



**Integrantes:**

Dennis Adolfo Jimenez Perez

Carlos Antonio Salazar Trinidad

Alan Oswaldo Toledo del Toro

Mauricio Abelardo Cordero Perez

Fecha: 30/abril/2025

# Especificaciones de Requerimientos del software

## Presentación de la Información. (Antecedentes)

En los últimos años, el uso de herramientas digitales para mejorar la atención al cliente y los procesos de venta ha tomado gran relevancia, especialmente en plataformas de mensajería instantánea como WhatsApp. Esta plataforma se ha consolidado como uno de los canales de comunicación más utilizados por empresas para interactuar con sus clientes debido a su accesibilidad, inmediatez y alta tasa de respuesta.

Ante este contexto, surge la necesidad de desarrollar soluciones tecnológicas que permitan automatizar procesos de atención y ventas, sin sacrificar la personalización y calidad de la experiencia del usuario. La inteligencia artificial generativa (IA), particularmente los modelos de procesamiento de lenguaje natural, se presenta como una tecnología clave para atender esta demanda, permitiendo la creación de asistentes virtuales capaces de interactuar de forma coherente, personalizada y en tiempo real con los usuarios.

- **Requisitos no funcionales (Seguridad, acceso, Almacenaje, configuración, actuación, interoperabilidad, recuperación, accesibilidad.)**

| <b>REQUISITOS NO FUNCIONALES DEL SISTEMA</b> |                             |   |   |
|--|-----------------------------|---|---|
| <b>ID</b>                                    | <b>NOMBRE DEL REQUISITO</b> | <b>DESCRIPCIÓN</b>  | <b>OBSERVACIÓN</b>  |
| RNF01  | Seguridad                   | Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.                                      | Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un lugar físico externo (edificio) al que reside el sistema. |
| RNF02  | Acceso                      | Solo usuarios autenticados mediante credenciales válidas podrán acceder al panel de administración y modificar configuraciones del asistente. | Se recomienda el uso de autenticación de dos factores (2FA) para el administrador del sistema.  |
| RNF03  | Almacenaje                  | El sistema debe permitir el almacenamiento estructurado de conversaciones, leads y métricas de uso para su análisis posterior.                | Los datos deben almacenarse en bases de datos seguras como PostgreSQL o Firebase, cumpliendo con normativas de protección de datos  |

| ID    | NOMBRE DEL REQUISITO | DESCRIPCIÓN   | OBSERVACIÓN  |
|-------|----------------------|---|--|
| RNF04 | Configuración        | El asistente debe permitir configurar respuestas, flujos de conversación y horarios de atención desde un panel intuitivo.             | Las configuraciones deberán guardarse automáticamente y aplicarse sin necesidad de reiniciar el sistema.     |
| RNF05 | Actuación            | El asistente debe ofrecer respuestas en menos de 5 segundos luego de recibir un mensaje del usuario.                                  | El rendimiento será monitoreado constantemente para garantizar tiempos de respuesta adecuados.               |
| RNF06 | Interoperabilidad    | El sistema debe integrarse de manera fluida con la API de WhatsApp Business y otras plataformas como CRM o sistemas de análisis.      | Se debe asegurar compatibilidad con versiones actualizadas de la API y documentar los puntos de integración. |
| RNF07 | Recuperación         | En caso de falla, el sistema deberá recuperar su funcionamiento automáticamente en menos de 1 minuto, sin pérdida de datos recientes. | Se debe implementar un sistema de recuperación automática y monitoreo constante.                             |

- Casos de Uso:
  1. Actores
    - a) Descripción de actores primarios

| NOMBRE DEL ACTOR   | TIPO DE ACTOR   |
|--|---|
| cliente / Usuario Final  | Primario  |
| Actividades  | Descripción   |
| Chat<br>Consulta de productos o servicios<br>Realización de pedidos<br>Recepción de respuestas | Interactúa directamente con el asistente por WhatsApp. Consulta información sobre productos o servicios, realiza pedidos, y recibe respuestas automáticas o personalizadas en tiempo real. También puede ser redirigido a un agente humano si es necesario. |

