

MANUAL DE USUARIO
APLICATIVO DE CITAS - SERVICIOS DE APOYO

VICEDECANATURA DE INVESTIGACIÓN Y EXTENSIÓN
FACULTAD DE INGENIERÍA

YOVANY ESNEIDER VARGAS GUTIÉRREZ
INGENIERO MECATRÓNICO

UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE BOGOTÁ

2025

ÍNDICE

ÍNDICE	2
INTRODUCCIÓN	3
APLICATIVO PRINCIPAL	4
APLICATIVO DE SANCIONES	6
FUNCIONES AUTOMÁTICAS	6
ACTUALIZACIONES	7
DOCUMENTACIÓN Y HERRAMIENTAS UTILIZADAS	10
ANEXO A, DIAGRAMAS	11
ANEXO B, TABLAS.....	12

INTRODUCCIÓN

La tarea de crear un aplicativo web capaz de manipular el Google Calendar de los profesionales de apoyo, exige por parte del desarrollador encontrar la manera más eficiente en la que los usuarios puedan interactuar con el calendario de los profesionales de apoyo. Al observar cómo una persona realizaría manualmente la tarea de agendar una cita, los pasos que debe seguir son los siguientes.

1. Encontrar las franjas horarias disponibles en Google Calendar e informarlas al usuario
2. Reunir los datos del usuario solicitante y la franja horaria deseada
3. Verificar que el usuario no tenga sanciones vigentes
4. Bloquear la franja horaria solicitada
5. Notificar a los involucrados
6. Agregar la cita al historial

Considerando estos factores, es imprescindible la creación de dos aplicativos principales con funcionalidades puntuales, los cuales son: El aplicativo principal, es decir, la página web, donde se muestra a los usuarios las franjas horarias disponibles y se solicitan los datos necesarios para agendar la cita, esta página consta de dos partes, el *FrontEnd* compuesta por la presentación frente a los usuarios, diseño e imagen. Y el *BackEnd*, compuesto por la programación interna de la página, recolección de datos, envío y recepción de información desde el servidor principal. Seguidamente, el aplicativo de sanciones, donde una vez reportada la inasistencia en una cita, por parte del profesional de apoyo encargado, y basándose en su historial de sanciones, se sancionará debidamente al usuario en cuestión. Por lo tanto, el aplicativo de citas, está compuesto por cuatro etapas, consulta de franjas horarias, reserva de cita, consulta de sanciones y historial de citas, con una constante retroalimentación hacia el usuario. Este proceso se puede observar en el diagrama de la **Figura 1**.

