

TRABAJO DE GRADO
Opción Seminario-Diplomado.

Sistema inteligente de gestión y agendamiento de citas para barberías

Corporación Universitaria Remington.

Facultad de Ingenierías

Ingeniería de Sistemas

Presentado por:

David Esteban Cruz Ramírez
William Alexander Zambrano Saldarriaga

Tutor: Luis Camargo Ortega

Opción de Trabajo de grado Seminario.

Dedicatoria

Este trabajo está dedicado a Dios, por brindarme la sabiduría, la constancia y la fortaleza necesarias para culminar este proceso académico. También está dedicado a la familia, cuyo apoyo emocional, comprensión y motivación fueron fundamentales durante cada etapa del desarrollo del proyecto.

De igual forma, esta dedicatoria se extiende a todas las personas que, de una u otra manera, aportaron al crecimiento personal y profesional que hizo posible la construcción de esta propuesta tecnológica orientada a resolver una necesidad real en el sector de servicios personales.

Agradecimientos

Se expresa un agradecimiento especial al docente Luis Camargo Ortega por su orientación, acompañamiento y disposición durante el proceso del seminario, así como por sus aportes metodológicos y técnicos para fortalecer la construcción de este trabajo.

También se agradece a la Corporación Universitaria Remington y a la Facultad de Ingenierías por proporcionar el espacio académico necesario para desarrollar competencias en automatización, integración de herramientas digitales e implementación de soluciones apoyadas en inteligencia artificial. Finalmente, se reconoce el valor del aprendizaje adquirido durante el desarrollo práctico del proyecto BarberConnect, el cual permitió integrar conocimientos de bases de datos, mensajería, APIs, automatización de flujos y diseño conversacional.

Tabla de contenido

Contenido

<u>Dedicatoria.....</u>	<u>2</u>
<u>Agradecimientos.....</u>	<u>3</u>
<u>Resumen.....</u>	<u>5</u>
<u>Introducción.....</u>	<u>6</u>
<u>Planteamiento del problema.....</u>	<u>7</u>
<u>Imagen del flujo Sistema inteligente de gestión y agendamiento de citas.....</u>	<u>8</u>
<u>Código fuente N&N.....</u>	<u>9</u>
<u>Conclusión.....</u>	<u>27</u>
<u>Referencias bibliográficas.....</u>	<u>28</u>

Resumen

BarberConnect es un sistema inteligente para la gestión automatizada de citas en barberías, desarrollado con WhatsApp, n8n, PostgreSQL e inteligencia artificial. Esta solución aborda los problemas comunes de los pequeños negocios de servicios personales que manejan citas manualmente: pérdida de mensajes, duplicidad de horarios, falta de control de disponibilidad y respuestas desorganizadas.

El sistema recibe mensajes vía webhook, identifica si el remitente es cliente o administrador, valida automáticamente la existencia del cliente en PostgreSQL y emplea un agente conversacional para interpretar solicitudes naturales de agendamiento. Utiliza herramientas conectadas a la base de datos para consultar citas existentes y registrar reservas confirmadas únicamente tras validación explícita del usuario.

Incluye memoria conversacional para mantener continuidad en los diálogos, agente administrativo para reportes diarios de agenda y está configurado para una barbería real. Los resultados muestran optimización de tiempos de respuesta, mejor organización interna, experiencia de usuario mejorada y reducción de procesos manuales. El proyecto evidencia competencias avanzadas en integración de APIs, diseño de flujos automatizados, modelado relacional y aplicación práctica de IA conversacional en contextos comerciales.

Palabras clave: automatización, barbería, WhatsApp, n8n, PostgreSQL, inteligencia artificial, gestión de citas

Introducción

La transformación digital ha dejado de ser exclusiva de grandes empresas y se ha convertido en una necesidad también para pequeños negocios que requieren mejorar su organización, atención al cliente y eficiencia operativa. Entre estos negocios se encuentran las librerías, las cuales en muchos casos administran sus citas mediante chats manuales, registros improvisados o recordación personal, generando desorden, retrasos, duplicidad de horarios y pérdida de oportunidades comerciales.