| NOMBRE DEL ACTOR  | TIPO DE ACTOR   |
|---|---|
| Administrador del Sistema   | Primario  |
| Actividades   | Descripción   |
| Login al panel de administración<br>Configuración del asistente<br>Entrenamiento del modelo<br>Visualización de reportes y estadísticas | Encargado de la gestión y configuración del asistente. Ajusta las respuestas, entrena el modelo con nueva información, administra la base de conocimientos, y analiza el desempeño del sistema con base en métricas y reportes generados. |

| NOMBRE DEL ACTOR   | TIPO DE ACTOR   |
|--|---|
| Agente Humano  | Primario  |
| Actividades  | Descripción   |
| Toma control de conversaciones<br>Atención personalizada | Interviene en la conversación cuando el asistente detecta que no puede resolver una solicitud del cliente. Ofrece atención personalizada a través del mismo canal (WhatsApp). |

a) Descripción de actores secundarios

| NOMBRE DEL ACTOR   | TIPO DE ACTOR  |
|--|--|
| Jefe de Proyectos  | Secundario   |
| Actividades  | Descripción  |
| Supervisión del desarrollo<br>Revisión de avances<br>Gestión de recursos | Supervisa el avance general del proyecto, revisa entregables de cada sprint, valida que se cumplan los requisitos funcionales y no funcionales, y gestiona los recursos necesarios para el desarrollo eficiente del asistente virtual. |

# Ingeniería de Requisitos

- **Objetivo del Proyecto**

Diseñar y desarrollar un asistente virtual con inteligencia artificial generativa que automatice la atención al cliente y el proceso de ventas a través de la plataforma WhatsApp, brindando respuestas personalizadas y en tiempo real. Este sistema busca mejorar la experiencia del usuario, reducir los tiempos de respuesta, y aumentar la conversión de ventas.

- **Justificación del Proyecto**

En un entorno empresarial cada vez más digitalizado, ofrecer atención al cliente inmediata y personalizada es clave para diferenciarse. WhatsApp se ha convertido en una plataforma preferida por los usuarios para comunicarse con empresas. Un asistente basado en IA generativa permitirá automatizar estas interacciones de forma inteligente, mejorando la eficiencia operativa, reduciendo costos en atención al cliente, y aumentando la satisfacción y fidelización de los clientes. Este proyecto responde a la necesidad de integrar soluciones tecnológicas modernas en la comunicación empresarial.

- **Calendario de Actividades (Diagrama de Gantt, avance en días, debe tener avance programado y real)**



- **Requisitos del Software** (define cómo va a interactuar el software, que quiere crearse con el hardware, las interfaces externas, la velocidad operativa, el tiempo de respuesta del sistema, la portabilidad del software en las diversas plataformas, el mantenimiento, la velocidad de reponerse después de un fallo, su seguridad, calidad, limitaciones, etc.)

| <b>Categoría</b>         | <b>Descripción</b>  |
|--------------------------|---|
| Interacción con Hardware | El sistema se ejecutará en servidores con capacidad de procesamiento medio-alto (nube), y se conectará con dispositivos móviles vía WhatsApp.               |
| Interfaces Externas      | WhatsApp Business API (Twilio o Meta API), OpenAI GPT-4 o Llama 2, bases de datos PostgreSQL o Firebase, plataforma analítica (Google Analytics, Power BI). |
| Velocidad Operativa      | El asistente debe procesar y responder en un rango de 1 a 3 segundos en condiciones óptimas.  |
| Tiempo de Respuesta      | Las respuestas generadas por la IA deben mantenerse por debajo de los 5 segundos, incluso con carga concurrente.  |
| Portabilidad             | El sistema debe ser portable y adaptable a diferentes plataformas en la nube (AWS, Google Cloud, Azure).  |
| <b>Categoría</b>         | <b>Descripción</b>  |
| Recuperación ante Fallos | El sistema deberá contar con copias de seguridad automáticas diarias y recuperación en menos de 2 horas.  |
| Seguridad                | Acceso restringido por roles, cifrado de datos sensibles, y autenticación en el panel de administración.  |



| Categoría            | Descripción   |
|----------------------|---|
| Calidad del Software | El sistema debe asegurar coherencia en respuestas, disponibilidad del 99.9%, y facilidad de uso para el cliente y el administrador.                     |
| Limitaciones         | La efectividad del modelo dependerá de la calidad y cantidad de datos de entrenamiento. Puede requerir ajustes constantes durante la operación inicial. |

- **Instrumento(s) de Recopilación de Información:**

- a) **Análisis bibliográfico**

Se realizó una revisión de literatura relacionada con:

- Asistentes virtuales y chatbots en el comercio electrónico.
- Implementaciones de IA generativa en atención al cliente.
- Integración de WhatsApp Business API con sistemas externos.
- Buenas prácticas en la experiencia conversacional basada en IA.

