



**TRABAJO DE GRADO  
Opción Seminario-Diplomado.**

**Asistente Financiero Automatizado**

**Corporación Universitaria Remington.**

**Facultad de Ingenierías**

**Ingeniería de Sistemas**

**Presentado por:**

**Cristian David Otalvaro Almanza**

**Tutor: Luis Camargo Ortega**

**Opción de Trabajo de grado Seminario.**

**2026.**

## Tabla de contenido

Descripción General.....	3
Objetivos del Sistema.....	3
Flujo del Asistente Financiero Automatizado.....	3
Flujo de Procesamiento.....	4
Manejo de Errores.....	7
Ventajas del Sistema.....	7
Tecnologías Utilizadas.....	7
Posibles Mejoras.....	8
Respuestas del Sistema.....	8
Conclusión.....	60

## **Descripción General**

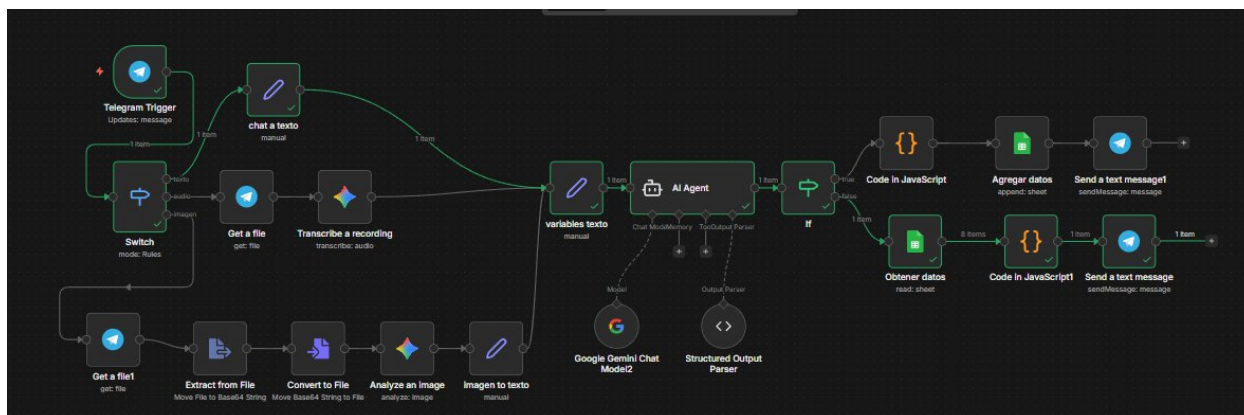
Este proyecto consiste en el desarrollo de un asistente financiero automatizado que permite a los usuarios registrar, consultar sus gastos personales a través de Telegram.

Se implementan múltiples fuentes de ingreso (texto, audio e imágenes) y almacenando los datos en Google Sheets, donde posteriormente se procesan para generar respuestas inteligentes.

## **Objetivos del Sistema**

- Registrar los gastos personales de forma rápida mediante un Bot Telegram
- Soportar diferentes formatos de entrada:
  - Texto
  - Audio
  - Imagen
- Almacenar los datos en Google Sheets
- Permitir consultas sobre los gastos
- Generar análisis básicos (totales, categorías, comparaciones)

## **Flujo del Asistente Financiero Automatizado**



## Flujo de Procesamiento

### Entrada del Usuario

El sistema inicia con un Trigger de Telegram, que recibe:

- Mensajes de texto
- Audios
- Imágenes

### Normalización de Entrada

Dependiendo del tipo de entrada:

- Texto Se envía directamente
- Audio se transcribe a texto
- Imagen se convierte de imagen a texto

Todos los caminos convergen en el nodo: **variables texto**

## Procesamiento con IA

El AI Agent analiza el texto y clasifica la intención:

Tipos de operación:

### Registro de gasto

```
{  
  
  "tipo": "gasto",  
  
  "data": {  
  
    "monto": 5000,  
  
    "categoria": "cafe",  
  
    "descripcion": "cafe"  
  
  }  
  
}
```

### Consulta

```
{  
  
  "tipo": "consulta",  
  
  "data": {  
  
    "tipo_consulta": "categoria",  
  
    "categoria": "cafe"  
  
  }  
  
}
```

```
}
```

```
}
```

### **Decisión (IF Node)**

El sistema separa el flujo en dos ramas:

#### **Rama 1: Registro de gasto**

- Transformación de datos
- Guarda en Google Sheets
- Confirma al usuario

#### **Rama 2: Consulta**

- Consulta de los datos desde Google Sheets
- Respuesta analítica

### **Lógica de Procesamiento (JavaScript)**

El sistema realiza:

- Suma todos los gastos
- Agrupa por descripción/categoría
- Identifica la categoría con mayor gasto

## **Manejo de Errores**

El sistema incluye:

- Validación de datos
- Manejo de campos vacíos
- Normalización de texto (lowercase)

## **Ventajas del Sistema**

Automatización completa

- Costo operativo bajo
- Escalable
- Multiformato (texto, voz, imagen)
- Fácil integración con otras herramientas

