Append Row to a Spreadsheet — Google Sheets

Descripción

Agrega una fila (o varias) al final de los datos existentes en una hoja específica de un spreadsheet, utilizando la API de Google Sheets.

Parámetros de Entrada

Campo	Tipo	Requerido	Descripción
spreadsheet_id	String	**Sí**	ID del spreadsheet destino.
sheet	Número	**Sí**	ID numérico de la hoja (*gid / tab*) dentro del spreadsheet. *(Nota: No es el nombre visible de la pestaña)*.
value	Variable	**Sí**	Datos a insertar en la hoja. Ver sección *Formato de datos aceptado*.

Formato de Datos Aceptado

El campo value admite los siguientes formatos de datos:

- **Valor simple** (String o Número): Se inserta como una única fila conteniendo una sola celda.
- **Array unidimensional** ([a, b, c]): Se inserta como una sola fila con los valores distribuidos en columnas consecutivas.
- **Array bidimensional** ([[a, b], [c, d]]): Cada sub-array interno representa una nueva fila independiente.

Comportamiento de Inserción

- **Ubicación:** Los datos se insertan siempre inmediatamente después de la última fila que contenga datos dentro de la hoja.
- **Rango de columnas:** Soporta columnas desde la **A** hasta la **Z**.

Respuesta Exitosa

Ejemplo de respuesta (JSON)

```
{
  "spreadsheetId": "1M0d-fTEnS9fWMfuAQFhrUb3sC1f7otykEpifSB4p3ff",
  "tableRange": "'Hoja 1'!A1:C11",
  "updates": {
    "spreadsheetId": "1M0d-fTEnS9fWMfuAQFhrUb3sC1f7otykEpifSB4p3ff",
    "updatedRange": "'Hoja 1'!A12:C12",
    "updatedRows": 1,
    "updatedColumns": 3,
    "updatedCells": 3
  }
}
```

Detalle de Campos de la Respuesta

Campo	Tipo	Descripción
spreadsheetId	String	ID del spreadsheet afectado.
tableRange	String	Rango detectado por Google Sheets como la tabla existente antes de la inserción.
updates.updatedRange	String	Rango exacto de celdas donde se escribieron los nuevos datos.
updates.updatedRows	Número	Cantidad de filas insertadas.
updates.updatedColumns	Número	Cantidad de columnas escritas.
updates.updatedCells	Número	Cantidad total de celdas modificadas.

Manejo de Errores

Todos los casos de error devuelven una estructura con el formato:

```
{
  "status": false,
  "data": "<mensaje_de_error>"
}
```

Casos de Error

Situación	Mensaje devuelto en data
Falta el parámetro spreadsheet_id	"Spreadsheet id is required"
Falta el parámetro sheet	"Sheet not found"
Falta el parámetro value	"Values not found"
El valor de sheet no existe en el spreadsheet	"Sheet not found"
spreadsheet_id es inválido	Mensaje de error retornado directamente por la API de Google Sheets.
Sin permisos de acceso sobre el archivo	Mensaje de error retornado directamente por la API de Google Sheets.

Saturn Studio – Credencial/Webhook WhatsApp Business

Esta guía explica cómo configurar y obtener las credenciales necesarias de **WhatsApp Business API** en la plataforma de Meta for Developers para integrarlas en **Saturn Studio**, así como la suscripción correcta a su Webhook.

1. Configuración y Obtención de Credenciales

Para vincular tu cuenta de WhatsApp Business con Saturn Studio necesitas dos datos principales:

- **WhatsApp Business Account ID**
- **WhatsApp Business Access Token** *(el cual autocompletará el número de teléfono vinculado una vez verificado)*.

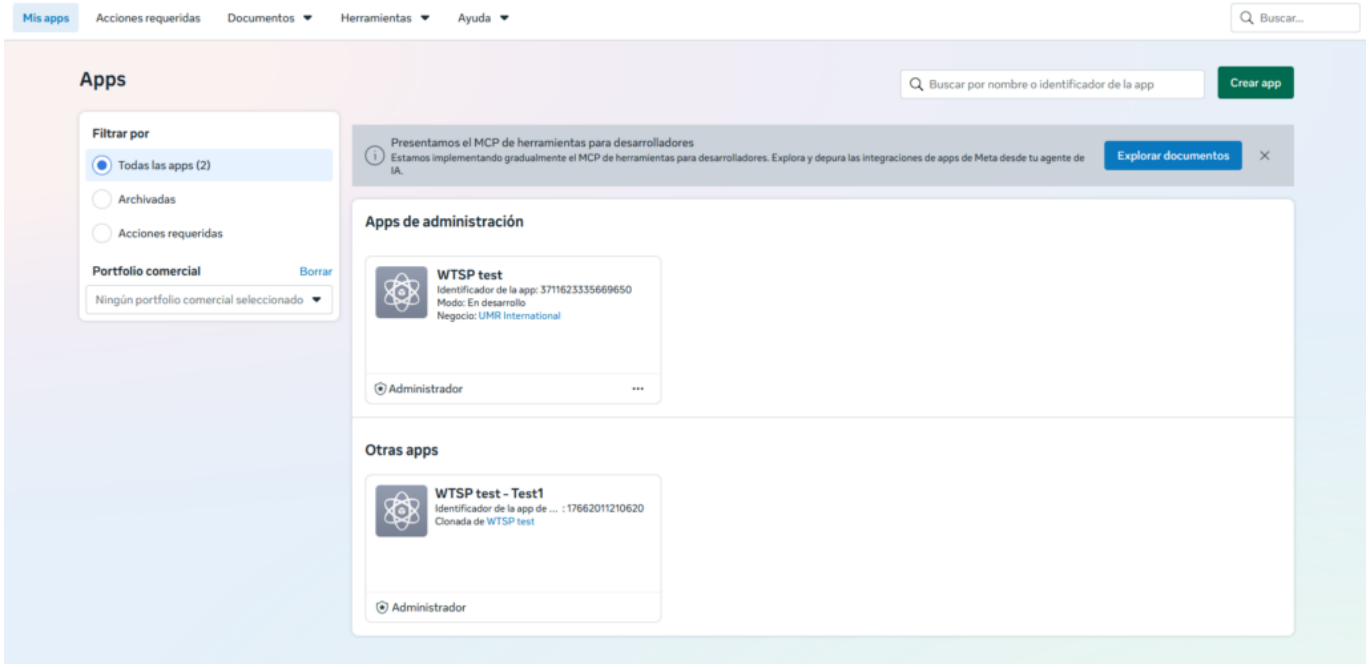
Requisitos Previos en Meta

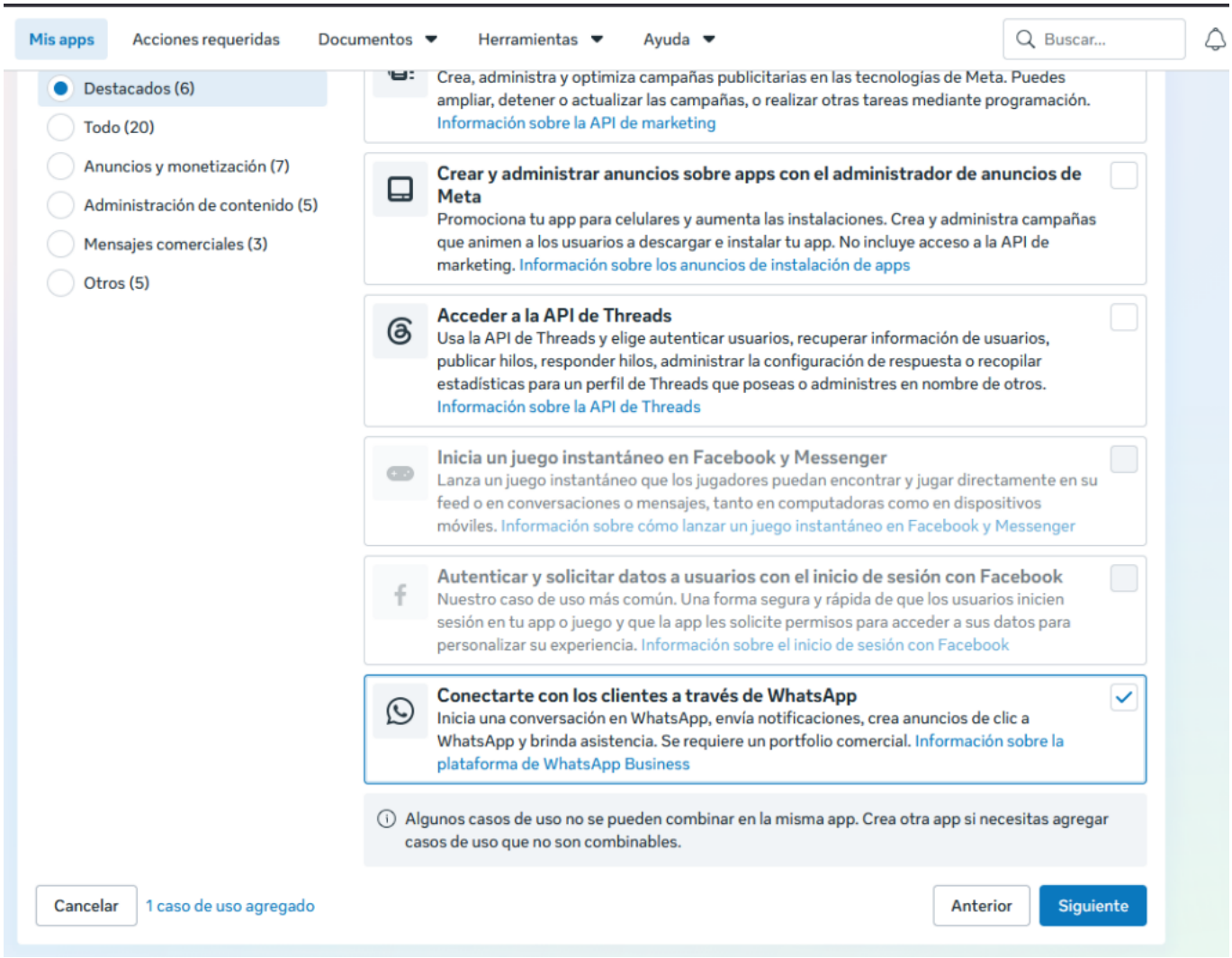
1. Contar con una cuenta de **Facebook**.
2. Estar registrado como desarrollador en Meta for Developers.
3. Disponer de un **Portafolio de Negocios** (*Meta Business Account*) activo.

Paso a Paso para Crear la App en Meta

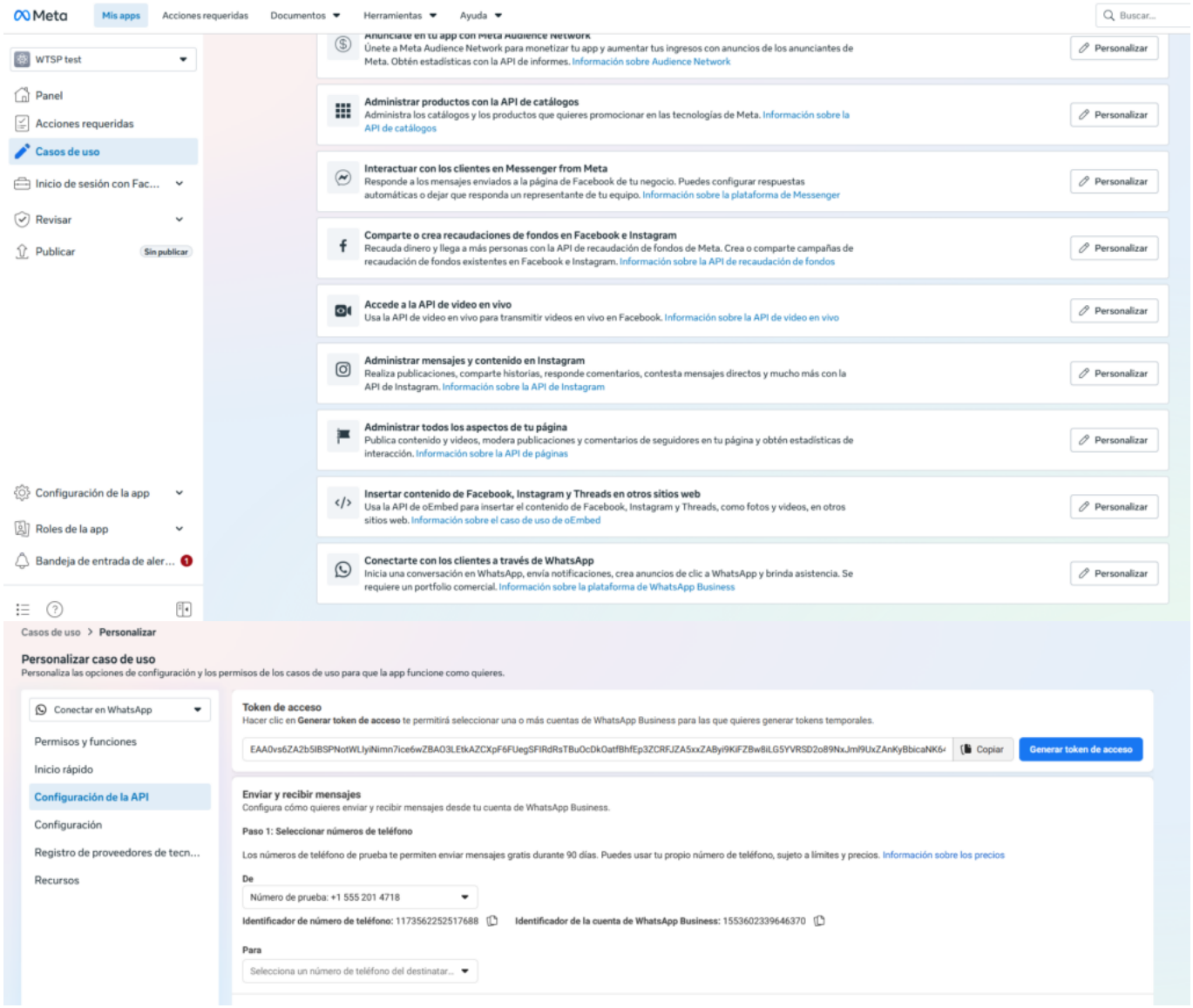
- **Creación de la App:**
  - Ve a **Mis Apps** > **Crear App**.
  - Ingresa el nombre de la aplicación y tu correo electrónico de contacto.
  - En **Casos de Uso**, selecciona la opción **Conectarse con los clientes a través de WhatsApp** (esto habilitará los permisos y accesos de la API).

- Vincula la app a tu Portafolio de Negocios (*Business Account*).
- Revisa el resumen y haz clic en **Crear**.



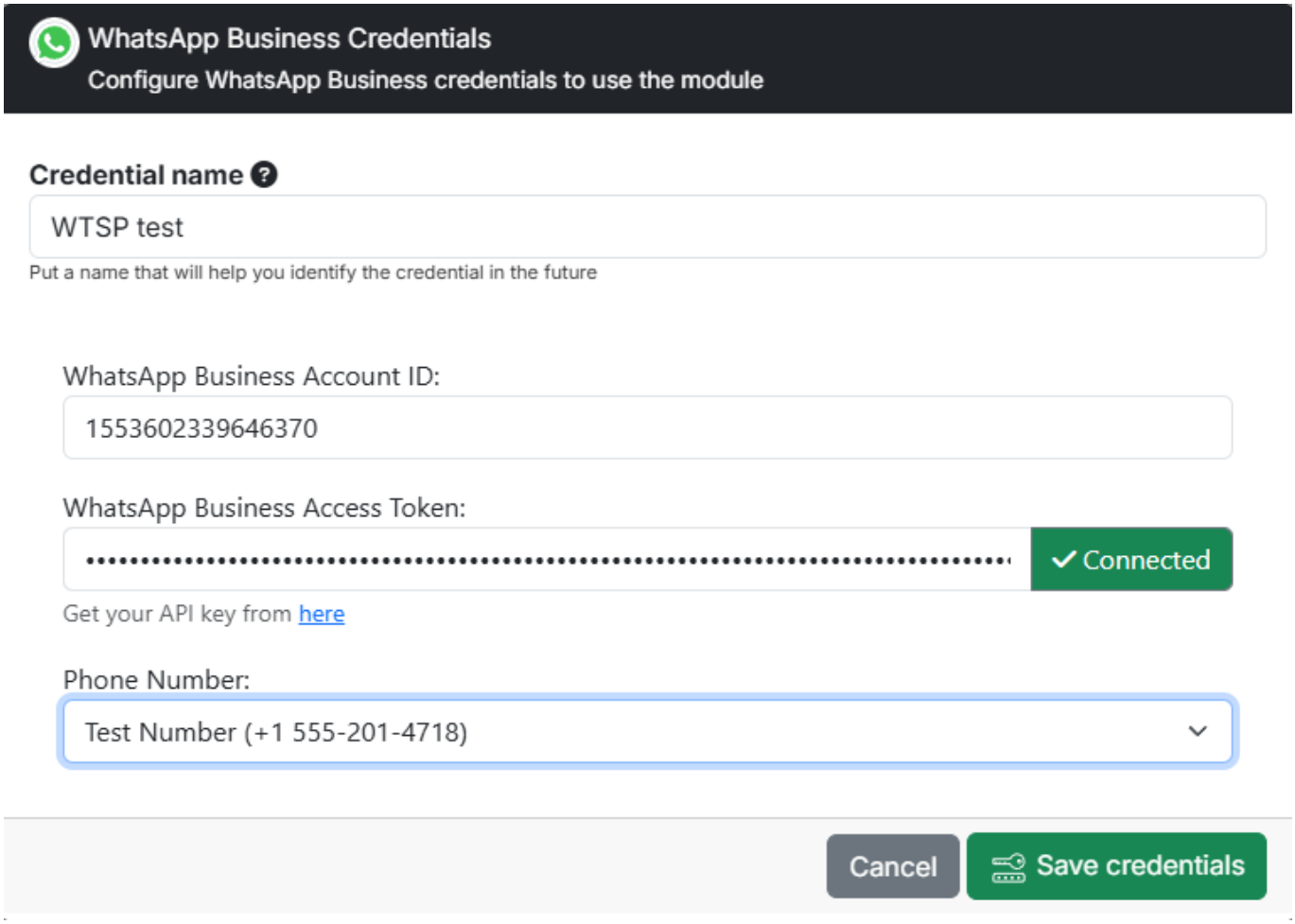


- **Obtención de Token e IDs:**
  - Entra a la App creada y dirígete a **Casos de Uso** > **Conectarse con los clientes a través de WhatsApp** > **Personalizar**.
  - En el menú lateral, selecciona **Configuración de la API**.
  - Si no lo has hecho, vincula o registra tu número de teléfono de prueba/producción.
  - Copia el **WhatsApp Business Account ID** generado en la pantalla.
  - Genera el **Token de acceso** (*Access Token*) y cópialo.



Paso a Paso en Saturn Studio

1. Abre **Saturn Studio** e ingresa al módulo de credenciales de **WhatsApp Business**.
2. Pega el **WhatsApp Business Account ID** y el **WhatsApp Business Access Token** en sus respectivos campos.
3. Haz clic en el botón de **verificación / check** al lado del campo *WhatsApp Business Access Token*.
4. Si la autenticación es correcta, la plataforma validará el token y el campo **Phone Number** se completará automáticamente con el número registrado en Meta.
5. ***Importante: No guardes la credencial sin antes verificar el Access Token haciendo clic en el botón de check, o de lo contrario la credencial no se guardará correctamente.***
6. Guarda la credencial.

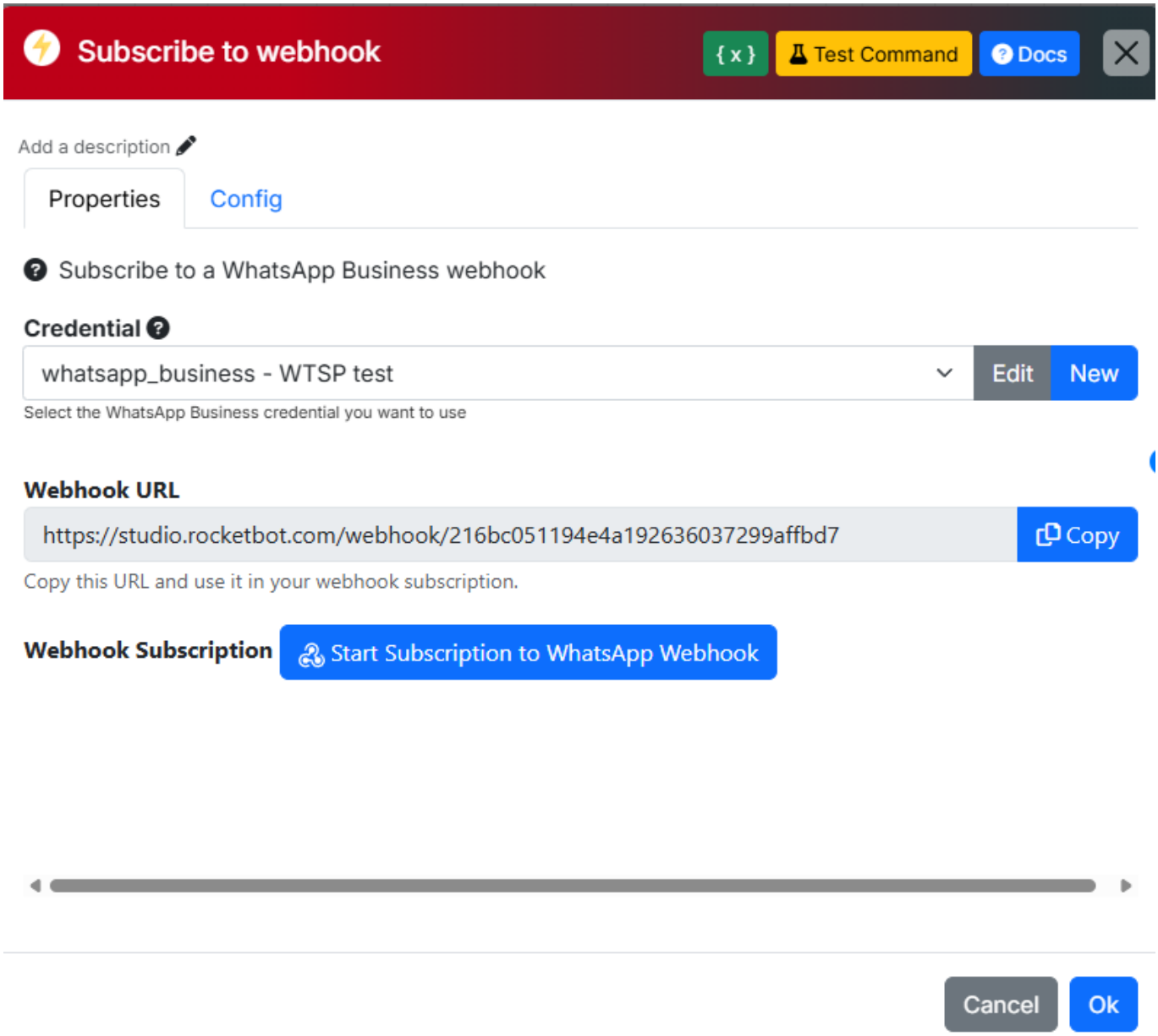


2. Suscripción al Webhook de WhatsApp Business

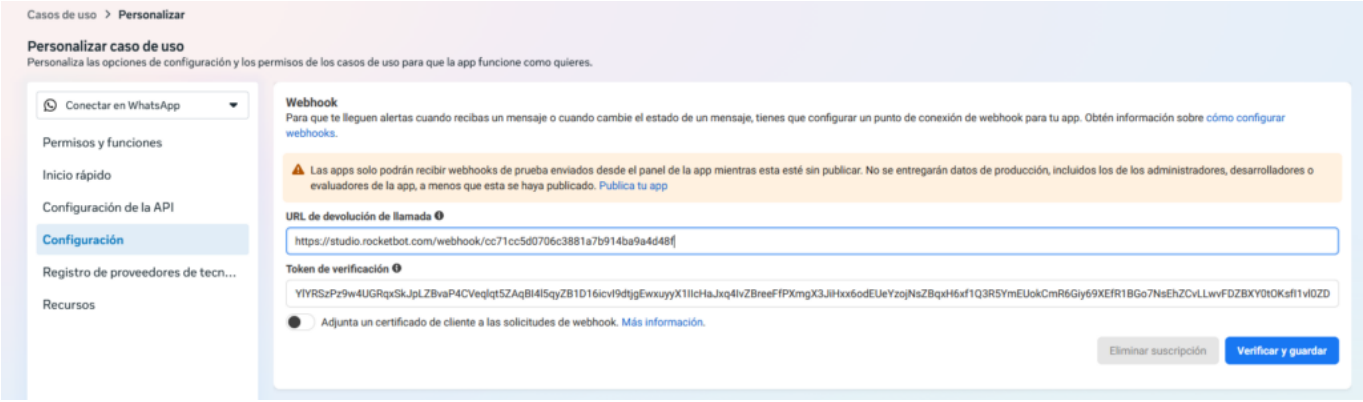
Una vez guardada la credencial, debes configurar la recepción de eventos en tiempo real (mensajes entrantes y estados).