Se analizaron fuentes académicas, artículos técnicos y documentación oficial de OpenAI, Meta, y Twilio. Esto permitió establecer los fundamentos técnicos y éticos para el diseño del asistente inteligente.

- b) **Entrevista (explicar a quien se entrevista, la finalidad, como se hará el seguimiento de la entrevista por ejemplo con un cuestionario o una lista de comprobación) poner los resultados.**

**Persona Entrevistada:**

– Ingeniero en Sistemas y proponente del proyecto.

**Finalidad de la Entrevista:**

Obtener información detallada sobre el flujo de interacción deseado entre usuarios y el asistente, así como los requerimientos funcionales y no funcionales del sistema.

**Metodología de Seguimiento:**

Se utilizó un **cuestionario estructurado** con preguntas cerradas y abiertas, y se complementó con una **lista de comprobación** para validar los aspectos técnicos y funcionales.

### **Preguntas del Cuestionario (Resumen):**

1. ¿Qué funcionalidades debe tener el asistente virtual?
2. ¿Qué tipo de usuarios interactuarán con el sistema?
3. ¿Cómo deben almacenarse y protegerse los datos?
4. ¿Qué tipos de respuestas debe generar la IA?
5. ¿Cómo se gestionarán los errores o preguntas fuera de contexto?

### **Resultados de la Entrevista:**

- Se identificó la necesidad de implementar módulos para recuperación de contraseña, formulario de registro, verificación de datos y visualización de bolsa de trabajo.
- Se recomendó incluir una interfaz de administración para monitorear conversaciones y estadísticas.
- Se estableció la integración inicial con WhatsApp como canal principal.
- El sistema debe generar respuestas empáticas, claras y segmentadas por tipo de consulta (ventas, soporte, información general).

- **Requisitos funcionales (Definen las funciones y la funcionalidad en y desde el sistema de software.)**

| <b>REQUISITOS FUNCIONALES DEL SISTEMA</b> |                                 |  |  |
|---|---------------------------------|--|--|
| <b>ID</b>                                 | <b>NOMBRE DEL REQUISITO</b>     | <b>DESCRIPCIÓN</b>   | <b>OBSERVACIÓN</b>   |
| RF-1                                      | Recepción de mensajes WhatsApp  | El sistema debe poder recibir mensajes entrantes a través de WhatsApp.                     | Se aceptarán únicamente mensajes de tipo texto.  |
| RF-2                                      | Procesamiento de texto          | El sistema debe procesar únicamente mensajes de tipo texto.                                | Otros tipos de mensajes recibirán una respuesta indicando que solo se procesan textos. |
| RF-3                                      | Envío de mensajes               | El sistema debe poder enviar mensajes de respuesta al usuario.                             | Las respuestas serán generadas automáticamente.  |
| RF-4                                      | Registro de clientes            | El sistema debe mantener un registro de clientes en una base de datos (Postgres).          | Se incluirán datos como número telefónico, nombre, email y dirección.                  |
| RF-5                                      | Historial de conversaciones     | El sistema debe almacenar un historial de conversación por cliente.                        | Se guardará cada mensaje entrante y saliente.  |
| RF-6                                      | Generación de respuestas con IA | El sistema utilizará un modelo de IA (OpenAI) para analizar mensajes y generar respuestas. | El modelo analizará intención, sentimiento y palabras clave.                           |