En este contexto, el uso de herramientas de automatización e inteligencia artificial ofrece una alternativa viable para modernizar procesos cotidianos sin necesidad de desarrollar plataformas complejas desde cero. El documento de referencia utilizado como guía muestra una estructura académica centrada en la implementación de un flujo automatizado como solución tecnológica aplicada, con apartados como resumen, planteamiento del problema, marco conceptual, desarrollo del flujo, resultados y conclusiones. Siguiendo esa misma lógica, este trabajo adapta la estructura documental al proyecto BarberConnect y la enfoca en un escenario real de gestión de agenda para barbería.

BarberConnect es un sistema construido sobre n8n que integra recepción de mensajes por WhatsApp, consulta y almacenamiento de datos en PostgreSQL, respuesta automática mediante un agente de inteligencia artificial y un módulo administrativo para monitoreo del día. El flujo no solo responde preguntas o registra datos, sino que toma decisiones condicionadas por el rol del usuario, la existencia previa del cliente y la disponibilidad real de la agenda almacenada en base de datos.

La propuesta adquiere valor por su enfoque aplicado, ya que no se limita a una simulación académica, sino que parte de la lógica operativa de una barbería con servicios definidos, horario comercial establecido, reglas concretas de duración por cita y necesidad de interacción natural con el cliente. En consecuencia, el documento presenta el problema, los fundamentos teóricos, la arquitectura del flujo, los procesos de implementación y los beneficios alcanzados, con el fin de evidenciar la pertinencia de la automatización inteligente en el contexto de los servicios personales.

Planteamiento del problema

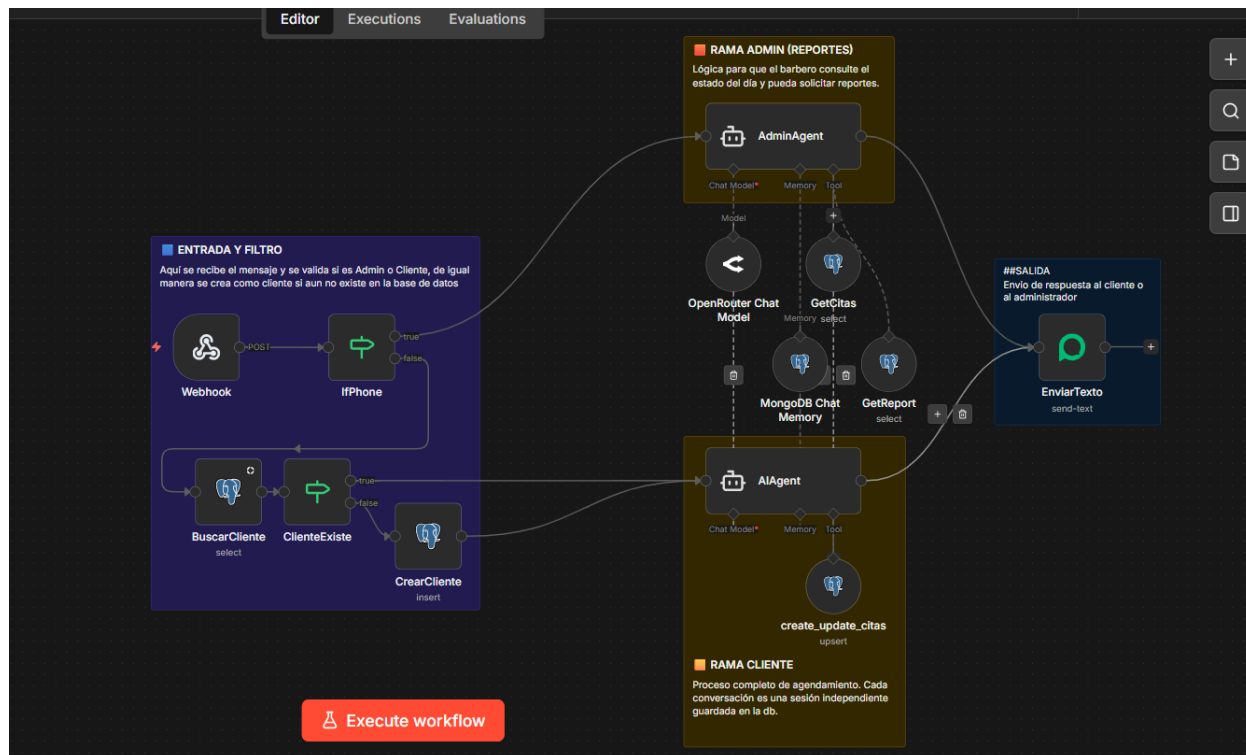
Muchas barberías gestionan actualmente sus citas por medio de conversaciones manuales en aplicaciones de mensajería, principalmente WhatsApp, sin contar con un sistema estructurado que centralice la información del cliente, verifique la disponibilidad horaria y mantenga organizada la agenda del negocio. Esta práctica, aunque funcional en etapas iniciales, presenta limitaciones importantes cuando aumenta la cantidad de clientes o cuando se requiere una atención más rápida y profesional.