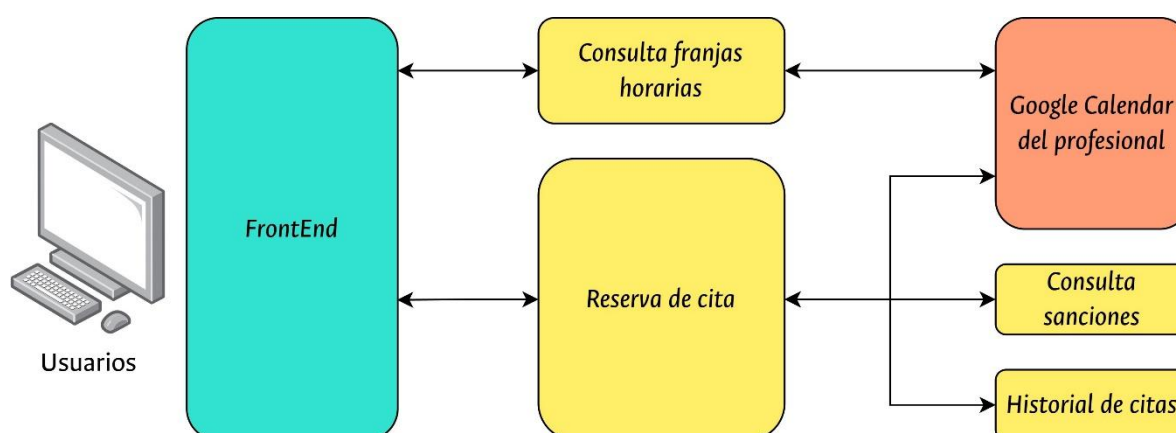


Figura 1. Diagrama, aplicativo de citas

APLICATIVO PRINCIPAL

El aplicativo principal se compone de un formulario HTML, donde se solicitan los datos relevantes al usuario, datos relacionados al proyecto o documento a revisar y la franja horaria deseada, mediante CSS se generan los estilos y animaciones de todos los objetos HTML como radios, inputs, divs, etc. Todos los iconos e imágenes utilizadas están almacenados en el repositorio <https://github.com/VicedecanaturaUnal/AppCitas>, vinculado al correo de la Vicedecanatura de Investigación y Extensión de la Facultad de Ingeniería (vinnyext_fibog@unal.edu.co), se optó por este servicio, ya que Google Drive bloquea el acceso a archivos que presentan un alto número de solicitudes de acceso, si bien, es un medio externo presenta un desempeño excelente, además de ser un servicio gratuito.

Respecto a la programación propia de la página web (FrontEnd), se compone de funciones en JavaScript que permiten el despliegue de opciones en entradas como: *Tipo de vinculación con la Facultad de Ingeniería*, *Tipo de documento* y *Asunto a tratar*. También funciones relacionadas con la validación de datos en las entradas: *Correo institucional*, *Código QUIPU o HERMES del proyecto* y *Convocatoria*. Gracias a que el aplicativo utiliza Google Apps Script como Host, este ofusca el código y otorga niveles de ciberseguridad adecuados, además se puede acceder a las funciones en el BackEnd con un solo comando.

```
google.script.run.withSuccessHandler(response).backEndFunction(parameters)
```

El script asociado al aplicativo lleva por nombre WebSites - App. Citas, allí se encuentra el código relacionado al FrontEnd en HTML, CSS y JS además del BackEnd con extensión .gs (Google Script). Se optó por tener todos los servicios asociados a un mismo proyecto e implementarlo tantas veces como servicios de apoyo existan. Esto se logra gracias a la asociación de una base de datos con el aplicativo, de manera que, al cambiar una única constante en los dos archivos antes mencionados, variable que hace referencia a la posición que ocupa en la base de datos el servicio en cuestión. La base de datos es un archivo sheets con nombre *Database – Aplicativos*, donde las dos primeras hojas contienen toda la información necesaria para el correcto funcionamiento del aplicativo.

Las franjas horarias disponibles y toda la información correspondiente al formulario HTML como elementos, títulos, tamaños. Se obtienen mediante el llamado a una pequeña API creada con este propósito, su funcionamiento es el siguiente.

```
let urlAPI = 'https://script.google.com/macros/s/AKfycbye20biAa8cDEQrKF0ZNcWN0zki3fTsvugnvdLhWv2ARf385zkfEHBmXfBT6jXIdeQfFA/exec'
let requestCode = XY
let urlRequest = `${urlAPI}?code=${requestCode}`
```

X corresponde a la posición del servicio en la base de datos, Y corresponde a los datos a solicitar, siendo 1 el título del servicio, 2 los elementos HTML y 3 las franjas horarias disponibles en ese momento, la API está contenida en el archivo *DataAPI - App. Citas*. Una vez obtenidos todos los datos, se reflejan en el formulario de forma ordenada de acuerdo con los estilos en CSS, y las franjas horarias se exponen al usuario como se muestra en la **Figura 2**.

● Virtual

Fecha	05/may/25	06/may/25	07/may/25	08/may/25
Hora	Lunes	Martes	Miércoles	Jueves
09:00 a.m.	Ocupado	Disponibile	Disponibile	Ocupado
10:00 a.m.	Ocupado	Disponibile	Disponibile	Ocupado
11:00 a.m.	Ocupado	Disponibile	Disponibile	Ocupado
01:00 p.m.	Disponibile	Disponibile	Disponibile	Disponibile
02:00 p.m.	Disponibile	Disponibile	Su cita	Disponibile
03:00 p.m.	Disponibile	Disponibile	Disponibile	Disponibile
04:00 p.m.	Ocupado	Disponibile	Disponibile	Ocupado

Figura 2. Franjas horarias expuestas en el formulario HTML.

