

TRABAJO DE GRADO
Opción Seminario-Diplomado.

Sistema de Inventario y Procesamiento de Registros

Corporación Universitaria Remington.

Facultad de Ingenierías

Ingeniería de Sistemas

Presentado por:

Miguel Angel Ibarra Ocampo

Tutor: Luis Camargo Ortega

Opción de Trabajo de grado Seminario.

2026

Tabla de contenido

Resumen.....	3
Introducción	4
Planteamiento del problema.....	5
Código fuente n8n.....	6
Figura del flujo de Sistema de Inventario y Procesamiento de Registros.....	21
Conclusión	22
Referencias bibliográficas.....	24

Resumen

El Sistema de Inventario y Procesamiento de Registros es una infraestructura digital de gestión operativa diseñada para automatizar el flujo de ventas, el control de inventario y la administración de clientes en un entorno de manufactura personalizada. El sistema actúa como un puente inteligente entre la comunicación humana informal a través de mensajería instantánea y una arquitectura de datos estructurada en la nube.

Introducción

El presente documento detalla el desarrollo e implementación del Sistema de Inventario y Procesamiento de Registros. Esta solución surge con el objetivo de centralizar y automatizar la operativa del negocio mediante una arquitectura basada en Inteligencia Artificial y Orquestación de Flujos de Trabajo n8n. A diferencia de los sistemas de gestión tradicionales que requieren interfaces complejas, se utiliza el Procesamiento de Lenguaje Natural (NLP) para convertir conversaciones cotidianas en datos estructurados.

A través de la integración de tecnologías de vanguardia como n8n, PostgreSQL y la API de Gemini, se ha logrado crear un ecosistema autónomo donde un agente de IA interpreta las necesidades del administrador y ejecuta acciones directas sobre la base de datos. Esta integración no solo optimiza el tiempo de respuesta, sino que profesionaliza la estructura administrativa del emprendimiento, permitiendo una escalabilidad sostenible apoyada en la eficiencia tecnológica.

Planteamiento del problema

En el sector del estampado y la personalización de prendas, la eficiencia operativa depende de la precisión con la que se capturan los detalles de cada pedido. Actualmente, muchos emprendimientos y microempresas gestionan sus interacciones con los clientes principalmente a través de canales de mensajería instantánea. Si bien esto facilita la comunicación, genera una desconexión crítica entre la toma del pedido y la actualización de los sistemas de inventario y registro contable.

Fragmentación de la Información: Los datos de los clientes, los estados de los pedidos y los niveles de stock se encuentran dispersos, lo que dificulta una consulta rápida y veraz.

Carga Operativa Manual: El proceso de registrar cada venta, verificar disponibilidad de prendas y actualizar abonos requiere una intervención humana constante y repetitiva, restando tiempo a la producción creativa (el estampado).

Margen de Error Elevado: La transcripción manual de datos desde un chat hacia una base de datos o cuaderno aumenta el riesgo de errores en las tallas, nombres de clientes o montos adeudados, lo que se traduce en pérdidas económicas y fallos en la experiencia del cliente.

Dificultad en el Seguimiento: La falta de un sistema centralizado impide conocer con exactitud y en tiempo real qué pedidos están pendientes por entregar o qué clientes tienen saldos pendientes.

Código fuente n8n

```

{
  "name": "Asistente Virtual",
  "nodes": [
    {
      "parameters": {
        "promptType": "define",
        "text": "={{ $json.message.text }}",
        "options": {
          "systemMessage": "=Eres el cerebro operativo de \"Tribu de Judah\", experto en gestión de pedidos de estampados. Tu objetivo es la eficiencia absoluta.\n\n## 1. PROTOCOLOS DE HERRAMIENTAS (LÓGICA DE INTENCIÓN)\nNo inventes datos ni escribas código SQL. Usa las herramientas según la necesidad específica de Daniel. **Identifica si la orden es un registro parcial o una venta completa.**\n\n### A. Registro de Cliente (Intención única)\n- Si Daniel solo pide registrar, buscar o agregar a un cliente: Usa `buscar_o_crear_cliente`.\n- **SALIDA INMEDIATA:** Una vez obtenida la respuesta de la herramienta, confirma el éxito y **DETENTE**.