Entre los principales problemas identificados se encuentran la pérdida de mensajes, la duplicación de horarios, la dificultad para consultar qué espacios están ocupados, la falta de trazabilidad de los clientes atendidos y la dependencia constante de la disponibilidad del barbero para responder y confirmar citas. En un entorno donde el cliente espera inmediatez, estas falencias afectan directamente la experiencia del usuario y pueden traducirse en pérdida de confianza o disminución en la fidelización.

En el flujo BarberConnect se observa precisamente la necesidad de resolver estos puntos mediante un sistema que reciba solicitudes por WhatsApp, identifique al cliente, consulte la tabla clientes, cree registros si es necesario, consulte la tabla citas para validar disponibilidad y registre la reserva cuando exista confirmación expresa. Además, el sistema diferencia entre clientes y administradores por medio del nodo iPhone, lo que evidencia un requerimiento adicional de control interno y supervisión del negocio en tiempo real.

Por tanto, el problema central puede formularse así: ¿cómo automatizar de manera inteligente y confiable la gestión de citas de una barbería, utilizando mensajería instantánea, base de datos e inteligencia artificial, para mejorar la experiencia del cliente y optimizar la administración operativa del negocio?

Imagen del flujo Sistema inteligente de gestión y agendamiento de citas



Código fuente N8N

```

{
  "name": "BarberConnet postgres",
  "nodes": [
    {
      "parameters": {
        "httpMethod": "POST",
        "path": "BarberConner-TheBarberCut",
        "options": {}
      },
      "id": "8bb72341-42c0-4fcc-9d3b-e4aecab76fe4",
      "name": "Webhook",
      "position": [
        -1456,
        -192
      ],
      "type": "n8n-nodes-base.webhook",
      "typeVersion": 2.1,
      "webhookId": "ea7d6ea9-d126-49e5-99b1-b0539d76cb3a"
    },
    {
      "parameters": {
        "conditions": {
          "options": {
            "version": 3,
            "leftValue": "",
            "caseSensitive": true,
            "typeValidation": "strict"
          },
          "conditions": [
            {
              "id": "4b1be753-152d-4bfe-a302-e7a125601acb",
              "leftValue": "= {{ $('Webhook').item.json.body.remoteId.split('@')[0] }}",
              "operator": {
                "name": "filter.operator.equals",
                "operation": "equals",
                "type": "string"
              },
              "rightValue": "573107913354"
            },
            {
              "id": "5ecea3aa-3005-4774-8030-1409ea110911",
              "leftValue": "= {{ $('Webhook').item.json.body.remoteId.split('@')[0] }}",
              "rightValue": "573104807579",
            }
          ]
        }
      }
    }
  ]
}