## **Tecnologías Utilizadas**

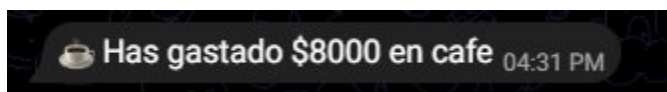
- n8n (automatización)
- Telegram Bot
- Google Sheets
- Convertir Imagen a texto
- Convertir Audio a texto
- JavaScript
- AI Agent

### Posibles Mejoras

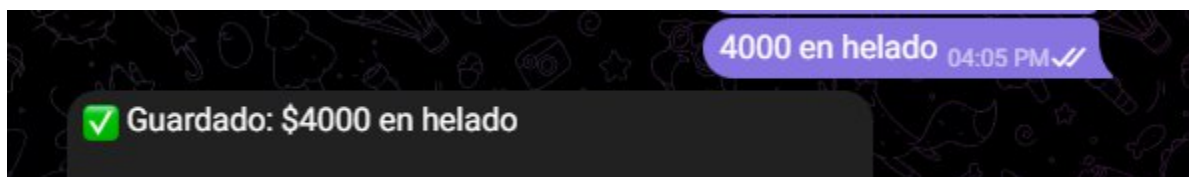
- Filtros por fecha (mensual, semanal)
- Soporte multiusuario
- Dashboard
- Alertas automáticas de gasto
- Clasificación automática de categorías
- Integración con bases de datos

### Respuestas del Sistema

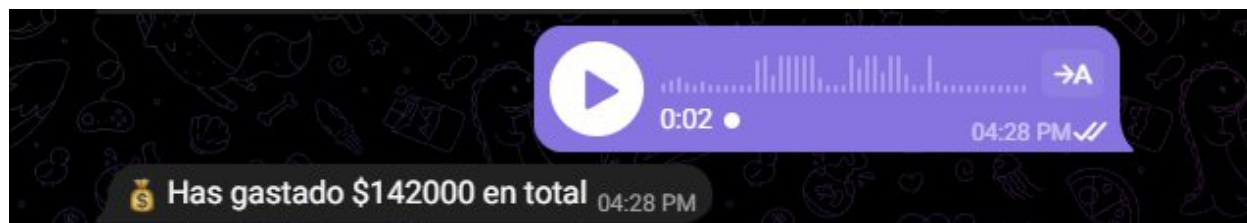
Consulta sobre los gastos realizados en Cafe



