

Saturn Studio – Webhooks



Los **Webhooks** son herramientas de comunicación en tiempo real que permiten a aplicaciones externas (como Facebook, GitHub o servicios de pago) enviar notificaciones automáticas a **Saturn Studio**. En lugar de que tu robot esté preguntando constantemente si hay novedades, el sistema externo envía la información de manera inmediata a una dirección URL única, lo que dispara la ejecución del bot al instante.

Interfaz de Gestión (Panel Lateral)

Una vez que has configurado y guardado un robot que contenga un comando de Webhook, la información de control aparecerá automáticamente en la sección **Webhooks** del menú lateral izquierdo.




Columna	Descripción
Robot	El nombre del bot que tiene el receptor activo.
Method	El tipo de petición (POST, GET, etc.) que espera recibir.
URL	La dirección única que debes copiar y pegar en la aplicación externa.
Tasks	El contador de cuántas veces se ha ejecutado el flujo por este medio.
Success / Pending	Muestra cuántas solicitudes terminaron bien y cuántas están en proceso.
Actions	Aparecerán las opciones para eliminar el webhook, borrar tareas o ver la cola de tareas.

Visualizar cola de tareas / View tasks queue

SaturnStudio by Rocketbot Create now

Webhooks

Available Webhooks: Here you can view all the webhooks configured for your robots. You can recreate the webhook by editing the robot or by testing the webhook data retrieval command within the robot.

Robot	Method	URL	Tasks	Success	Pending	Actions
_webhooks	GET	https://studio.rocketbot.com/webhook/dd5ea4e19aab52b6882a79458df8263a	5	0	0	  

Botones de Acción (Actions)

- **Papelera (Rojo):** Elimina permanentemente el Webhook y su URL del servidor.
- **Escoba/Borrador (Amarillo):** Limpia el historial de tareas y pone los contadores a cero.
- **Ojo (Verde):** Permite ver los detalles técnicos de la configuración actual del Webhook.

Al inspeccionar las ejecuciones de un Webhook, se despliega este modal que lista cada petición recibida de forma individual.

- **ID:** Identificador único alfanumérico de la petición recibida.
- **Header / Body / Query / Data:** Indican la cantidad de parámetros o el peso de la información recibida en cada sección de la petición HTTP.
- **Status:** Muestra si la tarea específica fue completada o si presentó algún error.
- **Updated:** Tiempo transcurrido desde que se recibió la petición.
- **Botón de Ver (Ojo Azul):** Abre el desglose detallado de los datos crudos (raw data) de esa tarea específica.

ID	Header	Body	Query	Data	Status	Updated	
84adcf1bf0cdaa039d6ffad5d3...	20	0	0	0		35 minutes ago	
c1c6868bdc700625066287f17...	18	0	0	0		35 minutes ago	
c443c9ced3220c900232394...	20	0	0	0		35 minutes ago	
e85689e06c54820539cd3192...	20	0	0	0		35 minutes ago	
3d223cdb2045e98ad65a7569...	20	0	0	0		35 minutes ago	

Page Size 10 | First | Prev 1 Next | Last

Detalle de Datos de Tarea (View tasks data)

Este panel es fundamental para la depuración (debugging), ya que muestra exactamente qué información envió la aplicación externa a Saturn Studio.

- **Headers:** Contiene los metadatos de la conexión (IP de origen, tipo de navegador, tokens de autenticación, etc.).
- **Body:** El contenido principal del mensaje, generalmente en formato JSON.
- **Query:** Parámetros enviados directamente en la URL (ej: ?id=123).
- **Params:** Parámetros de ruta procesados.
- **Response:** La respuesta que Saturn Studio devolvió al servidor emisor tras recibir el mensaje.

Headers

```
{"host":"studio.rocketbot.com","x-real-ip":"200.105.110.230","x-forwarded-for":"200.105.110.230","x-forwarded-proto":"https","connection":"close","cache-control":"max-age=0","sec-ch-ua":\\"Not:A-Brand\\";v=\\\"99\\\", \\\"Google Chrome\\";v=\\\"145\\\", \\\"Chromium\\";v=\\\"145\\\"","sec-ch-ua-mobile":"?0","sec-ch-ua-platform":\\"Windows\\","upgrade-insecure-requests":"1","user-agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/145.0.0.0 Safari/537.36","accept":"text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7","sec-fetch-site":"none","sec-fetch-mode":"navigate","sec-fetch-user":"?1","sec-fetch-dest":"document","accept-encoding":"gzip, deflate, br, zstd","accept-language":"en-US,en;q=0.9,es;q=0.8","cookie": "_gid=GA1.2.519211266.1772549334; _legacy_auth0.Bq16mHIGo7LF2uXbQfLbgJGX282tEe5w.is.authenticated=true; auth0.Bq16mHIGo7LF2uXbQfLbgJGX282tEe5w.is.authenticated=true; _ga=GA1.1.916201240.1771594458; _ga_JF6Z76QC5Q=GS2.1.s1772799177$o27$g1$t1772799575$j27$l0$h0; _ga_RMYQMPF73F=GS2.1.s1772799542$o9$g1$t1772799575$j27$l0$h0","if-none-match":"W/\\\"15-BVEW84Yxx4bH1R0h00COoTmUjFw\\\""}}
```

Body

```
{}
```

Query

```
{}
```

Params

```
{}
```

Response

Descripción de los Comandos

Para usar esta tecnología, encontrarás los siguientes módulos dentro de la categoría **Webhooks** del editor.



1. Recibir Webhook (Receive Webhook)

- **Webhook URL:** Es la dirección estática que genera el sistema para tu bot.
- **HTTP Method:** El protocolo de comunicación. Puedes elegir entre **GET**, **POST**, **PUT** o **DELETE**.
- **Response mode:** Define cómo se le avisa al emisor que el dato llegó. Puede ser **Immediately** (al instante), **After robot execution** (al terminar el bot) o de forma manual con el comando de respuesta.
- **Assign result to Variable:** El nombre de la variable donde se guardará toda la información recibida (el cuerpo del mensaje).

2. Responder Webhook (Response Webhook)


Se utiliza cuando necesitas enviar una respuesta personalizada o específica después de procesar los datos.


- **Task ID:** El identificador de la tarea a la que estás respondiendo.
- **Response:** El texto o JSON que quieres devolver a la aplicación original.
- **Headers:** Encabezados técnicos opcionales para la respuesta.

Configuración Avanzada (Pestaña Config)

Dentro de los módulos, puedes ajustar el comportamiento ante fallos:

- **Retry on Error:** Número de intentos (de 0 a 10) si ocurre un fallo al recibir.
- **Wait Between Retries:** Segundos de espera entre cada intento de recuperación.

 **Receive Webhook** { x } Test Command Docs ✕

Add a description 

Properties **Config**

Retry on Error

Number of times to retry this command if an error occurs (0-10)

Retry Scope

What to retry on error

Wait Between Retries (seconds)

Seconds to wait between each retry attempt

Cancel Ok

Cómo funciona todo en conjunto

1. **En el Editor:** Colocas el módulo **Receive Webhook** al principio de tu robot. Al configurarlo, copias la **URL**.
2. **En la App Externa:** Pegas esa URL en la configuración de la aplicación que quieres conectar (por ejemplo, en las opciones de desarrollador de WhatsApp o Facebook).
3. **El Evento:** Cuando sucede algo afuera, esa app envía un mensaje a la URL.
4. **La Ejecución:** Saturn Studio detecta el mensaje, “despierta” al robot y guarda los datos en la **Variable** que elegiste.
5. **El Monitoreo:** Puedes ir al menú principal, entrar a **Webhooks** y ver cuántas veces ha funcionado y si hubo errores.

[Saturn Studio – Global Variables](#)

Las **Variables Globales** son un recurso centralizado diseñado para simplificar la gestión de datos que necesitan ser compartidos entre múltiples procesos. En lugar de configurar la misma información en cada robot, puedes definirla una sola vez y acceder a ella desde cualquier lugar de tu cuenta.