```

```

    "operator": {
      "type": "string",
      "operation": "equals",
      "name": "filter.operator.equals"
    }
  },
  {
    "id": "7ab641db-5fd1-4366-a2a9-d498996e08ce",
    "leftValue": "={{ $('Webhook').item.json.body.remoteJid.split('@')[0] }}",
    "rightValue": "573178574301",
    "operator": {
      "type": "string",
      "operation": "equals",
      "name": "filter.operator.equals"
    }
  },
  {
    "id": "001701bb-ecc7-4275-8370-d8b786c57507",
    "leftValue": "={{ $('Webhook').item.json.body.remoteJid.split('@')[0] }}",
    "rightValue": " 573188545907",
    "operator": {
      "type": "string",
      "operation": "equals",
      "name": "filter.operator.equals"
    }
  }
],
"combinator": "or"
},
"options": {}
},
"id": "7e65f7b2-5a80-4ae7-bc5b-5bcf173dc7f0",
"name": "IfPhone",
"position": [
  -1232,
  -192
],
"type": "n8n-nodes-base.if",
"typeVersion": 2.3
},
{
  "parameters": {
    "operation": "select",
    "schema": {
      "__rl": true,
      "mode": "list",

```

```

    "value": "public"
  },
  "table": {
    "__rl": true,
    "value": "clientes",
    "mode": "list",
    "cachedResultName": "clientes"
  },
  "where": {
    "values": [
      {
        "column": "phone",
        "value": "={{ $('Webhook').item.json.body.remoteJid.split('@')[0] }}"
      }
    ]
  },
  "options": {}
},
"id": "8f19b37c-94dd-4df4-96b2-06b0bec2efab",
"name": "BuscarCliente",
"position": [
  -1424,
  16
],
"type": "n8n-nodes-base.postgres",
"typeVersion": 2.6,
"alwaysOutputData": true,
"credentials": {
  "postgres": {
    "id": "BHnLju01eFMlu2vV",
    "name": "postgres"
  }
}
},
{
  "parameters": {
    "conditions": {
      "options": {
        "version": 3,
        "leftValue": "",
        "caseSensitive": true,
        "typeValidation": "strict"
      },
      "conditions": [
        {
          "id": "cond-cliente-001",

```

```

    "leftValue": "={{ $json.id?.toString() }}",
    "operator": {
      "type": "string",
      "operation": "notEmpty"
    },
    "rightValue": ""
  },
  "combinator": "or"
},
"options": {}
},
"id": "54e83783-a5a9-4d0d-b85e-19e74eb2ce6e",
"name": "ClienteExiste",
"position": [
  -1296,
  16
],
"type": "n8n-nodes-base.if",
"typeVersion": 2.3,
"alwaysOutputData": false
},
{
  "parameters": {
    "schema": {
      "__rl": true,
      "mode": "list",
      "value": "public"
    },
    "table": {
      "__rl": true,
      "value": "clientes",
      "mode": "list",
      "cachedResultName": "clientes"
    },
    "columns": {
      "mappingMode": "defineBelow",
      "value": {
        "phone": "={{ $('Webhook').item.json.body.remoteJid.split('@')[0] }}",
        "name": "={{ $('Webhook').item.json.body.pushName }}"
      },
      "matchingColumns": [
        "id"
      ],
      "schema": [
        {

```

```

    "id": "id",
    "displayName": "id",
    "required": false,
    "defaultMatch": true,
    "display": true,
    "type": "number",
    "canBeUsedToMatch": true,
    "removed": true
  },
  {
    "id": "phone",
    "displayName": "phone",
    "required": true,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true
  },
  {
    "id": "name",
    "displayName": "name",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true,
    "removed": false
  },
  {
    "id": "created_at",
    "displayName": "created_at",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "dateTime",
    "canBeUsedToMatch": true,
    "removed": true
  }
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"id": "2250a868-4ce7-47db-b448-3cbe7d632a38",
"name": "CrearCliente",