```

\"Running...\"). Ejecuta las herramientas primero y responde ÚNICAMENTE con el reporte final una vez tengas los resultados.\n- **Error de Formato:** No uses guiones bajos `_` ni `**` en el texto de respuesta, ya que rompen el formato de Telegram.\n- **Tono:** Directo y profesional.\n- **Confirmación:** Usa siempre el check: [Acción realizada]\n.\n\n## 4. FORMATO DE REPORTE (Ejemplo)\n📦 [Nombre Cliente]\n• [Cantidad] [Prenda] - [Descripción Estampado]\n📅 Entrega: [Fecha] | 💰 Saldo: \$[Monto]\n\n## 5. NO USES FORMATO MARKDOWN\n(EJEMPLO) NO USES **PARA ENVIAR TU RESPUESTA**, deja el texto simple\n--\n**Contexto Temporal:**\nHoy es {{ \$now.setZone('America/Bogota').format('dddd, DD [de] MMMM [de] YYYY') }}.",

```

      "maxIterations": 7
    }
  },
  "type": "@n8n/n8n-nodes-langchain.agent",
  "typeVersion": 3.1,
  "position": [
    -304,
    -224
  ],
  "id": "4c211c79-ad19-4d55-af62-229967a21f0d",
  "name": "AI Agent",
  "alwaysOutputData": false,
  "executeOnce": false,
  "notesInFlow": false
},
{
  "parameters": {
    "modelName": "models/gemini-2.5-flash-lite",
    "options": {}
  },
  "type": "@n8n/n8n-nodes-langchain.lmChatGoogleGemini",
  "typeVersion": 1,
  "position": [
    -512,
    48
  ],
  "id": "8289d403-1573-453f-8f05-d945ea968694",
  "name": "Google Gemini Chat Model",
  "credentials": {
    "googlePalmApi": {
      "id": "ZvooaVHu5Jb0oHGu",
      "name": "geminimm"
    }
  }
}
},

```

```

{
  "parameters": {
    "sessionIdType": "customKey",
    "sessionKey": "={{ $json.message.from.id }}",
    "contextWindowLength": 2
  },
  "type": "@n8n/n8n-nodes-langchain.memoryBufferWindow",
  "typeVersion": 1.3,
  "position": [
    -368,
    48
  ],
  "id": "41b21d03-87fb-4670-8053-02197ac0122e",
  "name": "Simple Memory"
},
{
  "parameters": {
    "chatId": "={{ $('Recibir_mensaje').item.json.message.chat.id }}",
    "text": "={{\n$json.output\n .replace(/[*_`~]/g, '')\n}}",
    "additionalFields": {
      "appendAttribution": false
    }
  },
  "type": "n8n-nodes-base.telegram",
  "typeVersion": 1.2,
  "position": [
    320,
    -224
  ],
  "id": "16b30261-46ba-400a-9362-151069718635",
  "name": "Respuesta",
  "webhookId": "f969a3e3-0525-4ae6-a5cb-fed56993fe89",
  "notesInFlow": false,
  "credentials": {
    "telegramApi": {
      "id": "AY0kdYrCcvCdebpm",
      "name": "BotTribeOfJuda"
    }
  }
},
{
  "parameters": {
    "rule": {
      "interval": [
        {