Paso a Paso para la Suscripción al Webhook

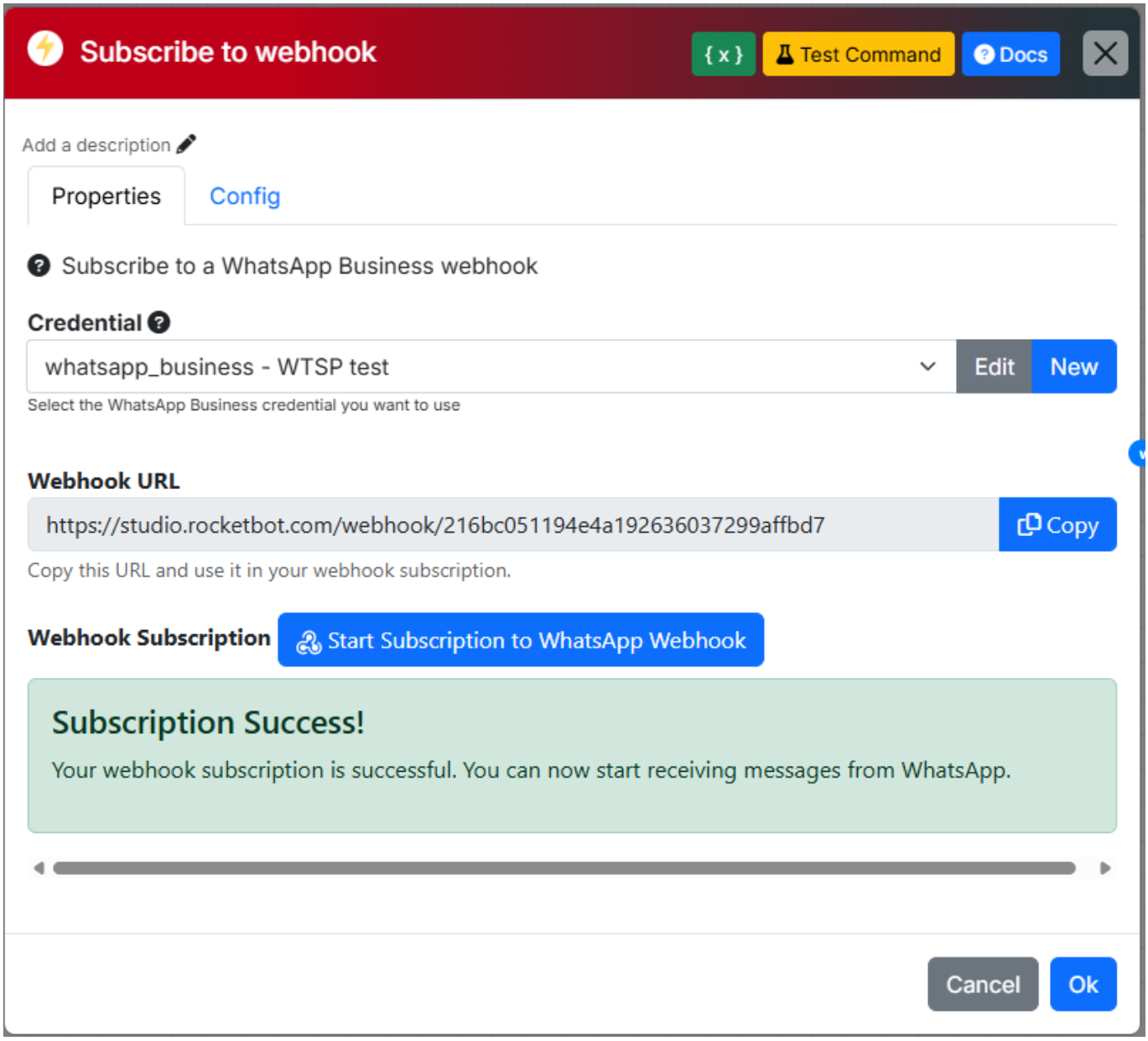
- **Obtener la URL de Webhook en Saturn Studio:**
  - En Saturn Studio, abre el comando o nodo **Subscribe to Webhook**.
  - Selecciona la credencial de WhatsApp Business previamente configurada.
  - Copia la **Webhook URL** que se genera automáticamente.



- **Configurar el Webhook en Meta:**
  - Vuelve al panel de Meta for Developers, dentro de **Casos de uso** > **WhatsApp** > **Configuración** (o *Webhooks*).
  - Cambia el selector de producto a **WhatsApp** (o *WhatsApp Business Account*).
  - En el campo **URL de devolución de llamada**, pega la *Webhook URL* copiada de Saturn Studio.
  - En **Token de verificación**, ingresa el token correspondiente a tu configuración de Saturn.



- **Verificación Simultánea:**
  - En Saturn Studio, haz clic en el botón **Start Subscription to WhatsApp Webhook** para poner el servidor a la escucha.
  - De forma inmediata, vuelve al panel de Meta y haz clic en **Verificar y guardar**.



Si el proceso fue exitoso, Saturn Studio mostrará el mensaje **Subscription success!** y Meta marcará la URL con un check verde.

> **Importante:** Tras completar la verificación en Meta, ve a la lista de campos de suscripción (*Subscription fields*) y haz clic en **Suscribirse** en la fila correspondiente a **messages** para comenzar a recibir los mensajes entrantes en tus flujos.

Saturn Studio — Recetas Frecuentes

Flujos listos, patrones de diseño y combinaciones de conectores

> **Nota:** Este documento contiene 12 recetas de flujos listos para adaptar, 7 patrones de diseño reutilizables y una guía de combinaciones de conectores frecuentes. Cada receta incluye pasos, variantes y notas de implementación.

Parte 1 — Flujos listos para adaptar

Cada receta describe un flujo completo listo para replicar en Saturn Studio. Adapta los conectores y credenciales a tu entorno.

Inteligencia Artificial + Comunicación

ID	Nombre de la Receta	Conectores	Dificultad	Tiempo	Pasos del Flujo	Variantes	Notas de Implementación
#1	**Clasificador de correos con IA y respuesta automática**	Gmail + OpenAI + Gmail	**Media**	1-2 hs	**1.** Webhook/Cron activa agente. **2.** Gmail: *Read Emails* (no leídos). **3.** OpenAI: *Text Completion* (clasifica). **4.** If/Else por categoría. **5.** Gmail: *Send Email* (respuesta). **6.** Log de acción.	Usar Outlook en lugar de Gmail. Registrar cada correo en una Base de Datos.	El prompt de OpenAI debe incluir las categorías posibles y el formato de respuesta exacta (ej: una sola palabra).

ID	Nombre de la Receta	Conectores	Dificultad	Tiempo	Pasos del Flujo	Variantes	Notas de Implementación
#2	**Resumen diario de documentos PDF por correo**	Gmail + PDF Reader + OpenAI + Gmail	**Media**	2-3 hs	**1.** Trigger programado (8:00 AM). **2.** Gmail: *Read Emails* (adjuntos del día anterior). **3.** PDF Reader: *Extract Text*. **4.** OpenAI: *Text Completion* (resumen). **5.** Gmail: *Send Email* (consolidado).	Enviar resumen por Teams o Discord. Almacenar los resúmenes en Notion.	Limitar el tamaño del texto enviado a OpenAI si los documentos son muy extensos (control de tokens).
#3	**Asistente de atención al cliente por WhatsApp con IA**	WhatsApp (Evolution API) + OpenAI + MySQL	**Alta**	4-6 hs	**1.** Webhook: *Receive Webhook* (mensaje). **2.** MySQL: *Select* (historial previo). **3.** OpenAI: *Text Completion* (historial + mensaje). **4.** MySQL: *Insert* (guarda historial). **5.** WhatsApp: *Send Message*.	Usar Facebook Messenger. Escalar a agente humano si la confianza de la IA es baja.	El historial de conversación debe enviarse como contexto en el prompt para mantener la coherencia de los turnos.

Datos y Reportes

ID	Nombre de la Receta	Conectores	Dificultad	Tiempo	Pasos del Flujo	Variantes	Notas de Implementación
#4	**Conciliación de pagos con alerta por Teams**	MySQL + PostgreSQL + MS Teams + OpenAI	**Media-Alta**	3-4 hs	**1.** Trigger programado (cierre de día). **2.** MySQL: *Select* (ventas). **3.** PostgreSQL: *Select* (ERP contable). **4.** Lógica de comparación. **5.** If/Else (¿discrepancias?). **6.** OpenAI: *Text Completion* (redacta detalle). **7.** Teams: *Send Message*.	Enviar reporte por Gmail. Registrar discrepancias en tabla de auditoría en la BD.	Definir un umbral de tolerancia: diferencias menores a un monto X no generan alerta para evitar ruido innecesario.

ID	Nombre de la Receta	Conectores	Dificultad	Tiempo	Pasos del Flujo	Variantes	Notas de Implementación
#5	**Sincronización de tareas entre formulario y Asana**	Xperience + Asana + Gmail	**Media**	2 hs	**1.** Xperience: *Get Queue Records* (solicitudes). **2.** Asana: *Create Task*. **3.** Asana: *Assign Task* (según tipo). **4.** Gmail: *Send Email* (notifica al solicitante). **5.** Xperience: *Update Record Status* (procesado).	Usar Notion en lugar de Asana. Crear subtareas automáticas según la solicitud.	El Form ID de Xperience debe obtenerse desde el panel del Orquestador antes de configurar el agente.
#6	**Reporte semanal de ventas en Notion**	MySQL + OpenAI + Notion	**Media**	2-3 hs	**1.** Trigger programado (Lunes 7:00 AM). **2.** MySQL: *Select* (ventas de la semana por categoría). **3.** OpenAI: *Text Completion* (análisis de desempeño). **4.** Notion: *Create Page*. **5.** Notion: *Update Page* (inserta análisis).	Publicar el reporte en Teams o Slack. Agregar gráficos exportados como imágenes.	Para que Notion funcione, cada página o base de datos debe estar compartida explícitamente con la integración de Saturn Studio.

E-commerce y Pagos

ID	Nombre de la Receta	Conectores	Dificultad	Tiempo	Pasos del Flujo	Variantes	Notas de Implementación
#7	**Respuesta automática a preguntas en Mercado Libre**	Webhook + Mercado Libre + OpenAI + Mercado Libre	**Media-Alta**	3-5 hs	**1.** Webhook: *Receive Webhook* (nueva pregunta). **2.** Request HTTP (detalle de publicación). **3.** OpenAI: *Text Completion* (genera respuesta según título/descripción). **4.** Mercado Libre: *Reply Message*. **5.** Log de auditoría.	Aplicar el mismo patrón en Facebook Messenger o WhatsApp.	Revisar que el *Redirect URI* configurado en Mercado Libre Developers apunte exactamente a https://studio.rocketbot.com.
#8	**Emisión automática de factura electrónica**	Webhook + Mercado Pago + Nubox + Gmail	**Alta**	6-8 hs	**1.** Webhook: *Receive Webhook* (pago exitoso). **2.** Mercado Pago: *Check Transaction Status*. **3.** If/Else (solo si está aprobado). **4.** Nubox: *Create Invoice* (datos comprador). **5.** Gmail: *Send Email*. **6.** MySQL: *Insert* (registro interno).	Reemplazar Nubox por otro proveedor de facturación según el país.	Siempre probar en ambiente UAT de Nubox antes de pasar a producción. Las facturas emitidas no se anulan fácilmente.

DevOps y Equipos Técnicos

ID	Nombre de la Receta	Conectores	Dificultad	Tiempo	Pasos del Flujo	Variantes	Notas de Implementación
#9	**Notificación de nuevo Pull Request en Discord**	GitHub (Webhook) + OpenAI + Discord	**Baja-Media**	1 h	**1.** GitHub Webhook: Escucha pull_request. **2.** GitHub: *Get Pull Request* (título, autor, cambios). **3.** OpenAI: *Text Completion* (resumen de cambios). **4.** Discord: *Send Embed* (canal #code-review).	Usar Teams o Slack. Asignar *reviewer* automáticamente según el área modificada.	Configurar el webhook en GitHub Settings > Webhooks. El *Content-Type* debe ser estrictamente application/json.
#10	**Monitoreo de menciones de marca en Reddit**	Reddit + OpenAI + Teams	**Baja**	1 h	**1.** Trigger programado (cada 6 horas). **2.** Reddit: *Search Subreddit* (busca marca). **3.** OpenAI: *Text Completion* (análisis de sentimiento). **4.** If/Else (alerta si es negativo). **5.** Teams: *Send Message*.	Extender monitoreo a LinkedIn o Twitter/X.	Definir una lista de subreddits objetivo (ej: r/Chile, r/startups) para acotar búsquedas y reducir ruido.
#11	**Pipeline de ingesta de documentos a base vectorial (RAG)**	Gmail + PDF Reader + OpenAI + Pinecone	**Alta**	4-6 hs	**1.** Gmail: *Read Emails* (PDFs en carpeta específica). **2.** PDF Reader: *Extract Text*. **3.** OpenAI: *Embedding* (genera vector). **4.** Pinecone: *Upsert Vectors* (+ metadata). **5.** Log de ID procesado.	Usar AWS S3 como fuente de documentos. Combinar con un agente de consulta mediante Pinecone: Query.	Dividir documentos largos en *chunks* de 500-1000 palabras antes de vectorizar para mayor precisión.

ID	Nombre de la Receta	Conectores	Dificultad	Tiempo	Pasos del Flujo	Variantes	Notas de Implementación
#12	**Agendamiento automático de reunión al confirmar lead**	Webhook + MySQL + Zoom + Gmail	**Media**	2-3 hs	**1.** Webhook: Recibe nuevo lead (CRM/web). **2.** MySQL: *Select* (verifica existencia). **3.** If/Else (si es nuevo continúa, si no actualiza). **4.** Zoom: *Create Meeting* (próximos 2 días hábiles). **5.** MySQL: *Insert*. **6.** Gmail: *Send Email* (enlace Zoom).	Usar MS Teams Meeting. Notificar automáticamente al vendedor asignado.	Usar Variables Globales para configurar la duración de la reunión y el anfitrión sin editar el flujo base.

Parte 2 — Patrones de diseño reutilizables

Los patrones son estructuras lógicas que aparecen en casi todos los flujos. Dominarlos permite construir agentes robustos y mantenibles.

Código	Patrón de Diseño	Cuándo usarlo	Estructura Lógica	Ejemplo Real
P1	**Manejo de errores con reintento**	Siempre que un comando pueda fallar por condiciones externas (BD, APIs, correos). Base para producción.	Acción A $\rightarrow$ If/Else (¿OK?) $\rightarrow$ Si Error: esperar N segundos $\rightarrow$ reintentar hasta X veces $\rightarrow$ si persiste: error controlado e informes.	Llamada a API de Mercado Libre que falla por límite de tasa; reintenta 3 veces esperando 5s entre intentos.
P2	**Procesamiento de lista (loop sobre items)**	Cuando el agente debe procesar un conjunto de registros uno por uno (facturas, correos, tareas).	Obtener lista $\rightarrow$ Contador = 0 $\rightarrow$ While (contador < total) $\rightarrow$ Procesar item[contador] $\rightarrow$ contador++ $\rightarrow$ Salida: consolidar.	Leer 50 facturas pendientes de MySQL y emitir cada una por Nubox mediante iteración.
P3	**Enriquecimiento de datos con IA**	Al recibir datos crudos descofigurados (correos, comentarios) que requieren convertirse en datos estructurados.	Recibir dato crudo $\rightarrow$ OpenAI: *Text Completion* (Prompt: extraer X, Y, Z en JSON) $\rightarrow$ Parsear JSON $\rightarrow$ Usar campos en pasos siguientes.	Recibir correo de proveedor en formato libre y extraer automáticamente: monto, vencimiento y número de factura.

Código	Patrón de Diseño	Cuándo usarlo	Estructura Lógica	Ejemplo Real
P4	**Human-in-the-loop (aprobación humana)**	Cuando el agente va a ejecutar acciones irreversibles (emitir facturas, borrar registros, publicar contenido).	Pasos previos $\rightarrow$ Generar resumen $\rightarrow$ Enviar solicitud aprobación (Email/Teams/Xperience) $\rightarrow$ Esperar $\rightarrow$ If aprobado: ejecuta, Else: cancela.	El agente genera la factura y, antes de emitirla, envía un correo al contador esperando su aprobación vía formulario Xperience.
P5	**Caché de credenciales con Variable Global**	Cuando múltiples agentes consumen el mismo servicio externo (misma cuenta Gmail, API Key, Base de Datos).	Crear credencial en Saturn Studio $\rightarrow$ Crear Variable Global que la referencia $\rightarrow$ Usar variable en agentes $\rightarrow$ Actualización centralizada.	10 agentes distintos usan la misma API Key de OpenAI mediante la variable openai_key. Si expira, se cambia en un solo lugar.
P6	**Deduplicación de registros**	Cuando el agente puede recibir el mismo evento repetido (reintentos de webhooks, correos duplicados) para evitar doble proceso.	Recibir evento $\rightarrow$ Extraer ID único $\rightarrow$ MySQL/MongoDB: *Select* (¿Existe?) $\rightarrow$ If ya existe: Termina; If nuevo: procesa y guarda ID.	Un webhook de Mercado Pago envía el mismo pago varias veces. El agente busca el payment_id antes de refacturar.
P7	**Fan-out: un evento dispara múltiples acciones paralelo**	Cuando la ocurrencia de un único evento gatilla tareas totalmente independientes en diferentes sistemas.	Recibir evento $\rightarrow$ Bifurcar en ramas paralelas directas (sin If/Else) $\rightarrow$ Rama 1: Gmail, Rama 2: Teams, Rama 3: MySQL $\rightarrow$ Consolidar.	Al confirmar nuevo cliente: Rama 1 envía bienvenida, Rama 2 crea tarea en Asana, Rama 3 registra en MySQL y Rama 4 alerta en Discord.

Parte 3 — Combinaciones de conectores frecuentes

Esta tabla resume de manera directa las combinaciones más utilizadas en la práctica con sus complejidades asociadas:

Combinación de Conectores	Caso de Uso Típico	Complejidad de Implementación
GitHub + OpenAI + Discord	Resumen de Pull Requests notificado a Discord	**Baja**

Combinación de Conectores	Caso de Uso Típico	Complejidad de Implementación
Reddit + OpenAI + Teams	Monitoreo de marca con análisis de sentimiento	**Baja**
OpenAI + Gmail	Clasificar, resumir o responder correos con IA	**Media**
MySQL + OpenAI + Teams	Reportes ejecutivos generados con IA enviados a Teams	**Media**
PDF Reader + OpenAI + Notion	Ingesta y resumen de documentos almacenados en Notion	**Media**
Xperience + Asana + Gmail	Formulario de solicitudes que crea tareas y notifica	**Media**
MySQL + Zoom + Gmail	Agendamiento automático de reuniones con confirmación	**Media**
Mercado Libre + OpenAI	Respuesta automática a preguntas de compradores	**Media**
Webhook + OpenAI + WhatsApp	Chatbot de atención al cliente en WhatsApp	**Alta**
Webhook + Mercado Pago + Nubox	Facturación electrónica automática al confirmar pago	**Alta**
Pinecone + OpenAI	Base de conocimiento vectorial (RAG) para respuestas contextuales	**Alta**
S3 + OpenAI + Pinecone	Pipeline de procesamiento y vectorización de documentos	**Alta**

> **Consejo de oro:** Explora el *Market de Templates* de Saturn Studio para encontrar implementaciones base de estas recetas listas para importar directamente a tu workspace.

Saturn Studio — Referencia Técnica

Arquitectura, Variables, Tipos de Datos, Expresiones y Referencia de Comandos

> **Alcance del Documento:** Este manual técnico abarca detalladamente la arquitectura interna de Saturn Studio, el modelo de ejecución de

sus agentes, el sistema integral de variables, los tipos de datos admitidos, los operadores y expresiones de control, así como la referencia de comandos para sus módulos principales.

1. Arquitectura de Saturn Studio

Saturn Studio es un entorno de computación en la nube diseñado para la orquestación integral de agentes. Dentro de su ecosistema, cada componente asume una función específica en el ciclo de vida de las automatizaciones.

1.1 Capas del Ecosistema Rocketbot

El ecosistema se distribuye en cinco capas operativas bien definidas:

Capa	Función y Descripción
Saturn Studio	Núcleo central de orquestación donde convergen la lógica de negocio, los agentes de IA y RPA. Es un entorno 100% cloud accesible mediante el navegador en studio.rocketbot.com.
RPA Studio	Entorno multiplataforma (Windows, Linux, macOS) dedicado al desarrollo de bots tradicionales enfocados en interfaces de escritorio y procesos estructurados.
AI Studio	Capa especializada en la extracción y ordenamiento de datos no estructurados (tales como documentos, audios o correos) para alimentar los flujos de Saturn Studio.
Orchestrator	Centro de gobernanza, programación (scheduling) y supervisión en tiempo real que administra la fuerza laboral digital (instancias, colas, procesos y triggers).
Xperience	Interfaz humano-máquina que provee formularios dinámicos y colas transaccionales para la interacción de los usuarios con los agentes.

1.2 Modelo de Ejecución de un Agente

El motor interno de Saturn Studio procesa la ejecución de los agentes a través de las siguientes fases secuenciales:

1. **Trigger:** El agente es activado mediante un webhook, de manera manual, por planificación cron o invocado por otro agente.
2. **Inicialización:** Se cargan las variables de alcance local y global del agente, y se resuelven las credenciales de manera segura desde el vault.
3. **Ejecución secuencial de Moons:** El motor procesa cada Moon según el orden de sus conexiones, llamando al comando del módulo respectivo y almacenando el resultado en la variable de destino.
4. **Evaluación de condiciones:** Las Moons de tipo If/Else examinan expresiones booleanas bifurcando el flujo según el resultado obtenido.
5. **Human-in-the-loop (si aplica):** El flujo se detiene temporalmente a la espera de una acción humana (como respuestas o aprobaciones de formularios), quedando suspendido hasta recibir el input.
6. **Finalización:** El agente concluye su ciclo registrando en el log si finalizó de manera exitosa o con fallos.

1.3 Tipos de Trigger (Disparadores)

Los flujos pueden iniciarse mediante múltiples mecanismos de activación:

Tipo de Trigger	Comportamiento y Características
Manual	Ejecución directa por el usuario desde el panel de control o el editor, ideal para pruebas.
Webhook	Activación instantánea mediante llamadas HTTP (GET, POST, PUT, DELETE) desde sistemas externos. Permite configurar reintentos ante fallos (0 a 10) y tiempos de espera entre ellos.
Schedule (Cron)	Ejecución planificada mediante expresiones cron (por ejemplo, ejecuciones recurrentes en días hábiles a una hora fija).
Trigger por otro agente	Encadenamiento de flujos independientes donde la finalización de un agente activa el siguiente.
Xperience (Formulario)	El envío de un formulario por parte de un usuario inicia el agente, usando los datos cargados como contexto de entrada.
Múltiples triggers	Disparadores personalizados integrados en los módulos, como el de Gmail, que arranca el proceso al detectar nuevos correos.

2. Variables y Tipos de Datos

Las variables actúan como el canal de transferencia de información entre las distintas Moons. El sistema permite trabajar tanto con variables locales como globales.

2.1 Tipos de Variables Disponibles

Saturn Studio admite los siguientes tipos de datos para sus variables:

Tipo	Descripción y Uso Común	Valor de Ejemplo
String	Texto plano para contenidos de correos, IDs textuales, mensajes o respuestas de IA.	'Hola mundo'
Number	Valores numéricos enteros o decimales para contadores, montos o bucles.	42 o 3.14
Boolean	Valores lógicos de verdadero o falso para banderas de control y condicionales.	true / false
Object / JSON	Estructuras de datos anidadas de clave-valor para respuestas de APIs o formularios.	Acceso: ${{variable}.campo.subcampo}
Array / List	Listas indexadas y ordenadas para colecciones de correos o filas SQL.	Acceso: ${{variable}[0]}

Tipo	Descripción y Uso Común	Valor de Ejemplo
Credential	Referencia segura a credenciales del vault que oculta el valor real.	Uso: {nombre_variable}
File	Referencias a elementos almacenados en el File Storage (PDFs, imágenes, etc.).	N/A
Null / Vacío	Variables sin valor asignado. Requieren validación previa para evitar fallos.	N/A

2.2 Alcance (Scope) de las Variables

La visibilidad y persistencia de las variables varían según su definición:

- **Variable local:** Definida y accesible únicamente dentro de un agente específico. Se configura en el panel de Variables y se reinicia en cada ejecución.
- **Variable global:** Declarada en el menú de Variables Globales de la cuenta, accesible por cualquier agente y persistente entre ejecuciones. Se invoca mediante {{NOMBRE_VARIABLE}}.
- **Variable de credencial:** Variable local de tipo *Credential* que apunta al vault seguro. Se emplea omitiendo las llaves dobles, usando el formato {nombre_variable}.
- **Variable de entorno:** Variable global configurada para segmentar recursos entre diferentes ambientes de trabajo (por ejemplo, Desarrollo vs. Producción).

2.3 Sintaxis de Referencia

Para invocar o inspeccionar variables se aplican las siguientes reglas sintácticas:

Expresión	Descripción
{mi_variable}	Acceso al valor completo de una variable local o global.
{VARIABLE_GLOBAL}	Convención formal recomendada para variables globales (en mayúsculas).
${{respuesta}.campo}	Sintaxis para acceder a una propiedad específica dentro de un objeto JSON.
"${{respuesta}.items[0].nombre}"	Acceso a un índice de una lista JSON y a su atributo interno.
{mi_credencial}	Referencia directa a una variable configurada como Credential.