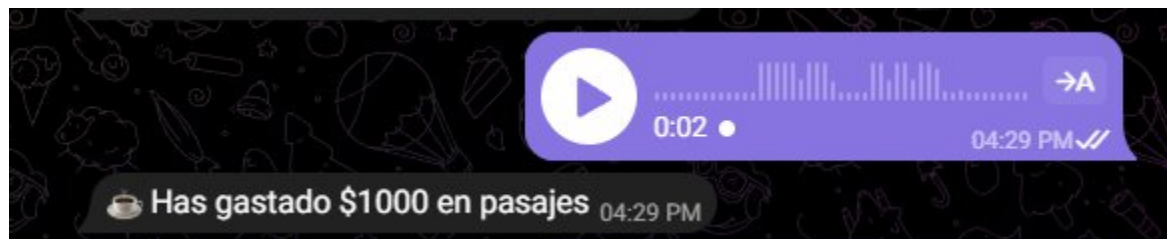
Solicitud para guardar 4000 en helado



Consulta sobre todos los gastos realizados



Consulta sobre los gastos realizados en pasaje



### Código fuente N8n

```
{
  "name": "My workflow",
  "nodes": [
    {
      "parameters": {
        "updates": [
          "message"
```

```
],  
  
  "additionalFields": {}  
  
},  
  
  "type": "n8n-nodes-base.telegramTrigger",  
  
  "typeVersion": 1.2,  
  
  "position": [  
  
    528,  
  
    -608  
  
  ],  
  
  "id": "85983c48-e39c-420f-a629-5ee9ed0e8b2c",  
  
  "name": "Telegram Trigger",  
  
  "webhookId": "c7a33a8c-7f6a-44bd-9e07-45a431cc7a52",  
  
  "credentials": {  
  
    "telegramApi": {  
  
      "id": "rEcHgB9avrO5uA39",  
  
      "name": "Telegram account"  
  
    }  
  
  }  
  
}
```

```
},  
  
{  
  
  "parameters": {  
  
    "rules": {  
  
      "values": [  
  
        {  
  
          "conditions": {  
  
            "options": {  
  
              "caseSensitive": true,  
  
              "leftValue": "",  
  
              "typeValidation": "strict",  
  
              "version": 3  
  
            },  
  
          },  
  
          "conditions": [  
  
            {  
  
              "leftValue": "={{ $json.message.text }}",  
  
              "rightValue": "",  
  
              "operator": {
```

```
"type": "string",

"operation": "exists",

"singleValue": true

},

"id": "5469a2d5-2a10-4815-9338-66f224f22385"

}

],

"combinator": "and"

},

"renameOutput": true,

"outputKey": "texto"

},

{

"conditions": {

"options": {

"caseSensitive": true,

"leftValue": "",

"typeValidation": "strict",
```

```
"version": 3

},

"conditions": [

  {

    "id": "e786384c-a3c8-4c24-9b42-65ad1657e070",

    "leftValue": "{{ $json.message.voice }}",

    "rightValue": "",

    "operator": {

      "type": "object",

      "operation": "exists",

      "singleValue": true

    }

  }

],

"combinator": "and"

},

"renameOutput": true,

"outputKey": "audio"
```

```
},  
  
{  
  
  "conditions": {  
  
    "options": {  
  
      "caseSensitive": true,  
  
      "leftValue": "",  
  
      "typeValidation": "strict",  
  
      "version": 3  
  
    },  
  
    "conditions": [  
  
      {  
  
        "id": "8f939dfc-d99c-4183-8820-723de55971ba",  
  
        "leftValue": "={{ $json.message.document }}",  
  
        "rightValue": "",  
  
        "operator": {  
  
          "type": "object",  
  
          "operation": "exists",  
  
          "singleValue": true  
  
        }  
  
      }  
  
    ]  
  
  }  
  
}
```

```
    }  
  }  
],  
  "combinator": "and"  
},  
  "renameOutput": true,  
  "outputKey": "imagen"  
}  
]  
},  
  "options": {}  
},  
  "type": "n8n-nodes-base.switch",  
  "typeVersion": 3.4,  
  "position": [  
    528,  
    -400  
  ],
```

```
"id": "12c674a9-c2dc-41c6-99ea-b368f808a9b0",  
  
"name": "Switch"  
  
},  
  
{  
  
"parameters": {  
  
"resource": "file",  
  
"fileId": "={{ $json.message.voice.file_id }}",  
  
"additionalFields": {}  
  
},  
  
"type": "n8n-nodes-base.telegram",  
  
"typeVersion": 1.2,  
  
"position": [  
  
720,  
  
-384  
  
],  
  
"id": "ec0d36e4-4456-445c-9c57-d50314469991",  
  
"name": "Get a file",  
  
"webhookId": "1274ecd9-0ba5-4f2b-9a8c-bf0ec73ab609",
```

```
"credentials": {  
  
  "telegramApi": {  
  
    "id": "rEcHgB9avrO5uA39",  
  
    "name": "Telegram account"  
  
  }  
  
}  
  
},  
  
{  
  
  "parameters": {  
  
    "resource": "audio",  
  
    "modelId": {  
  
      "__rl": true,  
  
      "value": "models/gemini-3.1-flash-lite-preview",  
  
      "mode": "list",  
  
      "cachedResultName": "models/gemini-3.1-flash-lite-preview"  
  
    },  
  
    "inputType": "binary",  
  
    "options": {}  
  
  }  
  
}
```

```
},  
  
"type": "@n8n/n8n-nodes-langchain.googleGemini",  
  
"typeVersion": 1.1,  
  
"position": [  
  
  896,  
  
  -384  
  
],  
  
"id": "9c3763b8-5406-4c79-839c-e248f244566f",  
  
"name": "Transcribe a recording",  
  
"credentials": {  
  
  "googlePalmApi": {  
  
    "id": "pO0oOmw4zofvrjc3",  
  
    "name": "Google Gemini(PaLM) Api account 5"  
  
  }  
  
}  
  
},  
  
{  
  
  "parameters": {
```

```
"resource": "file",

"fileId": "={{ $json.message.document.thumbnail.file_id }}",

"additionalFields": {}

},

"type": "n8n-nodes-base.telegram",

"typeVersion": 1.2,

"position": [

  480,

  -128

],

"id": "c7d70776-33ea-47c1-aba0-f22d10799fde",

"name": "Get a file1",

"webhookId": "745726f0-7a1a-4442-ad40-35de0dde4a87",

"credentials": {

  "telegramApi": {

    "id": "rEcHgB9avrO5uA39",

    "name": "Telegram account"

  }

}
```

```
}  
  
},  
  
{  
  
  "parameters": {  
  
    "operation": "binaryToProperty",  
  
    "options": {}  
  
  },  
  
  "type": "n8n-nodes-base.extractFromFile",  
  
  "typeVersion": 1.1,  
  
  "position": [  
  
    672,  
  
    -112  
  
  ],  
  
  "id": "b23b3dd9-04cc-4cc0-8fa7-441755f9f4e9",  
  
  "name": "Extract from File"  
  
},  
  
{  
  
  "parameters": {
```

```
"operation": "toBinary",

"sourceProperty": "data",

"options": {}

},

"type": "n8n-nodes-base.convertToFile",

"typeVersion": 1.1,

"position": [

  848,

  -112

],

"id": "63cb9072-82c3-4cc2-999d-a92084c7783b",

"name": "Convert to File"

},

{

"parameters": {

"resource": "image",

"operation": "analyze",

"modelId": {
```

```
"__rl": true,  
  
"value": "models/gemini-2.5-flash-lite",  
  
"mode": "list",  
  
"cachedResultName": "models/gemini-2.5-flash-lite"  
  
},  
  
"text": "Describe detalladamente la imagen",  
  
"inputType": "binary",  
  
"options": {}  
  
},  
  
"type": "@n8n/n8n-nodes-langchain.googleGemini",  
  
"typeVersion": 1.1,  
  
"position": [  
  
  1008,  
  
  -112  
  
],  
  
"id": "4b26d688-841f-4005-8998-dae2d6f4c813",  
  
"name": "Analyze an image",  
  
"credentials": {
```

```
"googlePalmApi": {  
  
  "id": "7XqiAWU3UJ9cLX1I",  
  
  "name": "Google Gemini(PaLM) Api account 2"  
  
}  
  
},  
  
{  
  
  "parameters": {  
  
    "assignments": {  
  
      "assignments": [  
  
        {  
  
          "id": "f84fc8ae-ada6-4740-b326-9ea5772674ad",  
  
          "name": "texto",  
  
          "value": "={{ $json.content.parts[0].text }}",  
  
          "type": "string"  
  
        }  
  
      ]  
  
    },  
  
  },  
  
}
```

```
"options": {}  
  
},  
  
"type": "n8n-nodes-base.set",  
  
"typeVersion": 3.4,  
  
"position": [  
  
  1168,  
  
  -112  
  
],  
  
"id": "9d77d44c-49e7-4d19-be9d-21bf438e7181",  
  
"name": "imagen to texto"  
  
},  
  
{  
  
  "parameters": {  
  
    "modelName": "models/gemini-2.0-flash",  
  
    "options": {}  
  
  },  
  
  "type": "@n8n/n8n-nodes-langchain.lmChatGoogleGemini",  
  
  "typeVersion": 1,
```

```
"position": [  
  
  1392,  
  
  -144  
  
],  
  
"id": "d24ac525-9603-422e-a8f5-29fa5b89c605",  
  
"name": "Google Gemini Chat Model2",  
  
"credentials": {  
  
  "googlePalmApi": {  
  
    "id": "pO0oOmw4zofvrjc3",  
  
    "name": "Google Gemini(PaLM) Api account 5"  
  
  }  
  
}  
  
},  
  
{  
  
  "parameters": {  
  
    "promptType": "define",  
  
    "text": "=INFORMACION:{{ $json.texto }}",  
  
    "hasOutputParser": true,
```

```

"options": {

  "systemMessage": "=Responde ÚNICAMENTE con JSON válido.\n\nNO uses la clave
\"output\"\n\nNO envuelvas el JSON en otro objeto\n\nNO uses ``json\n\nNO agregues texto antes o
después\n\n---\n\nEres un asistente financiero.\n\nTu tarea es:\n1. Detectar si el usuario registra
un gasto\n2. Detectar si el usuario hace una consulta sobre gastos\n3. Extraer datos
estructurados\n4. Responder SIEMPRE en JSON válido\n\n---

\n\nENTRADA:\n{{ $json.texto }}\n\nFECHA ACTUAL:\n{{ $now }}\n\n---\n\nFORMATO
EXACTO DE RESPUESTA:\n\n{\n  \"tipo\": \"gasto\" | \"consulta\" | \"error\",\n  \"data\":
{\n    \"fecha\": \"string\",\n    \"categoria\": \"string\",\n    \"monto\":
number,\n    \"descripcion\": \"string\",\n    \"tipo_consulta\": \"string\"\n  },\n  \"respuesta\":
\"string\"\n}\n\n---\n\n#### CASO 1: GASTO\n\nSi el mensaje contiene:\n- un número
(monto)\n- contexto de gasto\n\nEjemplos:\n- \"cafe 5000\"\n- \"gasté 20000 en comida\"\n-
\"5000 en transporte\"\n\nRespuesta:\n\n{\n  \"tipo\": \"gasto\",\n  \"data\": {\n    \"fecha\":
\"{{ $now }}\",\n    \"categoria\": \"cafe\",\n    \"monto\": 5000,\n    \"descripcion\":
\"cafe\",\n    \"tipo_consulta\": \"\"\n  },\n  \"respuesta\": \" Gasto guardado\n  cafe -
$5000\"\n}\n\n---\n\n REGLAS DE FECHA:\n\n- Si NO hay fecha → usar {{ $now }}\n\n-
Formato: YYYY-MM-DD\n\n---\n\n#### CASO 2: CONSULTA\n\nSi el usuario pregunta
sobre sus gastos, extrae:\n\n- tipo_consulta:\n  - \"total\"\n  - \"categoria\"\n  -
\"comparacion\"\n\n- categoria (si aplica)\n- fecha (si aplica: hoy, ayer, este mes)\n\n---
\n\nEjemplos:\n\n\"cuanto he gastado en cafe\"\n\n{\n  \"tipo\": \"consulta\",\n  \"data\":
{\n    \"fecha\": \"\",\n    \"categoria\": \"cafe\",\n    \"monto\": 0,\n    \"descripcion\":
\"\",\n    \"tipo_consulta\": \"categoria\"\n  },\n  \"respuesta\": \"Consultando gastos...\"\n}\n\n---
\n\n\"cuanto he gastado hoy\"\n\n{\n  \"tipo\": \"consulta\",\n  \"data\": {\n    \"fecha\":