¿Para qué sirven?

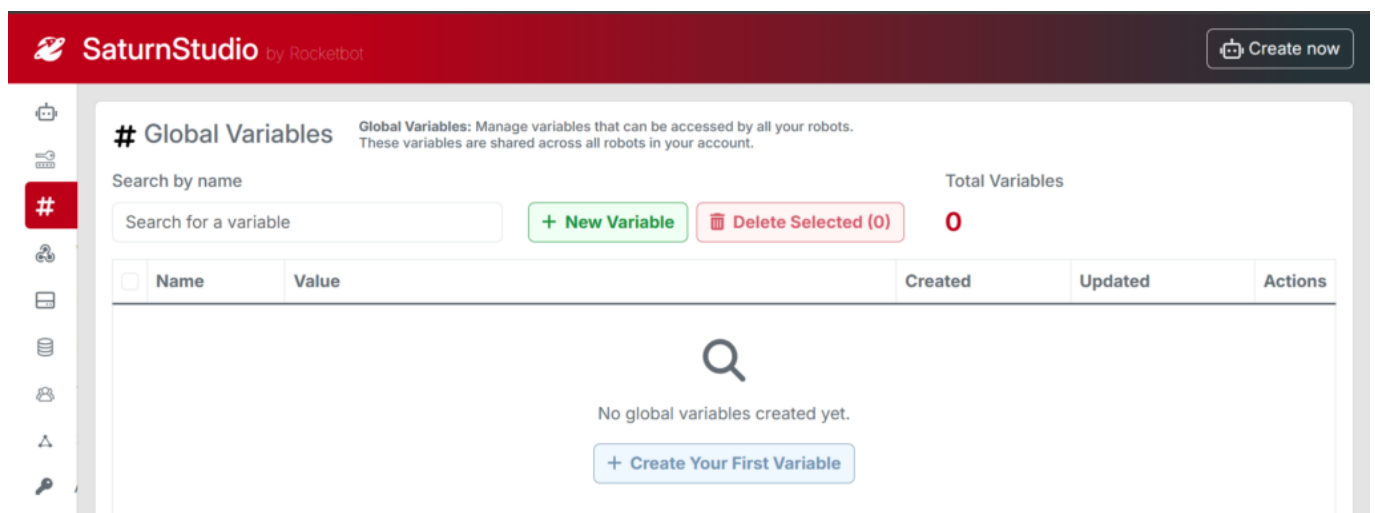
Estas variables están pensadas para almacenar datos constantes o configuraciones maestras que todos tus robots deben conocer. Son ideales

para:

- **URLs de API:** Si la dirección de tu servidor cambia, solo la actualizas en un lugar.
- **Credenciales compartidas o Hosts de Base de Datos:** Mantiene la consistencia técnica en todo tu ecosistema de automatización.
- **Tokens o Flags de sistema:** Información que debe ser idéntica para todos los procesos activos.

Interfaz de Gestión

Desde el panel lateral de Saturn Studio, puedes acceder al gestor de variables globales, donde encontrarás las siguientes herramientas:



The screenshot shows the 'Global Variables' management page in SaturnStudio. At the top, there's a red header with the 'SaturnStudio by Rocketbot' logo and a 'Create now' button. Below the header, the page title is '# Global Variables' with a subtitle: 'Global Variables: Manage variables that can be accessed by all your robots. These variables are shared across all robots in your account.' There's a search bar labeled 'Search by name' with a placeholder 'Search for a variable'. To the right of the search bar are two buttons: '+ New Variable' (green) and 'Delete Selected (0)' (red). Further right is a 'Total Variables' counter showing '0'. Below these elements is a table with columns: 'Name', 'Value', 'Created', 'Updated', and 'Actions'. The table is currently empty, displaying a large magnifying glass icon and the text 'No global variables created yet.' with a '+ Create Your First Variable' button below it.

Elemento	Función
Search by name	Permite filtrar y encontrar rápidamente una variable específica en una lista extensa.
+ New Variable	Abre el formulario para registrar una nueva variable global.
Delete Selected	Elimina de forma masiva las variables marcadas en la tabla.
Total Variables	Contador en tiempo real del número de variables creadas.

La Tabla de Variables

La tabla principal muestra información detallada para mantener el control sobre los datos:

- **Name:** El identificador único de la variable.
- **Value:** El valor que será devuelto a los robots cuando la invoquen.
- **Created / Updated:** Marcas de tiempo para rastrear cuándo se creó o modificó el dato por última vez.
- **Actions:** Botones para editar o eliminar variables de forma individual.

Cómo Crear una Variable Global

Al hacer clic en + **New Variable**, se desplegará un modal con los siguientes campos obligatorios:

Create Global Variable ✕

Variable Name *

Use descriptive names like API_URL, DATABASE_HOST, etc. Names must be unique.

Variable Value *

The value that will be returned when robots access this variable.

Usage in Robots: Access this variable in your robots using the variable name. Ex: `{global.variableName}`

Cancel **+ Create Variable**

1. **Variable Name (*):** Se recomienda usar nombres descriptivos en mayúsculas (ej: API_URL, DATABASE_HOST). **Importante:** Los nombres deben ser únicos dentro de tu cuenta.
2. **Variable Value (*):** Aquí introduces el dato específico que quieres que tus robots recuperen.
3. **Create Variable:** Guarda el registro y lo hace disponible inmediatamente para todos tus procesos.

Uso en los Robots (Sintaxis)

Una vez creada, invocar una variable global dentro del editor de cualquier

robot es muy sencillo. Solo debes usar la siguiente estructura:




```
{global.nombreDeTuVariable}
```


Por ejemplo, si creaste una variable llamada API_URL, dentro de tus comandos deberás escribir {global.API_URL} para que Saturn Studio reemplace ese texto por el valor real durante la ejecución.


Global Variables Global Variables: Manage variables that can be accessed by all your robots. These variables are shared across all robots in your account.

Search by name Total Variables


+ New Variable Delete Selected (0) 1

<input type="checkbox"/>	Name	Value	Created	Updated	Actions
<input type="checkbox"/>	test_variable	test	Mar 06, 2026 11:04	Mar 06, 2026 11:04	  

 **Set var** { x } Test Command Docs ✕

Add a description 

Properties Config

 Assign a value to a variable


Data






{global.test_variable}

Assign result to variable

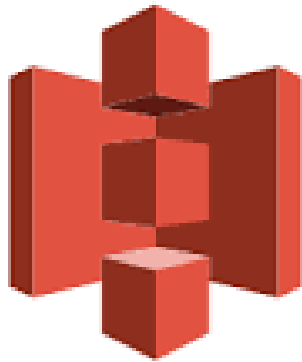
test_variable ▼

Cancel Ok

 **Variables {*}** + Add Variable

	Name	Data	
	test_variable	"test"	   

[Saturn Studio – AWS S3](#)



amazon S3

Amazon S3 (Simple Storage Service) es un servicio de almacenamiento en la nube que te permite guardar, proteger y recuperar archivos y datos de forma segura y escalable. Su integración en **Saturn Studio** facilita la gestión automatizada de activos digitales, copias de seguridad y transferencia de archivos a gran escala.

¿Cómo obtener las credenciales? [Haz clic aquí](#)

Descripción de los Comandos

Cargar Archivo (Upload File)

Sube un archivo local a un contenedor (bucket) específico en AWS S3.

Parámetro	Descripción	Ejemplo
Credential	Credencial de AWS S3 necesaria para cargar el archivo.	My AWS S3 credentials
Bucket Name	Nombre del bucket de S3 donde se guardará el archivo.	bucket_name
File	El archivo local que deseas subir a AWS S3.	Archivo a cargar
Assign result to variable	Variable donde se almacenará el resultado de la carga.	{var}

Descargar Archivo (Download File)

Descarga un archivo desde un bucket de AWS S3 a tu entorno local.

