**Documentación Técnica – Flujo Automatizado de Gestión de Correos y Eventos**

**🧩 1. Identificación del Flujo**

* **Nombre del Flujo:** Google Mail Calendario
* **Estado:** Inactivo ("active": false)
* **Versión de Ejecución:** executionOrder: v1
* **Entorno:** Plataforma n8n (Self-hosted/local)
* **Credenciales empleadas:**
  + gmailOAuth2
  + googleCalendarOAuth2Api
  + googlePalmApi (Gemini)
  + mistralCloudApi (Pixtral & Codestral)

**📨 2. Entrada del Flujo**

**Nodo: Gmail Trigger**

* Disparador por polling cada minuto (mode: everyMinute)
* Fuente: buzón Gmail autenticado
* Salida: JSON del correo entrante (incluye asunto, contenido, remitente, id)

**📊 3. Clasificación del Contenido**

**Nodo: Text Classifier**

* Clasifica el correo en una de las siguientes categorías:
  + **Trabajo**
  + **Personal**
  + **Evento**
  + **Otros**
* Basado en contenido del correo (subject + text)
* Redirige a nodos específicos de etiquetado según categoría

**🧠 4. Procesamiento con IA**

**Nodo: Google Gemini Chat Model**

* Asiste al nodo **Text Classifier** como motor de lenguaje natural

**Nodo: AI Agent**

* Prompt personalizado con dos objetivos:
  + Redactar una **respuesta al correo** en tono casual-profesional
  + Extraer posibles eventos con los campos: date, time, location, event\_description
* El output es un objeto JSON estructurado

**🔍 5. Postprocesamiento del Output**

**Parsers:**

* **Auto-fixing Output Parser:** Corrige desviaciones del formato esperado
* **Structured Output Parser:** Valida estructura contra ejemplo JSON predefinido

**🔧 6. Transformación con Código**

**Nodo: Code (JavaScript)**

* Convierte fechas y horas en formato RFC3339 (YYYY-MM-DDTHH:mm:ss±hh:mm)
* Lógica de parsing tolerante a formatos:
  + 5pm, 17:00, 17:00:00
  + Rangos como 17:00 - 19:00 o 5pm a 7pm
* Si el tiempo es único, se asume como misma hora de inicio y fin
* Ajuste horario: zona UTC−5

**✅ 7. Validación Final**

**Nodo: If**

* Comprueba integridad de campos:
  + calendar\_event no vacío
  + Campos válidos (sin contener “Invalid”)

**📤 8. Salida**

**Nodo: Gmail (Reply)**

* Envía al remitente la respuesta generada por IA

**Nodo: Google Calendar**

* Crea evento sólo si la validación fue exitosa
* Campos utilizados:
  + start\_time, end\_time
  + summary, location, description

**♻️ 9. Subprocesos por Categoría**

Aunque el JSON incluye solo la rama de **Trabajo**, se replica la misma lógica exacta para:

* **Personal**
* **Evento**
* **Otros**

Cada categoría cuenta con su propia ruta de etiquetado, pero el flujo IA, parseo, validación y acción es idéntico.

**📝 10. Notas Técnicas Complementarias**

* El nodo Code está escrito completamente en JavaScript moderno, modularizado y tolerante a errores de entrada.
* Se emplean múltiples modelos de lenguaje (*Gemini*, *Codestral*, *Pixtral*) para robustecer el procesamiento semántico y corregir ambigüedades.
* El flujo puede ser exportado, versionado y reimplementado vía JSON gracias a la estructura exportada desde n8n.