| ID    | NOMBRE DEL REQUISITO               | DESCRIPCIÓN   | OBSERVACIÓN   |
|-------|------------------------------------|---|---|
| RF-7  | Consulta a base de conocimiento    | El sistema debe consultar una base de conocimiento vectorial (Qdrant).                | Se generarán gráficas semanales de volumen de mensajes y tipos de consulta.         |
| RF-8  | Actualización de conocimiento      | El sistema podrá actualizar su base de conocimiento desde documentos de Google Drive. | e procesarán archivos, convirtiéndolos a texto, dividiéndolos y generando vectores. |
| RF-9  | Validación de tipo de mensaje      | El sistema validará que los mensajes sean de tipo texto.                              | En caso contrario, se enviará una respuesta predefinida.                            |
| RF-10 | Respuesta a mensajes no válidos    | El sistema enviará una respuesta informando que solo se aceptan mensajes de texto.    | Mensajes con imágenes, audios o archivos no serán procesados.                       |
| RF-11 | Verificación de cliente            | El sistema verificará si el número del remitente ya está registrado como cliente.     | Si no existe, iniciará el flujo de registro.  |
| RF-12 | Registro de nuevo cliente          | Si el número no está registrado, se solicitará nombre, correo y dirección.            | El cliente se marcará como pendiente de registro hasta completar los datos.         |
| RF-13 | Actualización de cliente existente | Si el cliente ya existe, se actualizarán fecha y contador de interacciones.           | También se verificará si hay datos faltantes en el registro.                        |

| ID    | NOMBRE DEL REQUISITO                  | DESCRIPCIÓN  | OBSERVACIÓN   |
|-------|---------------------------------------|--|---|
| RF-14 | Extracción de datos faltantes         | El sistema intentará obtener datos como nombre, email y dirección desde los mensajes.                      | Se usará regex para extraer y actualizar la información en la base de datos.        |
| RF-15 | Enriquecimiento continuo del perfil   | Durante el uso normal, se extraerán nuevos datos del cliente para actualizar su perfil.                    | Si se identifica información adicional, se añadirá automáticamente.                 |
| RF-16 | Análisis de intención del mensaje     | El sistema analizará cada mensaje usando IA para identificar intención, sentimiento, etc.                  | También detectará si se requiere intervención humana o si hay interés en productos. |
| RF-17 | Preparación del contexto              | Se armará un contexto para el agente IA con base en el historial del cliente y el análisis del mensaje.    | Este contexto se usará para generar respuestas personalizadas.                      |
| RF-18 | Generación de respuesta personalizada | El sistema generará respuestas usando Langchain + OpenAI con base en el contexto y memoria conversacional. | Las respuestas pueden incluir información de la base de conocimiento.               |
| RF-19 | Búsqueda semántica en Qdrant          | Se realizará una búsqueda por vectores en Qdrant para responder preguntas del usuario.                     | La búsqueda se hará mediante embeddings generados con IA.                           |
| RF-20 | Registro de interacción               | Cada mensaje y respuesta se registrarán en la base de datos.   | También se almacenará intención, sentimiento e interés si están presentes.          |

| ID    | NOMBRE DEL REQUISITO              | DESCRIPCIÓN  | OBSERVACIÓN   |
|-------|-----------------------------------|--|---|
| RF-21 | Registro de intereses del cliente | Si se detecta interés en productos, se asociará al perfil del cliente.                 | Esta funcionalidad está planeada como futura implementación.                              |
| RF-22 | Detección de escalamiento         | El sistema identificará si es necesaria una intervención humana.                       | Esta detección se basará en el análisis del mensaje por la IA.                            |
| RF-23 | Notificación a agente humano      | En caso de requerir ayuda humana, se generará una notificación.                        | Incluirá detalles del cliente y del mensaje que requiere atención.                        |
| RF-24 | Inicialización de base de datos   | El sistema debe poder crear las tablas e índices en Postgres si no existen.            | Incluye tablas como customers, conversations, etc.  |
| RF-25 | Inicialización de Qdrant          | El sistema podrá crear la colección de vectores (COMPANY_KNOWLEDGE) en Qdrant.         | Esto se hará bajo demanda.  |
| RF-26 | Refresco de base vectorial        | El sistema permitirá limpiar y recargar la base de conocimiento en Qdrant.             | Útil para cargar nueva documentación.   |
| RF-27 | Carga de documentos               | El sistema descargará y procesará documentos desde Google Drive para alimentar Qdrant. | Se dividirán en fragmentos, se generarán embeddings y se insertarán en la base vectorial. |