Parámetro	Descripción	Ejemplo
Credential	Credencial de AWS S3 necesaria para descargar el archivo.	My AWS S3 credentials
Bucket Name	Nombre del bucket de S3 donde está almacenado el archivo.	bucket_name
File Name	La clave (ruta) del archivo que se desea descargar de S3.	path/to/file.txt
Assign result to variable	Variable donde se almacenará el archivo descargado o el resultado.	{var}

Obtener Lista de Objetos en un Bucket (Get List of Objects in a Bucket)

Recupera un listado de todos los objetos almacenados dentro de un bucket específico.

Parámetro	Descripción	Ejemplo
Credential	Credencial de AWS S3 necesaria para consultar el bucket.	My AWS S3 credentials
Bucket Name	Nombre del bucket de S3 que se desea listar.	bucket_name
Assign result to variable	Variable donde se almacenará la lista de objetos.	{var}

Obtener Lista de Buckets (Get List of Buckets)

Enumera todos los buckets disponibles asociados a la cuenta de AWS.

Parámetro	Descripción	Ejemplo
Credential	Credencial de AWS S3 necesaria para listar los buckets.	My AWS S3 credentials
Assign result to variable	Variable donde se almacenará el listado de buckets encontrados.	{var}

[Saturn Studio – Credencial AWS S3](#)



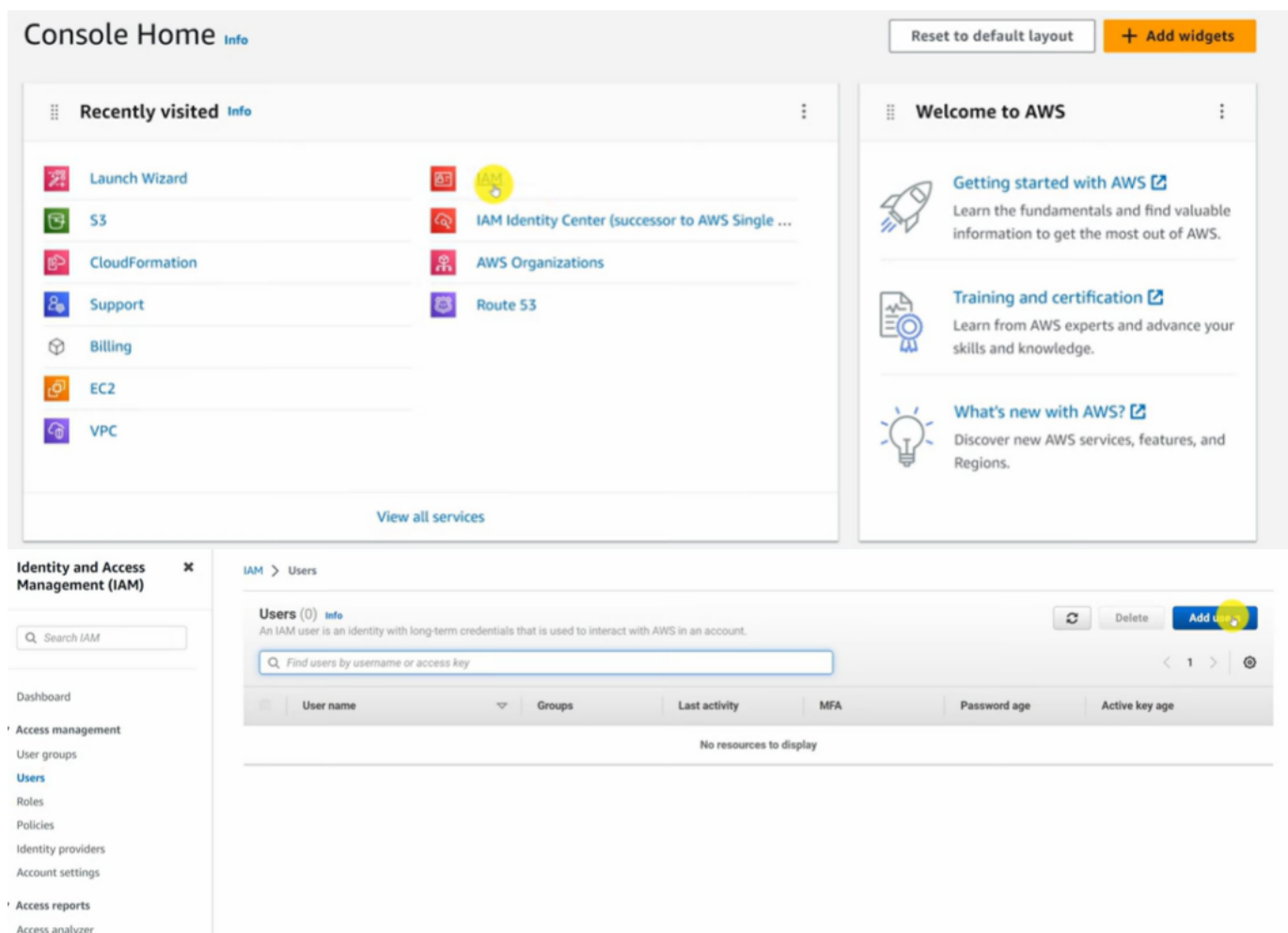
Amazon S3 (Simple Storage Service) es un servicio de almacenamiento en la nube que permite guardar, proteger y recuperar archivos y datos de forma segura y escalable. Su integración en **Saturn Studio** es fundamental para automatizar flujos que requieran el manejo persistente de activos digitales o copias de seguridad.

Cómo obtener las llaves de acceso (IAM)

El proceso de obtención de credenciales para S3 es idéntico al de otros servicios de AWS, variando únicamente el permiso específico que se le otorga al usuario.

1. Creación del Usuario en la Consola de AWS

- Inicia sesión en tu Consola de administración de AWS.
- En el buscador superior, selecciona el servicio **IAM**.
- En el menú lateral, dirígete a **Users** (Usuarios) y haz clic en el botón **Add Users**.



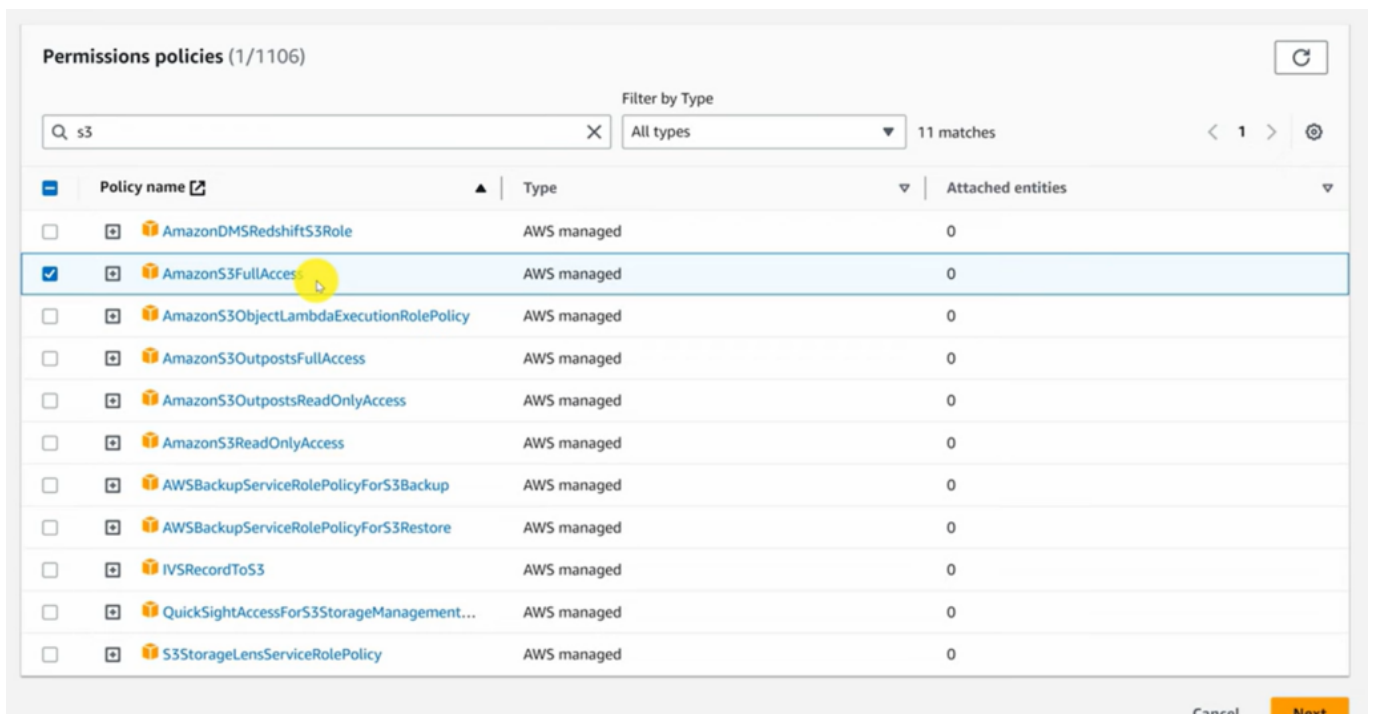
The screenshot displays the AWS IAM console interface. At the top, there's a 'Console Home' header with a search bar and buttons for 'Reset to default layout' and '+ Add widgets'. Below this, the 'Recently visited' section lists services like Launch Wizard, S3, CloudFormation, Support, Billing, EC2, and VPC. The 'Welcome to AWS' section offers links for 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. The main content area shows the 'IAM > Users' page, which includes a search bar for finding users by username or access key, a table with columns for User name, Groups, Last activity, MFA, Password age, and Active key age, and an 'Add user' button. The left sidebar contains navigation options for Identity and Access Management (IAM), including Dashboard, Access management, User groups, Users, Roles, Policies, Identity providers, Account settings, Access reports, and Access analyzer.