```

```

\ "hoy",\n  \ "categoria": \ "",\n  \ "monto": 0,\n  \ "descripcion": \ "",\n  \ "tipo_consulta":
\ "total"\n },\n  \ "respuesta": \ "Consultando gastos...\n}\n\n---\n\n"en que gasto
mas"\n\n{\n  \ "tipo": \ "consulta",\n  \ "data": {\n  \ "fecha": \ "",\n  \ "categoria":
\ "",\n  \ "monto": 0,\n  \ "descripcion": \ "",\n  \ "tipo_consulta":
\ "comparacion"\n },\n  \ "respuesta": \ "Analizando gastos...\n}\n\n---\n\n### CASO 3:
ERROR\n\nSi no es gasto ni consulta:\n\n{\n  \ "tipo": \ "error",\n  \ "data": {\n  \ "fecha":
\ "",\n  \ "categoria": \ "",\n  \ "monto": 0,\n  \ "descripcion": \ "",\n  \ "tipo_consulta":
\ ""\n },\n  \ "respuesta": \ " No entendí el mensaje. Ejemplo: cafe 5000"\n}\n\n---\n\n
REGLAS FINALES:\n\n- monto debe ser número\n- sin comas en números\n- NO inventar
datos\n- SIEMPRE JSON válido"