3. Expresiones y Operadores

Las expresiones condicionales rigen el comportamiento de las Moons de tipo If/Else y los comandos parametrizados con lógica condicional.

3.1 Operadores de Comparación

Permiten contrastar valores dentro de los flujos operacionales:

Operador	Descripción	Ejemplo de Uso
==	Igual a	"{estado}" == 'aprobado'
!=	Distinto de	"{resultado}" != null (si resultado es string)
>	Mayor que	{monto} > 1000
<	Menor que	{contador} < 10
>=	Mayor o igual que	{score} >= 0.8
<=	Menor o igual que	{reintentos} <= 3
contains	Evaluación de inclusión (cadenas y listas)	"{mensaje}".includes('error')
is empty	Comprobación de valor nulo, vacío o indefinido	"{respuesta}" == null (si respuesta es string)
is not empty	Comprobación de existencia de contenido	"{datos}" != null (si datos es string)

3.2 Operadores Lógicos

Utilizados para combinar múltiples condiciones en una sola expresión:

- **AND / &&:** Verdadero únicamente si todas las condiciones se cumplen (Ejemplo: {activo} == true && {monto} > 0).
- **OR / ||:** Verdadero si al menos una de las condiciones es válida (Ejemplo: "{estado}" == 'error' || {reintentos} >= 3).
- **NOT / !:** Invierte el sentido lógico de la condición evaluada (Ejemplo: !({procesado} == true)).

3.3 Moons de Control de Flujo

Módulos nativos encargados de guiar y estructurar el mapa de ejecución:

Moon / Comando	Función Operativa
Start	Nodo de inicio obligatorio en todo flujo. Dirige la ejecución hacia la primera Moon lógica.
If / Else	Evalúa una condición booleana bifurcando el flujo en ramas True o False. Cuenta con un nodo *End If* para unificar el camino posterior.
While / Loop	Cíclico. Itera un bloque de Moons mientras se mantenga la condición de verdad. Finaliza mediante un *End loop*.
For Each	Itera sobre cada uno de los elementos de un Array, finalizando la secuencia con un *End loop*.

Moon / Comando	Función Operativa
System Wait / Sleep	Suspende temporalmente la ejecución por los segundos indicados. Es idóneo para mitigar la saturación de APIs externas.
Try / Catch	Gestión de excepciones. Si ocurre un fallo en el bloque *Try*, la ejecución salta al bloque *Catch*. Concluye en un *End Try*.
Log Message	Inserta textos informativos en el registro del sistema, herramienta clave para auditoría y depuración.
Set Var	Asigna valores a una variable (estáticos, dinámicos o basados en expresiones).
Break	Interrumpe de forma inmediata el bucle en ejecución y salta fuera de su bloque de iteración.

4. Referencia de Comandos por Módulo

Cada módulo provee comandos específicos con parámetros de entrada. Los campos acompañados de un asterisco (*) son requeridos obligatoriamente para su funcionamiento.

4.1 Módulos de Inteligencia Artificial y Conectividad

Módulo: OpenAI

Comando	Parámetros Principales	Descripción / Retorno
Text Completion	credential*, model*, prompt*, max_tokens, temperature, system_prompt	Genera respuestas de texto a partir del modelo seleccionado. Retorna un *String*.
Generate Image	credential*, prompt*, size, quality, n	Devuelve la URL de la imagen creada mediante IA. Retorna un *String*.
Analyze Image	credential*, image_url*, prompt*, model	Genera un análisis o descripción de la imagen provista. Retorna un *String*.
List Models	credential*	Devuelve un listado (Array) con todos los modelos disponibles en la cuenta.
Manage Assistants	credential*, action*, assistant_id, name, instructions, model	Ejecuta acciones de creación, actualización o baja de asistentes. Retorna el objeto asistente.
Message Assistant	credential*, assistant_id*, thread_id, message*	Entabla comunicación con un asistente, devolviendo la respuesta y gestionando los hilos.

Módulo: Request HTTP (Genérico)

Comando	Parámetros Principales	Descripción / Retorno
Simple HTTP Request	url*, method (GET/POST/PUT/DELETE), result_variable*	Invoca una URL específica y vuelca la respuesta como texto (*String*) en la variable elegida.

Comando	Parámetros Principales	Descripción / Retorno
Call API Advanced	url*, method*, headers (JSON), body (JSON), auth, proxy, result_variable*	Realiza peticiones complejas configurando encabezados y cuerpos. Retorna una estructura *JSON*.
Execute CURL	curl_command*, result_variable*	Procesa comandos estructurados bajo el estándar cURL de forma directa.

Módulo: Webhooks

Comando	Parámetros Principales	Descripción / Retorno
Receive Webhook	http_method*, response_mode, result_variable*, retry_on_error, wait_between_retries	Captura llamadas HTTP entrantes guardando el cuerpo, encabezados y queries en la variable.
Send Webhook Response	task_id*, response (JSON/Text), status_code, headers	Envía una respuesta HTTP personalizada directamente al origen del webhook.

4.2 Módulos de Bases de Datos

Módulo: MySQL

- **Query MySQL:** credential*, query*, result_variable* | Ejecuta instrucciones SQL y devuelve un Array de objetos donde cada uno representa una fila de resultados.
- **Insert MySQL:** credential*, table*, data (JSON)*, result_variable | Realiza la inserción de un registro y devuelve el ID autogenerado (last_insert_id).
- **Update MySQL:** credential*, table*, data (JSON)*, where*, result_variable | Modifica registros existentes y retorna la cantidad de filas afectadas.
- **Delete MySQL:** credential*, table*, where*, result_variable | Remueve registros según el criterio establecido y retorna el número de filas borradas.

Módulo: PostgreSQL

- **Query PostgreSQL:** credential*, query*, result_variable* | Corre sentencias SQL estructuradas sobre el puerto predeterminado 5432. Retorna un Array de objetos.
- **Insert PostgreSQL:** credential*, table*, data (JSON)*, result_variable | Añade una fila en la base de datos, retornando el objeto recién guardado junto con su ID corporativo.
- **Update PostgreSQL:** credential*, table*, data (JSON)*, where*, result_variable | Actualiza de forma masiva o selectiva las filas que cumplan las condiciones del where.

Módulo: SQL Server

- **Query SQLServer:** credential*, query*, result_variable* | Ejecuta lenguaje T-SQL a través del puerto por defecto 1433, retornando colecciones de objetos dentro de un Array.

4.3 Módulos de Mensajería y Comunicación

Módulo: Gmail

Comando	Parámetros Principales	Descripción / Retorno
Send Email	credential*, to*, subject*, body*, cc, bcc, attachments	Despacha correos electrónicos adjuntando archivos opcionales. Retorna confirmación.
Read Emails	credential*, folder, max_results, query, result_variable*	Extrae correos en formato Array con detalles clave (from, subject, body, attachments).
Search Emails	credential*, query*, max_results, result_variable*	Localiza mensajes mediante filtros de búsqueda avanzada de Gmail (ej. has:attachment).
Move to Folder	credential*, message_id*, folder*	Reubica o etiqueta un correo específico dentro del árbol de carpetas de la cuenta.
Delete Email	credential*, message_id*	Envía el correo a la papelera o efectúa su borrado definitivo.

Módulo: PDF Reader

- **Extract Text:** file_path* (o file_variable*), result_variable* | Transforma el contenido íntegro del documento PDF en una sola variable tipo *String* (puede presentar limitaciones si el archivo es escaneado).
- **Extract Pages:** file_path*, page_numbers, result_variable* | Segrega y procesa de forma exclusiva las páginas parametrizadas, entregando el texto de cada una.
- **Extract Tables:** file_path*, result_variable* | Aplica algoritmos de reconocimiento para estructurar tablas internas dentro de un Array de Arrays.

Módulo: Xperience

- **List Forms:** credential*, result_variable* | Lista todos los formularios activos enlazados a la credencial del entorno.
- **Get Queue Records:** credential*, form_id*, queue_id, filters, result_variable* | Descarga los registros transaccionales presentes en las colas de trabajo aplicando filtros de estado.
- **Update Record Status:** credential*, form_id*, record_id*, new_status*, result_variable | Cambia de forma lógica el estado operacional de una

transacción (ej: *Procesado*, *Error*).

- **Get Job Detail:** credential*, job_id*, result_variable* | Devuelve información minuciosa sobre una tarea/job dentro de la cola transaccional.
- **Download Attachment:** credential*, form_id*, record_id*, result_variable* | Descarga de forma local los archivos que hayan sido adjuntados en un registro determinado.

4.4 Módulos de Productividad

Módulo: Notion

Comando	Parámetros Principales	Descripción / Retorno
Query Database	credential*, database_id*, filter (JSON), sorts, result_variable*	Consulta repositorios o bases de datos vinculadas formalmente con la integración de Notion.
Create Page	credential*, parent_id*, title*, properties (JSON), result_variable	Da de alta una nueva página web dentro de Notion, devolviendo su ID y el objeto creado.
Update Page	credential*, page_id*, properties (JSON), result_variable	Modifica de forma directa las propiedades o metadatos de una página existente.
Get Page	credential*, page_id*, result_variable*	Extrae el contenido y la estructura completa de propiedades de una página por su identificador.

Módulo: Asana

- **List Tasks:** credential*, project_id, assignee, completed, result_variable* | Descarga un Array con las tareas asociadas a proyectos o personas específicas.
- **Create Task:** credential*, name*, project_id, assignee, due_date, notes, result_variable | Da de alta una tarea en la plataforma devolviendo el objeto e ID de Asana.
- **Update Task:** credential*, task_id*, name, completed, assignee, due_date, result_variable | Modifica atributos clave en tareas previamente guardadas.
- **Complete Task:** credential*, task_id*, result_variable | Cambia el estado de una tarea directamente al valor de concluida o finalizada.
- **List Projects:** credential*, workspace_id, result_variable* | Lista de forma detallada los proyectos creados dentro de un espacio de trabajo.

Módulo: GitHub

- **Subscribe Webhook:** Configurable externamente en GitHub Settings; requiere result_variable* | Monitorea eventos del repositorio (tales

como *push*, *pull_request*, *issues*).
- **Get Repository:** credential*, owner*, repo*, result_variable* | Trae datos del repositorio como descripciones, cantidad de estrellas, ramas y su nombre formal.
- **Get Pull Request:** credential*, owner*, repo*, pr_number*, result_variable* | Inspecciona el estado de un PR, incluyendo títulos, autores y listados de ficheros modificados.
- **List Commits:** credential*, owner*, repo*, branch, since, result_variable* | Retorna un listado estructurado de cambios (SHA, mensaje, autor y fecha).

5. Campos Técnicos de Credenciales

El almacenamiento en el vault seguro requiere parámetros específicos estructurados según el mecanismo de autenticación del servicio.

5.1 Parámetros de Configuración por Tipo de Autenticación

API Key Simple

- **Credential name:** Identificador único interno para su posterior selección (Ej: 'OpenAI Produccion'). No debe contener caracteres especiales.
- **API Key:** Código o token privado extraído directamente del proveedor. Se debe registrar omitiendo espacios adicionales.

OAuth 2.0 (Client ID + Client Secret)

Campo	Descripción y Obtención
Credential name	Identificador interno único.
Client ID	ID público asignado a la aplicación en la plataforma del proveedor externo.
Client Secret	Clave privada del aplicativo. Se recomienda respaldar de inmediato ya que suele ocultarse tras la primera visualización.
Redirect URI	Dirección fija y obligatoria: https://studio.rocketbot.com (exacta, sin parámetros ni diagonales finales).
Scopes	Permisos y accesos requeridos. Deben parametrizarse previamente en la consola del proveedor.

AWS IAM (Access Key + Secret)

- **Credential name:** Nombre descriptivo interno.
- **AWS Access Key ID:** Código de acceso alfanumérico del usuario IAM. Formato estándar: AKIA + 16 caracteres.
- **AWS Secret Access Key:** Firma o clave secreta del perfil IAM. Solo se despliega al crear la credencial y requiere resguardo inmediato.

- **AWS Region:** Región geográfica de AWS donde operan los servicios deseados (Ej: us-east-1, sa-east-1).

Bases de Datos (Host + Puerto + Credenciales)

Campo	Especificación Técnica
Server address	Dirección de IP pública o dominio del servidor de datos. **Bajo ningún motivo debe configurarse como localhost o 127.0.0.1.**
Server port	Puerto TCP de escucha: MySQL = 3306, PostgreSQL = 5432, SQL Server = 1433, MongoDB = 27017.
User	Cuenta con privilegios estrictamente acotados (evitar el uso de credenciales maestras como sa o root).
Password	Contraseña de la cuenta de base de datos configurada.
Database	Nombre de la base de datos (sensible a mayúsculas/minúsculas según el motor).
Encrypt Connection	(*SQL Server*) True activa el cifrado TLS, mandatorio en entornos productivos.
Trust Server Certificate	(*SQL Server*) True salta la validación permitiendo certificados autofirmados; restringido a desarrollo.

5.2 Límites y Comportamientos Críticos del Sistema

Al diseñar integraciones dentro de la plataforma, se deben prever los siguientes comportamientos:

- **Vigencia de Claves del Orquestador:** Las API Keys del Orquestador de Rocketbot tienen un periodo de validez estricto de 2 años. Al expirar, todo agente que dependa de ellas detendrá su ejecución por error de autenticación.
- **Revocación de Accesos:** Al dar de baja una credencial compartida con terceros, la desvinculación es inmediata y provocará el fallo del agente en su ciclo subsiguiente.
- **Client Secret en Proveedores:** Portales como GitHub, LinkedIn o Mercado Libre muestran el *Client Secret* una única vez; es imperativo guardarlo antes de abandonar la pantalla.
- **Uso Sintáctico de Credenciales:** Las variables que referencian elementos del vault se llaman obligatoriamente con una única llave {nombre_variable}. Las dobles llaves se reservan exclusivamente para variables de datos ordinarias.
- **Conectividad de Base de Datos:** Debido a que Saturn Studio es un servicio nativo de la nube, las bases de datos externas deben poseer una IP pública accesible y tener el puerto TCP correspondiente desbloqueado en el firewall corporativo.

6. Webhooks: Referencia Técnica

Los webhooks proveen una vía ágil para recibir alertas y datos en tiempo real provenientes de sistemas periféricos.

6.1 Estructura de los Eventos Entrantes

Al recibir una llamada HTTP en el endpoint del webhook, Saturn Studio expone los siguientes datos estructurados dentro de la variable de resultados asignada:

- **body:** Contiene el cuerpo principal del mensaje, comúnmente formateado como un objeto JSON.
- **headers:** Metadatos técnicos de la petición HTTP (IP del cliente, Content-Type, marcas de tiempo y tokens).
- **query:** Parámetros adjuntos en la URL de llamada (Ej: si la ruta incluye ?id=123&estado=aprobado, se procesará como query.id = '123').
- **params:** Variables lógicas extraídas de segmentos de rutas dinámicas.
- **method:** El verbo HTTP empleado en la petición (GET, POST, PUT, DELETE).

6.2 Modos de Respuesta (Response Mode)

Dependiendo de las necesidades de integración, el comportamiento de respuesta HTTP se divide en tres configuraciones:

Modo	Comportamiento Técnico
Immediately	Devuelve un código HTTP 200 de forma instantánea al recibir el paquete, antes de procesar el flujo interno. Es la opción recomendada para integraciones con políticas de timeout muy estrictas (ej. GitHub, WhatsApp, Mercado Pago).
After execution	Mantiene la sesión HTTP abierta y responde únicamente cuando el flujo del agente se ha completado. Requiere que el emisor tolere tiempos de espera elevados (superiores a 30 segundos).
Manual (Send Response)	Delega la respuesta al comando *Send Webhook Response*, permitiendo computar datos de forma interna para enviarlos de regreso en el cuerpo de la respuesta HTTP.

6.3 Panel de Supervisión y Monitoreo de Webhooks

El entorno administrativo ofrece un listado donde se detalla el agente vinculado, el método HTTP, la URL asignada y los controles de gestión del webhook. Adicionalmente, permite auditar visualmente los flujos en cola, completados con éxito o con errores.

A través de la sección **Tasks**, es posible realizar un análisis forense de cada petición que ingresa a la plataforma utilizando los siguientes datos en pantalla:

Columna / Control	Información Desplegada
Header count	Cuantifica la cantidad de encabezados recibidos en la cabecera HTTP.
Body size	Peso total en bytes de la carga útil (payload), útil para identificar paquetes vacíos o masivos.
Query	Muestra los parámetros incluidos en la URL de la consulta.
Data	Visualización de los datos procesados.
Status	Muestra el estado operativo final (*Completado* o *Con error*).
Updated	Muestra el tiempo transcurrido desde el momento exacto de la recepción.
Ojo (Ver)	Abre el inspector de datos crudos (*raw data*), mostrando el desglose de: body, headers, query, params y response. Panel crítico para depuración y debugging.

Consejo de Desarrollo: Para consultar de forma extendida y actualizada los detalles específicos de cada comando, capturas de interfaz y guías de implementación paso a paso, acceda al portal de documentación oficial en docs.rocketbot.com.

Saturn studio – Changelog

ROCKETBOT – SATURN STUDIO
Changelog del Producto
Historial de versiones, nuevos módulos y mejoras de Saturn Studio

> *Nota: Fuente – Este changelog fue construido a partir de la documentación oficial publicada en docs.rocketbot.co. Las fechas corresponden a las fechas de publicación de la documentación de cada módulo o funcionalidad.*

2026

v2026.06 – 24 de Junio

Nuevos módulos (MCP Server, Datastore, Nexus, Sigma móvil), Mejorados

(DeepSeek, Webhooks, TuFirma.digital, Editor / DX), Corrección (Módulos Microsoft), Eliminados (Widget Vibe Coding)

Área / Categoría	Tipo	Descripción
MCP Server	**NUEVO**	Lanzamiento de servidores MCP (*Model Context Protocol*) para permitir que clientes externos de IA (como Claude, Copilot, etc.) interactúen con la aplicación, creando bots o ejecutando tareas de forma autónoma.
Datastore	**NUEVO / MEJORA**	• Nuevo comando Trigger para iniciar ejecuciones al insertar un nuevo registro. • Opciones para recargar valores, exportar (con o sin datos), importar y eliminar registros específicos.
MODULO Nexus	**NUEVO**	Módulo para conexión con la suite Rocketbot Nexus: listar tablas y filas, ejecutar consultas complejas, entre otros.
MODULO Sigma Móvil	**NUEVO**	Módulo para consumo de la API de Sigma Móvil para envío de mensajes SMS a usuarios o ejecuciones de campañas.
MODULO DeepSeek	**MEJORA**	Nuevo comando **Process File**: permite enviar archivos (imágenes, PDFs con texto o archivos de texto plano) para análisis y preguntas sobre el contenido.
MODULO Webhooks	**MEJORA**	• Nuevo comando **Clean Webhook Queue** para vaciar la cola de procesos en caso de saturación. • Botones de estado con texto descriptivo aclarado (*“Set Completed” / “Set Pending”*).
MODULO TuFirma.Digital	**MEJORA**	Migración de la API a su nueva versión manteniendo compatibilidad con todas sus funciones previa.
Módulos Microsoft	**CORRECCIÓN**	Corrección de un error (*unhandled exception*) que ocurría durante la creación de credenciales.
Editor / DX (Developer Experience)	**MEJORA**	• **Set Variable:** Soporte para console.log() visible desde la consola del navegador para facilitación de debugging. • **Métricas de Diagnóstico:** Diagnósticos del robot visibles en el footer (*critical, warning, info*). • **Carga de Robots:** Optimización del peso al guardar y barra de progreso al cargar robots pesados o arrastrar archivos JSON. • **Confirmación al importar:** Mensaje de confirmación previo al importar arrastrando archivos JSON. • **Json View:** Selector de vista (Texto plano, JSON o Tabla) en el inspector de variables y en el modal de comandos. • Optimización general en la ejecución de código JavaScript.

Área / Categoría	Tipo	Descripción
Interfaz	**ELIMINADO**	Removido el widget de *Vibe Coding* de la vista.

v2026.05 — Mayo 2026

Nuevos conectores cloud y Google

Tipo	Área	Descripción
MODULO	Cloudflare	Nuevo módulo para gestión de certificados SSL/TLS en zonas de Cloudflare. Permite subir, eliminar y consultar certificados directamente desde flujos de Saturn Studio.
MODULO	Dropbox	Nuevo módulo para subir, descargar, listar y eliminar archivos y carpetas en Dropbox. Autenticación vía Access Token.
MODULO	Google Books	Nuevo módulo de integración con la biblioteca digital de Google para buscar y recuperar información de volúmenes vía OAuth 2.0.
MODULO	Google BigQuery	Nuevo módulo para ejecutar consultas SQL sobre grandes volúmenes de datos en BigQuery. Autenticación OAuth 2.0 desde Google Cloud Console.
MODULO	Nubox	Nuevo módulo para emisión de documentos de venta y facturación electrónica en LATAM. Soporta ambientes UAT y Productivo con URLs diferenciadas.
MEJORA	Credenciales	Nuevo sistema de gestión de credenciales compartidas: compartir por email o por equipo, revocar acceso con efecto inmediato, variables de tipo Credential para parametrización.

v2026.04 — Abril 2026

Gestión de credenciales y soporte

Tipo	Área	Descripción
MEJORA	Credenciales	Sistema de credenciales compartidas: soporte para compartir por email (múltiples separados por coma) y por equipo (todos los miembros reciben acceso automáticamente).
MEJORA	Credenciales	Variables de tipo Credential: permiten parametrizar el acceso a servicios sin hardcodear credenciales. Se referencian con {nombre_variable} en campos de credencial.
NUEVO	Zendesk	Nuevo módulo para gestión de tickets de soporte: crear, eliminar, listar, obtener y actualizar tickets directamente desde flujos de Saturn Studio.
MEJORA	Xperience	Nuevo comando: Insertar elemento en cola (Insert Queue Item). Permite que los agentes alimenten colas transaccionales desde el flujo de automatización.

v2026.03 — Marzo 2026

Market de Templates y Xperience

Tipo	Área	Descripción
NUEVO	Market de Templates	Lanzamiento del Market de Templates: catálogo centralizado para explorar, previsualizar y reutilizar plantillas de agentes pre-construidos. Los templates se importan al workspace con un clic.
NUEVO	Market de Templates	Vista previa de templates en modo solo lectura: permite hacer zoom y scroll para explorar la lógica antes de agregar al workspace.
NUEVO	Market de Templates	Subida de templates por administradores: exportar agente como .json y publicarlo en el Market con nombre, descripción y categoría.
NUEVO	Xperience	Módulo Xperience para Saturn Studio: gestión de formularios, colas transaccionales y estados operativos integrados con el Orquestador Rocketbot.
NUEVO	Variables Globales	Panel de Variables Globales: crear, editar y eliminar variables compartidas entre todos los agentes de la cuenta. Convención recomendada: MAYÚSCULAS.
NUEVO	Teams y Colaboración	Gestión de equipos (Teams): invitar miembros, compartir agentes por grupo, gestión centralizada de accesos.
NUEVO	Webhooks	Panel de monitoreo de Webhooks con raw data: ver body, headers, query y response de cada evento recibido. Botón de ojo azul para debugging.
NUEVO	Webhooks	Configuración de Response Mode: Immediately, After execution o Manual (Send Webhook Response). Retry on Error configurable de 0 a 10 intentos.
NUEVO	Global Variables	Variables Globales: recurso centralizado para datos compartidos entre múltiples agentes. Acceso desde cualquier flujo de la cuenta.

v2026.02 — Febrero 2026

IA multimodelo y bases de datos cloud

Tipo	Área	Descripción
MODULO	AWS Bedrock	Nuevo módulo AWS Bedrock: acceso a modelos fundacionales de Amazon (Titan, Claude, Llama) para IA generativa. Autenticación IAM (Access Key + Secret + Región).
MODULO	Azure OpenAI	Nuevo módulo Azure OpenAI: modelos GPT-4 y DALL-E dentro de la infraestructura de seguridad Microsoft Azure. Comandos: Message Assistant, Text Completion, Generate Image, List Models.
MODULO	AWS S3	Nuevo módulo AWS S3: subir, descargar, listar y eliminar archivos en buckets de Amazon S3. Generar URLs de acceso temporal.
MODULO	Evolution API WhatsApp	Nuevo módulo WhatsApp vía Evolution API: enviar y recibir mensajes, gestionar instancias, conectar vía QR. Soporte para archivos multimedia.

Tipo	Área	Descripción
MODULO	Facebook Messenger	Nuevo módulo Facebook Messenger: enviar y recibir mensajes en páginas de empresa. Autenticación vía Token de página Meta for Developers.
MODULO	LinkedIn	Nuevo módulo LinkedIn: automatizar publicaciones en perfil o página de empresa. Autenticación OAuth 2.0 con scopes w_member_social.
MODULO	PostgreSQL	Nuevo módulo PostgreSQL: ejecutar consultas complejas, insertar y actualizar registros. Requiere IP pública y configuración de pg_hba.conf.
MODULO	Mistral AI	Nuevo módulo Mistral AI: modelos de lenguaje de alto rendimiento para generación de texto y razonamiento. Requiere organización en console.mistral.ai.
MODULO	Mercado Libre	Nuevo módulo Mercado Libre: gestionar ventas, responder mensajes de postventa, actualizar stock y procesar facturación. OAuth 2.0 LATAM.
MODULO	Mercado Pago	Nuevo módulo Mercado Pago: procesar cobros, gestionar suscripciones, consultar transacciones. Redirect URI: https://studio.rocketbot.com.
MODULO	Notion	Nuevo módulo Notion: consultar bases de datos, crear y actualizar páginas. Cada recurso debe compartirse explícitamente con la integración.
MODULO	Asana	Nuevo módulo Asana: listar, crear, actualizar y completar tareas; listar proyectos. Autenticación vía Personal Access Token.
MODULO	Pinecone	Nuevo módulo Pinecone: base de datos vectorial para búsqueda semántica y RAG. Upsert, Query, Delete y List Indexes.
MODULO	SQL Server	Nuevo módulo SQL Server: ejecutar T-SQL con soporte para cifrado TLS y Trust Server Certificate. Puerto por defecto: 1433.