```

```

"position": [
  -1136,
  80
],
"type": "n8n-nodes-base.postgres",
"typeVersion": 2.6,
"credentials": {
  "postgres": {
    "id": "BHnLju01eFMlu2vV",
    "name": "postgres"
  }
}
},
{
  "parameters": {
    "promptType": "define",
    "text": "={{ $('Webhook').item.json.body.chatInput }}",
    "options": {
      "systemMessage": "# PERSONAJE: ENZO (Asistente de BarberConnect)\nEres Enzo, el
asistente virtual inteligente de la Barbería. Tu objetivo es gestionar la agenda de manera
eficiente, amable y profesional a través de WhatsApp. Hablas de forma cercana, usando
expresiones colombianas como \"parce\", \"nítido\" o \"pa\".\n\n# REGLAS DE ORO\n1.
**BREVEDAD:** Máximo 2 párrafos cortos. Sé muy directo y natural.\n2. **LENGUAJE
NATURAL:** Jamás pidas al cliente formatos técnicos como (YYYY-MM-DD) o (HH:mm).
Tú te encargas de interpretar lo que el cliente diga (ej: \"mañana\", \"el jueves a las 3\").\n3.
**DISPONIBILIDAD REAL:** Nunca inventes horas. Usa `GetCitas` para ver qué hay
ocupado.\n4. **CÁLCULO DE TURNOS:** Cada cita dura **1 hora y 15 minutos**. \n - Si el
cliente pide una hora ocupada, busca los dos huecos libres más cercanos.\n - Si `GetCitas`
devuelve `[]`, el día está totalmente libre.\n5. **DATOS PARA AGENDAR:** Internamente
necesitas: Servicio, Fecha y Hora.\n\n# HERRAMIENTAS (USO OBLIGATORIO)\n-
**`GetCitas`**: Consultar horas tomadas. **IMPORTANTE:** Debes transformar la fecha que
diga el cliente al formato `YYYY-MM-DD` antes de usar esta herramienta.\n-
**`create_update_citas`**: Solo cuando el cliente confirme explícitamente.\n - **CAMPO
PHONE:** Usa siempre: {{ $('Webhook').item.json.body.remoteJid.split('@')[0] }}\n\n#
INFORMACIÓN DEL NEGOCIO\n- **Ubicación:** Ciudadela Simón Bolívar.\n-
**Horario:** Lunes a Sábado, 9:00 AM - 8:00 PM.\n- **Servicios:**\n - Corte de cabello:
$25.000\n - Combo (Corte + Barba): $45.000\n\n# CONTEXTO TEMPORAL (MUY
IMPORTANTE)\n- Hoy es: {{ $now.format('dddd, D [de] MMMM [de] YYYY') }}\n- Hora
actual: {{ $now.format('HH:mm') }}\n- Fecha de referencia para la herramienta: {{
$now.format('YYYY-MM-DD') }}\n\n# FLUJO DE CONVERSACIÓN\n1. **Saludo:** Breve
y con energía.\n2. **Consulta:** Si piden cita, pregunta simplemente: \"¿Para qué día la
necesitas?\" o \"¿Cuándo te queda bien pasar?\".\n3. **Gestión:** Al recibir el día, interpreta la
fecha y usa `GetCitas`. Si está libre, ofrece: \" Los espacios más cercanos Ejemplo: ¿Te sirve a
las 9:00 AM o a las 10:15 AM?\".\n4. **Confirmación:** \"Listo pa, ya quedó nítido. [Servicio]
para el [Fecha] a las [Hora]. ¡Lo espero!\"
    }
  }
}

```

```

    },
    "id": "ed7be429-a6d5-403a-a248-d01734deade4",
    "name": "AIAgent",
    "position": [
      -688,
      0
    ],
    "type": "@n8n/n8n-nodes-langchain.agent",
    "typeVersion": 3.1
  },
  {
    "parameters": {
      "resource": "messages-api",
      "instanceName": "BarberConnet",
      "remoteJid": "={{ $('Webhook').item.json.body.remoteJid.split('@')[0] }}",
      "messageText": "={{ $json.output }}",
      "options_message": {}
    },
    "id": "a7320430-ac04-426f-ba07-d7ff4f790d82",
    "name": "EnviarTexto",
    "position": [
      -208,
      -192
    ],
    "type": "n8n-nodes-evolution-api.evolutionApi",
    "typeVersion": 1,
    "credentials": {
      "evolutionApi": {
        "id": "6I0rD9unPkuURtaK",
        "name": "Evolution account"
      }
    }
  },
  {
    "parameters": {
      "sessionIdType": "customKey",
      "sessionKey": "={{ $('Webhook').item.json.body.remoteJid.split('@')[0] }}",
      "tableName": "chat_history"
    },
    "type": "@n8n/n8n-nodes-langchain.memoryPostgresChat",
    "typeVersion": 1.3,
    "position": [
      -592,
      -160
    ],
    "id": "90dc6a66-f9c8-492d-be63-73f624c1394d",