2. Configuración de Permisos

Para que Saturn Studio pueda gestionar tus archivos, el usuario debe tener acceso al servicio de almacenamiento:

- En la pantalla de **Set permissions**, selecciona la opción **Attach policies directly**.
- En el buscador de políticas, escribe **s3** y selecciona la casilla de **AmazonS3FullAccess**.

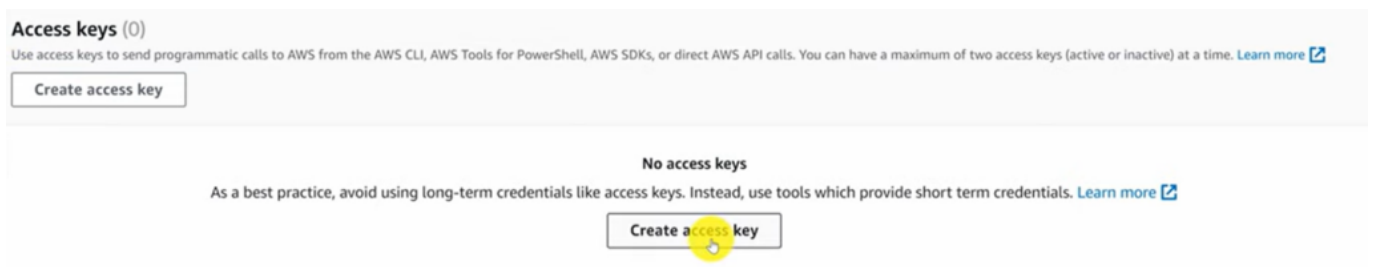
- Finaliza el proceso de creación del usuario.



3. Generación de las Access Keys

Una vez que el usuario esté creado, accede a su perfil para generar las llaves:

- Haz clic en la pestaña **Security credentials**.
- Desplázate hasta **Access keys** y haz clic en **Create access key**.
- Selecciona el caso de uso **Other** y asigna una descripción opcional (ej. Saturn_S3_Key).
- **IMPORTANTE:** Se mostrarán tu **Access key** y tu **Secret access key**. Debes copiarlas y pegarlas directamente en el módulo, ya que la llave secreta no volverá a mostrarse por seguridad.



IAM > Users > dev-user > Create access key



Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 AKIA3YGORTXJWJ23Z2BW	 ***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [Best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

Vinculación en Saturn Studio

Para finalizar la configuración, abre la ventana de credenciales del módulo de **AWS S3** y completa los campos:

1. **Credential name:** Ingresa un nombre para identificar esta conexión.
2. **AWS Region:** Selecciona la región donde se encuentran tus buckets de S3 (ej. us-east-1).
3. **AWS Access Key ID:** Pega el identificador de acceso que copiaste de la consola de AWS.
4. **AWS Secret Access Key:** Pega la llave secreta correspondiente.
5. Haz clic en el botón **Check** para verificar que los permisos de S3 sean válidos.
6. Presiona **Save credentials** para guardar los cambios.

[Saturn Studio – Credencial Asana](#)



Asana es una plataforma líder en gestión de trabajo y productividad diseñada para ayudar a los equipos a organizar tareas, proyectos y objetivos en un solo lugar. Al integrarla con **Saturn Studio**, puedes automatizar la creación de tareas y la gestión de flujos de trabajo colaborativos de manera eficiente.

Cómo obtener el Access Token

Para vincular Asana con Saturn Studio, necesitas generar un token de acceso desde la consola de desarrolladores de Asana siguiendo estos pasos:

1. Acceso desde Saturn Studio

- Dentro del módulo de Asana en Saturn Studio, localiza y haz clic en el enlace azul **“here”** ubicado debajo del campo de Access Token.
- Esto te redirigirá automáticamente al portal de **Asana para desarrolladores**.
- Inicia sesión con tu cuenta de Asana o regístrate si aún no posees una.

PARA DESARROLLADORES Y EQUIPOS

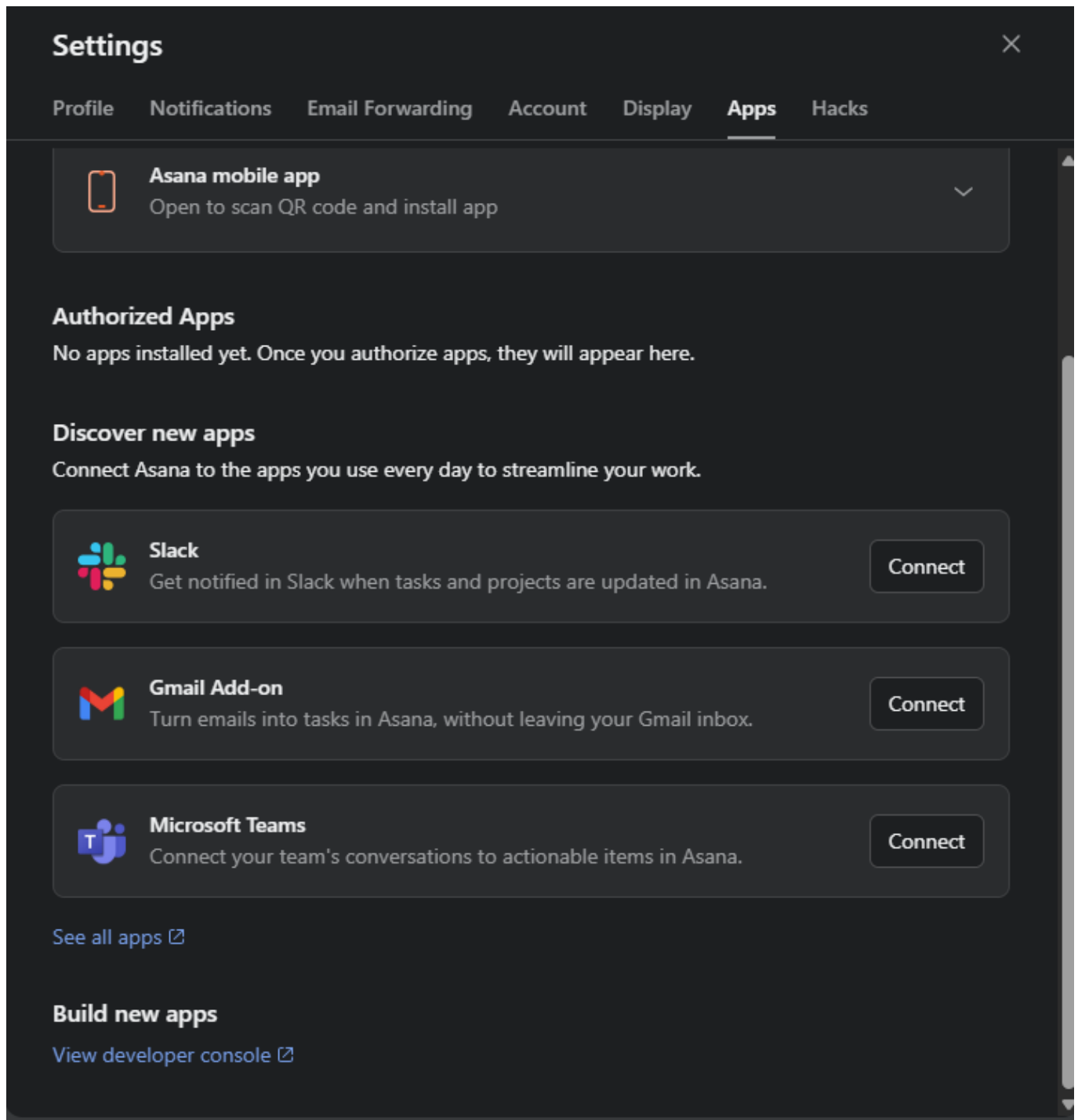
Una solución que ofrece claridad, conectividad y colaboración