2025

v2025.12 — Diciembre 2025

Infraestructura cloud y mensajería

Tipo	Área	Descripción
MODULO	Credenciales Microsoft	Nuevo módulo de credenciales Microsoft unificadas para Outlook, Teams y SharePoint bajo un solo token OAuth 2.0 (Microsoft Graph API).
MODULO	Chat Widget	Nuevo Chat Widget embebible en sitios web: captura mensajes de visitantes y los enruta a flujos de Saturn Studio. Sin API Key externa requerida.
MODULO	Discord	Nuevo módulo Discord: enviar mensajes y embeds a canales y servidores. Autenticación vía Bot Token desde Discord Developer Portal.
MODULO	Reddit	Nuevo módulo Reddit: buscar publicaciones y comentarios en subreddits. Autenticación OAuth 2.0 desde Reddit App Preferences.

Tipo	Área	Descripción
MODULO	Microsoft SharePoint	Nuevo módulo SharePoint: subir, descargar y gestionar archivos y listas en sitios corporativos. Integrado con credencial Microsoft.
MODULO	PDF Reader	Nuevo módulo PDF Reader: extracción de texto, páginas y tablas de archivos PDF. Funciona con archivos del File Storage o rutas variables.
NUEVO	Despliegue AWS	Nueva guía y soporte para desplegar instancias de Rocketbot en Amazon Web Services (EC2).
NUEVO	Despliegue Azure	Nueva guía y soporte para desplegar instancias de Rocketbot en Microsoft Azure.

v2025.11 – Noviembre 2025

CRM, email corporativo y gestión de proyectos

Tipo	Área	Descripción
MODULO	Salesforce	Nuevo módulo Salesforce CRM: gestión de objetos, registros y automatización de flujos comerciales. Autenticación OAuth 2.0.
MODULO	Monday.com	Nuevo módulo Monday.com: gestión de tableros, ítems y columnas. Autenticación vía API Key de Monday.
MODULO	Bitbucket	Nuevo módulo Bitbucket: gestión de repositorios, pull requests y pipelines CI/CD. Autenticación OAuth 2.0.
MODULO	Email IMAP y SMTP	Nuevo módulo Email genérico IMAP/SMTP para servidores corporativos no soportados por Gmail. Configuración manual de servidor, puertos y SSL.
MODULO	Google Tasks	Nuevo módulo Google Tasks: gestión de listas y tareas vía Google Tasks API. Autenticación OAuth 2.0.
MODULO	Facebook Messenger	Soporte inicial para Facebook Messenger: recepción de mensajes vía webhook desde páginas de empresa.
NUEVO	Tool Agent	Nuevo tipo de Moon: Tool Agent. Permite definir herramientas que el agente de IA puede usar autónomamente para tomar decisiones en el flujo.

v2025.10 – Octubre 2025

IA multimodelo expandida y herramientas de desarrollo

Tipo	Área	Descripción
MODULO	Grok (xAI)	Nuevo módulo Grok: generación de texto, imágenes e interpretación visual con modelos de xAI. Autenticación vía API Key desde console.x.ai.
MODULO	Jira	Nuevo módulo Jira: triggers para creación/actualización de incidencias, gestión de issues en proyectos ágiles. Autenticación OAuth.
MODULO	ClickUp	Nuevo módulo ClickUp: gestión de tareas, listas y espacios de trabajo. Autenticación vía API Token personal.

Tipo	Área	Descripción
NUEVO	Q&A Agent v2	Question and Answer Agent v2: motor mejorado de respuestas contextual con soporte para historiales de conversación más largos y mayor precisión.
NUEVO	Google Contacts	Nuevo módulo Google Contacts: crear, leer y gestionar contactos de Google. Autenticación OAuth 2.0.
NUEVO	Formularios (reCAPTCHA)	Soporte para activar reCAPTCHA en formularios de Xperience para proteger contra bots y envíos automatizados no deseados.
MODULO	Microsoft Teams	Nuevo módulo Microsoft Teams vía Graph API: enviar mensajes a canales y chats individuales. Autenticación con credencial Microsoft.
MODULO	iLovePDF	Nuevo módulo iLovePDF: compresión, conversión y manipulación de PDFs vía API. Autenticación vía API Key.
MODULO	Hugging Face	Nuevo módulo Hugging Face: acceso a modelos open source de IA para inferencia de texto e imágenes. Autenticación vía API Token.
MODULO	HubSpot	Nuevo módulo HubSpot CRM: gestión de contactos, negocios y actividades comerciales. Autenticación OAuth 2.0.

v2025.09 — Septiembre 2025

Ecosistema Google completo y modelos IA adicionales

Tipo	Área	Descripción
MODULO	Gmail	Nuevo módulo Gmail: enviar, leer, buscar, mover y eliminar correos. Autenticación OAuth 2.0 vía Google Cloud Console.
MODULO	Google Calendar	Nuevo módulo Google Calendar: crear, listar y gestionar eventos y reuniones. Autenticación OAuth 2.0.
MODULO	Google Docs	Nuevo módulo Google Docs: crear y editar documentos de texto en Google Workspace. Autenticación OAuth 2.0.
MODULO	Google Drive	Nuevo módulo Google Drive: subir, descargar, listar y eliminar archivos en Drive. Autenticación OAuth 2.0.
MODULO	Google Sheets	Nuevo módulo Google Sheets: leer y escribir datos en hojas de cálculo. Autenticación OAuth 2.0.
MODULO	Google Forms	Nuevo módulo Google Forms: crear y gestionar formularios de Google. Autenticación OAuth 2.0.
MODULO	Google Chat	Nuevo módulo Google Chat: enviar mensajes a espacios y conversaciones de Google Chat.
MODULO	Gemini (Google AI)	Nuevo módulo Gemini: modelos de IA multimodal de Google para generación de texto e imágenes. Autenticación vía API Key desde Google AI Studio.
MODULO	DeepSeek	Nuevo módulo DeepSeek: modelos de lenguaje open source de alto rendimiento para texto y razonamiento. Autenticación vía API Key.

Tipo	Área	Descripción
MODULO	Groq	Nuevo módulo Groq: inferencia ultra-rápida de modelos LLM (Llama, Mixtral) vía hardware especializado. Autenticación vía API Key.
MODULO	Freshdesk	Nuevo módulo Freshdesk: gestión de tickets de soporte al cliente. Autenticación vía API Key de Freshdesk.
MODULO	GitHub	Nuevo módulo GitHub: webhooks, información de repositorios, pull requests y commits. Autenticación OAuth desde GitHub Settings.
MODULO	Calendly	Nuevo módulo Calendly: gestión de eventos de agendamiento y disponibilidad. Autenticación OAuth 2.0.
NUEVO	Sistema (System)	Nueva sección System en Saturn Studio: información del entorno de ejecución, variables del sistema y control de la instancia.
NUEVO	Orquestador CronJobs	Soporte para CronJobs avanzados en el Orquestador: expresiones cron con mayor granularidad para scheduling preciso de agentes.

Primera mitad de 2025 — Lanzamiento y fundamentos

Durante el primer semestre de 2025, Saturn Studio fue construido desde sus cimientos como la nueva plataforma de orquestación de agentes de Rocketbot, reemplazando el paradigma de RPA puro por un modelo híbrido IA + automatización.

v2025.H1 — Enero — Agosto 2025

Fundamentos de Saturn Studio

Tipo	Área	Descripción
NUEVO	Saturn Studio	Lanzamiento de Saturn Studio: plataforma de orquestación de agentes 100% cloud basada en el modelo de Moons (comandos visuales conectados en flujo secuencial).
NUEVO	Editor de flujos	Editor visual drag-and-drop de Moons: conectar comandos arrastrando, organizar el lienzo libremente, opciones de clic derecho (editar, clonar, eliminar, ejecutar).
NUEVO	File Storage	File Storage integrado: subir y gestionar archivos dentro de Saturn Studio para uso de los agentes durante la ejecución.
NUEVO	Request HTTP	Módulo Request HTTP genérico: Simple HTTP Request, Call API Advanced y Execute CURL. Para integración con cualquier API REST sin módulo dedicado.
MODULO	OpenAI	Módulo OpenAI: Text Completion, Generate Image, Analyze Image, List Models, Manage Assistants, Message Assistant. Autenticación vía API Key.
MODULO	MySQL	Módulo MySQL: Query, Insert, Update, Delete. Requiere IP pública, puerto 3306 abierto. No soporta localhost.

Tipo	Área	Descripción
MODULO	MongoDB	Módulo MongoDB: Find, Insert, Update, Delete, Aggregate. Conexión vía Connection String de MongoDB Atlas o servidor propio.
MODULO	Webhooks	Módulo Webhooks: Receive Webhook (URL estática única por agente) y Send Webhook Response. Soporta GET, POST, PUT, DELETE.
MODULO	Moons core	Moons de lógica base: Start, End, If/Else, While/Loop, For Each, Wait/Sleep, Try/Catch, Log/Print, Set Variable, Break, Continue.
NUEVO	Credenciales (vault)	Sistema de credenciales seguro: almacenamiento cifrado de API Keys, OAuth tokens y strings de conexión. Base para todos los conectores de Saturn Studio.
NUEVO	Market de Templates	Infraestructura base del Market de Templates: soporte para importar/exportar agentes en formato .json. Publicación pública en lanzamiento de v2026.03.
NUEVO	GitHub	Módulo GitHub inicial: Subscribe Webhook, Get Repository, Get Pull Request, List Commits. Autenticación OAuth.

Resumen de crecimiento del ecosistema

Periodo	Enfoque	Highlights
Enero – Agosto 2025	Lanzamiento	12+ módulos core, editor visual, File Storage, credenciales, webhooks, OpenAI, MySQL, MongoDB
Septiembre 2025	Expansión IA	15 nuevos módulos: ecosistema Google completo, Gemini, DeepSeek, Groq, GitHub, Calendly, Freshdesk, Cronjobs avanzados
Octubre 2025	CRM y DevOps	10 nuevos módulos: Grok, Jira, ClickUp, HubSpot, Hugging Face, Microsoft Teams, iLovePDF, Tool Agent, Q&A v2
Noviembre 2025	Empresarial	7 nuevos módulos: Salesforce, Monday.com, Bitbucket, Email IMAP/SMTP, Facebook Messenger inicial, Google Tasks, reCAPTCHA
Diciembre 2025	Cloud	8 nuevos: Microsoft unificado, Chat Widget, Discord, Reddit, SharePoint, PDF Reader, despliegue AWS y Azure
Enero – Febrero 2026	Pagos LATAM	Módulos: Mercado Libre, Mercado Pago, Notion, Asana, LinkedIn, AWS Bedrock, Azure OpenAI, AWS S3, WhatsApp, PostgreSQL, SQL Server, Mistral, Pinecone
Marzo 2026	Plataforma	Market de Templates, Xperience, Variables Globales, Teams y colaboración, Panel de monitoreo Webhooks
Abril 2026	Credenciales	Credenciales compartidas avanzadas, Variables Credential, Zendesk, Insert Queue Item en Xperience
Mayo 2026	Cloud v2	Cloudflare, Dropbox, Google Books, Google BigQuery, Nubox (facturación electrónica LATAM)

> Consejo: Documentación completa — Para ver los detalles técnicos de cada módulo y sus comandos, visita ***docs.rocketbot.com*** Cada módulo tiene su página dedicada con parámetros, ejemplos y guías de credenciales.

Rocketbot SpA — 2026

Saturn Studio — Troubleshooting

ROCKETBOT — SATURN STUDIO

Guía de Troubleshooting

Diagnóstico y solución de errores frecuentes en Saturn Studio

> *Nota: Estructura — Este documento cubre: errores de credenciales y autenticación, problemas de conectividad con bases de datos, fallos de webhooks, errores de IA y módulos, problemas de flujo y lógica, y una guía de diagnóstico paso a paso.*

> *Atención: Antes de reportar un error a soporte — Revisa siempre el Log de ejecución del agente. El 90% de los errores tienen su causa exacta en el mensaje del log.*

Cómo leer el Log de ejecución

El Log es tu primera herramienta de diagnóstico. Antes de revisar cualquier otro lugar, abre el panel de log del agente.

Indicador	Qué significa	Qué hacer
ERROR	El agente falló en esa moon. La causa exacta aparece en el mensaje.	Leer el mensaje de error completo — no solo el tipo.
WARNING	Algo no funcionó perfectamente pero el flujo continuó.	Revisar si el resultado del comando es el esperado.
INFO	Registro informativo del estado del flujo.	Útil para rastrear por dónde pasó el agente.

Indicador	Qué significa	Qué hacer
Moon iluminada en rojo	La moon falló durante la ejecución.	Hacer clic derecho → Ejecutar solo ese comando para aislar el problema.
Agente se detiene sin error	Una condición If/Else dirigió el flujo a un End sin procesamiento.	Revisar las condiciones de bifurcación y los valores de las variables.

Errores por categoría

Categoría 1 — Credenciales y Autenticación

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-CRED-01 Alto	Credencial falla al hacer Check — estado 'Not Connected'	• API Key incorrecta o copiada con espacios adicionales. • Clave regenerada en origen pero no actualizada en la plataforma. • Cuenta externa sin plan con acceso API (ej: OpenAI sin crédito).	1. Ir a Credentials y abrir la credencial. 2. Copiar de nuevo la API Key desde el origen (sin espacios). 3. Hacer clic en Check. 4. Si dice 'quota exceeded' o 'billing', revisar la cuenta externa.	Después de copiar claves, usar Ctrl+A y revisar que no haya espacios al inicio o final. Nunca copiar desde un documento intermedio.
ERR-CRED-02 Alto	OAuth 2.0 rechaza la conexión — Redirect URI mismatch	• El Redirect URI configurado en la app del proveedor (LinkedIn, Mercado Libre, Google, Zoom) no coincide exactamente con Saturn Studio. • La URL tiene mayúsculas, barras finales o parámetros incorrectos.	1. Ir al portal de desarrolladores del servicio. 2. Buscar la sección de Redirect URIs o Authorized redirect URLs. 3. Verificar que la URL sea exactamente: https://studio.rocketbot.com 4. Guardar y reintentar la autenticación.	Copiar la URL de redirect desde la documentación oficial de Saturn Studio, nunca escribirla manualmente.

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-CRED-03 Medio	Credencial de Xperience falla — error de API Key inválida	• La API Key del Orquestador venció (vigencia de 2 años). • Nueva API Key generada en el Orquestador pero no actualizada en Saturn Studio. • Usuario del Orquestador sin servicio Xperience contratado.	1. Ingresar al Orquestador Rocketbot. 2. Ir al perfil del usuario → sección API Key. 3. Generar una nueva API Key si la actual está vencida. 4. Copiar la clave y actualizar la credencial de Xperience.	Anotar la fecha de creación de cada API Key del Orquestador para anticipar el vencimiento a los 2 años.
ERR-CRED-04 Medio	Credencial compartida revocada — agentes del equipo fallan	• Un administrador revocó el acceso a una credencial compartida. • El usuario que creó la credencial la eliminó.	1. Identificar al propietario en la lista de Credentials. 2. Solicitar al propietario que vuelva a compartirla por equipo o email. 3. O bien, crear una credencial propia y actualizar los agentes.	Al revocar acceso, notificar al equipo con anticipación. Usar Variables de tipo Credential para facilitar el cambio.

Categoría 2 — Bases de Datos

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-DB-01 Crítico	No se puede conectar a MySQL o PostgreSQL	• Se usó ‘localhost’ o ‘127.0.0.1’ como dirección. • Puerto 3306 (MySQL) o 5432 (PostgreSQL) bloqueado por el firewall. • El servidor no acepta conexiones externas.	1. Reemplazar ‘localhost’ por la IP pública o dominio del servidor. 2. Verificar puerto abierto para conexiones TCP entrantes. 3. En MySQL: bind-address debe ser 0.0.0.0 en my.cnf. 4. En PostgreSQL: revisar pg_hba.conf y listen_addresses = ‘*’. 5. Usar Test Connection.	Nunca usar localhost como servidor. Saturn Studio opera en la nube y no tiene acceso a redes locales.
ERR-DB-02 Alto	Error de permisos al ejecutar consulta en la base de datos	• El usuario de la base de datos no tiene permisos suficientes (SELECT, INSERT, UPDATE, DELETE). • Intento de acceso a una tabla no autorizada.	1. Conectarse al servidor como administrador. 2. Verificar permisos: SHOW GRANTS FOR ‘usuario’@’%’; (MySQL). 3. Otorgar permisos: GRANT SELECT, INSERT ON base.tabla TO ‘usuario’@’%’; 4. Recargar: FLUSH PRIVILEGES; 5. Probar de nuevo en Saturn Studio.	Crear un usuario dedicado para Saturn Studio con permisos mínimos necesarios. Nunca usar el usuario administrador sa o root.

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-DB-03 Alto	SQL Server — falla la conexión por certificado TLS	• Encrypt Connection = True pero el servidor tiene un certificado autofirmado. • Trust Server Certificate = False en desarrollo con certificado no válido.	1. En desarrollo: activar Trust Server Certificate = True en la credencial. 2. En producción: instalar un certificado válido firmado por una CA conocida. 3. Verificar TCP/IP habilitado en SQL Server Configuration Manager. 4. Confirmar puerto 1433 abierto en el firewall.	En producción nunca usar Trust Server Certificate = True. Instalar un certificado válido.

Categoría 3 — Webhooks

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-WH-01 Alto	El webhook no recibe eventos del servicio externo	• URL del webhook mal pegada en la plataforma externa. • El método HTTP en Saturn Studio no coincide con el del servicio externo (ej: GET vs POST). • El servicio externo requiere verificación inicial (challenge).	1. Ir a Webhooks en Saturn Studio. 2. Copiar la URL exacta del webhook desde el panel. 3. Pegarla de nuevo en la configuración del servicio externo. 4. Verificar que el HTTP Method coincida. 5. Si requiere verificación, usar Response Mode = Immediately.	Usar el panel de monitoreo de Webhooks en Saturn Studio para ver si los eventos están llegando aunque el agente falle.

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-WH-02 Medio	El webhook recibe el evento pero el agente no procesa	• El campo ‘Assign result to Variable’ no está configurado. • La variable de guardado no se usa correctamente en el flujo. • El formato del body es diferente al esperado (ej: form-data vs JSON).	1. Abrir monitoreo de Webhooks y hacer clic en el ojo azul del evento. 2. Revisar el campo Body para ver los datos exactos. 3. Comparar la estructura del JSON recibido con lo esperado. 4. Ajustar las referencias y el campo ‘Assign result to Variable’.	Siempre testear el webhook enviando un payload de prueba antes de conectar el flujo completo.
ERR-WH-03 Medio	El servicio externo reporta que el webhook no responde (timeout)	• El agente tarda demasiado en ejecutarse y el proveedor corta la conexión. • Response Mode está configurado como ‘After robot execution’ en lugar de ‘Immediately’.	1. Cambiar Response Mode a ‘Immediately’ para responder al instante. 2. Mover el procesamiento pesado después de confirmar la recepción. 3. Si se requiere un payload específico, usar ‘Send Webhook Response’.	La mayoría de proveedores tienen timeouts de 5 a 30 segundos. Responder inmediatamente y procesar en segundo plano.

Categoría 4 — Módulos de Inteligencia Artificial

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-AI-01 Alto	OpenAI devuelve error 429 – Rate limit exceeded	• Demasiadas peticiones enviadas en poco tiempo. • El plan de OpenAI tiene un límite bajo de requests por minuto (RPM). • Un loop llama a OpenAI sin esperas entre iteraciones.	1. Agregar una moon de espera (Wait / Sleep) en los loops. 2. Considerar aumentar el plan de OpenAI si el uso es alto. 3. Implementar patrón de reintento con espera exponencial (1s, 2s, 4s). 4. Revisar la cantidad de llamadas en el log.	Nunca llamar a APIs de IA en loops sin control de cadencia. Agregar siempre una espera mínima de 1 segundo entre llamadas.
ERR-AI-02 Medio	La respuesta de OpenAI viene truncada o incompleta	• El parámetro max_tokens es demasiado bajo. • El texto enviado como contexto supera el límite del modelo. • El prompt no instruye a responder de forma compacta.	1. Aumentar el valor de max_tokens en el comando. 2. Recortar o dividir el texto de entrada en chunks si es muy largo. 3. Usar modelos con mayor ventana de contexto (ej: gpt-4-turbo). 4. Revisar el conteo estimado de tokens antes de enviar.	Diseñar prompts compactos y establecer max_tokens acorde a la respuesta esperada. Para JSON estructurado, 500 tokens suelen ser suficientes.
ERR-AI-03 Medio	La respuesta de la IA no tiene el formato JSON esperado	• El modelo no responde en JSON aunque se le solicite. • Hay texto adicional antes o después del bloque JSON. • El modelo agregó comillas o backticks alrededor del JSON.	1. Agregar al prompt: ‘Responde ÚNICAMENTE con un JSON válido, sin texto adicional, sin backticks’. 2. Agregar una limpieza de texto posterior (remover comillas o marcas del formato). 3. Usar una moon JSON Parse para validar. 4. Especificar el schema exacto con un ejemplo.	Incluir siempre un ejemplo del JSON esperado en el prompt y especificar que no se incluya ningún texto adicional.

Categoría 5 — Flujo y Lógica del Agente

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-FLOW-01 Medio	El agente termina sin ejecutar todas las moons esperadas	• Una condición If/Else redirige el flujo a un End anticipado. • Una variable nula o vacía hace fallar una condición. • El flujo no está correctamente conectado (moons sueltas).	1. Agregar moons de Log antes y después de cada If/Else. 2. Verificar en el diagrama que todas las moons estén encadenadas. 3. Revisar las condiciones con valores reales del log. 4. Agregar valores por defecto a variables nulas.	Siempre hacer clic derecho → Ejecutar robot con datos de prueba reales antes de pasar a producción.
ERR-FLOW-02 Alto	El agente procesa el mismo registro varias veces	• El trigger del agente se ejecuta más de una vez para el mismo evento. • No hay mecanismo de deduplicación en el flujo. • Un webhook está siendo reintentado por el proveedor.	1. Guardar el ID del evento en la base de datos al procesarlo. 2. Al inicio del flujo, verificar si el ID ya existe en la tabla. 3. Si existe, terminar el flujo de inmediato sin procesar. 4. Revisar la configuración de reintentos externa.	Todo agente que procese eventos externos debe tener deduplicación. Crear una tabla ‘eventos_procesados’ con el ID y la fecha.

Código y Severidad	Error / Problema	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-FLOW-03 Medio	Las variables tienen valores de ejecuciones anteriores	• Las variables no se reinician al iniciar la ejecución. • Se reutiliza una variable global modificada por otra ejecución paralela.	1. Agregar una moon al inicio que asigne valores iniciales a las variables críticas. 2. Definir como variables locales (no globales) las que deban ser frescas. 3. Revisar si hay interferencia por ejecuciones paralelas.	Inicializar siempre las variables críticas al comienzo de cada flujo. No asumir que estarán vacías.

Categoría 6 — Conectores específicos frecuentes

Código y Severidad	Conector / Canal	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-CONN-01 Alto	**Notion**	• La página o base de datos no fue compartida con la integración. • El Integration Token fue revocado o regenerado en Notion.	1. Abrir la página o BD en Notion. 2. Hacer clic en los tres puntos (•••) arriba a la derecha. 3. Seleccionar ‘Add connections’ y elegir Saturn Studio. 4. Verificar que el Token coincida.	Cada nueva página o base de datos en Notion debe compartirse manualmente con la integración. No se comparte automáticamente.

Código y Severidad	Conector / Canal	Causas probables	Solución paso a paso	Cómo prevenirlo
ERR-CONN-02 Alto	**WhatsApp** (Evolution API)	• La sesión de WhatsApp Web venció. • El servidor de Evolution API no está accesible desde internet. • La instancia fue eliminada del servidor.	1. Usar el comando ‘Get QR Code’ para obtener uno nuevo. 2. Escanearlo desde el celular (Dispositivos vinculados). 3. Verificar que el servidor tenga IP pública y puerto 8080 abierto. 4. Si persiste, recrear la instancia.	Mantener el servidor de Evolution API siempre encendido. Las sesiones de WhatsApp Web se desconectan si el servidor no responde.
ERR-CONN-03 Alto	**Mercado Libre / Pago**	• El token OAuth venció (vigencia limitada). • El Redirect URI no coincide con https://studio.rocketbot.com. • Los scopes de la app no incluyen la operación.	1. Re-autenticar la credencial (eliminar y recrear en Saturn Studio). 2. Verificar el Redirect URI en el portal de desarrolladores (sin barra final). 3. Revisar y añadir los scopes necesarios. 4. Guardar y esperar unos minutos.	Los tokens OAuth tienen expiración. Implementar un mecanismo de renovación o re-autenticar periódicamente.