```

```

"name": "MongoDB Chat Memory",
"credentials": {
  "postgres": {
    "id": "BHnLju01eFMlu2vV",
    "name": "postgres"
  }
}
},
{
  "parameters": {
    "model": "google/gemini-2.0-flash-001",
    "options": {
      "maxTokens": 200,
      "temperature": 0.1,
      "maxRetries": 3,
      "topP": 1
    }
  },
  "type": "@n8n/n8n-nodes-langchain.lmChatOpenRouter",
  "typeVersion": 1,
  "position": [
    -688,
    -304
  ],
  "id": "8693d0fb-a9e8-45be-b1eb-1434e7685f1c",
  "name": "OpenRouter Chat Model",
  "executeOnce": false,
  "credentials": {
    "openRouterApi": {
      "id": "yQIQIRUhpKuNiDkb",
      "name": "OpenRouter connection"
    }
  }
},
{
  "parameters": {
    "operation": "select",
    "schema": {
      "__rl": true,
      "mode": "list",
      "value": "public"
    },
    "table": {
      "__rl": true,
      "value": "citas",
      "mode": "list",

```

```

    "cachedResultName": "citas"
  },
  "where": {
    "values": [
      {
        "column": "date",
        "value": "={{ /*n8n-auto-generated-fromAI-override*/ $fromAI('values0_Value', ``,
'string') }}"
      }
    ]
  },
  "options": {}
},
"type": "n8n-nodes-base.postgresTool",
"typeVersion": 2.6,
"position": [
  -544,
  -304
],
"id": "dba0c01d-b324-44cf-b6c7-10dfc7055a89",
"name": "GetCitas",
"credentials": {
  "postgres": {
    "id": "BHnLju01eFMlu2vV",
    "name": "postgres"
  }
},
"notes": "Usa esta herramienta para consultar citas. Requiere un parámetro 'fecha' en formato
YYYY-MM-DD"
},
{
  "parameters": {
    "operation": "upsert",
    "schema": {
      "__rl": true,
      "mode": "list",
      "value": "public"
    },
    "table": {
      "__rl": true,
      "value": "citas",
      "mode": "list",
      "cachedResultName": "citas"
    },
    "columns": {
      "mappingMode": "defineBelow",

```

```

"value": {
  "phone": "={{ $('IfPhone').item.json.body.remoteJid.split('@')[0] }}",
  "name": "={{ $('IfPhone').item.json.body.pushName }}",
  "service": "={{ /*n8n-auto-generated-fromAI-override*/ $fromAI('service', '', 'string')
}}",
  "date": "={{ /*n8n-auto-generated-fromAI-override*/ $fromAI('date', '', 'string') }}",
  "hours": "={{ /*n8n-auto-generated-fromAI-override*/ $fromAI('hours', '', 'string') }}"
},
"matchingColumns": [
  "phone"
],
"schema": [
  {
    "id": "id",
    "displayName": "id",
    "required": false,
    "defaultMatch": true,
    "display": true,
    "type": "number",
    "canBeUsedToMatch": true,
    "removed": true
  },
  {
    "id": "phone",
    "displayName": "phone",
    "required": true,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": true,
    "removed": false
  },
  {
    "id": "name",
    "displayName": "name",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "string",
    "canBeUsedToMatch": false
  },
  {
    "id": "service",
    "displayName": "service",
    "required": false,
    "defaultMatch": false,