Crea aplicaciones que se integren con Asana para conectar a los equipos con el trabajo.

Comenzar

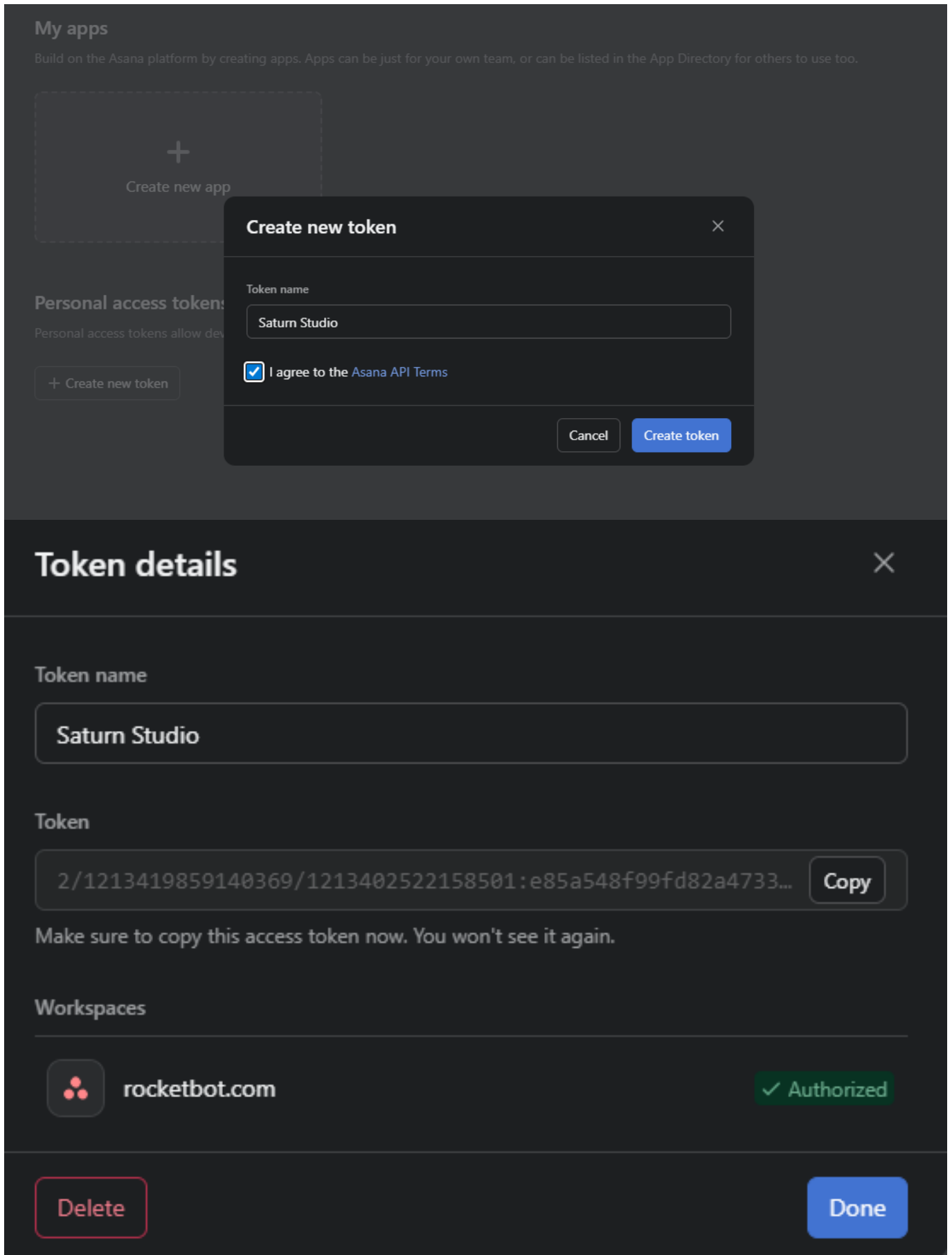
2. Navegación a la Consola de Desarrolladores

- Haz clic en tu **foto de perfil** para acceder a la **Configuración** (Settings).
- Dirígete a la pestaña de **Apps** (Aplicaciones).
- Al final de esta sección, busca y haz clic en el enlace **“View developer console”** (Ver consola de desarrolladores).



3. Generación del Access Token

- En la sección de **Personal Access tokens**, haz clic en el botón **“+ Create new token”**.
- **Token name:** Asigna un nombre descriptivo para identificar el token, por ejemplo, **“Saturn Studio”**.
- Marca la casilla de aceptación de los términos y haz clic en **“Create token”**.
- **Importante:** Se mostrará el token generado. Haz clic en **“Copy”** para copiarlo y guárdalo en un lugar seguro, ya que no se volverá a mostrar completo.



Vinculación en Saturn Studio

Regresa a Saturn Studio para finalizar la configuración de la credencial:

1. En el campo **Credential name**, ingresa un nombre para identificar esta conexión.
2. En el campo **Access Token**, pega el código alfanumérico que acabas de copiar desde Asana.
3. Haz clic en el botón azul **“Connect to Asana”**.
4. Si los datos son correctos, el sistema mostrará un recuadro verde con el mensaje **“Success! connected successfully”** y el estado cambiará a **Connected**.
5. Finalmente, haz clic en **“Save credentials”**.

The screenshot shows the Saturn Studio interface with a modal dialog for configuring Asana credentials. The dialog has a dark header with the Asana logo and the text 'Asana Credentials' and 'Configure Asana credentials to use the module'. Below the header, there is a 'Credential name' field with a question mark icon, containing the text 'Test'. A small note below the field says 'Put a name that will help you identify the credential in the future'. Underneath is the 'Access Token' field, which is currently empty and masked with dots. A link 'here' is provided for getting credentials. A green 'Connected' button with a checkmark is visible. Below that, a green message box says 'Success! connected successfully.'. At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Save credentials', with the latter being highlighted in green. The background shows a 'List Workspaces' page with a 'Config' tab selected and a sidebar with an 'Asana' section.

[Saturn Studio – Asana](#)



Asana es una plataforma de gestión de trabajo y productividad que permite organizar tareas, proyectos y objetivos en un solo lugar. Su integración facilita la coordinación de equipos, el seguimiento de plazos y la centralización de la información para mejorar el rendimiento colectivo.