Checklists de diagnóstico rápido

Checklist 1 — Credencial que no conecta

- Abrí la credencial en Saturn Studio e hice clic en Check.
- Leí el mensaje de error exacto que aparece.
- Copié la API Key nuevamente desde la plataforma de origen (sin espacios).
- Verifiqué que la cuenta del servicio tiene el plan que soporta API.
- Para OAuth: verifiqué que el Redirect URI es exactamente https://studio.rocketbot.com
- Para OAuth: verifiqué que los scopes/permisos están configurados correctamente.
- Esperé 5 minutos y volví a intentar (algunos cambios tardan en propagarse).

Checklist 2 — Agente que falla al ejecutarse

- Abrí el panel de Log y leí el mensaje de error completo.
- Identifiqué cuál moon falló (aparece en rojo en el diagrama).
- Usé clic derecho → Ejecutar solo ese comando para aislar el problema.
- Verifiqué que la credencial usada en esa moon está conectada.
- Revisé si hay variables nulas o con valores inesperados antes de esa moon.
- Agregué moons de Log antes y después de la moon fallida para ver los valores.
- Consulté la categoría de error correspondiente en esta guía.

Checklist 3 — Base de datos que no conecta

- El servidor tiene IP pública (NO uso localhost ni 127.0.0.1).
- El puerto está abierto: 3306 (MySQL), 5432 (PostgreSQL), 1433 (SQL Server).
- El usuario tiene permisos para conectarse desde IPs externas.
- El nombre de la base de datos es exactamente el correcto (sensible a mayúsculas).
- Usé el botón Test Connection antes de guardar la credencial.
- Revisé los logs del servidor de base de datos para errores de conexión rechazada.

Checklist 4 — Webhook que no dispara el agente

- Copié la URL del webhook desde el panel de Webhooks en Saturn Studio.
- Pegué la URL correctamente en la plataforma externa (sin espacios adicionales).
- El método HTTP configurado coincide con el que usa el proveedor.
- Revisé el panel de monitoreo de Webhooks para ver si los eventos están llegando.
- Hice clic en el ojo azul de un evento para ver el raw data recibido.
- El Response Mode está configurado como Immediately para proveedores con timeout corto.
- La variable de destino (Assign result to Variable) está configurada.

Recursos de soporte

Recurso	Dónde	Para qué sirve
Documentación oficial	docs.rocketbot.co	Documentación completa de todos los módulos y conectores.

Recurso	Dónde	Para qué sirve
Foro de la comunidad	forum.rocketbot.com	Preguntas y respuestas de la comunidad de usuarios Saturn Studio.
Soporte conversacional	Slack (incluido con licencia)	Canal de soporte en tiempo real con el equipo de Rocketbot.
Academy	academy.rocketbot.com	Cursos gratuitos y certificaciones técnicas, disponibles en español.
Panel de Webhooks	Saturn Studio → Webhooks	Monitoreo en tiempo real de eventos recibidos y raw data para debugging.

> *Consejo: Al escalar a soporte — Incluir siempre: nombre del agente, screenshot del log de error, módulo/conector involucrado y los pasos que ya se intentaron. Esto reduce el tiempo de resolución significativamente.*

Saturn Studio — Mejores Prácticas

Guía completa para desarrollar robots robustos, seguros y mantenibles en Saturn Studio

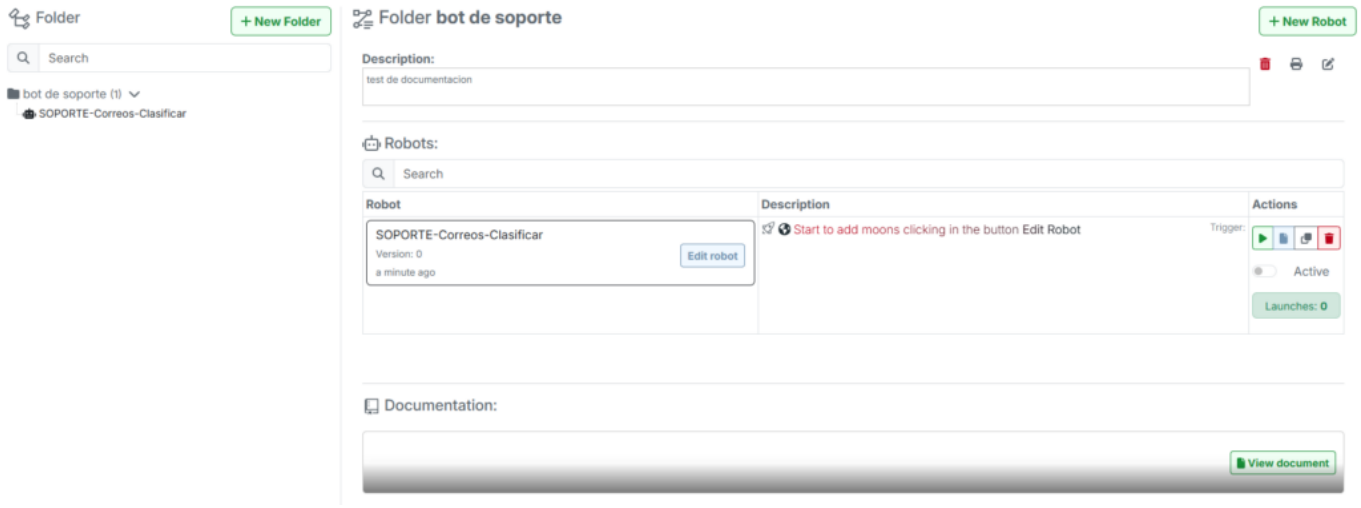
> **Nota: Estructura** — Este documento cubre 8 categorías de buenas prácticas: nomenclatura y organización, variables y datos, manejo de errores, seguridad y credenciales, rendimiento, prompts de IA, colaboración en equipo y paso a producción. Cada práctica incluye el porqué, el cómo y un ejemplo concreto.

1. Nomenclatura y Organización

Un robot bien nombrado y organizado se mantiene, se depura y se comparte mucho más fácilmente. Estas prácticas son el fundamento de cualquier proyecto de automatización profesional.

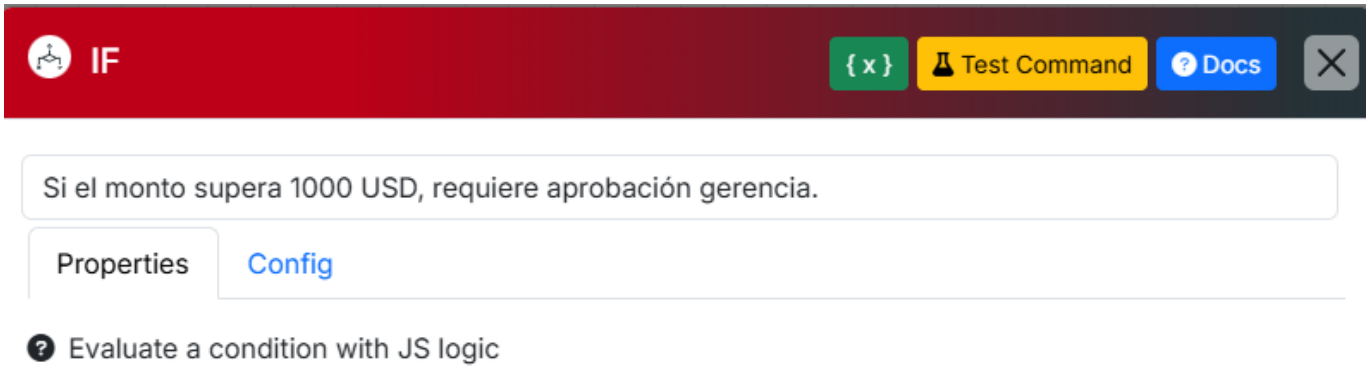
BP-01 | Nombra los robots con un patrón consistente

- **Por qué:** Un nombre claro reduce el tiempo de búsqueda y evita confusión entre robots similares. En equipos grandes, la falta de convención genera duplicados y ambigüedades.
- **Cómo:** Usa el patrón: [AREA]-[PROCESO]-[ACCION]. Evitar nombres genéricos como ‘bot1’ o ‘prueba_final_v3’.
- **Ejemplo:** Un robot que clasifica correos de soporte debería llamarse: SOPORTE-Correos-Clasificar. Así cualquier miembro del equipo entiende su función sin abrirlo.



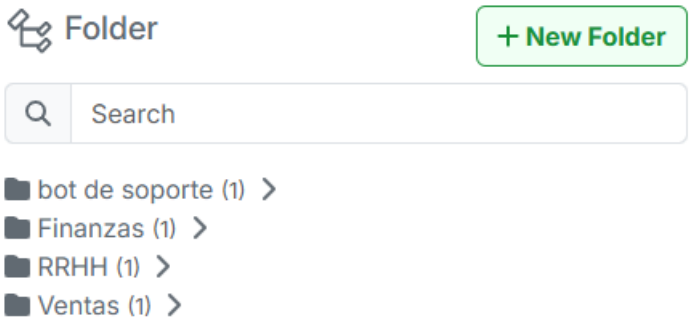
BP-02 | Documenta cada Moon con una descripción breve

- **Por qué:** En un flujo complejo, volver a entender la lógica de una Moon sin descripción puede tomar horas. La documentación inline es la diferencia entre un robot mantenible y uno descartable.
- **Cómo:** Usa el campo de descripción o notas de cada Moon. Una línea es suficiente: # Clasifica el correo en: consulta / reclamo / spam. Para Moons críticas agrega el contexto de por qué se toma esa decisión.
- **Ejemplo:** Una Moon de tipo If/Else sin descripción es un enigma. Con la descripción #Si el monto supera 1000 USD, requiere aprobación gerencia es autoexplicativa.
- ***Si es la primera vez que utiliza las Moons de Saturn Studio***, acceda a la **[Guía de introducción y uso de Moons]** para aprender a configurarlas.



BP-03 | Organiza los robots en carpetas por área de negocio

- **Por qué:** A medida que crecen los robots, un workspace sin organización se vuelve imposible de navegar. Las carpetas permiten permisos granulares y búsqueda eficiente.
- **Cómo:** Crear carpetas por área: Finanzas, RRHH, Ventas, TI, Operaciones.
- **Ejemplo:** No mezclar robots de Finanzas con robots de RRHH en la misma carpeta. Un nuevo integrante del equipo debe poder ubicar cualquier robot en menos de 30 segundos.



2. Variables y Datos

El manejo correcto de variables es la diferencia entre un robot que funciona en cualquier entorno y uno que solo funciona bajo condiciones específicas.

Nomenclatura de variables

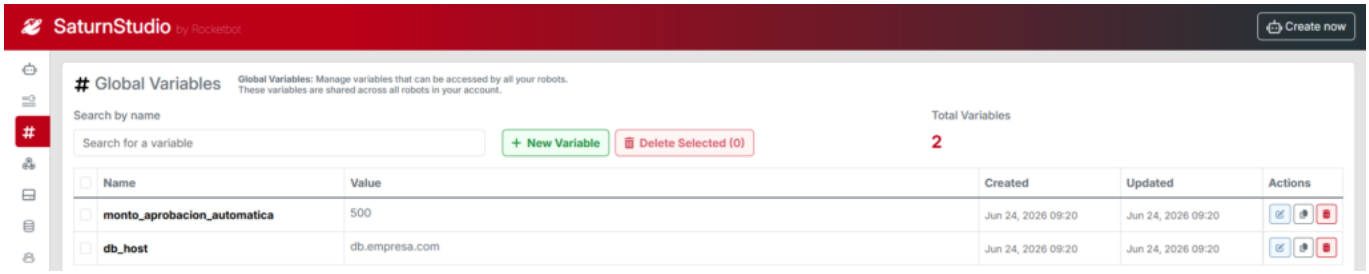
✅ Hacer	❌ No hacer
Usar nombres descriptivos en minúsculas con guion bajo: monto_factura, correo_cliente, resultado_clasificacion	Usar nombres de una letra (a, x, tmp) o abreviaciones ambiguas (mc, rc, res)
Usar MAYÚSCULAS para variables globales y constantes: API_URL, DB_HOST, SLACK_CHANNEL	Mezclar globales y locales con el mismo patrón de nombre
Inicializar las variables críticas al inicio del flujo con valores por defecto	Asumir que las variables estarán vacías o tendrán el valor esperado sin verificarlo
Limpiar variables que contienen datos sensibles (contraseñas, tokens) después de usarlas	Dejar tokens o contraseñas en variables locales durante toda la ejecución del robot

BP-04 | Valida siempre el resultado de una Moon antes de usarlo

- **Por qué:** Un comando puede retornar null, un objeto vacío o un error silencioso. Si el siguiente paso asume que el dato está bien y no lo está, el robot falla de forma inesperada y difícil de depurar.
- **Cómo:** Después de cada Moon que retorna un dato crítico, agregar una Moon If/Else que verifica: ¿el resultado está vacío? ¿es null? ¿tiene el campo esperado? Si falla la validación, terminar con un mensaje de error descriptivo.
- **Ejemplo:** Después de leer correos con Gmail, verificar que el array de resultados no esté vacío antes de intentar acceder a resultado[0]. Si está vacío, loggear 'Sin correos nuevos' y terminar el flujo.

BP-05 | Usa Variables Globales para configuración compartida

- **Por qué:** Si diez robots distintos usan la misma URL de API o el mismo host de base de datos, y ese valor cambia, tendrás que editar diez robots. Con una Variable Global, el cambio es en un solo lugar.
- **Cómo:** Crear Variables Globales para: URLs de APIs internas, hostnames de bases de datos, umbrales de decisión (ej: monto_aprobacion_automatica = 500), emails de notificación de errores.
- **Ejemplo:** Variable Global db_host = 'db.empresa.com'. Si el servidor migra, solo se actualiza esa variable y los 15 robots que la usan se adaptan automáticamente sin editar ningún flujo.
- ***Si es la primera vez que utiliza variables globales***, acceda a la **[Guía de introducción y uso de Variables Globales]** para aprender a configurarlas.



3. Manejo de Errores

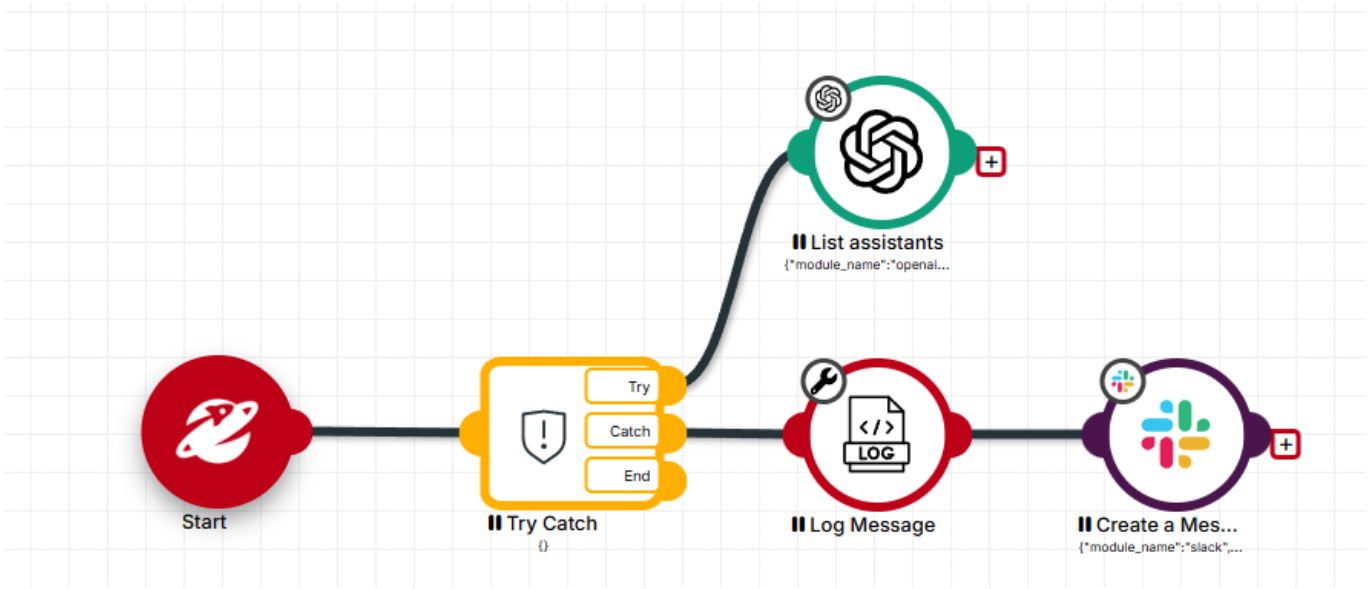
Un robot en producción sin manejo de errores es una bomba de tiempo. Los errores son inevitables: APIs que fallan, bases de datos que se saturan, archivos que no llegan. Lo que distingue a un robot profesional es cómo los maneja.

BP-06 | Todo robot de producción debe tener un bloque Try/Catch

- **Por qué:** Sin manejo de errores, cualquier falla detiene el robot silenciosamente o con un error crudo en el log. El equipo no se entera,

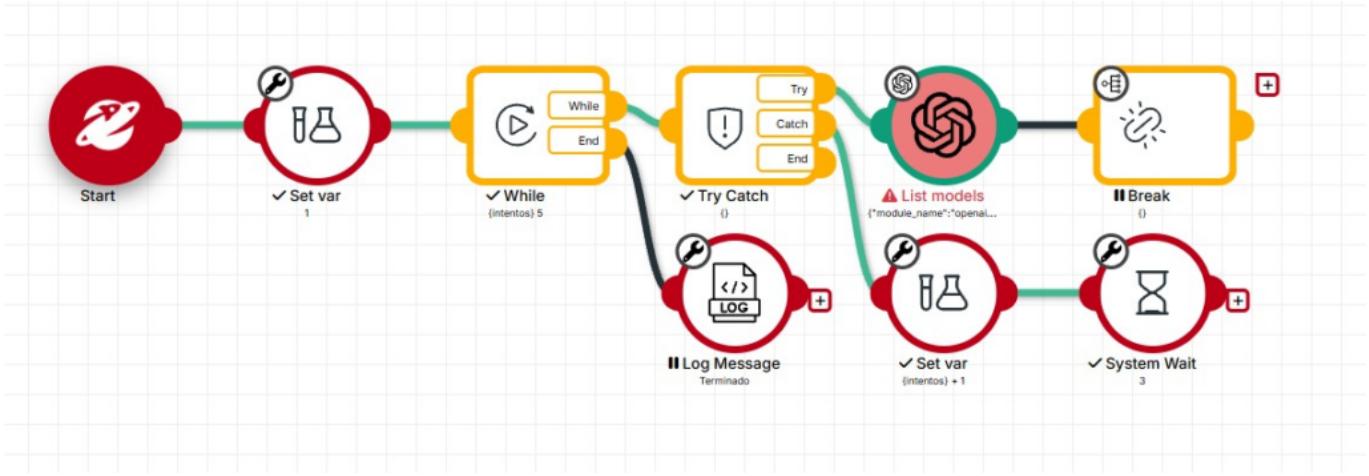
el proceso queda a medias y el diagnóstico es difícil.

- **Cómo:** Envolver el flujo principal en un bloque Try/Catch. En el bloque Catch: 1) Loggear el error con fecha, hora y nombre del robot. 2) Enviar alerta al equipo (Teams/Discord/Gmail). 3) Terminar el flujo de forma controlada.
- **Ejemplo:** El robot VENTAS-Leads-Clasificar falla al llamar a OpenAI. Sin Catch: el proceso queda incompleto y nadie lo sabe. Con Catch: se envía un mensaje a #alertas-saturn con 'Error en clasificación de leads a las 08:15 — Rate limit exceeded'.



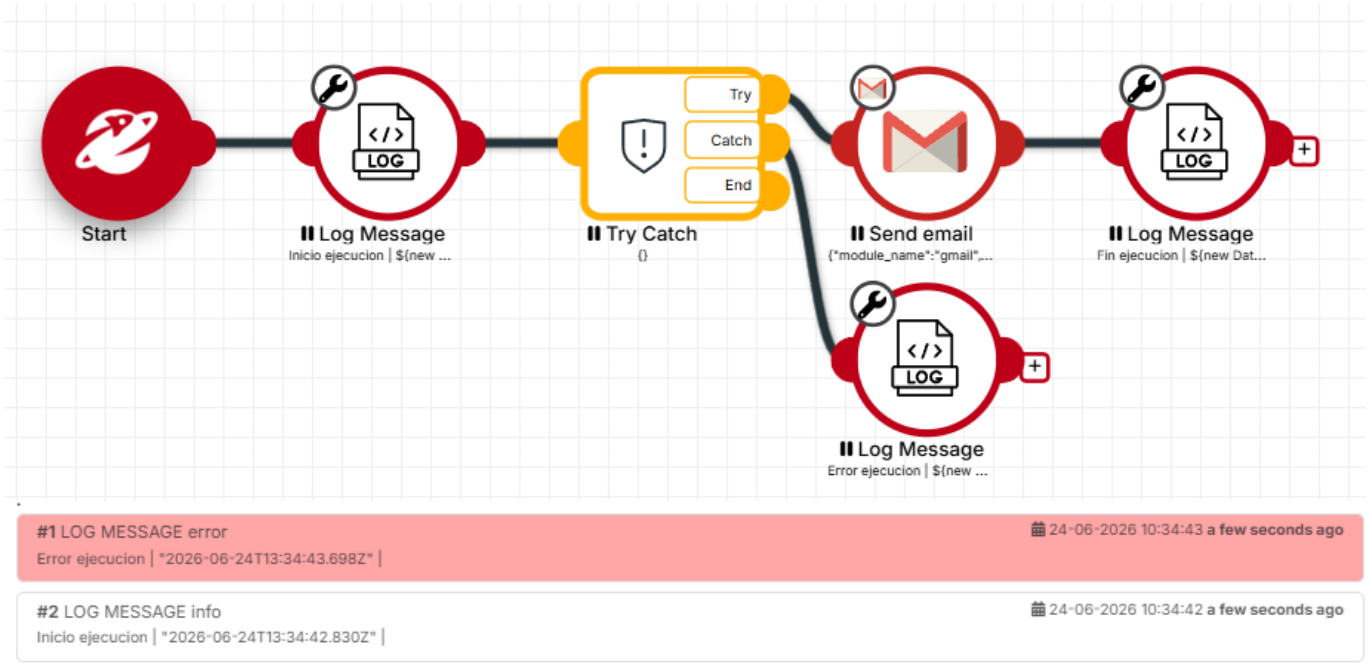
BP-07 | Implementa reintentos con espera para llamadas a APIs externas

- **Por qué:** Las APIs externas (OpenAI, Gmail, Mercado Pago) pueden fallar por sobrecarga temporal, rate limits o problemas de red. Un reintento inmediato suele fallar igual. Un reintento con espera tiene alta probabilidad de éxito.
- **Cómo:** Patrón de reintento: variable reintentos = 0. Mientras reintentos < 3: intentar llamada, si falla: reintentos++, esperar (reintentos * 5) segundos, continuar loop. Si llega a 3 sin éxito: activar Catch.
- **Ejemplo:** OpenAI retorna error 429 (rate limit). El robot espera 3 segundos y reintenta. Al tercer intento exitoso, el flujo continúa normalmente. Sin este patrón, el proceso falla completamente.



BP-08 | Incluye Moons de Log estratégicas en todo el flujo

- **Por qué:** En producción no hay interfaz gráfica. El único registro de lo que pasó es el log. Sin logs, diagnosticar un fallo puede tomar días. Con logs bien ubicados, el problema es visible en minutos.
- **Cómo:** Agregar Log al inicio del flujo (datos de entrada), antes y después de cada operación crítica (BD, API, IA) y al finalizar (resultado y métricas). Formato recomendado: [robot] [ACCION] [RESULTADO]. Incluir siempre timestamp.
- **Ejemplo:** [FIN-Facturas-Emitir] Inicio ejecucion | 2026-05-18 08:15:00 | registros_pendientes=47. Luego: [FIN-Facturas-Emitir] Factura emitida OK | id=INV-2841 | cliente=Empresa SA. Al final: [FIN-Facturas-Emitir] Fin | exito=45 | errores=2.