La lógica detrás de la obtención de franjas horarias disponibles se basa en la validación de tres casos puntuales mediante la función `slotsValidation()` en *DataAPI – App. Citas*, de cumplirse alguno, significa que la franja horaria no está disponible, ocupado de cara al usuario. Además, se obtienen todas las franjas horarias hasta la tercera semana a partir del lunes correspondiente a la semana de la fecha de consulta, las franjas horarias son consultadas en base a los horarios establecidos en la segunda hoja del *sheets* que contiene la base de datos de los aplicativos, de manera que, si se requiere hacer un cambio en el horario de algún servicio de apoyo, debe modificarse únicamente la segunda hoja en el área predefinida para dicho servicio.

Al enviar el formulario la solicitud se somete a validación, donde pueden surgir siete escenarios, el primer escenario es cuando al momento de hacer la solicitud por razones temporales o de intersección de solicitudes la franja horaria deja de estar disponible. Segundo escenario, es cuando el usuario solicitante aparece sancionado de por vida. Tercer escenario, es cuando el usuario solicitante esta sancionado por cierta cantidad de días. Cuarto escenario, es cuando el solicitante tiene una cita con el servicio en cuestión, no puede solicitar otra cita hasta cumplir dicha cita. Quinto escenario, es cuando el usuario ha cancelado una cita y aún no se ha eliminado en el Google Calendar del servicio. Sexto escenario, es cuando el usuario solicitante ya tuvo una cita y está solicitando una nueva el mismo día. Séptimo escenario, es cuando una vez validados los seis anteriores el usuario agenda la cita exitosamente.

Así se satisfacen los requerimientos planteados en el análisis hecho inicialmente, con las franjas horarias expuestas al usuario, este puede solicitar una cita en cualquiera que se encuentre disponible. Habiendo completado las demás entradas en el formulario HTML y elegido la franja horaria deseada, una vez el usuario envía la solicitud de cita, la función `request` almacenada en el *BackEnd*, somete los datos del usuario a verificación. Si no se encuentra ninguna novedad, como sanciones vigentes o cruce de citas. La franja horaria es bloqueada y la cita es agendada en el calendario del profesional de apoyo correspondiente, el usuario recibirá una notificación y los datos de la cita serán añadidos al historial, el historial es un documento *sheets* con nombre *Historial de citas YYYY* (Año actual).

APLICATIVO DE SANCIONES

Ante la posibilidad de inasistencias en las citas, se opta por sancionar a los usuarios que agenden una cita y no asistan a la misma. A la fecha, solo el servicio de Corrección de estilo de textos en inglés y español presenta periodos de sanción según el número de inasistencias acumuladas. Sin embargo, para los demás servicios el aplicativo funciona adecuadamente y bajo las mismas condiciones. De igual manera este aplicativo también está asociado a la base de datos, de manera que cualquier modificación en la misma alterará automáticamente el aplicativo.

Este aplicativo de sanciones está compuesto por un formulario HTML y un documento sheets asociado al mismo, allí se solicita la fecha de la cita, hora y el servicio asociado como se muestra en la **Figura 3**. Al igual que el aplicativo principal, este cuenta con funciones para el despliegue de opciones en la entrada relacionada con la Hora de la cita, además de la validación de la fecha seleccionada, de manera que si se otorga una fecha en la cual es fin de semana o no hay horario definido para ese día, el aplicativo lo notifique, también cuenta con casos puntuales para la consulta y la sanción, estos están comentados adecuadamente en el archivo con nombre *back.gs*, en el proyecto contenedor que lleva por nombre *Penalty - App. Sanciones*.

Formulario de Sanciones

Vicedecanatura de Investigación y Extensión
Facultad de Ingeniería
Sede Bogotá



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Indique los datos de la cita a la que no se asistió.

Fecha de la cita
03/10/2023

Hora de la cita
Seleccione un servicio de apoyo.

Servicio de apoyo

BUSCAR CITA

Figura 3. Interfaz de usuario, aplicativo de sanciones

FUNCIONES AUTOMÁTICAS

Debido a diferentes requerimientos del aplicativo, como la cancelación automática de citas, el levantamiento de sanciones una vez cumplidas, el cambio de año y la consulta del historial. Es necesaria la implementación de funciones de ejecución periódica o que sean activadas por un evento predefinido, para ello *Google Apps Script* cuenta con un tipo de implementación denominado *Activador*, este se puede configurar respecto a tiempo, eventos en calendario, interacción con documentos, entre otros. La función asociada a la cancelación de citas automáticamente corresponde a *appointmentEliminator()*, este activador se ejecuta cada 15 minutos, para un total de 96 ejecuciones diarias, siendo 100 el límite de ejecuciones diarias que Google permite para cuentas institucionales. Su funcionamiento es el siguiente, se obtienen los eventos de las siguientes tres semanas partiendo del lunes de la semana actual, y verifica que los invitados hayan rechazado la invitación al evento, es de vital importancia que si el anfitrión, invitó a otros usuarios todos rechacen la invitación, de no ser así no se detectará como cancelada, estos términos están presentes en los términos y condiciones de todos los servicios

de apoyo. En caso contrario, la función elimina la cita y deja la franja horaria disponible nuevamente.