```

```

}
```

```

},
```

```

"type": "@n8n/n8n-nodes-langchain.agent",
```

```

"typeVersion": 3.1,
```

```

"position": [
```

```

  1456,
```

```

  -400
```

```

],
```

```

"id": "063fc392-1ab4-4f52-b157-36076443e58c",
```

```

"name": "AI Agent"

```

```

    },
    {
      "parameters": {
        "jsCode": "const itemsOut = [];\n\nfor (const item of items) {\n  try {\n    // 1. Obtener respuesta del AI\n    let raw = item.json.output || item.json.respuesta || item.json;\n\n    // 2. Si viene como string, limpiarlo\n    if (typeof raw === \"string\") {\n      raw = raw.replace(/``json|``/g, \"\").trim();\n      raw = JSON.parse(raw);\n    }\n\n    // 3. Validar tipo\n    if (raw.tipo !== \"gasto\") {\n      return items; // no hacer nada si no es gasto\n    }\n\n    // 4. Extraer datos\n    const data = raw.data;\n\n    itemsOut.push({\n      json: {\n        fecha: data.fecha,\n        categoria: data.categoria,\n        monto: data.monto,\n        descripcion: data.descripcion\n      }\n    });\n\n  } catch (error) {\n    console.log(\"Error parsing:\", error);\n  }\n}\n\nreturn itemsOut;"
      },
      "type": "n8n-nodes-base.code",
      "typeVersion": 2,
      "position": [
        1904,
        -480
      ],
      "id": "b9ccbc81-0f80-4c5d-9eb9-eda2506140de",

```

```
"name": "Code in JavaScript"
},
{
  "parameters": {
    "assignments": {
      "assignments": [
        {
          "id": "194ae908-2f1c-46f4-808a-face883d15be",
          "name": "texto",
          "value": "={{ $json.message.text }}",
          "type": "string"
        }
      ]
    },
    "options": {}
  },
  "type": "n8n-nodes-base.set",
  "typeVersion": 3.4,
```

```
"position": [  
  
  768,  
  
  -576  
  
],  
  
"id": "d4305949-9fbd-43f4-af8b-e4f0116e2ee1",  
  
"name": "chat a texto"  
  
},  
  
{  
  
  "parameters": {  
  
    "assignments": {  
  
      "assignments": [  
  
        {  
  
          "id": "c5f770e0-8864-4973-b6cc-cd9587654839",  
  
          "name": "texto",  
  
          "value": "={{ $json.content.parts[0].text }}",  
  
          "type": "string"  
  
        }  
  
      ]  
  
    }  
  
  }  
  
}
```

```
    },  
  
    "options": {}  
  
  },  
  
  "type": "n8n-nodes-base.set",  
  
  "typeVersion": 3.4,  
  
  "position": [  
  
    1312,  
  
    -400  
  
  ],  
  
  "id": "a53773d1-8b30-404c-9d78-497644246638",  
  
  "name": "variables texto"  
  
},  
  
{  
  
  "parameters": {  
  
    "conditions": {  
  
      "options": {  
  
        "caseSensitive": true,  
  
        "leftValue": "",
```

```
"typeValidation": "strict",

"version": 3

},

"conditions": [

{

  "id": "7930ed89-4894-4734-97ba-87ab6bbb1adf",

  "leftValue": "={{ $json.tipo === \"gasto\" }}",

  "rightValue": "",

  "operator": {

    "type": "boolean",

    "operation": "true",

    "singleValue": true

  }

},

{

  "id": "41673413-cdaa-45da-b90c-23fd89b3caab",

  "leftValue": "={{ $json.tipo === \"consulta\" }}",

  "rightValue": "",
```

```
"operator": {  
  
  "type": "boolean",  
  
  "operation": "false",  
  
  "singleValue": true  
  
}  
  
},  
  
  "combinator": "and"  
  
},  
  
  "options": {}  
  
},  
  
  "type": "n8n-nodes-base.if",  
  
  "typeVersion": 2.3,  
  
  "position": [  
  
    1728,  
  
    -400  
  
  ],  
  
  "id": "5ceca948-c8fb-4722-a331-34b7fa421e7c",
```

```

"name": "If"

},

{

  "parameters": {

    "jsonSchemaExample": "{\n  \"tipo\": \"string\",\n  \"data\": {\n    \"fecha\":\n\"string\",\n    \"categoria\": \"string\",\n    \"monto\": \"number\",\n    \"descripcion\":\n\"string\"\n  },\n  \"respuesta\": \"string\"\n}"

  },

  "type": "@n8n/n8n-nodes-langchain.outputParserStructured",

  "typeVersion": 1.3,

  "position": [

    1584,

    -144

  ],

  "id": "03c7cf57-abec-496f-93b8-c6e112ce0567",

  "name": "Structured Output Parser"

},

{

  "parameters": {