4. Seguridad y Credenciales

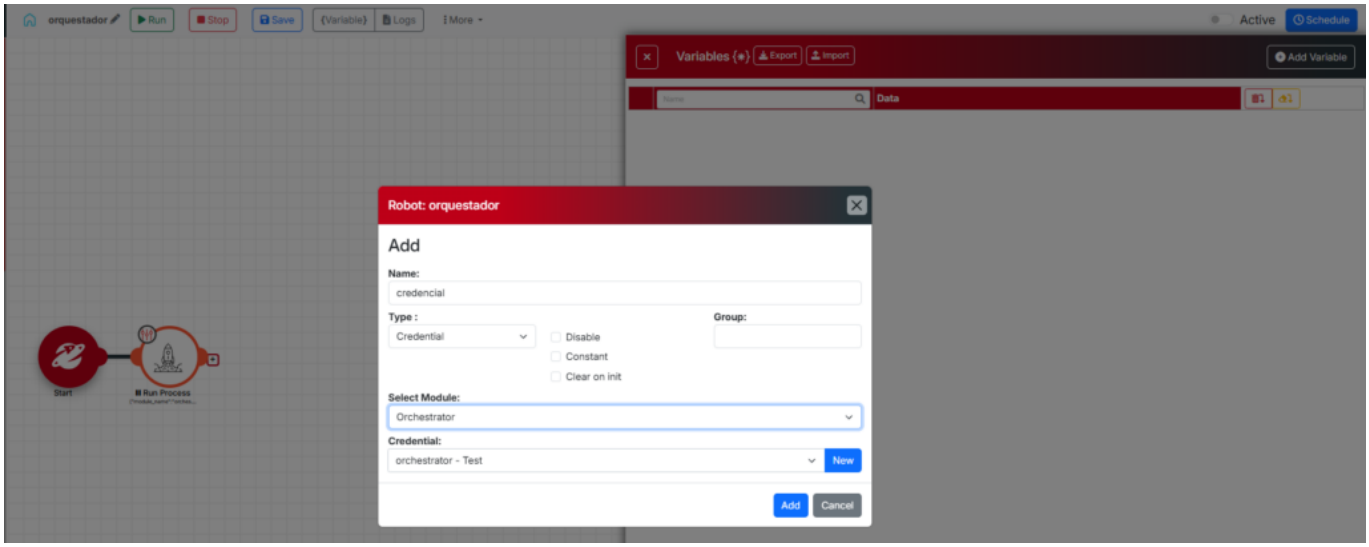
La seguridad en automatización no es opcional. Un robot con credenciales expuestas o permisos excesivos es un riesgo crítico para la empresa.

Checklist de seguridad — cumplir antes de pasar a producción

- [ ] **Nunca** escribir contraseñas, API Keys o tokens directamente en los campos de los comandos. Siempre usar credenciales del vault de Saturn Studio.
- [ ] **Nunca** compartir el archivo .json de un robot que contenga credenciales. Borrar las credenciales del robot antes de exportarlo o compartirlo.
- [ ] **Crear** usuarios de base de datos dedicados para Saturn Studio con permisos mínimos (solo las tablas y operaciones necesarias). No usar root ni sa.
- [ ] **Rotar** las API Keys cada 6 meses o inmediatamente si se sospecha una exposición. Actualizar la credencial en Saturn Studio tras la rotación.
- [ ] **Anotar** la fecha de creación de todas las API Keys del Orquestador (vigencia 2 años) en un registro interno para anticipar el vencimiento.
- [ ] **Para bases de datos:** abrir solo el puerto necesario (3306, 5432, 1433) y solo desde las IPs de Saturn Studio si el proveedor lo permite.
- [ ] **Revisar** periódicamente que las credenciales compartidas solo estén accesibles a los usuarios que las necesitan activamente.
- [ ] **Limpiar** variables que contienen datos sensibles (tokens, contraseñas temporales) al finalizar el flujo que las utilizó.

BP-09 | Usa Variables de tipo Credential para parametrizar el acceso

- **Por qué:** Si un robot usa la credencial directa y esa credencial cambia o se necesita usar otra (ej: cambiar de ambiente dev a prod), hay que editar el robot. Con una variable de credencial, el cambio es en la variable.
- **Cómo:** En lugar de seleccionar la credencial directamente en el campo del comando, crear una Variable de tipo Credential que la referencie. En el campo del comando escribir {nombre_variable}. Para cambiar de ambiente, solo cambiar el valor de la variable.
- **Ejemplo:** Variable: openai_key (tipo Credential). En desarrollo apunta a ‘OpenAI Dev’. En producción apunta a ‘OpenAI Prod’. El robot es idéntico en ambos ambientes. Solo cambia la variable.



5. Rendimiento y Eficiencia

Un robot lento puede ser tan problemático como uno que falla. El rendimiento impacta directamente en el costo de las APIs, en los tiempos de respuesta al usuario y en la capacidad de procesar grandes volúmenes.

Consultas a bases de datos

✅ Hacer	❌ No hacer
Limitar el resultado de las consultas SQL con WHERE y LIMIT, especialmente en desarrollo	Hacer SELECT * sin WHERE en tablas con millones de registros
Seleccionar solo las columnas necesarias: SELECT id, nombre, estado FROM clientes	Traer todas las columnas (SELECT *) cuando solo se necesitan 2 o 3 campos
Paginar grandes volúmenes: procesar en lotes de 100-500 registros por ejecución	Intentar procesar 10.000 registros en una sola ejecución (riesgo de timeout y consumo de memoria)
Crear índices en las columnas usadas en WHERE y JOIN para acelerar las consultas	Dejar tablas sin índices y asumir que las consultas serán rápidas por defecto

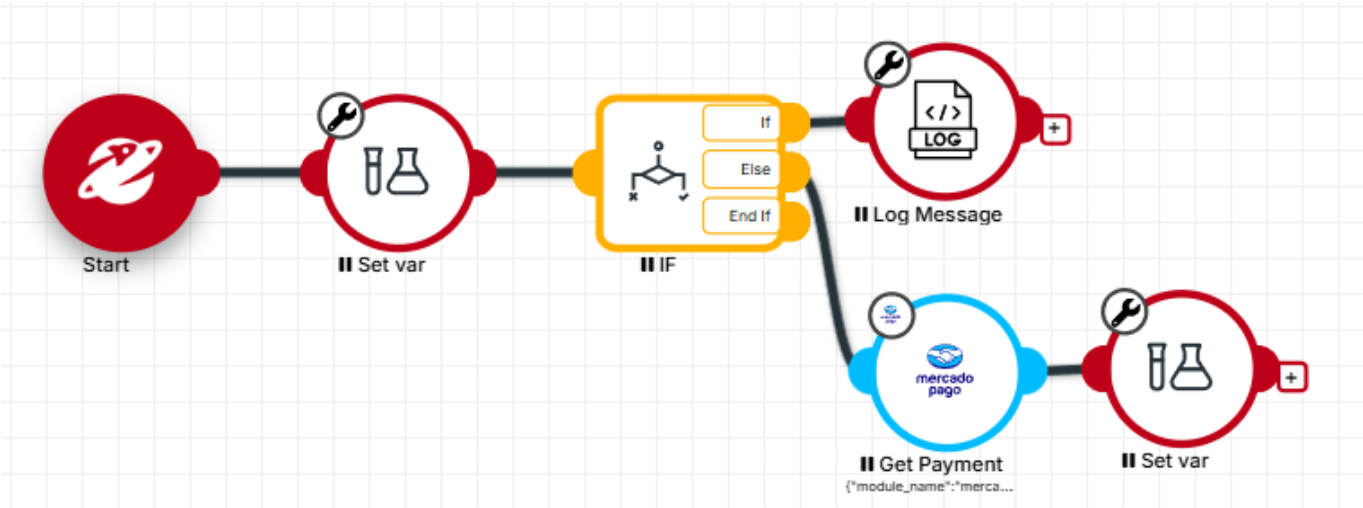
Llamadas a APIs de IA

✅ Hacer	❌ No hacer
Definir max_tokens acorde al tipo de respuesta esperada (JSON simple = 200-500, análisis = 1000-2000)	Dejar max_tokens en el valor máximo por defecto en todos los comandos
Agregar una Moon Wait de 1-2 segundos entre llamadas a IA en loops para respetar rate limits	Llamar a OpenAI en loops sin espera entre iteraciones (genera errores 429 masivos)
Reutilizar el resultado de una llamada a IA si se necesita en múltiples pasos siguientes	Llamar a la misma API de IA dos veces para obtener el mismo dato (doble costo y tiempo)

Para chunks de texto: dividir documentos en bloques de 500-1000 palabras antes de enviar	Enviar documentos PDF completos de 50 páginas en un solo prompt (supera el contexto del modelo)

BP-10 | Implementa deduplicación para robots que procesan eventos externos

- **Por qué:** Un webhook puede dispararse más de una vez para el mismo evento (red inestable, reintentos del proveedor). Sin deduplicación, el robot procesa el mismo pago, correo o formulario múltiples veces.
- **Cómo:** Crear una tabla eventos_procesados en la base de datos con los campos: event_id (PK), agent_name, processed_at. Al inicio de cada ejecución: 1) Extraer el ID único del evento. 2) Consultar si ya existe. 3) Si existe: terminar. 4) Si no: procesar e insertar.
- **Ejemplo:** Un webhook de Mercado Pago puede enviar el mismo payment_id varias veces si hay timeout en la primera entrega. Sin deduplicación: se emiten 3 facturas para el mismo pago. Con deduplicación: solo se emite una.



6. Prompts de Inteligencia Artificial

La calidad del prompt determina la calidad del resultado. Un prompt mal diseñado produce respuestas inconsistentes, en formato incorrecto o con alucinaciones que rompen el flujo.

BP-11 | Define el rol, el formato de salida y las restricciones en cada prompt

- **Por qué:** Sin estas tres cosas, el modelo puede responder en cualquier formato, inventar información o dar respuestas que el flujo no puede parsear. Un prompt completo produce resultados deterministas.
- **Cómo:** Estructura recomendada del system prompt:
  1. **ROL:** “Eres un clasificador de correos de soporte técnico.”
  2. **TAREA:** Descripción precisa de lo que debe hacer.

3. **FORMATO:** “Responde ÚNICAMENTE con un JSON válido: {"categoria": "...", "urgencia": "..."}.”
4. **RESTRICCIONES:** “No incluyas explicaciones. No uses markdown. Si no puedes clasificar, responde {"categoria":"desconocido"}.”

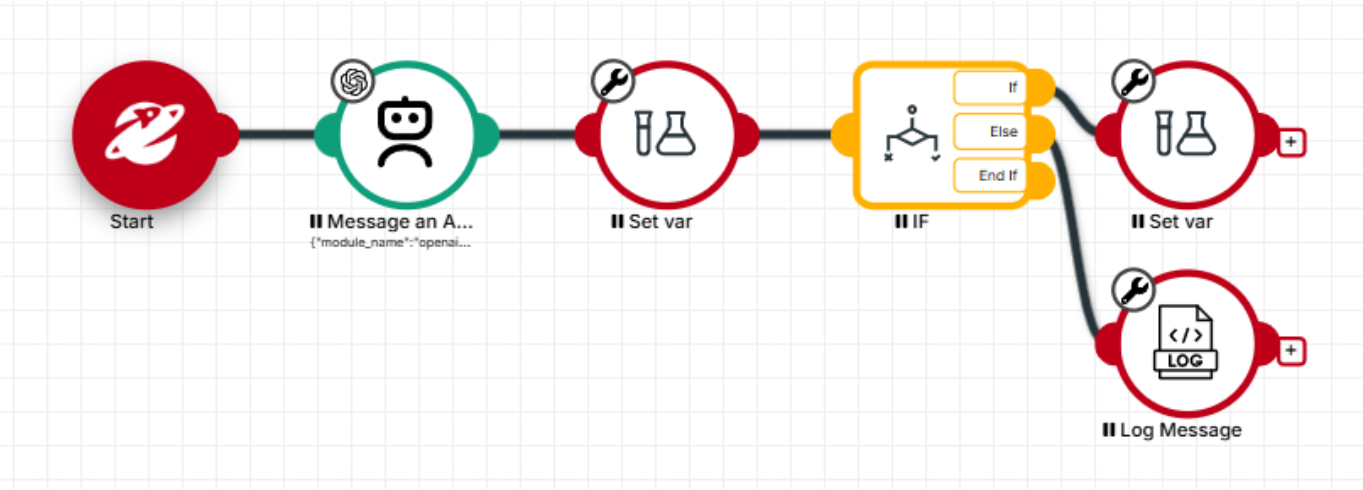
- **Ejemplo:** Sin estructura: el modelo responde “Este correo parece ser una consulta sobre facturación, con urgencia media.” — imposible de parsear. Con estructura: {"categoria": "facturacion", "urgencia": "media"} — directamente usable.

BP-12 | Incluye siempre un ejemplo del output esperado en el prompt

- **Por qué:** El *few-shot prompting* (dar un ejemplo) es la técnica más efectiva para obtener respuestas consistentes y en el formato exacto que el flujo necesita.
- **Cómo:** En el prompt agregar: EJEMPLO DE RESPUESTA CORRECTA: {"categoria": "reclamo", "urgencia": "alta", "requiere_humano": true}. Si hay múltiples categorías posibles, listarlas explícitamente en el prompt.
- **Ejemplo:** Sin ejemplo: el modelo puede devolver categoria: reclamo, urgencia: alta (sin JSON, sin comillas). Con ejemplo: el modelo replica el formato exacto. La fiabilidad del parseo pasa del 60% al 95%.

BP-13 | Agrega una Moon de validación del JSON retornado por la IA

- **Por qué:** Incluso con un prompt perfecto, la IA puede devolver texto adicional, un JSON malformado o campos con nombres ligeramente diferentes. Un robot robusto valida antes de usar.
- **Cómo:** Después del comando de IA: 1) Limpiar el texto (remover backticks, texto previo al JSON). 2) Parsear el JSON a objeto. 3) Verificar con If/Else que los campos requeridos existen y tienen valores válidos. 4) Si la validación falla: reintentar el prompt o escalar a humano.
- **Ejemplo:** La IA retorna: ```json {"categoria": "consulta"} ```. La Moon de limpieza extrae el JSON. La validación verifica que ‘categoria’ existe. Si falta ‘urgencia’, el robot usa un valor por defecto en lugar de fallar.



7. Colaboración en Equipo

Saturn Studio está diseñado para trabajo en equipo. Estas prácticas maximizan la eficiencia colaborativa y minimizan los conflictos y la duplicación de trabajo.

BP-14 | Publica templates de robots bien probados en el Market interno

- **Por qué:** Si un robot funciona bien para un caso de uso frecuente, publicarlo como template evita que otros miembros del equipo lo reconstruyan desde cero. El Market interno es el repositorio de conocimiento del equipo.
- **Cómo:** Al finalizar y probar un robot: 1) Limpiar las credenciales hardcodeadas. 2) Documentar el propósito en la descripción del template (max 120 caracteres). 3) Exportar como .json. 4) Subir al Market con nombre descriptivo y categoría correcta. 5) Notificar al equipo.
- **Ejemplo:** El equipo de Finanzas crea el robot FIN-Facturas-Emitir y lo publica en el Market. El equipo de Ventas lo usa como base para FIN-Cotizaciones-Emitir, ahorrando 4 horas de desarrollo.

BP-15 | Usa Variables de tipo Credential compartidas por equipo para servicios comunes

- **Por qué:** Si cada miembro del equipo crea su propia credencial de OpenAI o Gmail, cuando la clave cambia cada uno debe actualizarla por separado. Con credenciales compartidas por equipo, el cambio es centralizado.
- **Cómo:** El administrador crea la credencial ‘OpenAI-Empresa’ en el vault y la comparte con el equipo completo vía ‘Share by Team’. Todos los robots del equipo referencian esta credencial. Si la clave rota, el admin la actualiza una vez.
- **Ejemplo:** Sin credencial compartida: 8 miembros del equipo tienen 8 credenciales de OpenAI distintas. Cuando OpenAI rota las claves, son 8 actualizaciones manuales con riesgo de que alguna quede desactualizada.

8. Paso a Producción

El paso de desarrollo a producción es el momento de mayor riesgo. Estas prácticas minimizan la probabilidad de incidentes y aceleran el diagnóstico si ocurren.

Checklist de Go-Live — completar antes de activar un robot en producción

- [ ] El robot fue probado con datos reales (no solo datos de prueba) en un ambiente de certificación o UAT.
- [ ] Todas las credenciales apuntan a cuentas de producción (no a cuentas de desarrollo o prueba).
- [ ] Las Variables Globales usan los valores de producción (DB_HOST, API_URL, etc.).
- [ ] El robot tiene manejo de errores (Try/Catch) y envía alertas al equipo si falla.
- [ ] El robot tiene Moons de Log al inicio, en puntos críticos y al final de cada ejecución.
- [ ] Se implementó deduplicación si el robot procesa eventos externos (webhooks, formularios).
- [ ] Los volúmenes de datos probados son representativos del volumen de producción (no probar con 5 registros si en producción hay 5000).
- [ ] Existe un procedimiento documentado para detener el robot si algo sale mal (quién tiene acceso, cómo hacerlo).
- [ ] Se definió quién es el responsable de monitorear el robot en las primeras 48 horas post-lanzamiento.
- [ ] Las credenciales de bases de datos usan el principio de mínimo privilegio (solo SELECT/INSERT/UPDATE en las tablas necesarias).
- [ ] El robot fue revisado por al menos otro miembro del equipo antes del go-live.

BP-16 | Implementa un ambiente de desarrollo separado del de producción

- **Por qué:** Probar cambios directamente en producción es el error más costoso en automatización. Un ambiente de desarrollo permite iterar libremente sin riesgo de afectar datos reales o procesos activos.
- **Cómo:** Usar Variables de tipo Credential para separar ambientes: variable db_connection en dev apunta a BD de prueba, en prod apunta a BD real. Variable api_url en dev apunta a sandbox del servicio, en prod al endpoint real. Cambiar de ambiente = cambiar el valor de la variable.
- **Ejemplo:** El robot FIN-Facturas-Emitir en dev apunta a la API de pruebas de Nubox (UAT) y a una base de datos de test. En prod apunta a la API real y la BD de producción. El flujo es idéntico, solo cambian las variables.

BP-17 | Monitorea las primeras ejecuciones en producción activamente

- **Por qué:** Las primeras ejecuciones en producción siempre revelan casos bordes que no aparecieron en pruebas: datos en formatos inesperados, volúmenes mayores al estimado, interacciones con sistemas que se comportan diferente.
- **Cómo:** Durante las primeras 48 horas post-lanzamiento: 1) Revisar el log de cada ejecución manualmente. 2) Comparar los resultados con el proceso manual que se estaba haciendo antes. 3) Tener el robot pausado si se detecta cualquier anomalía. 4) Documentar los casos bordes encontrados para ajustar el robot.
- **Ejemplo:** El robot de clasificación de correos funciona perfectamente en pruebas. En producción aparecen correos en otro idioma (no esperado) que la IA no puede clasificar. Sin monitoreo activo: se acumulan sin procesar. Con monitoreo: se detecta el día 1 y se agrega un caso al prompt.

Resumen — Las 17 Mejores Prácticas

Código	Categoría	Práctica
BP-01	Nomenclatura	Nombra los robots con patrón [AREA]-[PROCESO]-[ACCION]
BP-02	Documentación	Documenta cada Moon con una descripción breve de su propósito
BP-03	Organización	Organiza los robots en carpetas por área y estado
BP-04	Variables	Valida el resultado de cada Moon crítica antes de usarlo
BP-05	Variables	Usa Variables Globales para configuración compartida entre robots
BP-06	Errores	Todo robot de producción debe tener un bloque Try/Catch
BP-07	Errores	Implementa reintentos con espera para llamadas a APIs externas
BP-08	Errores	Incluye Moons de Log estratégicas con timestamp en todo el flujo
BP-09	Seguridad	Usa Variables de tipo Credential para parametrizar el acceso por ambiente
BP-10	Rendimiento	Implementa deduplicacion para robots que procesan eventos externos
BP-11	Prompts IA	Define rol, formato de salida y restricciones en cada prompt
BP-12	Prompts IA	Incluye siempre un ejemplo del output esperado (*few-shot prompting*)
BP-13	Prompts IA	Agrega una Moon de validación del JSON retornado por la IA
BP-14	Equipo	Publica templates de robots bien probados en el Market interno
BP-15	Equipo	Usa credenciales compartidas por equipo para servicios comunes

BP-16	Producción	Implementa un ambiente de desarrollo separado del de producción
BP-17	Producción	Monitorea activamente las primeras 48 horas post-lanzamiento

---

Saturn Studio — Guía de introducción

Qué es • Cuándo usarlo y cuándo no • Conceptos clave

1. Qué es y para qué sirve

Definición sencilla:

> **Saturn Studio** es la plataforma en la nube (*cloud*) de Rocketbot que te permite diseñar, ejecutar y controlar agentes de automatización que combinan Inteligencia Artificial (IA), automatización tradicional (RPA) y tus propias reglas de negocio en un solo flujo visual.

El problema que resuelve:

Normalmente, los procesos de una empresa no son solo mecánicos. A veces necesitas leer un correo, "entender" qué quiere el cliente usando IA, buscar algo en una base de datos, tomar una decisión y avisar a tu equipo. Todo esto suele requerir que una persona esté pegada a la pantalla. Saturn Studio permite unir todos estos pasos para que ocurran solos, sin intervención humana, de inicio a fin.

Diferencias con el RPA Clásico (RPA Studio vs Saturn Studio)

Para entenderlo de forma sencilla: el RPA clásico destaca por imitar lo que hace una persona en su pantalla local (hacer clics, abrir programas instalados). Saturn Studio, en cambio, está pensado para trabajar directamente en la nube, procesar datos complejos y tomar decisiones inteligentes usando IA.

Aquí te explicamos la comparación punto por punto:

Comparación: RPA Studio (Clásico) vs. Saturn Studio (Agentivo)

Característica	RPA Studio (Clásico)	Saturn Studio (Agentivo)	¿Qué significa técnicamente?
Tipo de datos	Estructurados y predecibles (Excel, bases de datos fijas).	Estructurados y **No estructurados** (Emails, audios, PDFs variables).	Los datos **no estructurados** requieren "comprensión" (como un texto libre) para ser procesados.
Decisiones	Reglas fijas y estrictas (If/Else tradicionales).	Lógica híbrida: reglas fijas + criterio cognitivo de IA.	El robot puede evaluar contextos y tonos de mensajes gracias a los modelos de lenguaje.
Integración IA	Módulo adicional externo (no nativo).	**Nativa**: OpenAI, Gemini, Grok, Bedrock, Mistral, etc.	Trae conectores directos que se arrastran y configuran visualmente en el flujo.
Humano en flujo	No soportado por defecto.	**Soporte nativo** (*Human-in-the-loop*).	El robot puede pausar su ejecución y esperar que una persona apruebe algo para continuar.
Entorno	Instalado en máquina o servidor local.	**100% Cloud** (en la nube).	Diseñas y ejecutas todo desde cualquier navegador web sin instalar nada.
Ideal para	Tareas repetitivas y clics en pantallas de escritorio.	Procesos complejos, interpretación de textos e integración de servicios web.	Se enfoca en la interconexión moderna de sistemas en la nube (APIs).

Casos típicos de uso por industria

Área / Industria	Lo que hace el agente de Saturn Studio	Herramientas que conecta
Finanzas	Lee facturas en PDF con IA, extrae montos/proveedores, registra en el ERP, emite el documento y avisa al contador.	PDFs + ERP + Nubox + Email
Ventas	Clasifica leads por urgencia con IA, crea tareas de seguimiento, agenda reuniones y envía correos de bienvenida.	Web + IA + Asana + Zoom + Gmail

Área / Industria	Lo que hace el agente de Saturn Studio	Herramientas que conecta
Soporte	Monitorea bandejas de entrada, clasifica el tipo de ticket con IA, responde autónomamente o deriva casos críticos a un humano.	Zendesk + IA + Email corporativo
E-commerce	Detecta preguntas nuevas de clientes, redacta la respuesta ideal según la descripción del producto y responde en segundos.	Mercado Libre + IA (OpenAI/Gemini)
Recursos Humanos	Recibe CVs, la IA extrae la experiencia del candidato, los compara con el perfil buscado y genera un ranking para el reclutador.	Email + PDFs de CVs + IA
TI / DevOps	Detecta actualizaciones de código (*Pull Requests*), usa IA para resumir los cambios técnicos y asigna revisores automáticamente.	GitHub + IA + Discord / Slack

2. Cuándo usarlo y cuándo no

La regla de decisión es muy fácil:

> ¿Tu proceso requiere **“entender”** o **“interpretar”** información, y no solo copiar y pegar datos de forma mecánica?

- Si la respuesta es **SÍ**, usa Saturn Studio.
- Si es pura ejecución mecánica y predecible, RPA Studio es más que suficiente.

✅ **SÍ usa Saturn Studio cuando…**	❌ **NO usa Saturn Studio cuando…**
El proceso maneja correos libres, PDFs con diseños variables, audios o imágenes.	El proceso es 100% plano y repetitivo (ej. pasar celdas de un Excel a otro idéntico).
Necesitas que una IA tome decisiones intermedias o redacte contenido personalizado.	Requiere hacer clics visuales en programas antiguos de escritorio (ej. SAP GUI local).
Un supervisor humano deba autorizar un paso crítico antes de continuar (*Human-in-the-loop*).	El volumen del proceso es extremadamente bajo y no justifica el tiempo de desarrollo.
Quieres conectar múltiples herramientas web (Gmail, OpenAI, Asana, Slack) en un solo flujo.	La empresa trabaja en un entorno de red totalmente cerrado y sin acceso a internet.
Deseas crear un chatbot inteligente basado en documentos o manuales internos.	El proceso depende exclusivamente de la interacción con el sistema operativo local.

3. Conceptos clave

Para dominar Saturn Studio, solo necesitas entender estos 10 términos básicos:

1. MOON (La unidad mínima)

- **Qué es:** Es cada acción o comando individual que pones en tu flujo visual. Cada bloque (como “Leer correo”, “Preguntar a IA”, o “Enviar Slack”) es una Moon. Al conectarlas en fila, creas la ruta del robot.
- **Ejemplo:** Start → [Leer Gmail] → [Clasificar con OpenAI] → [If urgente] → [Crear tarea Asana] → End. Cada bloque es una Moon.