## **Diagramas de Caso de Uso**

### **Actores Identificados:**

- Cliente / Usuario Final
- Administrador del Sistema
- Agente Humano
- Jefe de Proyectos (Secundario)

### **Casos de Uso Principales:**

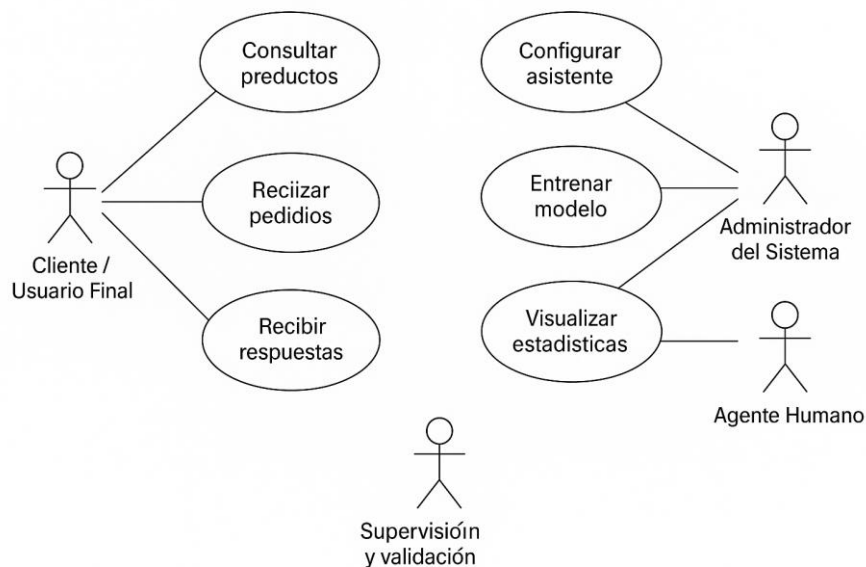
- **Cliente/Usuario Final:**
  - Consultar productos
  - Realizar pedidos
  - Recibir respuestas del asistente o de un agente
- **Administrador del Sistema:**
  - Configurar asistente
  - Entrenar modelo IA
  - Visualizar estadísticas
- **Agente Humano:**
  - Tomar control de conversaciones complejas
- **Jefe de Proyecto:**
  - Supervisión y validación del desarrollo

## Caso de Uso: Consultar productos

- **Actor Principal:** Cliente
- **Flujo Principal:**
  1. Cliente inicia conversación en WhatsApp
  2. El asistente recibe el mensaje
  3. Busca información en base de datos
  4. Envía respuesta
- **Requisitos:** RNF05 (Actuación), RF-4 (Chat con IA)

## Especificaciones del Producto

- **Tipo:** Aplicación Web + Integración con WhatsApp
- **Objetivo:** Automatizar atención al cliente y procesos de venta usando IA generativa
- **Audiencia:** Empresas, usuarios finales, personal administrativo





## 1. Caso de Uso: Consultar productos

|                         |  |
|-------------------------|--|
| <b>Nombre</b>           | Consultar productos  |
| <b>Versión</b>          | 1.0  |
| <b>Descripción</b>      | Permite al cliente consultar productos a través de un chat de WhatsApp gestionado por un asistente de IA.  |
| <b>Precondición</b>     | El cliente debe enviar un mensaje válido vía WhatsApp.   |
| <b>Flujo de eventos</b> | <ol style="list-style-type: none"><li>1. Cliente inicia conversación.</li><li>2. El asistente recibe y analiza el mensaje.</li><li>3. Se consulta la base de datos.</li><li>4. Se responde al cliente.</li></ol> |
| <b>Postcondición</b>    | El cliente recibe información sobre productos.   |
| <b>Excepciones</b>      | <ol style="list-style-type: none"><li>3A. Si no se encuentran productos, se informa al cliente.</li><li>3B. Si falla la base de datos, se muestra un mensaje de error.</li></ol>                                 |