¿Cómo obtener las credenciales? [Haz clic aquí](#)

Descripción de los Comandos

Listar Espacios de Trabajo (List Workspaces)

Enumera todos los espacios de trabajo disponibles en la cuenta vinculada de Asana.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para listar los espacios de trabajo.	credential
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: <pre>{“Workspaces”:[{“id”:“9034647833”, “gid”:“9034647833”, “name”:“SaturnStudio”}]}</pre>	{var}

Listar Todas las Tareas (List All Tasks)

Muestra todas las tareas contenidas en un espacio de trabajo o proyecto específico.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para listar las tareas.	credential
Workspace	Espacio de trabajo desde el cual se listarán las tareas.	My Workspace
Project	Proyecto desde el cual se listarán las tareas (opcional).	Marketing Project
Task Status	Permite filtrar las tareas según su estado de finalización.	

Parámetro	Descripción	Ejemplo
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: [{"id": "12034567890", "name": "name", "completed": "true", "due_on": "2023-12-31", "assignee": "John Doe", "assignee_id": "1122334455"}]	{var}

Obtener Espacio de Trabajo (Get Workspace)

Recupera los detalles técnicos de un espacio de trabajo específico.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para obtener los detalles.	credential
Workspace ID	ID único del espacio de trabajo que se desea consultar.	9034647833
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: { "id": "9034647833", "name": "SaturnStudio", "is_organization": "false", "archived": "false" }	{var}

Obtener Proyecto (Get Project)

Obtiene la información detallada de un proyecto mediante su identificador.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para obtener los detalles del proyecto.	credential
Project ID	ID del proyecto del cual se desea obtener la información.	12034567890
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: { "id": "12034567890", "name": "Project", "archived": "false", "owner": "John Doe" }	{var}

Crear Tarea (Create Task)

Genera una nueva tarea dentro de un espacio de trabajo o proyecto definido.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para crear la tarea.	credential
Workspace	Espacio de trabajo donde se creará la tarea.	My Workspace
Project	Proyecto donde se alojará la tarea.	Marketing Project
Assignee	Usuario a quien se le asignará la tarea.	nombre_usuario
Task Name	Nombre de la tarea a crear.	Revisar diseño de la interfaz
Description	Descripción detallada de las acciones a realizar en la tarea.	Revisar el diseño de la nueva interfaz
Due Date	Fecha de vencimiento (Formato YYYY-MM-DD o DD-MM-YYYY).	31-12-2026
Custom Fields	Campos personalizados en formato JSON.	mi_valor

Parámetro	Descripción	Ejemplo
Assign result to variable	Variable donde se almacenará el resultado de la creación.	{var}

Obtener Campos Personalizados de Proyecto (Get Project Custom Fields)

Extrae los campos personalizados configurados para un proyecto en particular.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para la consulta.	credential
Workspace	Espacio de trabajo desde donde se obtendrán los campos.	My Workspace
Project	Proyecto del cual se extraerán los campos personalizados.	Marketing Project
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: [{"name": "Priority", "id": "12034567890", "type": "enum", "options": [{"id": "111", "name": "High"}]}	{var}

Obtener Tarea (Get Task)

Recupera los detalles completos de una tarea específica mediante su ID.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para obtener los detalles.	credential
Task ID	ID único de la tarea a consultar.	12034567890
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: {"id": "12034567890", "name": "Nombre de la tarea", "notes": "descripción", "due_on": "2023-12-31", "assignee": "John Doe", "assignee_id": "1122334455", "project_id": "9988776655"}	{var}

Eliminar Tarea (Delete Task)

Borra de forma permanente una tarea de la plataforma.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para eliminar la tarea.	credential
Task ID	ID de la tarea que se desea borrar.	12034567890
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: {"success": true, "task_id": "12034567890", "message": "Tarea eliminada con éxito"}	{var}

Listar Usuarios (List Users)

Obtiene el listado de todos los usuarios vinculados a un espacio de trabajo.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para listar los usuarios.	credential
Workspace	Espacio de trabajo desde el cual se listarán los usuarios.	My Workspace
Project	Proyecto desde el cual se listarán los usuarios (opcional).	Marketing Project
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: {"id":"1122334455","name":"John Doe"}	{var}

Actualizar Tarea (Update Task)

Modifica los datos de una tarea ya existente.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para la actualización.	credential
Task ID	ID único de la tarea a modificar.	1212531686562092
Task Name	El nuevo nombre para la tarea.	Nuevo nombre de tarea
Workspace	Espacio de trabajo donde se ubica la tarea.	My Workspace
Project	Proyecto donde se encuentra la tarea.	Marketing Project
Assignee	Nuevo usuario asignado a la tarea.	nombre_usuario
Description	Nueva descripción para la tarea.	Nueva descripción
Due Date	Nueva fecha de vencimiento (Formato DD-MM-YYYY).	31-12-2026
Custom Fields	Campos personalizados en formato JSON.	mi_valor
Mark as Completed	Marca la tarea como completada automáticamente.	
Assign result to variable	Variable donde se almacenará el resultado de la actualización.	{var}

Cargar Archivo Adjunto a Tarea (Upload Attachment to Task)

Sube un archivo local y lo adjunta a una tarea específica en Asana.

Parámetro	Descripción	Ejemplo
Credential	Credencial de Asana necesaria para cargar el archivo.	credential
Task ID	ID de la tarea a la cual se le subirá el adjunto.	12034567890
Workspace	Espacio de trabajo donde se ubica la tarea.	My Workspace
Project	Proyecto donde se encuentra la tarea.	Marketing Project
Local File	Archivo local que se desea cargar como adjunto.	seleccionar archivos

Parámetro	Descripción	Ejemplo
Assign result to variable	Variable donde se almacenará el resultado. Ejemplo: {"id": "1212461679463197", "resource_type": "attachment", "name": "Nombre del archivo"}	{var}

Saturn Studio – AWS Bedrock



Amazon Bedrock

Amazon Bedrock es una plataforma integral, segura y flexible para construir aplicaciones y agentes de inteligencia artificial generativa. Esta herramienta facilita el acceso a modelos fundacionales de alto rendimiento, permitiendo escalar soluciones de IA de manera eficiente y protegida.

¿Cómo obtener las credenciales? [Haz clic aquí](#)

Descripción de los Comandos

Listar modelos (List models)

Enumera los modelos de AWS Bedrock disponibles en tu cuenta.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de AWS necesaria para listar los modelos.	Mis credenciales de AWS
Asignar resultado a variable	Variable donde se almacenará el resultado de la operación.	{var}

Completado de texto (Text completion)

Completa un texto utilizando un modelo específico de Bedrock.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de AWS necesaria para completar el texto.	Mis credenciales de AWS
Modelo	Modelo de Bedrock seleccionado para completar el texto.	Model
Texto	El texto inicial que se desea completar.	Texto
Máximo de tokens	Cantidad máxima de tokens que el modelo puede generar.	300
Temperatura	Controla la creatividad del modelo. Valores altos lo hacen más creativo y valores bajos más conservador. Rango entre 0 y 2 (por defecto 1.1).	1.1
URL de imagen	URL de la imagen que se enviará al modelo para su procesamiento.	https://www.example.com/image.jpg
Formato de respuesta	Define el formato en el que se entregará la respuesta del modelo.	Texto
Retornar uso de tokens	Si se activa, la respuesta incluirá el número exacto de tokens utilizados.	
Asignar resultado a variable	Variable donde se almacenará el resultado obtenido del modelo.	{var}

[Saturn Studio – Credencial AWS Bedrock](#)