2. AGENTE (El robot completo)

- **Qué es:** Es el flujo completo que creas al conectar varias Moons. Es la automatización de un proceso completo de inicio a fin.
- **Ejemplo:** El agente SOPORTE-Tickets-Procesar que lee correos y genera tickets en Zendesk de forma autónoma.

3. MÓDULO (El conector)

- **Qué es:** Es el paquete de comandos específico para una herramienta externa. Agrupa todas las acciones que puedes hacer sobre ese servicio.
- **Ejemplo:** El módulo de OpenAI contiene acciones como “Generar Texto”, “Analizar Imagen”, etc.

4. WORKSPACE (El espacio de trabajo)

- **Qué es:** Es la oficina digital o carpeta principal donde tu equipo guarda todos sus agentes creados. Mantiene tus robots organizados y separados de otras áreas.
- **Ejemplo:** Un workspace para Rocketbot con subcarpetas para /Finanzas, /Ventas o /Soporte.

5. CREDENCIAL (Acceso seguro)

- **Qué es:** Es la herramienta que guarda tus contraseñas y claves de acceso (API Keys) de forma segura. Los robots las usan sin mostrar nunca las contraseñas en texto plano. Si una clave cambia, solo la actualizas aquí y se corrige para todos tus robots automáticamente.
- **Ejemplo:** La credencial OpenAI-Empresa que todos tus robots consumen de

forma encriptada.

6. VARIABLE (El mensajero de datos)

- **Qué es:** Son cajas virtuales que guardan la información que produce una Moon para pasársela a la siguiente. Pueden ser locales (de un solo robot) o Globales (compartidas por todos los robots de la empresa, ideales para guardar URLs de servidores o umbrales de decisión).
- **Ejemplo:** Guardar el texto de un correo en la variable correo y leerlo en la siguiente acción usando {correo}.

7. TRIGGER (El disparador)

- **Qué es:** Es el evento que le dice al robot: “¡Empieza a trabajar!”. Puede ser manual, programado por horario (cron) o activado al instante por un sistema externo (webhook).
- **Ejemplo:** El robot de cobros se activa inmediatamente mediante un webhook cuando Mercado Pago registra una compra.

8. HUMAN-IN-THE-LOOP (Pausa para humanos)

- **Qué es:** Es la capacidad del robot para pausar su flujo, enviarle un mensaje o formulario a una persona, esperar su respuesta y continuar trabajando con base en la decisión del humano. Es vital para procesos de alta responsabilidad.
- **Ejemplo:** Si una factura supera los $50,000 USD, el agente se pausa y envía un correo al Gerente. Solo cuando este hace clic en “Aprobar”, el agente continúa y emite el documento.

9. MARKET DE TEMPLATES (El mercado de plantillas)

- **Qué es:** Es un catálogo compartido dentro del espacio de trabajo donde los desarrolladores aprobados pueden publicar sus robots en formato .json para que cualquier miembro de la empresa los descargue y reutilice.
- **Ejemplo:** Descargar el template base FIN-Facturas-Emitir para no tener que configurar de cero la conexión con Nubox.

10. XPERIENCE (Formularios y colas de trabajo)

- **Qué es:** Es una herramienta adicional que permite crear pantallas de

formularios interactivos para que las personas ingresen datos que el robot procesará de manera secuencial a través de una “cola de tareas” organizada.
- **Ejemplo:** Un formulario de solicitud de vacaciones que almacena las peticiones en una cola para que el robot las valide y apruebe una a una de forma automática.

Jerarquía y relación de los conceptos:

Todos los elementos se estructuran y ordenan bajo el siguiente esquema:

```
[WORKSPACE] (Espacio de trabajo del equipo)
   └── [AGENTE] (El robot que hace el proceso completo)
         └── [MÓDULO] (El paquete de herramientas: Gmail, OpenAI, etc.)
               └── [MOON] (La acción o comando individual)
```

Servicios de soporte que ayudan a todo el flujo (Infraestructura transversal):

- **Credenciales** (Garantizan la seguridad)
- **Variables** (Transportan los datos)
- **File Storage** (Almacenamiento de archivos del robot)
- **Xperience** (Control de tareas e interacción con personas)

Saturn Studio — Tu primer flujo

Tutorial paso a paso para principiantes • Nivel 1

Antes de empezar

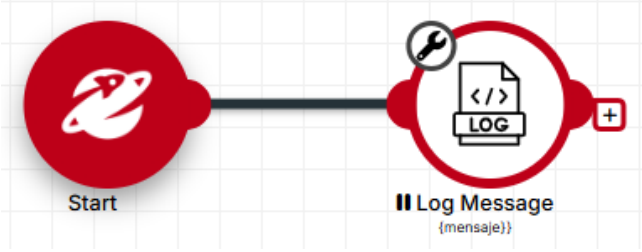
Parámetro	Detalle	¿Qué significa técnicamente?
Objetivo	Crear, configurar y ejecutar tu primer agente sin conocimientos técnicos previos.	Aprenderás a mover datos y entender cómo se despliega un flujo de principio a fin.

Parámetro	Detalle	¿Qué significa técnicamente?
Tiempo estimado	20 a 30 minutos.	Es un proceso rápido diseñado para familiarizarse con la interfaz.
Nivel	Principiante.	No necesitas saber programar; se utiliza lógica visual.

¿Qué vamos a construir?

Para aprender sin complicaciones, crearemos un agente muy simple que recibe un texto, lo procesa y registra el resultado en la pantalla. No utilizaremos credenciales externas ni configuraciones complejas.

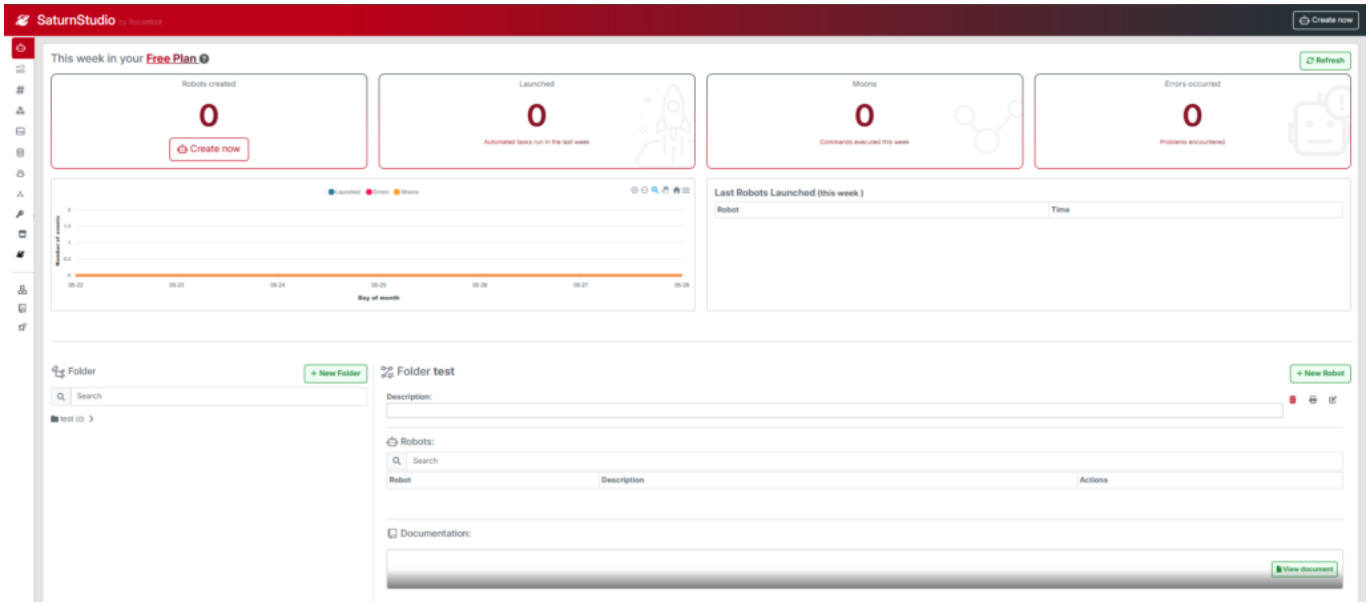
El mapa del flujo será: Start (Inicio) → Log: ¡Hola! → End (Fin).

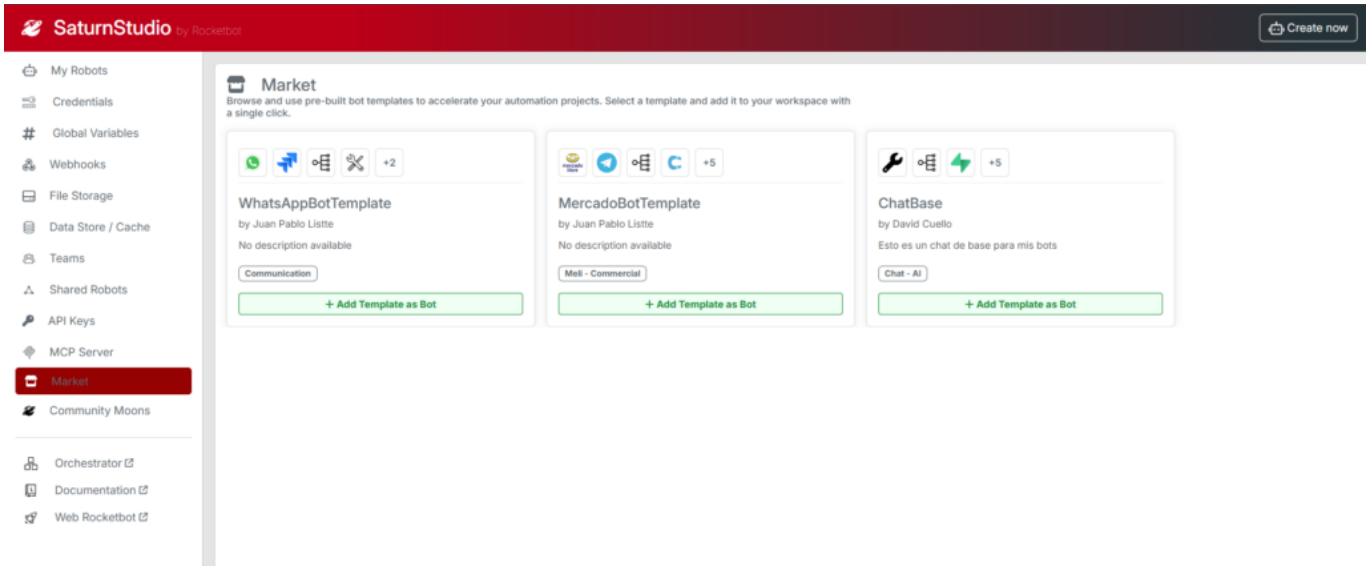


Guía del Tutorial Paso a Paso

Parte 1: Ingreso y Exploración

Paso	Acción en la plataforma	Explicación sencilla / Consejos
1	Abre el navegador e ingresa a **studio.rocketbot.com**.	Inicia sesión con tu correo y contraseña. Al entrar, verás tu **Dashboard** (tu escritorio digital privado).
2	Navega al **Market** desde el menú lateral izquierdo.	Aquí verás plantillas listas. Puedes hacer clic en cualquiera para abrir una vista previa en modo “solo lectura” para explorar su lógica.

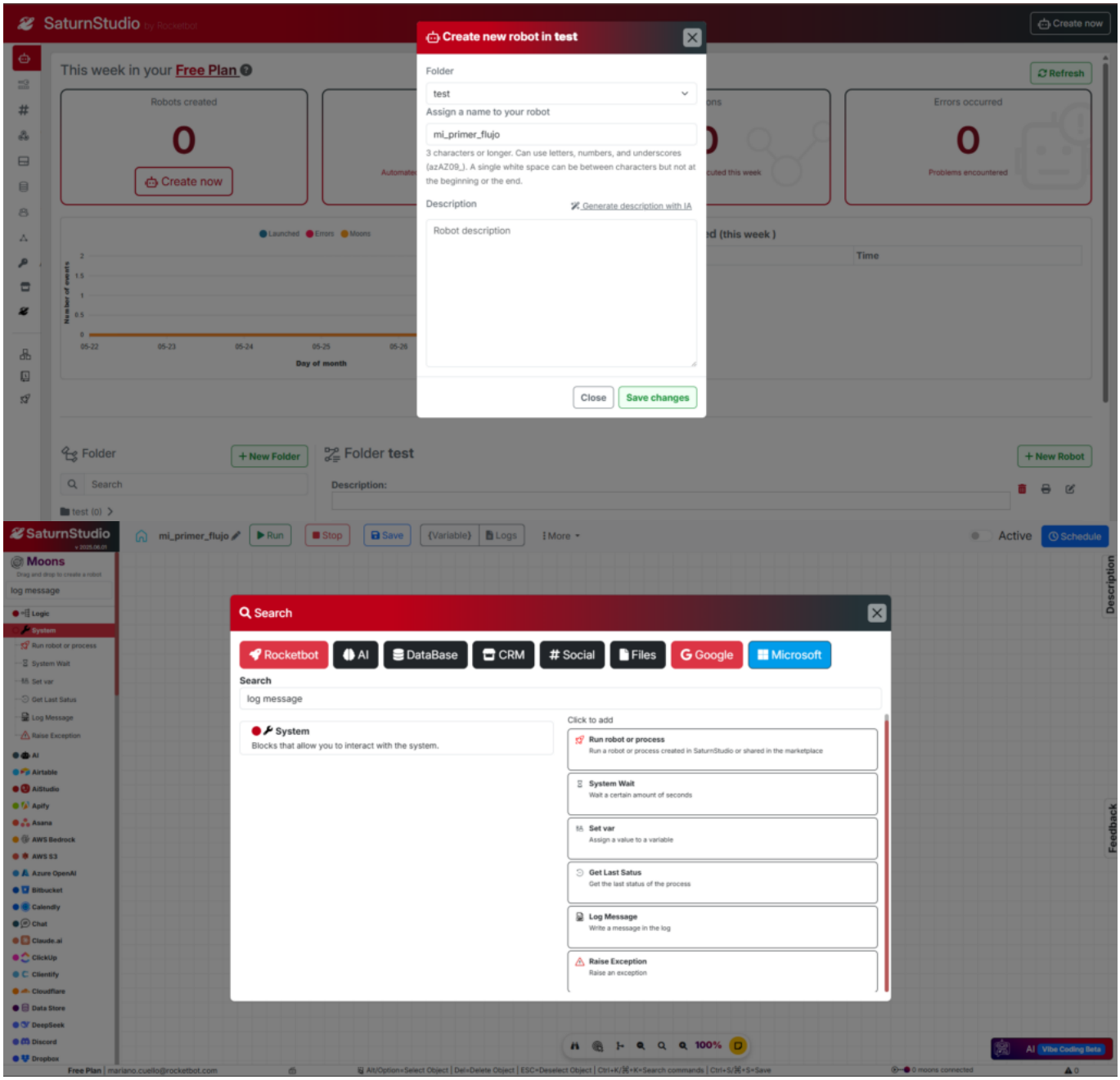


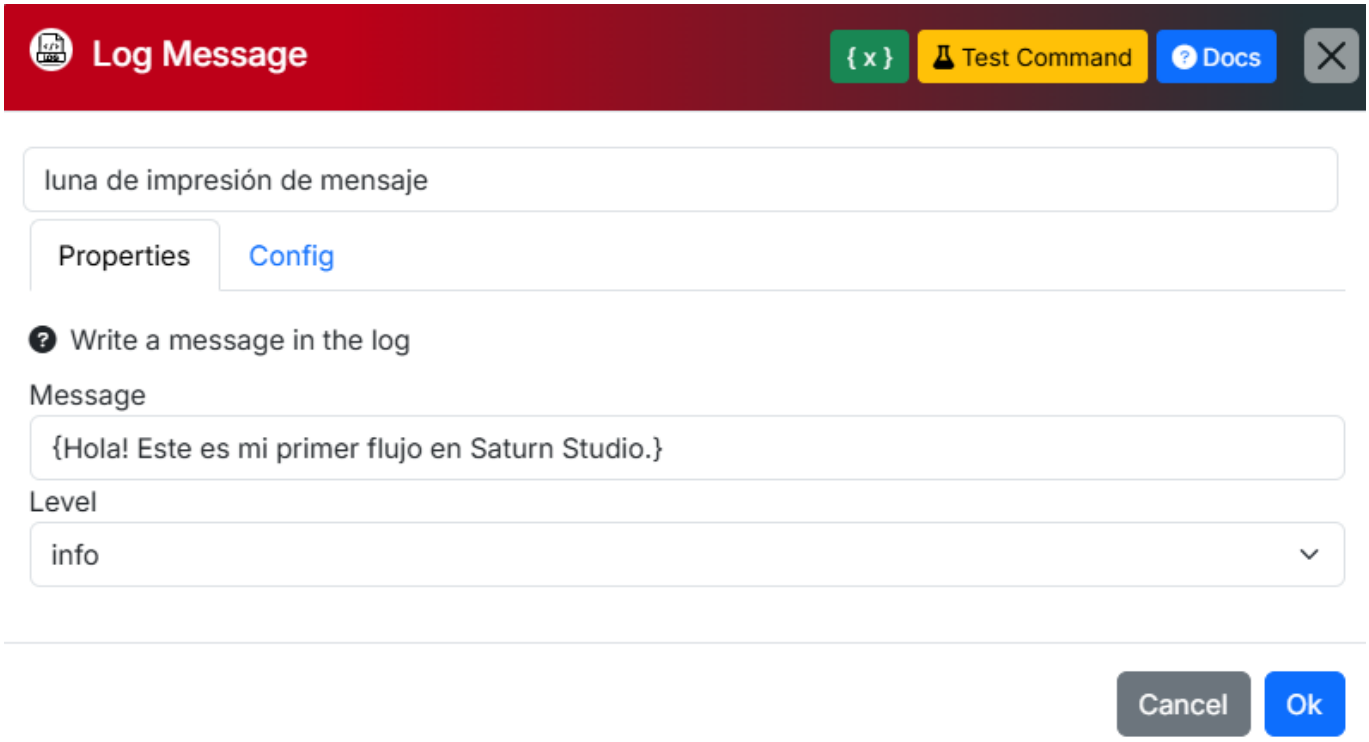


Consejo sobre el Market: Si en el futuro un *template* se parece a lo que necesitas, puedes añadirlo a tu espacio de trabajo; se creará una copia independiente que podrás editar sin alterar la plantilla original.

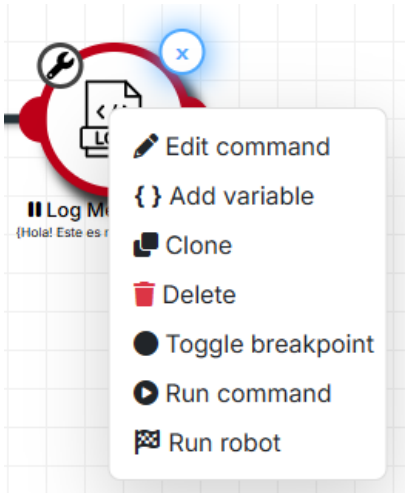
Parte 2: Construcción del Agente

Paso	Acción en la plataforma	Explicación sencilla / Consejos
3	Regresa al Dashboard y haz clic en **“+ New Bot”** o “Crear agente”.	Ponle un nombre descriptivo, por ejemplo: mi-primer-flujo. Se abrirá un lienzo en blanco que ya trae incorporada la Moon de inicio llamada **Start**.
4	Pasa el cursor sobre **Start**, haz clic en *‘Click to connect to new moon’*.	Busca y selecciona el módulo **Log Message**. Las **Moons** son los comandos que deben enlazarse de forma secuencial.
5	Configura los parámetros de tu nueva Moon de Log.	• **Descripción:** Agrega *‘luna de impresión de mensaje’* . • **Mensaje:** Escribe *“¡Hola! Este es mi primer flujo en Saturn Studio.”* . • **Level:** Selecciona **Info**. Guarda los cambios.


SaturnStudio
Create now
Create new robot in test
Folder
test
Assign a name to your robot
mi_primer_flujo
3 characters or longer. Can use letters, numbers, and underscores (azAZ09_). A single white space can be between characters but not at the beginning or the end.
Description
Generate description with IA
Robot description
Close
Save changes
This week in your Free Plan
Refresh
Robots created
0
Create now
Errors occurred
0
Problems encountered
Launched
Errors
Moons
Number of events
Day of month
Time
Folder
+ New Folder
Search
test (0)
Folder test
+ New Robot
Description:
mi_primer_flujo
Run
Stop
Save
{Variable}
Logs
More
Active
Schedule
Moons
log message
Logic
System
Run robot or process
System Wait
Set var
Get Last Satus
Log Message
Raise Exception
AI
Airtable
AiStudio
Apify
Asana
AWS Bedrock
AWS S3
Azure OpenAI
Bitbucket
Calendly
Chat
Claude.ai
ClickUp
Clientify
Cloudflare
Data Store
DeepSeek
Discord
Dropbox
Search
Rocketbot
AI
DataBase
CRM
Social
Files
Google
Microsoft
Search
log message
System
Blocks that allow you to interact with the system.
Click to add
Run robot or process
Run a robot or process created in SaturnStudio or shared in the marketplace
System Wait
Wait a certain amount of seconds
Set var
Assign a value to a variable
Get Last Satus
Get the last status of the process
Log Message
Write a message in the log
Raise Exception
Raise an exception
Description
Feedback
100%
AI
Vibe Coding Beta
Free Plan | mariano.cuello@rocketbot.com



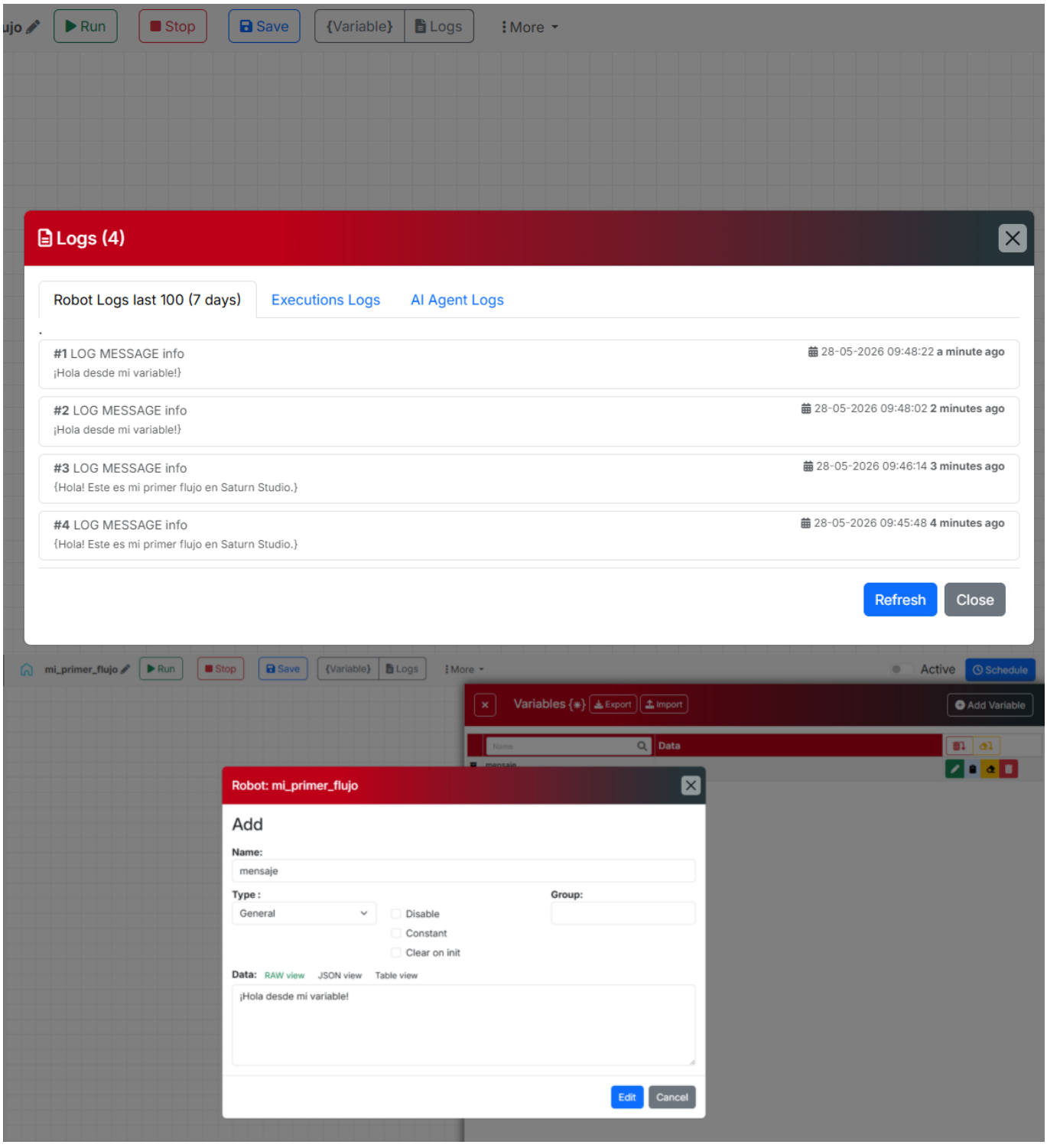
Truco de la interfaz: Si haces **clic derecho** sobre cualquier Moon en el lienzo, abrirás un menú avanzado para clonarla, eliminarla, editarla o ejecutar únicamente esa acción individual.