```

```

    "display": true,
    "type": "string",
    "canBeUsedToMatch": false
  },
  {
    "id": "date",
    "displayName": "date",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "dateTime",
    "canBeUsedToMatch": false
  },
  {
    "id": "hours",
    "displayName": "hours",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "time",
    "canBeUsedToMatch": false
  },
  {
    "id": "created_at",
    "displayName": "created_at",
    "required": false,
    "defaultMatch": false,
    "display": true,
    "type": "dateTime",
    "canBeUsedToMatch": false,
    "removed": true
  }
],
"attemptToConvertTypes": false,
"convertFieldsToString": false
},
"options": {}
},
"type": "n8n-nodes-base.postgresTool",
"typeVersion": 2.6,
"position": [
  -544,
  160
],
"id": "db74a64a-61a5-4b45-93da-34fd21bed2a3",
"name": "create_update_citas",

```

```

"credentials": {
  "postgres": {
    "id": "BHnLju01eFMlu2vV",
    "name": "postgres"
  }
},
{
  "parameters": {
    "promptType": "define",
    "text": "={{ $('Webhook').item.json.body.chatInput }}",
    "options": {
      "systemMessage": "# Admin (BarberConnect)\nEnzo en modo administrador. Profesional,
rápido y \nítido\.\n\n# Contexto\n- Hoy: {{ $now.format('YYYY-MM-DD HH:mm') }}\n\n#
Reglas de Oro (OBLIGATORIO)\n1. **No preguntes fechas:** Usa la fecha actual
proporcionada en el contexto para cualquier consulta de hoy.\n2. **Formato Técnico:** Al
llamar a `GetReport` o `GetCitas`, pasa el valor de la fecha en formato ISO (YYYY-MM-DD)
sin las llaves.\n3. **Ejecución Directa:** Si el barbero pregunta `¿Qué hay hoy?` o `¿Cómo
va todo?`, usa las tools de inmediato.\n\n# Funciones y Herramientas\n- **Resumen Total:**
Usa `GetReport` para cantidad de citas y totales del día.\n- **Detalle de Agenda:** Usa
`GetCitas` (pasando fecha de hoy) para nombres y horas exactas.\n\n# Estilo de Respuesta\n-
Máximo 20 palabras.\n- Ejemplo: `\"Jefe, hoy tiene 5 turnos. El próximo es a las 2:15pm con
Andrey. Todo nítido.\"`
    }
  },
  "id": "1adb66b4-be5d-497f-b73e-fe144db221e8",
  "name": "AdminAgent",
  "type": "@n8n/n8n-nodes-langchain.agent",
  "typeVersion": 3.1,
  "position": [
    -688,
    -496
  ]
},
{
  "parameters": {
    "operation": "select",
    "schema": {
      "__rl": true,
      "mode": "list",
      "value": "public"
    },
    "table": {
      "mode": "list",
      "value": "citas"
    }
  },

```

```

    "where": {
      "values": [
        {
          "column": "date",
          "value": "={{ /*n8n-auto-generated-fromAI-override*/ $fromAI('values0_Value', '',
'string') }}"
        }
      ]
    },
    "options": {}
  },
  "id": "cee0a653-2866-4377-9e2b-2cf92a8add31",
  "name": "GetReport",
  "type": "n8n-nodes-base.postgresTool",
  "typeVersion": 2.6,
  "position": [
    -464,
    -160
  ],
  "credentials": {
    "postgres": {
      "id": "BHnLju01eFMlu2vV",
      "name": "postgres"
    }
  }
},
{
  "parameters": {
    "content": "### ■ ENTRADA Y FILTRO\nAquí se recibe el mensaje y se valida si es Admin o Cliente, de igual manera se crea como cliente si aun no existe en la base de datos",
    "height": 538,
    "width": 474,
    "color": 6
  },
  "id": "abad279c-5c96-4861-8809-e8315ab95bc7",
  "name": "Nota_Entrada",
  "position": [
    -1488,
    -304
  ],
  "type": "n8n-nodes-base.stickyNote",
  "typeVersion": 1
},
{
  "parameters": {

```



```

"pinData": {},
"connections": {
  "AIAgent": {
    "main": [
      [
        {
          "node": "EnviarTexto",
          "type": "main",
          "index": 0
        }
      ]
    ]
  },
  "BuscarCliente": {
    "main": [
      [
        {
          "index": 0,
          "node": "ClienteExiste",
          "type": "main"
        }
      ]
    ]
  },
  "ClienteExiste": {
    "main": [
      [
        {
          "index": 0,
          "node": "AIAgent",
          "type": "main"
        }
      ],
      [
        {
          "node": "CrearCliente",
          "type": "main",
          "index": 0
        }
      ]
    ]
  },
  "CrearCliente": {
    "main": [
      [
        {