Amazon Bedrock

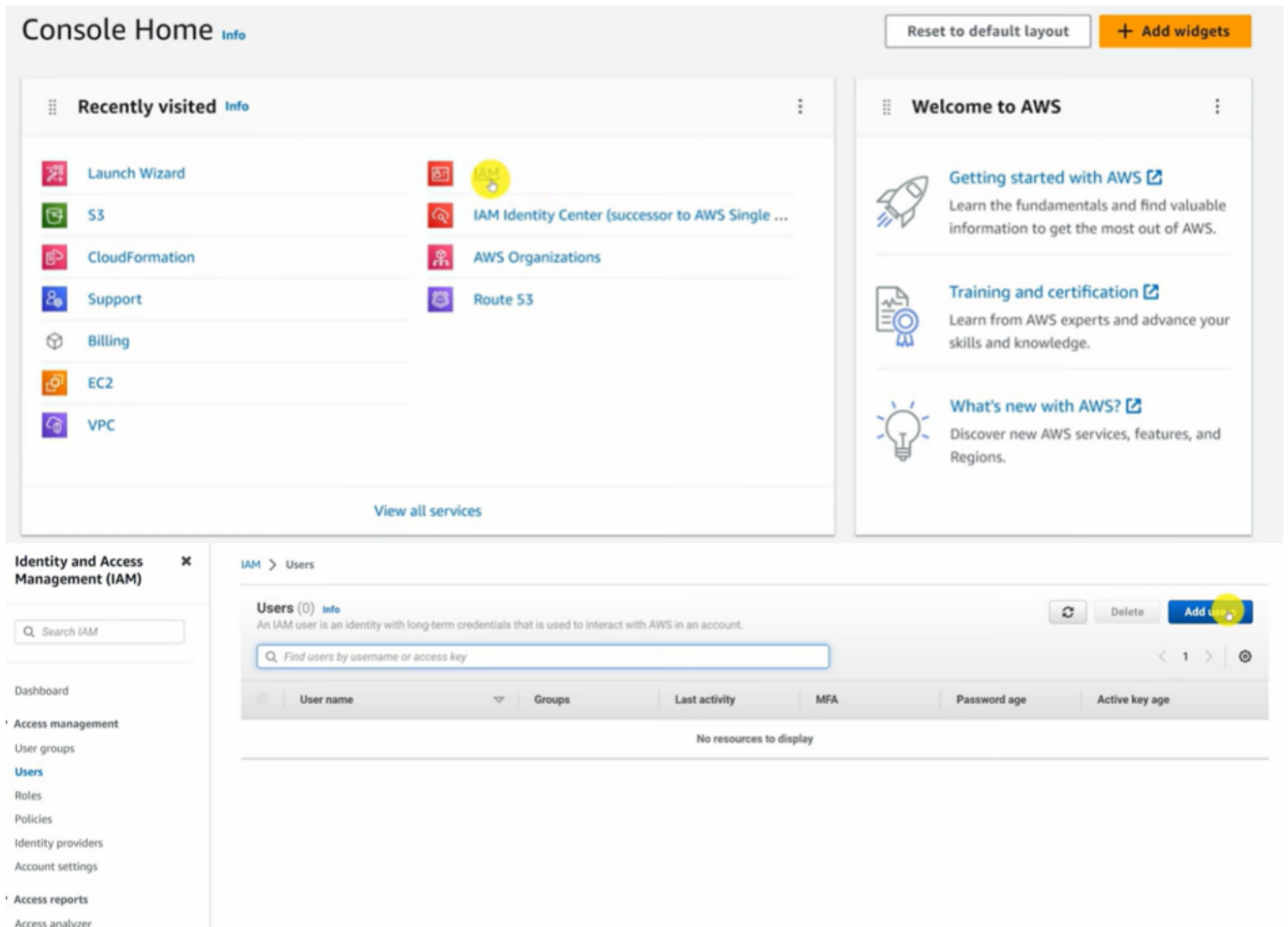
Amazon Bedrock es un servicio de infraestructura de AWS que facilita el acceso a modelos fundacionales de alto rendimiento para construir aplicaciones de IA generativa. Para conectar este servicio con **Saturn Studio**, es necesario configurar un usuario de **IAM (Identity and Access Management)** con los permisos adecuados para garantizar una comunicación segura y eficiente.

Cómo obtener las llaves de acceso (IAM)

El proceso de autenticación en AWS se realiza mediante un par de llaves: la **Access Key ID** y la **Secret Access Key**. Sigue estos pasos para generarlas correctamente:

1. Creación del Usuario en la Consola de AWS

- Inicia sesión en tu Consola de administración de AWS.
- En el buscador de servicios o en el panel de inicio, selecciona **IAM**.
- En el menú lateral izquierdo, dirígete a **Users** (Usuarios) y haz clic en el botón **Add Users**.



2. Configuración de Permisos

Para que Saturn Studio pueda interactuar con Bedrock, el usuario debe tener permisos activos:

- En la pantalla de **Set permissions**, selecciona la opción **Attach policies directly**.
- En el buscador de políticas, escribe **AmazonBedrockFullAccess** y marca la casilla correspondiente para adjuntarla al usuario.
- Finaliza la creación del usuario.

3. Generación de las Access Keys

Una vez creado el usuario, entra a su perfil detallado:

- Haz clic en la pestaña superior **Security credentials**.
- Desplázate hacia abajo hasta la sección **Access keys** y haz clic en **Create access key**.
- En el formulario de selección de caso de uso, elige **Other** y presiona Next.
- Define un nombre para el **Description tag value** (por ejemplo: my developer keys) y haz clic en **Create access key**.
- **IMPORTANTE:** Se mostrarán tu **Access key** y tu **Secret access key**. Debes

copiarlas y pegarlas directamente en el módulo de Saturn Studio, ya que por seguridad el sistema no volverá a mostrar la llave secreta después de cerrar esta ventana.

The screenshot shows the AWS IAM console interface. At the top, there are navigation tabs: 'Permissions', 'Groups', 'Tags', 'Security credentials', and 'Access Advisor'. The 'Security credentials' tab is selected and highlighted with a yellow circle. Below the tabs, there is a section titled 'Permissions policies (1)' with a sub-header 'Permissions are defined by policies attached to the user directly or through groups.' Below this, there is a section for 'Access keys (0)' with a 'Create access key' button. A message states 'No access keys' and provides a 'Create access key' button. The main content area shows the 'Retrieve access keys' step of the 'Create access key' process. It displays the 'Access key' as 'AKIA3YGORTXJWJ23ZZBW' and the 'Secret access key' as a masked string with a 'Show' button. Below this, there are 'Access key best practices' listed as bullet points. At the bottom right, there are buttons for 'Download .csv file' and 'Done'.

Vinculación en Saturn Studio

Regresa al módulo de **AWS Bedrock** en Saturn Studio y completa los campos en la ventana de credenciales con las variables completas:

1. **Credential name:** Ingresa un nombre identificador, por ejemplo: My AWS Bedrock Credential.
2. **AWS Access Key ID:** Pega el identificador de acceso que generaste, por ejemplo: AKIA3YGORTXJWJ23ZZBW.
3. **AWS Secret Access Key:** Pega la llave secreta correspondiente que acabas de copiar de la consola de AWS.
4. **AWS Region:** Selecciona la región donde tienes habilitados tus modelos de Bedrock, por ejemplo: us-east-1.

5. Haz clic en el botón **Check** para verificar que los datos sean correctos.
6. Si el estado es exitoso, presiona **Save credentials**.

Saturn Studio – Evolution API WhatsApp



Evolution API es una solución de código abierto y gratuita diseñada para integrar **WhatsApp** en diversas aplicaciones. Permite realizar una gestión avanzada de la comunicación, incluyendo el envío y recepción de mensajes, la administración de múltiples instancias y otras funcionalidades potentes para la automatización de flujos de trabajo.