```

```

        "triggerAtHour": 8
      }
    ]
  },
  "type": "n8n-nodes-base.scheduleTrigger",
  "typeVersion": 1.3,
  "position": [
    -288,
    592
  ],
  "id": "575e7c60-9d31-4559-8a2e-26e78cb8234d",
  "name": "Schedule Trigger"
},
{
  "parameters": {
    "chatId": "=8605583464",
    "text": "=📅 REPORTE DE ENTREGAS DEL DIA DE HOY\n{{$json.pedido_formateado }}",
    "additionalFields": {
      "appendAttribution": false,
      "parse_mode": "Markdown"
    }
  },
  "type": "n8n-nodes-base.telegram",
  "typeVersion": 1.2,
  "position": [
    384,
    592
  ],
  "id": "f5b95ce2-8291-4c87-954a-58cc505e8990",
  "name": "Reporte diario",
  "webhookId": "f969a3e3-0525-4ae6-a5cb-fed56993fe89",
  "credentials": {
    "telegramApi": {
      "id": "AY0kdYrCcvCdebpm",
      "name": "BotTribeOfJuda"
    }
  }
},
{
  "parameters": {
    "operation": "executeQuery",
    "query": "SELECT \n      '• *' || c.nombre || '*: ' || \n      d.cantidad ||
' ' || i.tipo_prenda || ' - ' || \n      d.descripcion_estampado || \n      ' [Saldo:"

```

```

'$' || p.saldo_pendiente || ']' as pedido_formateado\nFROM pedidos p\nJOIN
clientes c ON p.cliente_id = c.id\nJOIN detalles_pedido d ON d.pedido_id =
p.id\nJOIN inventario i ON d.camiseta_id = i.id\nWHERE p.fecha_promesa_entrega =
CURRENT_DATE;";
  "options": {}
},
"type": "n8n-nodes-base.postgres",
"typeVersion": 2.6,
"position": [
  -64,
  592
],
"id": "14fd954a-d11a-48a1-9ad3-602a378799ce",
"name": "Execute a SQL query",
"credentials": {
  "postgres": {
    "id": "xzFsNG2I5CeqBC1M",
    "name": "bd_supabase"
  }
}
},
{
  "parameters": {
    "fieldsToAggregate": {
      "fieldToAggregate": [
        {
          "fieldToAggregate": "pedido_formateado"
        }
      ]
    },
    "options": {
      "mergeLists": true
    }
  },
  "type": "n8n-nodes-base.aggregate",
  "typeVersion": 1,
  "position": [
    160,
    592
  ],
  "id": "1c172295-709a-4afb-b74b-ff78b59fd710",
  "name": "Aggregate"
},
{
  "parameters": {

```

```

      "descriptionType": "manual",
      "toolDescription": "Busca un cliente por nombre. Si existe, devuelve el
ID. Si no existe, crea uno nuevo y devuelve el ID.",
      "operation": "executeQuery",
      "query": "INSERT INTO clientes (nombre, telefono)\nVALUES
('{{$json.nombre}}', '{{$json.telefono}}')\nON CONFLICT (nombre) \nDO UPDATE SET
telefono = EXCLUDED.telefono\nRETURNING id;",
      "options": {}
    },
    "type": "n8n-nodes-base.postgresTool",
    "typeVersion": 2.6,
    "position": [
      -224,
      16
    ],
    "id": "f1790401-52fa-4314-ba16-875220ccf1c0",
    "name": "buscar_o_crear_cliente",
    "credentials": {
      "postgres": {
        "id": "xzFsNG2I5CeqBC1M",
        "name": "bd_supabase"
      }
    }
  },
  {
    "parameters": {
      "descriptionType": "manual",
      "toolDescription": "Consulta el stock, precio de venta y prendas. DEBES
enviar el parámetro 'palabra_clave' con el nombre de la prenda que el usuario
solicita.",
      "operation": "executeQuery",
      "query": "SELECT id, tipo_prenda, precio_venta,
unidades_disponibles\nFROM inventario\nWHERE unaccent(tipo_prenda) ILIKE '%' ||
unaccent('{{ $fromAI(\"palabra_clave\") }}') || '%';",
      "options": {}
    },
    "type": "n8n-nodes-base.postgresTool",
    "typeVersion": 2.6,
    "position": [
      -64,
      16
    ],
    "id": "f066f427-546b-4404-9255-dd87f270b790",
    "name": "consultar_precios",
    "credentials": {