## 2. Caso de Uso: Obtener datos

|                  |   |
|------------------|---|
| Nombre           | Obtener datos   |
| Versión          | 1.0   |
| Descripción      | Recuperar datos específicos desde la base de datos para responder a solicitudes del cliente.  |
| Precondición     | El sistema debe estar conectado y tener acceso a la base de datos.  |
| Flujo de eventos | 1. La IA detecta la intención de consulta.<br>2. Se consulta la base de datos.<br>3. Se obtiene la información y se envía al usuario. |
| Postcondición    | Se entrega al usuario la información solicitada.  |
| Excepciones      | 2A. Error en la conexión a la base de datos.<br>2B. No hay coincidencias en los datos.  |

### 3. Caso de Uso: Iniciar conversación

|                         |   |
|-------------------------|---|
| <b>Nombre</b>           | <b>Iniciar conversación</b>   |
| <b>Versión</b>          | <b>1.0</b>  |
| <b>Descripción</b>      | <b>El cliente inicia contacto vía WhatsApp para interactuar con el asistente.</b>   |
| <b>Precondición</b>     | <b>El cliente debe tener acceso a WhatsApp.</b>   |
| <b>Flujo de eventos</b> | <b>1. El cliente envía un mensaje inicial.<br/>2. El mensaje llega a través de la API de WhatsApp.<br/>3. El backend lo redirige al módulo de IA.</b> |
| <b>Postcondición</b>    | <b>Se establece una sesión de conversación.</b>   |
| <b>Excepciones</b>      | <b>2A. Error de red en el envío del mensaje.<br/>3A. API de WhatsApp no responde.</b>   |

#### 4. Caso de Uso: Generar respuesta con IA

|                         |   |
|-------------------------|---|
| <b>Nombre</b>           | <b>Generar respuesta con IA</b>   |
| <b>Versión</b>          | <b>1.0</b>  |
| <b>Descripción</b>      | <b>La IA procesa el mensaje del cliente y genera una respuesta natural.</b>   |
| <b>Precondición</b>     | <b>El backend debe haber recibido un mensaje válido.</b>  |
| <b>Flujo de eventos</b> | <b>1. Se recibe el mensaje.<br/>2. Se envía al modelo de IA (GPT-4).<br/>3. Se recibe la respuesta.<br/>4. Se envía al cliente.</b> |
| <b>Postcondición</b>    | <b>El cliente recibe una respuesta generada por IA.</b>   |
| <b>Excepciones</b>      | <b>2A. Error de tiempo de espera con la API de IA.<br/>3A. Respuesta vacía o mal formada.</b>                                       |

## 5. Caso de Uso: Consultar métricas

|                         |   |
|-------------------------|---|
| <b>Nombre</b>           | <b>Consultar métricas</b>   |
| <b>Versión</b>          | <b>1.0</b>  |
| <b>Descripción</b>      | <b>Permite al administrador visualizar datos estadísticos del sistema.</b>  |
| <b>Precondición</b>     | <b>El administrador debe iniciar sesión en el panel web.</b>  |
| <b>Flujo de eventos</b> | <b>1. El administrador accede al módulo analítico.<br/>2. Solicita métricas.<br/>3. El backend consulta y muestra los resultados.</b> |
| <b>Postcondición</b>    | <b>Se muestran gráficas o datos al administrador.</b>   |
| <b>Excepciones</b>      | <b>2A. No hay datos disponibles.<br/>3A. Error en carga del módulo analítico.</b>   |

## 6. Caso de Uso: Configurar sistema

|                  |  |
|------------------|--|
| Nombre           | Configurar sistema   |
| Versión          | 1.0  |
| Descripción      | El administrador configura parámetros del sistema desde el panel web.  |
| Precondición     | El administrador debe haber iniciado sesión.   |
| Flujo de eventos | 1. Accede al panel.<br>2. Modifica parámetros (respuestas, productos, horarios, etc.).<br>3. Guarda los cambios. |
| Postcondición    | El sistema se actualiza con la nueva configuración.  |
| Excepciones      | 2A. Error al guardar configuración.<br>3A. Validaciones fallidas en los datos ingresados.                        |

### 1. Diagrama de Componente

#### a) Componentes Principales:

- **Frontend Web (Administrador):** Panel de configuración
- **Backend API:** Gestión lógica, usuarios y flujo de conversación
- **Módulo de IA:** Generación de respuestas usando modelos como GPT-4
- **Base de Datos:** PostgreSQL / Firebase
- **WhatsApp API:** Canal de comunicación
- **Módulo Analítico:** Estadísticas y métricas

#### b) Descripción de Componentes:

**Interfaz de Usuario WhatsApp:** El punto de entrada/salida para el usuario final (cliente).

**Motor de Flujo de Trabajo n8n:** El orquestador central que ejecuta la lógica del "V4-Improved". Contiene nodos específicos para:

- Recibir/Enviar mensajes de WhatsApp.
- Controlar el flujo (Switch, If).
- Ejecutar lógica personalizada (Function, Code).
- Interactuar con bases de datos y APIs.
- Integrarse con Langchain.