La función *subtractDay()* se encarga de llevar el conteo y levantamiento de sanciones al cumplirse el plazo establecido, su funcionamiento se basa en acceder al historial de sanciones, donde la hoja con nombre *Lista Temporal*, almacena las sanciones vigentes. Seguidamente se recolectan los datos y resta un día a todas las sanciones vigentes, si dicha resta llega a cero quiere decir que el tiempo de sanción ha terminado y el usuario puede acceder al servicio nuevamente. Este activador se ejecuta pasada la medianoche, entre las 00:00 a.m. y 01:00 a.m.

La función *annualOrganizer()*, es la encargada de organizar los archivos cada 31 de diciembre, su funcionalidad se resume en la creación de las nuevas carpetas asociadas al siguiente año, tales como COMITÉ UAB YYYY, APLICATIVOS YYYY, etc. Seguidamente se hace el traspaso del historial de citas y el de sanciones, donde el primero se limpia para dar paso al siguiente año. Antes de limpiar el historial se crea una copia la cual permanece en la carpeta correspondiente al año pasado. Mientras que el historial de sanciones sí debe ser el mismo. Así, se logra mantener la programación asociada a cada documento y los enlaces de acceso permanecen igual. Finalmente, es necesario una función que genere un reporte actualizado del historial de citas, para ello se vinculó el historial con la función *consolidated()*, la cual está programada para ejecutarse cada vez que se abra el archivo Historial de citas YYYY, la función se encarga de crear un consolidado el cual alimenta varias tablas dinámicas. También existe un activador el cual se ejecuta el 31 de diciembre, una hora antes que *annualOrganizer()*, con el propósito de garantizar el traspaso del consolidado actualizado. En el **Anexo A.1.** se puede observar la distribución y ubicación de estas funciones, almacenadas en proyectos o archivos asociados respectivamente.

ACTUALIZACIONES

Todas las funciones y aplicativos están asociados con la base de datos, de manera que cualquier actualización debe llevarse a cabo modificando esta. Sin embargo, existen ciertos procedimientos que requieren pasos adicionales para tener en cuenta dado el caso, al crear un nuevo servicio se hace un barrido de todos estos procedimientos.

Paso a paso para la creación de un nuevo servicio

Para la creación de un nuevo servicio, en la primera hoja de la base de datos, se inserta una fila a la derecha de la última fila ya existente en la hoja. El índice asociado a ese servicio será el número de columna, A=1, B=2, C=3, etc. Existen 17 campos a rellenar para crear el servicio ver **Anexo B.1.** estos campos alimentan los aplicativos y funciones que componen el aplicativo de citas, **la creación de un nuevo servicio involucra crear un nuevo horario en la hoja Horarios, una nueva pestaña en el historial de citas y nuevas columnas en el historial de sanciones.** Con todos estos elementos creados se hace clic en el menú **Opciones/Actualizar**, esta es una opción personalizada, que actualiza todos los documentos y genera arreglos que se utilizan en los aplicativos. Antes de crear una nueva implementación en el proyecto contenedor, se deben asegurar los permisos de modificación

en el Google Calendar del profesional de apoyo, para ello el profesional debe dirigirse a la ruta en Google Calendar Configuración/Configuración de mis calendarios/Calendario con el nombre del profesional/Compartir con, una vez allí, debe otorgar el permiso de Hacer cambios y gestionar el uso compartido, a dos usuarios nerojasb@unal.edu.co y vinnyext_fibog@unal.edu.co. Como se muestra en la **Figura 4**.