```

```

"jsCode": "let total = 0;\nlet categorias = {};\n\n//  TRAER DATOS DEL AI\nconst ai =
$items(\"AI Agent\")[0]?.json?.output || {};\n\n//  EXTRAER DATOS\nlet tipo =
ai?.data?.tipo_consulta || \"\";\nlet categoriaFiltro = (ai?.data?.categoria ||
\"\" ).toLowerCase();\n\n//  FALLBACK (si el AI falla)\nif (!tipo) {\n  if (categoriaFiltro)
{\n    tipo = \"categoria\";\n  } else {\n    tipo = \"total\";\n  }\n}\n\n//  RECORRER DATOS
DEL SHEET\nfor (const item of items) {\n  const categoria = (\n    item.json.Descripcion || //
prioridad correcta\n    item.json.Categoria || \"\" ).toLowerCase();\n\n  const monto =
Number(item.json.Monto || 0);\n\n  total += monto;\n\n  if (!categorias[categoria])
{\n    categorias[categoria] = 0;\n  }\n\n  categorias[categoria] += monto;\n}\n\n//
RESPUESTAS\n\n//  TOTAL GENERAL\nif (tipo === \"total\") {\n  return [\n    {\n      json:
{\n        respuesta: ` Has gastado $$${total} en total` \n      }\n    }];\n}\n\n//  POR
CATEGORÍA (ej: café)\nif (tipo === \"categoria\") {\n  return [\n    {\n      json:
{\n        respuesta: `☹️ Has gastado $$${categorias[categoriaFiltro] || 0} en
${categoriaFiltro} \n      }\n    }];\n}\n\n//  COMPARACIÓN (en qué gastas más)\nif (tipo
=== \"comparacion\") {\n  let maxCategoria = \"\";\n  let maxValor = 0;\n\n  for (const cat in
categorias) {\n    if (categorias[cat] > maxValor) {\n      maxValor =
categorias[cat];\n      maxCategoria = cat;\n    }\n  }\n\n  return [\n    {\n      json:
{\n        respuesta: ` Gastas más en $${maxCategoria}
($${maxValor})` \n      }\n    }];\n}\n\n//  FALLBACK FINAL\nreturn [\n  {\n    json:
{\n      respuesta: ` No entendí la consulta` \n    }\n  }];"
},
"type": "n8n-nodes-base.code",

```

```
"typeVersion": 2,  
  
"position": [  
  
  2128,  
  
  -304  
  
],  
  
"id": "f763750b-d754-4f74-83ab-98808dbd7e41",  
  
"name": "Code in JavaScript1"  
  
},  
  
{  
  
  "parameters": {  
  
    "chatId": "={{ $('Telegram Trigger').item.json.message.chat.id }}",  
  
    "text": "={{ $json.respuesta }}",  
  
    "additionalFields": {  
  
      "appendAttribution": false  
  
    }  
  
  },  
  
  "type": "n8n-nodes-base.telegram",  
  
  "typeVersion": 1.2,
```

```
"position": [  
  
  2304,  
  
  -304  
  
],  
  
"id": "7e81d55a-f385-45d9-a742-aff437195395",  
  
"name": "Send a text message",  
  
"webhookId": "172ad220-45b6-4bd6-b45c-dce8052d801a",  
  
"credentials": {  
  
  "telegramApi": {  
  
    "id": "rEcHgB9avrO5uA39",  
  
    "name": "Telegram account"  
  
  }  
  
}  
  
},  
  
{  
  
  "parameters": {  
  
    "operation": "append",  
  
    "documentId": {
```

```

    "__rl": true,

    "value": "1HEnbjvtBDlixibplgafZdZpDgwNN-fS9VE822wWnhaY",

    "mode": "list",

    "cachedResultName": "Control de gastos",

    "cachedResultUrl":

    "https://docs.google.com/spreadsheets/d/1HEnbjvtBDlixibplgafZdZpDgwNN-
    fS9VE822wWnhaY/edit?usp=drivesdk"

    },

    "sheetName": {

        "__rl": true,

        "value": "gid=0",

        "mode": "list",

        "cachedResultName": "Hoja 1",

        "cachedResultUrl":

        "https://docs.google.com/spreadsheets/d/1HEnbjvtBDlixibplgafZdZpDgwNN-
        fS9VE822wWnhaY/edit#gid=0"

        },

    "columns": {

        "mappingMode": "defineBelow",