```

```

    "postgres": {
      "id": "xzFsNG2I5CeqBC1M",
      "name": "bd_supabase"
    }
  },
  {
    "parameters": {
      "descriptionType": "manual",
      "toolDescription": "Muestra la lista de pedidos pendientes con nombres de
clientes y detalles de prendas. No muestra IDs técnicos.",
      "operation": "executeQuery",
      "query": "SELECT \n      c.nombre AS
cliente,\n      i.tipo_prenda,\n      d.cantidad,\n      d.descripcion_ estampado AS
estampado,\n      p.total,\n      p.saldo_pendiente,\n      p.fecha_promesa_entrega AS
entrega\nFROM pedidos p\nJOIN clientes c ON p.cliente_id = c.id\nJOIN
detalles_pedido d ON d.pedido_id = p.id\nJOIN inventario i ON d.camiseta_id =
i.id\nWHERE p.estado_entrega = 'Pendiente'\nORDER BY p.fecha_promesa_entrega
ASC;",
      "options": {}
    },
    "type": "n8n-nodes-base.postgresTool",
    "typeVersion": 2.6,
    "position": [
      96,
      16
    ],
    "id": "8e2ab0b5-80f8-4102-b670-413ad56df0ab",
    "name": "ver_pedidos_activos",
    "credentials": {
      "postgres": {
        "id": "xzFsNG2I5CeqBC1M",
        "name": "bd_supabase"
      }
    }
  },
  {
    "parameters": {
      "descriptionType": "manual",
      "toolDescription": "Registra una venta nueva en la tabla pedidos
Necesita: cliente_id, fecha_promesa_entrega, total, saldo_pendiente, estado de
entrega \ne inserta los detalles del pedido, necesita: camiseta_id camiseta_id,
cantidad, descripcion_ estampado.",
      "operation": "executeQuery",

```

```

      "query": "WITH nuevo_pedido AS (\n      INSERT INTO pedidos (cliente_id,\nfecha_promesa_entrega, total, saldo_pendiente, estado_entrega)\n      VALUES\n(\n      '{{ $json.cliente_id }}',\n      '{{ $json.fecha_promesa_entrega\n}}',\n      {{ $json.total}},\n      {{ $json.saldo_pendiente\n}},\n      'Pendiente'\n      )\n      RETURNING id)\n\nINSERT INTO detalles_pedido\n(pedido_id, camiseta_id, cantidad, descripcion_estampado)\nSELECT \n      id,\n      '{{ $json.camiseta_id }}',\n      {{ $json.cantidad }},\n      '{{\n$json.descripcion_estampado }}'\nFROM nuevo_pedido;",
      "options": {}
    },
    "type": "n8n-nodes-base.postgresTool",
    "typeVersion": 2.6,
    "position": [
      80,
      176
    ],
    "id": "40ce81d3-d69c-4709-9d01-9ee8b191e002",
    "name": "registrar_venta",
    "credentials": {
      "postgres": {
        "id": "xzFsNG2I5CeqBC1M",
        "name": "bd_supabase"
      }
    }
  },
  {
    "parameters": {
      "descriptionType": "manual",
      "toolDescription": "Cambia el estado de entrega de un pedido (ej: de\n'Pendiente' a 'Entregado'). Requiere el id del pedido.",
      "operation": "executeQuery",
      "query": "UPDATE pedidos\nSET \n      estado_entrega =\n:nuevo_estado,\n      saldo_pendiente = CASE WHEN :pago_completo = true THEN 0 ELSE\nsaldo_pendiente END\nWHERE id = :id_pedido;",
      "options": {}
    },
    "type": "n8n-nodes-base.postgresTool",
    "typeVersion": 2.6,
    "position": [
      -176,
      160
    ],
    "id": "5342dea5-8884-4f76-bdc6-ed77de4b2f94",
    "name": "actualizar_estado_pedido",
    "credentials": {