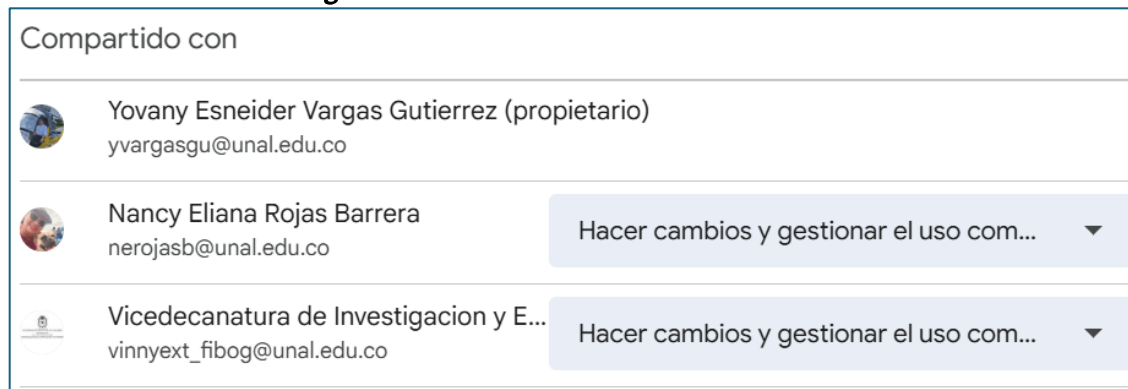


Figura 4. Permisos Google Calendar

Con los permisos obtenidos se da paso a la implementación para un nuevo servicio, en primer lugar, se accede al proyecto contenedor (WebSites - App. Citas), allí se encuentran expuestos dos archivos a la izquierda del espacio de trabajo, en el primer archivo *Código.gs*, se debe modificar la constante ubicada en la línea número dos, *serviceIndex* por el valor de la columna que ocupa el nuevo servicio, de igual manera en el archivo *webSite.html*, se modifica la constante ubicada en la línea número siete. Se guardan los cambios, para dirigirse a la parte superior derecha, donde se hace clic en el botón *Implementar*, seguidamente, clic en la opción *Nueva Implementación*.

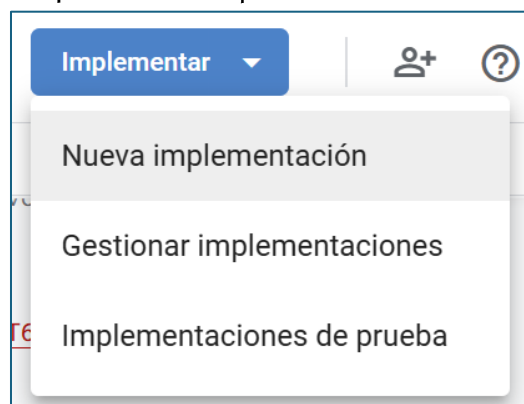


Figura 5. Primer paso, nueva implementación

Se despliega una nueva ventana donde se solicita la descripción de la implementación, la nomenclatura manejada es ID del servicio Año.Versión, en el ejemplo se visualiza CT 2025.1.0 ver **Figura 6**. correspondiente a *Corrección de estilo de textos en ingles y español*, año 2025 y versión 1.0. El usuario que ejecuta debe ser la Vicedecanatura de Investigación y Extensión, con el correo electrónico vinnyext_fibog@unal.edu.co y el acceso debe ser solo para la Universidad Nacional de Colombia, así se garantiza que el aplicativo web por sí mismo

valide que el usuario que accede al servicio pertenece a la Universidad Nacional de Colombia sin funciones o validaciones adicionales por parte del BackEnd.

Nueva implementación

Seleccionar tipo Configuración

Aplicación web

Descripción

Nueva descripción

CT 2025.1.0

Aplicación web

Ejecutar como

Yo (vinnyext_fibog@unal.edu.co)

La ejecución de la aplicación web se autorizará con los datos de tu cuenta.

Quién tiene acceso

Cualquier usuario de Universidad Nacional de Colombia

También se puede usar como una biblioteca. [Más información](#)

Cancelar Implementar

Figura 6. Segundo paso, nueva implementación

Haciendo clic en el botón implementar, se despliega una nueva ventana que contiene la URL del aplicativo web, esta URL es única y es equivalente a una página web.

Nueva implementación

La implementación se ha actualizado correctamente.