```

```
"value": {  
  
  "Fecha": "={{ $json.fecha }}",  
  
  "Categoria": "={{ $json.categoria }}",  
  
  "Monto": "={{ $json.monto }}",  
  
  "Descripcion": "={{ $json.descripcion }}"  
  
},  
  
"matchingColumns": [],  
  
"schema": [  
  
  {  
  
    "id": "Fecha",  
  
    "displayName": "Fecha",  
  
    "required": false,  
  
    "defaultMatch": false,  
  
    "display": true,  
  
    "type": "string",  
  
    "canBeUsedToMatch": true  
  
  },  
  
  {
```

```
"id": "Categoria",  
  
"displayName": "Categoria",  
  
"required": false,  
  
"defaultMatch": false,  
  
"display": true,  
  
"type": "string",  
  
"canBeUsedToMatch": true  
},  
  
{  
  
"id": "Monto",  
  
"displayName": "Monto",  
  
"required": false,  
  
"defaultMatch": false,  
  
"display": true,  
  
"type": "string",  
  
"canBeUsedToMatch": true  
},  
  
{
```

```
"id": "Descripcion",  
  
"displayName": "Descripcion",  
  
"required": false,  
  
"defaultMatch": false,  
  
"display": true,  
  
"type": "string",  
  
"canBeUsedToMatch": true  
  
}  
  
],  
  
"attemptToConvertTypes": false,  
  
"convertFieldsToString": false  
  
},  
  
"options": {}  
  
},  
  
"type": "n8n-nodes-base.googleSheets",  
  
"typeVersion": 4.7,  
  
"position": [  
  
2112,
```

-480

],

"id": "94cbe5b4-279c-4a36-9ebc-bea1c9608bf6",

"name": "Agregar datos",

"credentials": {

"googleSheetsOAuth2Api": {

"id": "i9NvIujvyNCqOS1e",

"name": "Google Sheets account"

}

}

},

{

"parameters": {

"documentId": {

"\_\_rl": true,

"value": "1HEnbjvtBDlixibplgafZdzpDgwNN-fS9VE822wWnhaY",

"mode": "list",

"cachedResultName": "Control de gastos",

```
"cachedResultUrl":  
"https://docs.google.com/spreadsheets/d/1HEnbjvtBDlixibplgafZd zpDgwNN-  
fS9VE822wWnhaY/edit?usp=drivesdk"  
  
},  
  
"sheetName": {  
  
  "__rl": true,  
  
  "value": "gid=0",  
  
  "mode": "list",  
  
  "cachedResultName": "Hoja 1",  
  
  "cachedResultUrl":  
"https://docs.google.com/spreadsheets/d/1HEnbjvtBDlixibplgafZd zpDgwNN-  
fS9VE822wWnhaY/edit#gid=0"  
  
},  
  
"options": {}  
  
},  
  
"type": "n8n-nodes-base.googleSheets",  
  
"typeVersion": 4.7,  
  
"position": [  
  
  1920,
```

-304

],

"id": "ea5723b0-8ea7-44fc-9fc7-8e16ab96c7c7",

"name": "Obtener datos",

"credentials": {

"googleSheetsOAuth2Api": {

"id": "i9NvIujvyNCqOS1e",

"name": "Google Sheets account"

}

}

},

{

"parameters": {

"chatId": "={{ \$('Telegram Trigger').item.json.message.chat.id }}",

"text": "={{ \" Guardado: \$\" + \$json.Monto + \" en \" + (\$json.Descripcion ||