```

```

    "postgres": {
      "id": "xzFsNG2I5CeqBC1M",
      "name": "bd_supabase"
    }
  },
  {
    "parameters": {
      "descriptionType": "manual",
      "toolDescription": "Consulta el stock, precio de venta y tipos de prendas disponibles en el inventario",
      "operation": "executeQuery",
      "query": "SELECT tipo_prenda, precio_venta, unidades_disponibles \nFROM inventario \n",
      "options": {}
    },
    "type": "n8n-nodes-base.postgresTool",
    "typeVersion": 2.6,
    "position": [
      -48,
      160
    ],
    "id": "345b968e-190f-462a-a4a5-ad6cc7c9215c",
    "name": "Consultar _Inventario",
    "credentials": {
      "postgres": {
        "id": "xzFsNG2I5CeqBC1M",
        "name": "bd_supabase"
      }
    }
  },
  {
    "parameters": {
      "content": "## Disparador del Flujo\n**Al recibir un mensaje inicia el flujo del programa y se lo envia al agente** ",
      "height": 240,
      "width": 320
    },
    "type": "n8n-nodes-base.stickyNote",
    "position": [
      -912,
      -320
    ],
    "typeVersion": 1,
    "id": "5b290efb-023a-4687-810d-55de1bf3544a",

```

```

    "name": "Sticky Note"
  },
  {
    "parameters": {
      "updates": [
        "message"
      ],
      "additionalFields": {}
    },
    "type": "n8n-nodes-base.telegramTrigger",
    "typeVersion": 1.2,
    "position": [
      -816,
      -224
    ],
    "id": "a2c5f2fb-e79f-457b-9764-3b76da99e062",
    "name": "Recibir_mensaje",
    "webhookId": "47167c72-c174-4e5c-93e7-52220a774bb9",
    "credentials": {
      "telegramApi": {
        "id": "AY0kdYrCcvCdebpm",
        "name": "BotTribeOfJuda"
      }
    }
  },
  {
    "parameters": {
      "content": "## Agente de AI\n**El agente recibe el mensaje y dependiendo de la petición decide cuál herramienta debe usar** ",
      "height": 624,
      "width": 816,
      "color": 5
    },
    "type": "n8n-nodes-base.stickyNote",
    "position": [
      -560,
      -320
    ],
    "typeVersion": 1,
    "id": "4a522703-8341-4772-ba81-ccf8b6f2de1f",
    "name": "Sticky Note1"
  },
  {
    "parameters": {

```

```

    "content": "## Respuesta\n**El nodo recibe una respuesta y se la envía al usuario** ",
    "height": 256,
    "width": 272,
    "color": 4
  },
  "type": "n8n-nodes-base.stickyNote",
  "position": [
    256,
    -320
  ],
  "typeVersion": 1,
  "id": "e2e0a0eb-b9e7-4498-b987-a2d01e0eee54",
  "name": "Sticky Note2"
},
{
  "parameters": {
    "content": "## Flujo de recordatorio Diario\n**Este flujo diariamente le envia un mensaje al usuario indicandole los pedidos que tiene por entregar para la fecha de ese dia**\n",
    "height": 400,
    "width": 1040
  },
  "type": "n8n-nodes-base.stickyNote",
  "position": [
    -352,
    416
  ],
  "typeVersion": 1,
  "id": "0e003560-3a01-407a-bc64-4fb81a0efe8f",
  "name": "Sticky Note3"
}
],
"pinData": {},
"connections": {
  "Google Gemini Chat Model": {
    "ai_languageModel": [
      [
        {
          "node": "AI Agent",
          "type": "ai_languageModel",
          "index": 0
        }
      ]
    ]
  }
}
]

