarea + " (fecha: " + datos[i][1] + ")");
27    }
28  }
29
30  // Si hay tareas pendientes, envía un correo con el resumen
31  if (tareasPendientes.length > 0) {
32    var mensaje = "Estas son las tareas pendientes o vencidas:\n\n" + tareasPendientes.join("\n");
33
34    // Envía el correo al usuario que está usando el script
35    MailApp.sendEmail(Session.getActiveUser().getEmail(), "🔔 Recordatorio de tareas", mensaje);
36  }
37 }
38
```

Luego lo corrimos en el app script y después de darle permisos para acceder a nuestra cuenta, el código compiló.



The screenshot shows the Google Apps Script editor interface. On the left, a sidebar lists 'Archivos' (Files) with 'Ejercicio-google-app-script.gs' selected, 'Bibliotecas' (Libraries) with a plus icon, and 'Servicios' (Services) with a plus icon. The main editor area displays the JavaScript code for the `enviarRecordatorios()` function. On the right, a 'Registro de ejecución' (Execution Log) window is open, showing two log entries:

Time	Message	Action
21:04:32	Aviso	Se ha iniciado la ejecución
21:04:34	Aviso	Se ha completado la ejecución

En el reloj creamos un activador seleccionando la función a correr, los días, la hora, etc en que nos llegara la notificación. Con fines de prueba se seleccionó que llegara el mensaje al abrir la hoja de cálculo.

Editar Activador de Ejercicio_google_app_script

enviarRecordatorios ▼

Qué debe ejecutarse durante el despliegue

Principal ▼

Selecciona la fuente del evento

De una hoja de cálculo ▼

Selecciona el tipo de evento

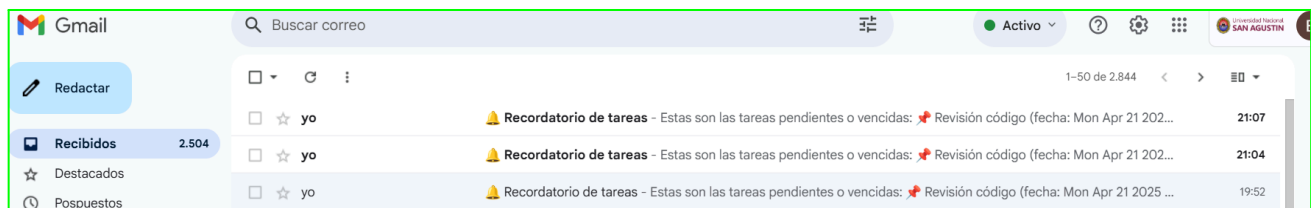
Al abrirse ▼

Notifícame inmediatamente

Cancelar

Guardar

Finalmente se abrió la hoja de cálculo y se comprobó la bandeja de entrada del correo para comprobar de que si estaba llegando el mensaje de alerta.



4. Link Github

- <https://github.com/BCarlosAC/Ejercicio-google-app-script.git>

5. Referencias

- <https://script.google.com/home>
- https://www.youtube.com/watch?v=GD-WxvnWXXs&list=PLG1qjdD__qH4dyXq4sM03Rf0RFhB_4tbm&index=3