\$json.Categoria) }}",

"additionalFields": {

"appendAttribution": false

}

```
    },  
  
    "type": "n8n-nodes-base.telegram",  
  
    "typeVersion": 1.2,  
  
    "position": [  
  
        2288,  
  
        -480  
  
    ],  
  
    "id": "cdabf87f-0d1a-4902-8806-739f6154ea42",  
  
    "name": "Send a text message1",  
  
    "webhookId": "dcf3cd48-8c02-43da-a238-705f9760800d",  
  
    "credentials": {  
  
        "telegramApi": {  
  
            "id": "rEcHgB9avrO5uA39",  
  
            "name": "Telegram account"  
  
        }  
  
    }  
  
    }  
  
    ],
```

```
"pinData": {},  
  
"connections": {  
  
  "Telegram Trigger": {  
  
    "main": [  
  
      [  
  
        {  
  
          "node": "Switch",  
  
          "type": "main",  
  
          "index": 0  
  
        }  
  
      ]  
  
    ]  
  
  },  
  
  "Switch": {  
  
    "main": [  
  
      [  
  
        {  
  
          "node": "chat a texto",
```

```
"type": "main",  
  
"index": 0  
  
}  
  
],  
  
[  
  
{  
  
"node": "Get a file",  
  
"type": "main",  
  
"index": 0  
  
}  
  
],  
  
[  
  
{  
  
"node": "Get a file1",  
  
"type": "main",  
  
"index": 0  
  
}  
  
]
```

```
]
},
"Get a file": {
  "main": [
    [
      {
        "node": "Transcribe a recording",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Get a file1": {
  "main": [
    [
      {
        "node": "Extract from File",
```

```
    "type": "main",  
    "index": 0  
  }  
]  
]  
},  
"Extract from File": {  
  "main": [  
    [  
      {  
        "node": "Convert to File",  
        "type": "main",  
        "index": 0  
      }  
    ]  
  ]  
},  
"Convert to File": {
```

```
"main": [  
  
  [  
  
    {  
  
      "node": "Analyze an image",  
  
      "type": "main",  
  
      "index": 0  
  
    }  
  
  ]  
  
],  
  
},
```

```
"Analyze an image": {  
  
  "main": [  
  
    [  
  
      {  
  
        "node": "imagen to texto",  
  
        "type": "main",  
  
        "index": 0  
  
      }  
  
    ]  
  
  ]  
  
}
```

```
]
]
},
"Transcribe a recording": {
  "main": [
    [
      {
        "node": "variables texto",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"imagen to texto": {
  "main": [
    [
      {
```

```
"node": "variables texto",

"type": "main",

"index": 0

}

]

]

},

"Google Gemini Chat Model2": {

  "ai_languageModel": [

    [

      {

        "node": "AI Agent",

        "type": "ai_languageModel",

        "index": 0

      }

    ]

  ]

},
```

```
"AI Agent": {  
  
  "main": [  
  
    [  
  
      {  
  
        "node": "If",  
  
        "type": "main",  
  
        "index": 0  
  
      }  
  
    ]  
  
  ]  
  
},  
  
"Code in JavaScript": {  
  
  "main": [  
  
    [  
  
      {  
  
        "node": "Agregar datos",  
  
        "type": "main",  
  
        "index": 0
```

```
    }  
  ]  
]  
},  
"chat a texto": {  
  "main": [  
    [  
      {  
        "node": "variables texto",  
        "type": "main",  
        "index": 0  
      }  
    ]  
  ]  
},  
"variables texto": {  
  "main": [  
    [  

```

```
{  
  "node": "AI Agent",  
  "type": "main",  
  "index": 0  
}  
]  
]  
},  
"If": {  
  "main": [  
    [  
      {  
        "node": "Code in JavaScript",  
        "type": "main",  
        "index": 0  
      }  
    ],  
    [  

```

```
{  
  "node": "Obtener datos",  
  "type": "main",  
  "index": 0  
}  
]  
]  
},  
"Structured Output Parser": {  
  "ai_outputParser": [  
    [  
      {  
        "node": "AI Agent",  
        "type": "ai_outputParser",  
        "index": 0  
      }  
    ]  
  ]  
}
```

```
},
```

```
"Code in JavaScript1": {
```

```
  "main": [
```

```
    [
```

```
      {
```

```
        "node": "Send a text message",
```

```
        "type": "main",
```

```
        "index": 0
```

```
      }  
    ]
```

```
  ]
```

```
]
```

```
},
```

```
"Obtener datos": {
```

```
  "main": [
```

```
    [
```

```
      {
```

```
        "node": "Code in JavaScript1",
```

```
        "type": "main",
```

```
    "index": 0
  }
]
],
},
"Agregar datos": {
  "main": [
    [
      {
        "node": "Send a text message1",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"active": false,
```

```
"settings": {  
  
  "executionOrder": "v1",  
  
  "binaryMode": "separate"  
  
},  
  
"versionId": "f258d14f-9544-45b9-b48d-58548eea60e3",  
  
"meta": {  
  
  "templateCredsSetupCompleted": true,  
  
  "instanceId": "98062e833ff3473ea6b60a60d03ce36c5152247bd04884a2bbe148eb16ec7839"  
  
},  
  
"id": "JHv5pZttl35432FA",  
  
"tags": []  
  
}
```

## **Conclusión**

El desarrollo de este asistente demuestra cómo la integración de herramientas de automatización, inteligencia artificial y servicios en la nube permite construir soluciones prácticas para la gestión de información personal.

A través del uso de flujos automatizados, el sistema logra transformar entradas no estructuradas (texto, voz e imágenes) en datos organizados, permitiendo el registro de gastos.

Además, la arquitectura permite una fácil evolución del sistema. Componentes como el agente de IA, la lógica de procesamiento y el almacenamiento pueden adaptarse o ampliarse sin afectar el funcionamiento general, lo que lo convierte en una solución robusta y mantenible.