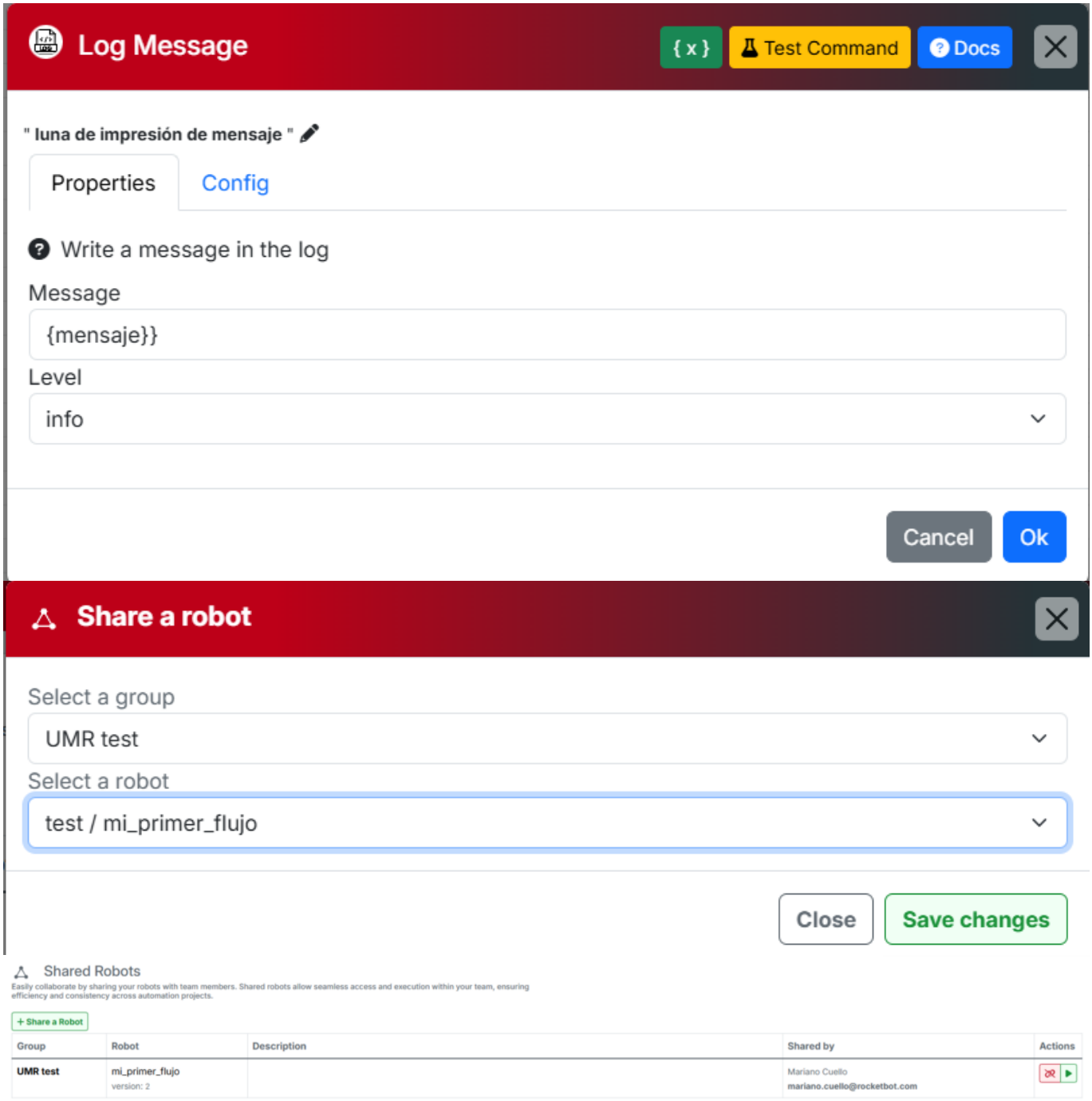


Parte 3: Ejecución, Variables y Almacenamiento

Paso	Acción en la plataforma	Explicación sencilla / Consejos
6	Haz clic en el botón **Ejecutar (Run)** en la barra superior.	Las Moons se iluminarán en orden confirmando que el flujo avanza. Al terminar, revisa el panel de **Logs** arriba para ver impreso tu mensaje.

Paso	Acción en la plataforma	Explicación sencilla / Consejos
7	**(Opcional)** Reemplaza el texto fijo usando una **Variable**.	Haz clic en el menú de variables de la cabecera (**{Variable}**) y luego selecciona **“Add variable”**. Nómbrala mensaje y asígnale el dato ¡Hola desde mi variable!. Luego, edita la Moon de Log y escribe {mensaje} en su campo de texto.
8	Guarda el agente y compártelo con tus colaboradores.	Presiona **Save**. Luego, regresa al **Home** y, en el menú lateral, ve a **Shared Robots**. Haz clic en **“+ Share a Robot”**, elige el equipo y asigna tu agente. Tu equipo podrá ejecutarlo pero no modificarlo.





¿Por qué usar variables? Si el día de mañana necesitas cambiar el texto, solo editas el valor de la variable una vez en su panel central, sin necesidad de abrir, buscar y modificar cada Moon de forma individual.

Resumen de lo aprendido

Concepto técnico	Lo que ejecutaste en este tutorial
Dashboard	Ingresaste y aprendiste a navegar en tu panel de control principal.

Concepto técnico	Lo que ejecutaste en este tutorial
Agente	Creaste un robot con identidad propia desde cero en el lienzo.
Moon	Añadiste un comando de Log, lo conectaste en secuencia y le asignaste una descripción.
Flujo	Validaste el ciclo de procesamiento ejecutando el orden Start → Log → End.
Variable	Convertiste un texto rígido en un contenedor de datos flexible y reutilizable.
Compartir	Distribuiste tu flujo terminado hacia la biblioteca compartida de tu equipo de trabajo.

¿Qué niveles siguen ahora?

Una vez que domines las bases de este flujo inicial, tu camino natural de aprendizaje en Saturn Studio incluye los siguientes módulos:

Etapa	Módulo / Tema	¿Qué aprenderás a hacer?
Nivel 2	Conectar Gmail	El robot enviará correos de reporte automáticos al finalizar sus tareas.
Nivel 3	Lógica condicional	Añadirás Moons de tipo **If/Else** para que el agente tome rutas distintas según los datos.
Nivel 4	Integrar IA	Conectarás OpenAI para interpretar el contexto de documentos o clasificar correos.
Nivel 5	Human-in-the-loop	Lograrás que el agente pause el flujo y espere una confirmación humana antes de seguir.

> *"Cada automatización grande empieza con un primer flujo simple. Lo que acabas de construir es la base de todo lo que vendrá."*

Saturn Studio — Conectores e integraciones

Documentación técnica de referencia • Versión: Mayo 2026

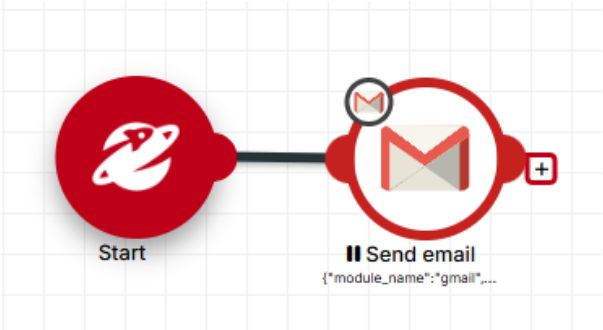
> **Nota crítica de red:** Saturn Studio opera 100% desde la nube. Para conectar bases de datos locales o APIs internas, el servidor de tu empresa debe ser accesible desde internet (IP pública) y el firewall debe permitir conexiones entrantes desde las IPs de Saturn Studio. **No se puede usar "localhost"** como dirección.

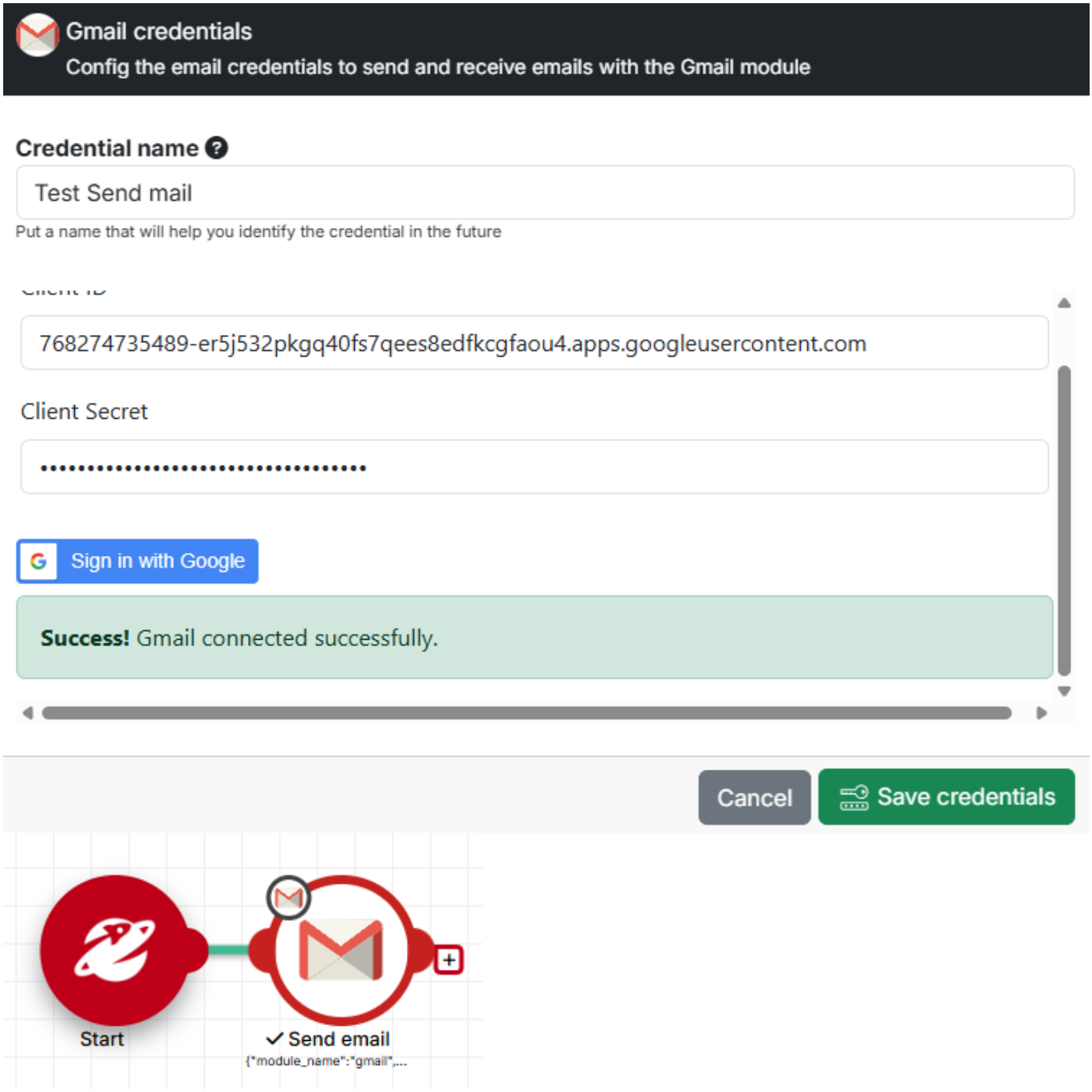
¿Cómo funcionan los conectores?

Los conectores son **módulos** que empaquetan toda la programación necesaria para que Saturn Studio hable con servicios externos. Cada conector contiene una serie de **Moons** (comandos) para realizar acciones específicas.

El flujo de trabajo es simple:

Agregar conector al flujo → Crear credencial en el Vault → Seleccionar credencial en el comando → Configurar parámetros → Ejecutar





- **Credenciales compartidas:** Las claves se guardan de forma centralizada. Si una contraseña o API Key cambia, solo la actualizas en un lugar y todos los robots se sincronizan solos.

Resumen del Catálogo de Conectores

Disponemos de **31 conectores oficiales** organizados en 6 categorías estratégicas:

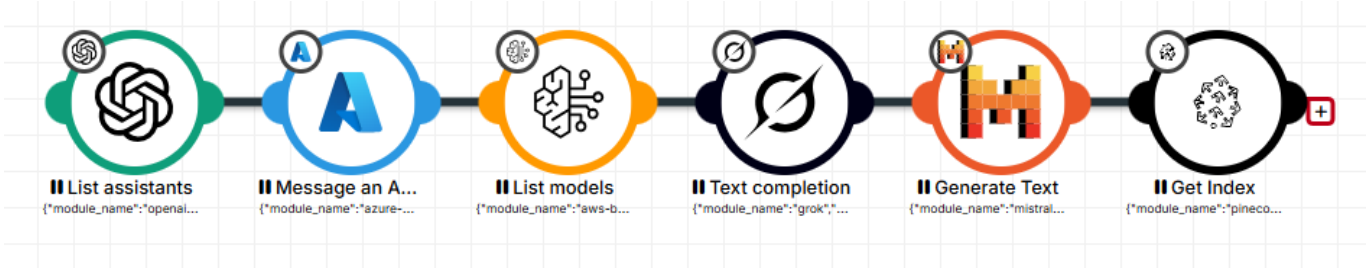
Categoría	Conectores Incluidos	Total
1. Inteligencia Artificial	OpenAI, Azure OpenAI, AWS Bedrock, Grok, Mistral AI, Pinecone	6

Categoría	Conectores Incluidos	Total
2. Mensajería y Comunicación	Gmail, Microsoft (Outlook/Teams), SharePoint, Discord, WhatsApp, Facebook, LinkedIn, Reddit, Chat Widget	9
3. Bases de Datos y Almacenamiento	MySQL, PostgreSQL, MongoDB, AWS S3	4
4. Productividad y Gestión	Notion, Asana, GitHub, Zoom, Xperience	5
5. Pagos y E-commerce	Mercado Libre, Mercado Pago, Nubox	3
6. Infraestructura y Utilidades	Webhooks, Request HTTP, PDF Reader, File Storage	4

Detalle de Conectores por Categoría

1. Inteligencia Artificial

Conectores diseñados para dotar de criterio cognitivo, visión y memoria semántica a los agentes.



Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
OpenAI	Integra modelos GPT y DALL-E para entender texto, imágenes y generar contenido.	Text Completion, Analyze Image, Manage Assistants.	Clasificar correos entrantes, resumir PDFs y responder a clientes.
Azure OpenAI	Los mismos modelos de OpenAI pero ejecutados dentro de la nube segura de Microsoft corporativo.	Message an Assistant, Text Completion.	Procesamiento de datos en grandes empresas con alta seguridad en Azure.
AWS Bedrock	Plataforma de Amazon para usar modelos de alto rendimiento (como Claude) de forma segura.	List Models, Text Completion.	Agentes empresariales que ya operan con infraestructura en Amazon (AWS).
Grok (xAI)	Conecta con los modelos avanzados de Grok desarrollados por la compañía xAI.	Text Completion, Generate Image.	Redacción de contenido y análisis visual con modelos alternativos.
Mistral AI	Modelos de lenguaje ultra eficientes y rápidos para razonamiento lógico a través de API.	Text Completion, List Models.	Automatizar informes técnicos y clasificación de datos a gran escala.

Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
Pinecone	**Base de datos vectorial** que funciona como la memoria a largo plazo de la IA, guardando “embeddings”.	Upsert Vectors, Query.	Búsqueda semántica inteligente (buscar por significado, no por palabra exacta).

2. Mensajería y Comunicación

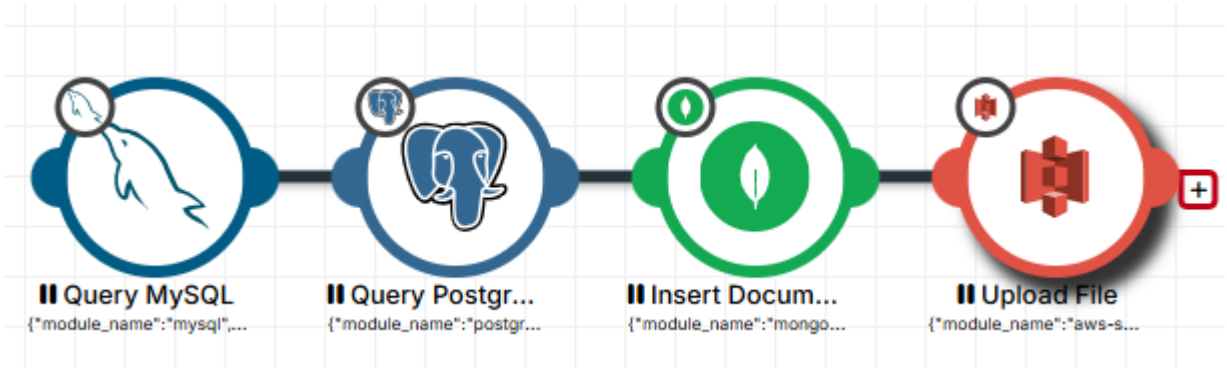
Conectores para interactuar con canales de soporte, correos corporativos y redes sociales.

Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
Gmail	Conexión nativa con correos de Google para leer, enviar y archivar mensajes.	Send Email, Read Emails, Move to Folder.	Revisar correos de proveedores, descargar archivos adjuntos automáticamente.
Microsoft 365	Integra correos de Outlook, mensajes en Teams y gestión de calendarios.	Send Email (Outlook), Send Message (Teams).	Enviar alertas internas a canales de Teams o agendar reuniones de trabajo.
SharePoint	Conector para interactuar con el gestor documental corporativo de Microsoft.	Upload File, Download File, Create List Item.	Guardar informes generados por el bot directamente en carpetas compartidas.
Discord	Envío de mensajes y alertas estructuradas a servidores de comunicación.	Send Message, Send Embed.	Notificar instantáneamente al equipo de TI si un proceso crítico falla.
WhatsApp	Automatización de chat a través de Evolution API (servidor puente open-source).	Send Message, Receive Message, Get QR Code.	Enviar alertas de despacho o confirmaciones de compra directamente al cliente.
Facebook Messenger	Automatiza la lectura y respuesta de chats de tus páginas de empresa.	Send Message, Receive Message.	Respuestas instantáneas y automáticas a preguntas frecuentes de clientes.
LinkedIn	Vincula Saturn Studio con perfiles o páginas corporativas profesionales.	Create Post, Manage Page Content.	Publicación automatizada de reportes de actividad o noticias de la empresa.
Reddit	Permite leer e interactuar con subreddits (comunidades temáticas).	Get Posts, Search Subreddit.	Monitorear en tiempo real qué se dice de tu marca o analizar tendencias.

Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
Chat Widget	Un cuadro de chat que puedes pegar en tu sitio web para capturar visitas.	Receive Message, Route to Flow.	Capturar clientes interesados (*leads*) y responderles usando un flujo de IA.

3. Bases de Datos y Almacenamiento

Conectores destinados al resguardo, lectura y actualización de información transaccional.



Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
MySQL	Ejecución de consultas estándar sobre bases de datos relacionales MySQL/MariaDB.	Execute Query, Select, Insert.	Extraer la lista de clientes pendientes de cobro desde el sistema interno.
PostgreSQL	Conexión avanzada con bases de datos relacionales PostgreSQL de gran volumen.	Execute Query, Update, Delete.	Consultar stocks de inventario actualizados y modificar estados de órdenes.
MongoDB	Conexión con bases de datos NoSQL basadas en documentos flexibles estilo JSON.	Find, Insert, Aggregate.	Almacenar registros técnicos, configuraciones de usuarios o datos web variables.
AWS S3	Almacenamiento en la nube de Amazon para guardar cualquier tipo de archivo físico.	Upload File, Download File, Generate URL.	Guardar respaldos de reportes mensuales y generar enlaces de descarga seguros.

4. Productividad y Gestión

Conectores enfocados en la organización de tareas, control de versiones y operaciones internas.

Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
Notion	Conexión con páginas y bases de datos organizativas de Notion.	Query Database, Create Page.	Crear registros automáticos de control o tableros de resumen de proyectos.
Asana	Gestión automatizada de tableros de tareas y asignación de pendientes.	Create Task, Update Task, Complete Task.	Generar un ticket de trabajo al equipo cuando un formulario web requiere atención.
GitHub	Interacción con repositorios de código. Permite activar robots mediante webhooks.	Subscribe Webhook, Get Pull Request.	Desencadenar flujos automáticos de revisión cada vez que un programador sube código.
Zoom	Automatización completa del calendario de salas de videoconferencia.	Create Meeting, Delete Meeting.	Crear enlaces de Zoom solos al confirmar una cita de ventas con un cliente.
Xperience	Módulo interno para leer los datos de los formularios y colas del Orquestador Rocketbot.	Get Queue Records, Update Record Status.	Procesar uno a uno los datos ingresados por usuarios en pantallas de autoservicio.

5. Pagos y E-commerce

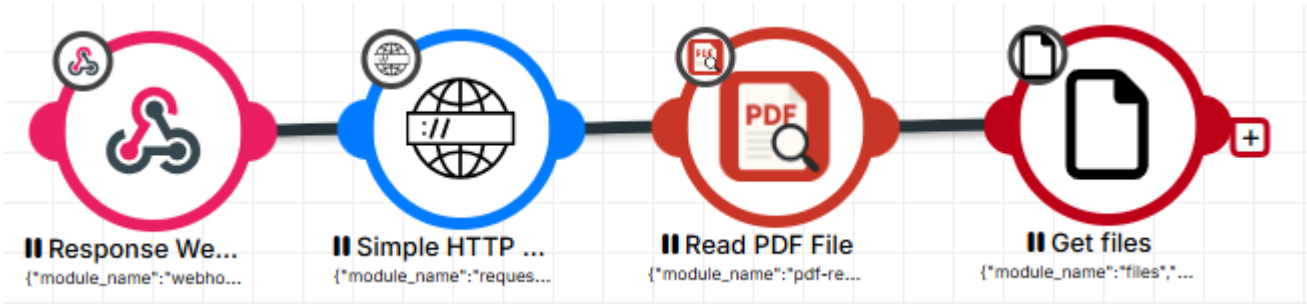
Conectores especializados en el ecosistema financiero y comercial de Latinoamérica.

Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico

Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
Mercado Libre	Gestión comercial de publicaciones, inventarios y preguntas en la plataforma.	Manage Listings, Reply Messages, Update Stock.	Responder de forma inmediata los mensajes post-venta y sincronizar precios.
Mercado Pago	Operaciones fintech para control de cobros, transacciones y links de pago.	Process Payment, Check Transaction Status.	Conciliación bancaria automática (revisar qué clientes ya pagaron y cuáles no).
Nubox	Conexión con el software contable de emisión de boletas y facturas electrónicas.	Create Invoice, Submit to SII.	Emitir y enviar al SII la factura legal de forma automática al registrar un pago exitoso.

6. Infraestructura y Utilidades

Comandos transversales y herramientas genéricas que potencian la arquitectura de cualquier robot.



Conector	¿Para qué sirve? (Descripción)	Comandos Clave	Caso de Uso Típico
Webhooks	**Disparador en tiempo real.** Permite que un sistema externo le avise al robot que actúe empujando los datos.	Receive Webhook, Send Webhook Response.	Iniciar el robot de inmediato cuando se completa un formulario web o un pago.
Request (HTTP)	**El conector comodín.** Sirve para hacer peticiones técnicas (GET, POST, etc.) a cualquier sistema en internet.	Simple HTTP Request, Call API Advanced.	Conectarse a cualquier software o sistema propio de tu empresa que no tenga conector listo.
PDF Reader	Extractor de datos local. Lee archivos PDF y los transforma en texto plano estructurado.	Extract Text, Extract Tables.	Leer tablas de datos dentro de contratos escaneados o estados de cuenta.
File Storage	Disco duro interno temporal de Saturn Studio para guardar archivos durante la ejecución.	Upload File, Download File, List Files.	Retener temporalmente una imagen descargada para que la IA la analice en el siguiente paso.

Métodos de Autenticación Utilizados

Todos los conectores de la plataforma se aseguran utilizando uno de estos métodos estándares:

Método Técnico	¿En qué consiste de forma sencilla?	Conectores que lo usan
API Key	Una clave de texto larga generada en el panel del servicio externo (funciona como contraseña maestra).	OpenAI, Grok, Mistral AI, Pinecone, Discord, Asana.
OAuth 2.0	Ventana de autorización donde inicias sesión con tu cuenta y autorizas a Saturn Studio a actuar en tu nombre.	Gmail, Microsoft, LinkedIn, Mercado Libre, Mercado Pago, Zoom.
Server-to-Server OAuth	Conexión segura directa entre servidores de fondo utilizando credenciales fijas de aplicación (sin intervención de usuario).	Zoom (servidor a servidor), Facebook Messenger.
Host + Puerto / Connection String	Datos directos de dirección IP, puerto, usuario y contraseña para ingresar a un servidor de base de datos corporativo.	MySQL, PostgreSQL, MongoDB.
AWS IAM	Credenciales de seguridad nativas de Amazon (Access Key + Secret Key) con permisos controlados.	AWS Bedrock, AWS S3.
Integration Token	Un código de vinculación interno que se genera en la plataforma y requiere que compartas el recurso a mano.	Notion.
Sin credencial externa	No requieren configurar claves fuera de la plataforma; funcionan con el motor interno de Saturn Studio.	Webhooks, Request HTTP, PDF Reader, File Storage, Chat Widget.