Versión 141 del 24 abr 2025, 9:14

ID de implementación

AKfycbxPz7FqyNU8n_GeHHg1ddsWpnoDr8B7uzcoOMkQRytb9_kyoKCqveqnY_wwwAo1QhWAjQ

[Copiar](#)

Aplicación web

URL

https://script.google.com/a/macros/unal.edu.co/s/AKfycbxPz7FqyNU8n_GeHHg1ddsWpnoDr8B7uzcoOMkQRytb9_kyoK...

[Copiar](#)

Hecho

Figura 7. Tercer paso, nueva implementación

El aplicativo funciona idealmente y el código ya ha sido refactorizado, sin embargo, si el desarrollador ve conveniente seguir mejorándolo, debe garantizar que las URLs sean las mismas, esto se hace mediante la opción gestión de implementaciones que se muestra en la **Figura 5**. Debe realizar las implementaciones sobre las ya existentes, respetando el manejo de versiones.

DOCUMENTACIÓN Y HERRAMIENTAS UTILIZADAS

Las herramientas utilizadas en el desarrollo del aplicativo son las siguientes.

- Google Apps Script
Este aplicativo permite el acceso y modificación directa en la mayoría de servicios de Google, en este caso Google Calendar y Google Sheets. Aquí es donde se encuentran implementadas los aplicativos para evitar el uso de servicios externos.
- Visual Studio Code
Gracias a su amplia lista de extensiones, la eficiencia en el diseño de páginas web aumenta haciendo uso de esta aplicación de escritorio.
- Google Drive
Este aplicativo se utiliza como servidor principal, donde se almacenan archivos de texto, manuales y están almacenados los proyectos de Google Apps Script.
- GitHub
Aquí se encuentra almacenado todo el aplicativo, dividido en carpetas y archivos, junto a todas las imágenes utilizadas.

Para el desarrollo total del aplicativo se hizo uso de JavaScript como lenguaje de programación principal, HTML y CSS, para el diseño de las páginas web. Toda la documentación de Google Apps Script se encuentra en su página web en el apartado de referencias developers.google.com/apps-script/reference, allí se encuentra la descripción de todos los objetos, clases, métodos y servicios de Google que Google Apps Script puede manipular o acceder. Muy útil a la hora de utilizar métodos o clases nuevas para el desarrollador, además de existir gran cantidad de material didáctico en YouTube y blogs de la comunidad. Cabe aclarar que Google puede realizar actualizaciones de toda esta documentación e incluso cambiar el funcionamiento de métodos utilizados en el aplicativo, generando errores inesperados, para lo cual el desarrollador debe estar preparado para implementar soluciones ideales y que se ajusten a las necesidades de los servicios de apoyo de la Vicedecanatura de Investigación y Extensión de la Facultad de Ingeniería de la Universidad Nacional de Colombia Sede Bogotá.