```

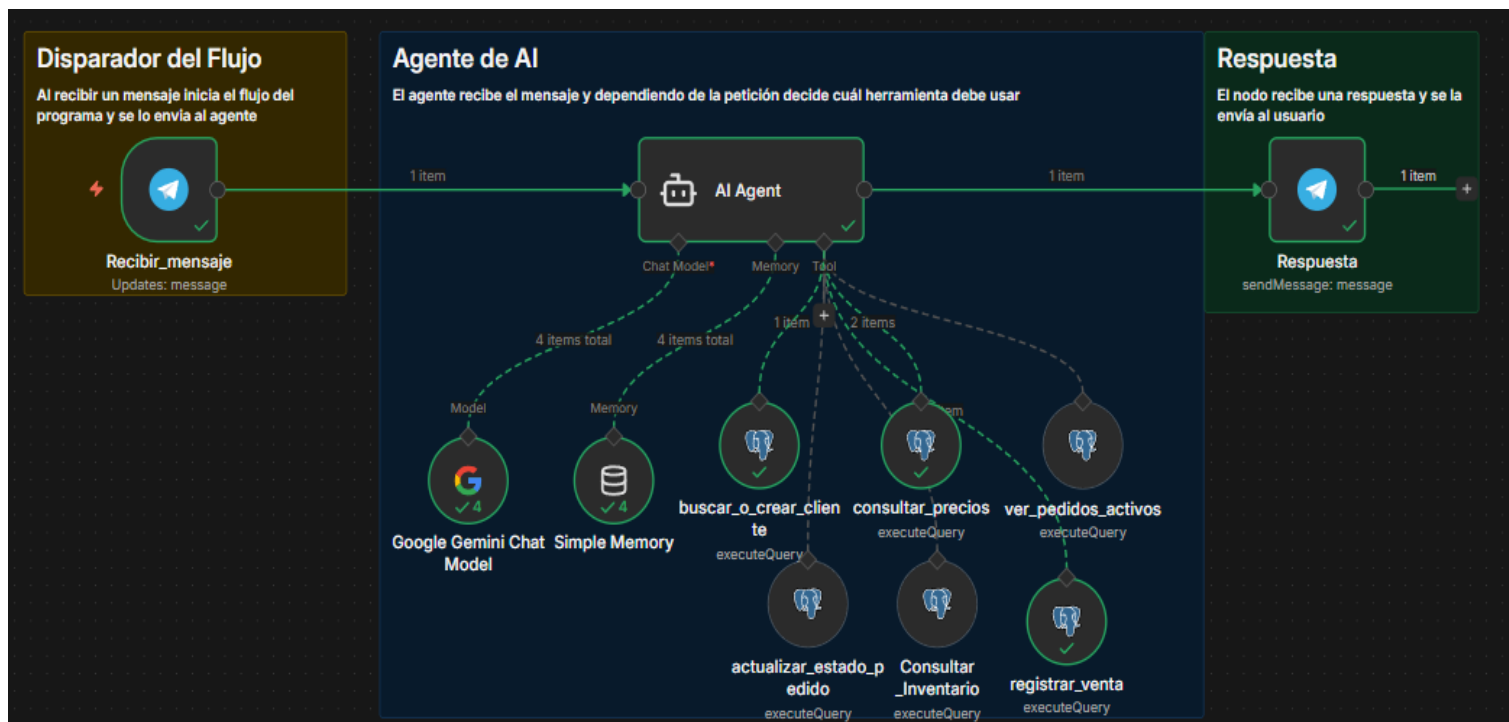
```
},
"AI Agent": {
  "main": [
    [
      {
        "node": "Respuesta",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Simple Memory": {
  "ai_memory": [
    [
      {
        "node": "AI Agent",
        "type": "ai_memory",
        "index": 0
      }
    ]
  ]
},
"Schedule Trigger": {
  "main": [
    [
      {
        "node": "Execute a SQL query",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Reporte diario": {
  "main": [
    []
  ]
},
"Execute a SQL query": {
  "main": [
    [
      {
        "node": "Aggregate",
        "type": "main",
```

```
        "index": 0
      }
    ]
  ],
},
"Aggregate": {
  "main": [
    [
      {
        "node": "Reporte diario",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"buscar_o_crear_cliente": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"consultar_precios": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"ver_pedidos_activos": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
}
```

```
    }
  ]
]
},
"registrar_venta": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"actualizar_estado_pedido": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"Consultar _Inventario": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"Recibir_mensaje": {
  "main": [
    [
      {
        "node": "AI Agent",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
```

```
    ]
  ]
}
},
"active": false,
"settings": {
  "executionOrder": "v1",
  "binaryMode": "separate",
  "availableInMCP": false,
  "timeSavedMode": "fixed",
  "timezone": "America/Bogota",
  "callerPolicy": "workflowsFromSameOwner"
},
"versionId": "b0bd5528-5b7b-4603-8dc3-8e50a4dad400",
"meta": {
  "templateCredsSetupCompleted": true,
  "instanceId":
"effd31c9690fbddcff4e00ff30faced1db343fc4567425ce80df54dad578ccba"
},
"id": "J51HsRvqzLHq5bC5",
"tags": []
}
```