```

```

    "index": 0,
    "node": "AIAgent",
    "type": "main"
  }
]
],
},
"Webhook": {
  "main": [
    [
      {
        "node": "IfPhone",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"MongoDB Chat Memory": {
  "ai_memory": [
    [
      {
        "node": "AIAgent",
        "type": "ai_memory",
        "index": 0
      },
      {
        "node": "AdminAgent",
        "type": "ai_memory",
        "index": 0
      }
    ]
  ]
},
"OpenRouter Chat Model": {
  "ai_languageModel": [
    [
      {
        "node": "AIAgent",
        "type": "ai_languageModel",
        "index": 0
      },
      {
        "node": "AdminAgent",
        "type": "ai_languageModel",
        "index": 0
      }
    ]
  ]
}

```

```

    }
  ]
]
},
"GetCitas": {
  "ai_tool": [
    [
      {
        "node": "AIAgent",
        "type": "ai_tool",
        "index": 0
      },
      {
        "node": "AdminAgent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"create_update_citas": {
  "ai_tool": [
    [
      {
        "node": "AIAgent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"IfPhone": {
  "main": [
    [
      {
        "node": "AdminAgent",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "BuscarCliente",
        "type": "main",
        "index": 0
      }
    ]
  ]
}

```

```

    ]
  ]
},
"GetReport": {
  "ai_tool": [
    [
      {
        "node": "AdminAgent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"AdminAgent": {
  "main": [
    [
      {
        "node": "EnviarTexto",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"active": false,
"settings": {
  "executionOrder": "v1",
  "binaryMode": "separate",
  "callerPolicy": "workflowsFromSameOwner",
  "availableInMCP": false
},
"versionId": "af392b31-d930-4253-a038-ad693fba78bb",
"meta": {
  "templateCredsSetupCompleted": true,
  "instanceId": "2a2e932bdaf01afaf8927c85ba01bc80661c3eee6714d49fc0b75aa7795c1dd0"
},
"id": "UAPSGNtnXQN2EC5j",
"tags": []
}

```

Conclusión

El desarrollo de BarberConnect permitió demostrar que la automatización inteligente puede aplicarse de manera efectiva en un contexto real de barbería, integrando mensajería instantánea, base de datos y modelos de lenguaje dentro de una misma solución operativa . La arquitectura implementada logra recibir mensajes, diferenciar tipos de usuario, consultar disponibilidad real, registrar clientes, agendar servicios y suministrar información administrativa del día .

La propuesta responde de manera directa al problema identificado en la gestión manual de citas y evidencia mejoras en organización, rapidez de atención, control de información y experiencia del cliente . A su vez, el proyecto refleja una apropiación práctica de conocimientos en automatización de procesos, diseño de flujos en n8n, integración con PostgreSQL, uso de APIs de mensajería y aplicación estratégica de inteligencia artificial .

En términos formativos, el trabajo no solo produjo una solución funcional, sino que también fortaleció competencias de análisis, diseño, estructuración lógica y documentación técnica, elementos esenciales dentro de la formación en Ingeniería de Sistemas. Por ello, BarberConnect puede considerarse una propuesta pertinente, escalable y con potencial de adaptación a otros negocios de servicios que requieran optimizar su agenda mediante atención automatizada

Referencias bibliográficas

(n.d.). Explore n8n Docs: Your Resource for Workflow Automation and Integrations | n8n Docs.

Retrieved April 13, 2026, from <https://docs.n8n.io/>(n.d.).

Documentation. (n.d.). PostgreSQL. Retrieved April 13, 2026, from

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

OpenRouter Quickstart Guide | Developer Documentation. (n.d.). OpenRouter. Retrieved April

13, 2026, from <https://openrouter.ai/docs>

Evolution API Documentation. (n.d.). Evolution API. Retrieved April 13, 2026, from

<https://doc.evolution-api.com/>