ANEXO A, DIAGRAMAS

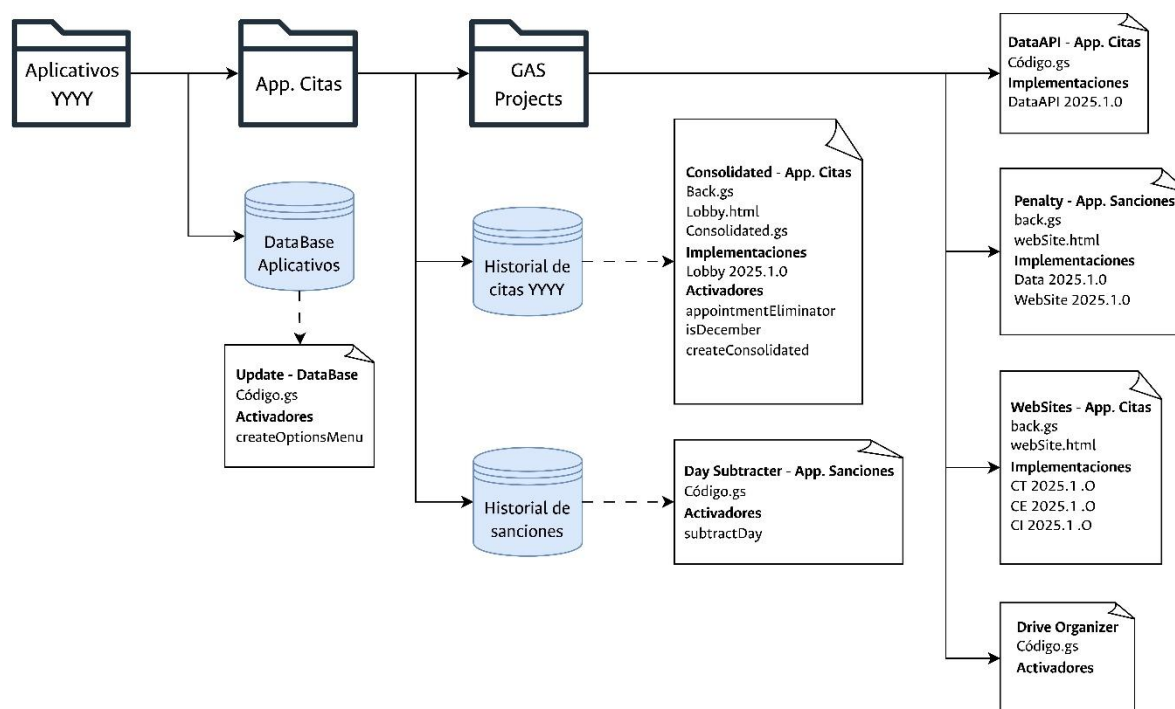


Figura 8. Distribución de carpetas, archivos y proyectos Google Apps Script

ANEXO B, TABLAS

Nombre del servicio	Nombre del servicio de apoyo.
Nombre del profesional de apoyo	Nombre del profesional responsable del servicio de apoyo.
Correo del profesional de apoyo	Correo del profesional responsable del servicio de apoyo.
ID	ID correspondiente a un abreviado del nombre del servicio de apoyo.
Semanas de servicio	Numero de semanas que muestra el servicio.
Duración de la cita	Duración de la cita en minutos.
Usuarios siempre invitados	Correos electrónicos de usuarios ajenos al profesional de apoyo, que necesite presentes por una u otra razón.
Título del evento	Título que tendrán los eventos creados en el Google Calendar del profesional de apoyo.
Ubicación por modalidad	Arreglo de tamaño dos, lugar del evento por modalidad.
Elementos HTML	Arreglo de tamaño dos, que contiene los elementos HTML a insertar en la sección del formulario, correspondiente con el servicio de apoyo. La primera posición es el título de la sección, la segunda es un arreglo de tamaño n, con los elementos HTML cada uno descrito por un arreglo de tamaño nueve. Name del elemento, url de la imagen usada por el label, id del elemento, tipo de input, en caso de ser <i>select</i> un arreglo con las opciones del <i>select</i> , arreglo con la id y el valor de un elemento del cual dependa el elemento nuevo a añadir (este elemento debe ser definido antes del nuevo), test de valor once posibles validaciones (Función con nombre <i>logicValidation()</i> en el código JavaScript del formulario HTML, texto de alerta, test de campo vacío. Esto últimos dos tests se activan con un 1 y se desactivan con un "" ya que se usa <i>JSON.parse()</i> para el traspaso de texto a array, <i>alertText</i> es el texto que aparece en caso de no cumplirse el test de valor, mientras que el texto para el test de campo vacío está predefinido en el código.
Datos de la cita - Cuerpo de invitación	Arreglo de tamaño dos, que contiene los datos que se van a mostrar en el cuerpo de la invitación que emite el evento en Google Calendar. En primera posición un arreglo con el índice y título del dato, en segunda posición un arreglo con los tests de

	validación de la información, donde cada elemento contiene dos índices y un valor.
Columnas sanciones	Arreglo de tamaño 3, primera posición, columna donde se ubica el conteo del número de sanciones por usuario, segunda posición, fecha de inasistencia en la lista temporal y tercera posición, días de sanción.
Días de sanción	Arreglo de tamaño definido por la tipología de sanciones del servicio, allí se almacenan en orden ascendente los días de sanción respectivamente con el número de sanciones en el historial de sanciones.
Categorías del servicio	Arreglo de tamaño definido por las categorías de atención que ofrece el servicio, que almacena los tipos de atención a los que se puede acceder.
Columna de categorías en historial	Letra única que indica la columna donde se ubica la categoría de las citas por servicio en el historial de citas.
Servicio activo	Contenedor binario, Sí o No.

Tabla 1. Campos base de datos, servicio de apoyo