Figura del flujo de Sistema de Inventario y Procesamiento de Registros



Conclusión

El desarrollo del Sistema de Inventario y Procesamiento de Registros representa una solución avanzada a la gestión operativa de emprendimientos de manufactura personalizada. El mayor logro de este proyecto no es solo la digitalización de la información, sino la eliminación de la fricción entre el lenguaje humano y las bases de datos relacionales. Gracias a n8n se logró la Integración de Herramientas. La potencia de este sistema reside en la orquestación de cuatro pilares tecnológicos distintos:

Interfaz de Usuario (Telegram): Utilizada como una terminal móvil de alta disponibilidad, permitiendo el registro de datos en tiempo real desde cualquier ubicación.

Cerebro Lógico IA Agent con Gemini: Actúa como el intérprete de lenguaje natural, transformando peticiones ambiguas en parámetros precisos y consultas estructuradas.

Orquestador de Flujos (n8n): El núcleo del sistema que gestiona la lógica de negocio, las condiciones de salida y la comunicación bidireccional entre la IA y los servicios externos.

Persistencia de Datos (PostgreSQL): Una base de datos robusta que garantiza la integridad de la información, permitiendo consultas complejas de inventario y el seguimiento histórico de cada pedido.

Gracias a esto es una infraestructura escalable. La arquitectura modular implementada permite que, en el futuro, se puedan integrar pasarelas de pago automáticas, generación de guías

de envío o dashboards de análisis de ventas. En conclusión, hemos transformado un chat convencional en un ERP inteligente y autónomo, demostrando que la inteligencia artificial aplicada es la herramienta definitiva para la competitividad de las pequeñas y medianas empresas.

El desarrollo del Sistema de Inventario y Procesamiento de Registros ha representado profundo de aprendizaje adaptativo y resolución de problemas complejos. Desarrollar la arquitectura integral. Superar desafíos como la gestión de estados de la IA y la orquestación de flujos de trabajo en n8n ha fortalecido mi capacidad para enfrentar retos tecnológicos desde un enfoque analítico y autodidacta. Este proyecto marca un hito en mi formación, consolidando la habilidad de integrar tecnologías emergentes para resolver necesidades tangibles del entorno real.

Referencias bibliográficas

n8n.io. (2026). n8n Documentation: AI Agent Node and Tooling.

<https://docs.n8n.io/>

The PostgreSQL Global Development Group. (2026). *PostgreSQL 16* Documentation.

<https://www.postgresql.org/docs/>

Google for Developers. (2026). Gemini API Documentation: Build with Google's most capable

<https://ai.google.dev/gemini-api/docs?hl=es-419>

Imran, M., & Almusharraf, N. (2024). Google Gemini as a next-generation AI educational tool: A review of emerging educational technology. *Smart Learning Environments*, 11, Article 22. <https://doi.org/10.1186/s40561-024-00310-z>