Descripción de los Comandos

Crear Instancia (Create instance)

Genera una nueva instancia de WhatsApp dentro de Evolution API para comenzar a operar.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API configurada para la creación de instancias.	credencial
Nombre de la instancia	Identificador único para reconocer la instancia de WhatsApp.	mi_instancia_whatsapp
Token (opcional)	Clave de seguridad adicional para la autenticación de la instancia.	token_secreto
URL del Webhook (opcional)	Dirección URL del servidor que recibirá las notificaciones de eventos.	https://mi-servidor.com/webhook
Asignar resultado a variable	Variable donde se almacenará el resultado de la operación.	{var}

Conectar a WhatsApp (Connect to WhatsApp)

Conectar la instancia a WhatsApp y generar código QR

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API necesaria para la conexión.	credencial
Nombre de la instancia	Nombre de la instancia creada previamente que se desea conectar.	mi_instancia_whatsapp
Asignar resultado a variable	Variable donde se almacenará el estado de la conexión. Ejemplo: {var}	{var}

Obtener Código QR (Get QR code)

Genera y recupera el código QR necesario para realizar el escaneo desde la aplicación de WhatsApp.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API autorizada.	credencial
Nombre de la instancia	Nombre de la instancia para la cual se requiere el código QR.	mi_instancia_whatsapp
Asignar resultado a variable	Variable donde se almacenará la imagen o el código QR generado. Ejemplo: {var}	{var}

Enviar Mensaje (Send message)

Envía un mensaje de texto simple a un destinatario específico.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API necesaria para el envío.	credencial
Nombre de la instancia	Nombre de la instancia conectada y activa.	mi_instancia_whatsapp
Número de teléfono	Número de destino con código de país (sin el signo +).	5491123456789
Mensaje	Texto informativo o contenido que se desea enviar.	Hola, ¿cómo estás?
Asignar resultado a variable	Variable donde se almacenará la confirmación del envío. Ejemplo: {var}	{var}

Esperar Mensaje (Wait for message)

Mantiene el flujo en espera hasta recibir un mensaje entrante a través del webhook configurado.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API para la recepción.	credencial
Asignar resultado a variable	Variable donde se guardará el contenido del mensaje recibido. Ejemplo: {var}	{var}

Descargar Media (Download media)

Permite bajar archivos multimedia (fotos, audios, documentos) contenidos en un mensaje.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API autorizada.	credencial
Nombre de la instancia	Instancia desde la cual se descargará el archivo.	mi_instancia_whatsapp
Clave del mensaje	Identificador único del mensaje que contiene el archivo multimedia.	mensaje_key_123
Asignar resultado a variable	Variable que contendrá el archivo descargado para su uso posterior. Ejemplo: {var}	{var}

Enviar Media (Send media)

Envía archivos como imágenes, videos o documentos a un contacto.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API para la transferencia.	credencial
Nombre de la instancia	Instancia conectada que realizará el envío.	mi_instancia_whatsapp
Número de teléfono	Número de WhatsApp del destinatario.	5491123456789
Variable del archivo	Variable de Saturn Studio que contiene el archivo a transmitir.	archivo_variable
Leyenda (opcional)	Texto descriptivo o pie de foto que acompaña al archivo.	Descripción del archivo
Asignar resultado a variable	Variable donde se almacenará el estado del envío multimedia. Ejemplo: {var}	{var}

Información de Instancia (Instance info)

Consulta el estado actual y los detalles de configuración de una instancia.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API autorizada.	credencial
Nombre de la instancia	Nombre de la instancia de la cual se desea obtener datos.	mi_instancia_whatsapp

Parámetro	Descripción	Ejemplo
Asignar resultado a variable	Variable donde se almacenará el objeto con la información. Ejemplo: {var}{var}	

Eliminar Instancia (Delete instance)

Borra de forma definitiva una instancia de WhatsApp del servidor.

Parámetro	Descripción	Ejemplo
Credencial	Credencial de Evolution API configurada.	credencial
Nombre de la instancia	Nombre de la instancia que se desea remover.	mi_instancia_whatsapp
Asignar resultado a variable	Variable donde se guardará la respuesta de la eliminación. Ejemplo: {var}	{var}

Configuración de Credenciales

Para utilizar este módulo en Saturn Studio, debe crear una credencial de tipo **Evolution API** completando los siguientes datos:

- **URL Base de la API:** Dirección completa de su servidor (ej. <https://api.evolution-api.com>).
- **API Key:** Clave de seguridad proporcionada por la plataforma para la autenticación de las peticiones.

Instalación de Evolution API

Si desea alojar su propio servidor, puede realizar una instalación rápida utilizando Docker mediante el siguiente comando:

```
docker run -d \
  --name evolution_api \
  -p 8080:8080 \
  -e AUTHENTICATION_API_KEY=su-api-key-aqui \
  atendai/evolution-api:latest
```

Ejemplos de uso

1. Crear y conectar una instancia

1. Usar el comando "Crear instancia" con el nombre "mi_bot"
2. Usar el comando "Conectar a WhatsApp" para generar el QR
3. Usar el comando "Obtener código QR" para mostrar el QR al usuario
4. Escanear el QR con WhatsApp

2. Enviar un mensaje

Una vez conectada la instancia, usar el comando "Enviar mensaje" con:

- Nombre de instancia: "mi_bot"
- Número: "5491123456789"
- Mensaje: "¡Hola! Este es un mensaje desde Evolution API"

3. Recibir mensajes

Usar el comando "Esperar mensaje" para recibir mensajes entrantes a través de webhooks.

Notas Importantes

- Cada instancia generada debe poseer un **nombre único** para evitar conflictos.
- Los números de teléfono deben ingresarse con el código de país correspondiente, **sin incluir el símbolo +**.
- El uso de webhooks requiere una **URL con acceso público** desde internet.
- El módulo es compatible con las funciones principales de la aplicación y gestiona la sesión de forma similar a WhatsApp Web.

[**Saturn Studio – Credencial Facebook Messenger**](#)



facebook Messenger

¿Qué es la API de Facebook Messenger?

La **API de Facebook Messenger** permite que **Saturn Studio** se integre con tus **Páginas de Facebook** para automatizar la gestión de mensajes entrantes y salientes. Esta conexión es fundamental para crear bots de respuesta automática, gestionar atención al cliente y enviar notificaciones directas a través de la infraestructura de Meta.

Cómo obtener el Facebook Page Access Token

Para obtener este token, es indispensable ser administrador de la página de Facebook que deseas vincular y seguir estos pasos técnicos en la plataforma de Meta:

1. Preparación en Meta for Developers

- Inicia sesión en tu cuenta de [Meta for Developers](#).
- Haz clic en **Crear aplicación**.
- **Tipo de aplicación:** Selecciona obligatoriamente el tipo **Negocio** (Business).

2. Configuración de Permisos y Producto

- Una vez dentro de tu app, busca y añade el producto **Messenger**.
- Para generar el token con los permisos correctos, utiliza el **Explorador de API de Gráficos** (Graph API Explorer).
- En la sección de permisos (Scopes), debes añadir y autorizar los siguientes:
 - **pages_show_list:** Obligatorio para ver y seleccionar la página de

Facebook que vas a conectar.

- **pages_messaging**: El permiso principal; permite que el bot envíe y reciba mensajes en nombre de la página.
- **pages_manage_metadata**: Necesario para el comando “Subscribe to Webhook”, permitiendo que la App reciba notificaciones automáticas cuando llega un mensaje.
- **pages_read_engagement**: Recomendado para leer el contenido de los chats y obtener información de los hilos de conversación.

3. Generación del Token de Página

- En el Explorador de API, selecciona la **Página de Facebook** específica para la que necesitas el token.
- Haz clic en generar y obtendrás una cadena larga alfanumérica que comienza con **EA...**
- Copia este token inmediatamente.

Configuración en Saturn Studio

Para vincular tu cuenta, sigue estos pasos dentro del módulo:

1. Abre el módulo de **Facebook Messenger** y haz clic en **New** para crear una nueva credencial.
2. **Credential name**: Ingresa un nombre que te ayude a identificar la página en el futuro.
3. **Facebook Page Access Token**: Pega el código generado anteriormente.
4. Haz clic en el botón **Check** para validar que el token tenga los permisos de administrador necesarios.
5. Si el sistema arroja el mensaje de éxito, haz clic en **Save credentials**.