**Base de Datos PostgreSQL:** Sistema externo para almacenar datos estructurados persistentes (clientes, conversaciones, productos, intereses).

**Servicio API OpenAI:** Sistema externo que proporciona modelos para la generación de embeddings, análisis de texto (intención/sentimiento) y generación de respuestas conversacionales.

**Módulo Langchain (integrado en n8n):** Framework utilizado internamente por nodos específicos de n8n para orquestar interacciones complejas con LLMs, incluyendo:

- Gestión de Agentes (Agent Executor).
- Interfaz con Modelos de Chat.
- Módulo de Memoria conversacional.
- Interfaz con Embeddings.
- Interfaz con Vector Stores (Qdrant).
- Herramientas (Búsqueda en Base de Conocimientos).
- Cargadores/Divisores de Documentos (para indexación).

**Vector Store Qdrant:** Sistema externo para almacenar y buscar vectores de embeddings (la base de conocimientos semántica). Se accede vía HTTP (inicialización) y Langchain (búsqueda).

**Servicio API Google Drive:** Sistema externo utilizado como fuente de los documentos para la base de conocimientos durante la inicialización.

**Interfaz de Usuario del Editor n8n:** Interfaz implícita para el desarrollo, monitorización y administración del flujo de trabajo por parte del equipo técnico/administrador.

## 2. Diagrama de Arquitectura del Software

[https://mermaid.live/edit#pako:eNgNVc1uEzEQfhXL51BCCN1kD0hpg1CIFgoRVEK5THcnG1OvvbW9oaXqw\\_AAnLhxzYsx9v5kmx9gD8nG-ebvm2\\_GDzzRKfKYW7wtUSU4FZAZyOeK0QOJ04Z9siUYoaujAowTiShAOXa1BGcnRcHAssnIWft7F6IGyoP815U2Nwupv7E3KhMKd7HnOhOJR7\\_TqbYsRXa-\\_kVH4M134dMTj73U1mUGZx\\_O6WAX9L5ANTnzwPrNpztDsxlJvgxAZckSRMh5kqFYuDnbhX9ljf8ibP32GQNvM\\_govVefNZHPXr9umlqJhdX6J7AcIYWvNbr5l4BUssc4g1kLYiDZfV4vtb6p8AQiaOAtZh8xEdfI7llhdIJ2y3UAEXh6ErNTrWwpHbAUXCBaskQKX22FnZ4827id4qpEuUIm1EKbHBKx\\_qW6xYF07A01rcSVZqddP924hwr3bb4mmxUqkUJlh9Kid\\_kRM2Gd0ewSVXrAr6\\_nsi743uu2BCm-16X9RxxqJVgvxtKwmfF1LkwaketdfpaqYzbQkVRClONY\\_pLDCEI90AnJjU2H3Udsa7QZo5efLRFJfSNnhndNUR5SQg5dFJdaNecdR46CT7ScnAkk5rQCpfQskqqsxbKV7SaFt6iQMjBoixKtg30BdeGIsttSoKHWrn9b2i8pMFTsBGzoN-IPJyL37HZ7dCjblvMrzFNhcq2jPam2pJ7yKobgppfP0OvejZpr8JIP-mjfyqDQyGJHz9ffosZILgiKHaik5r\\_3qk9EmkpZ1noQQR7Z2HSTMBzOs0QSlaO4EIH339pVwDTKAY0W5Y5KNhirRPrAgwt56DJJ4b\\_rrR2EpZcl-tNhQuhunxXW25ncvelsLM\\_65W72aFbcMgM93iONPoipWvwwR\\_PuVtijnMe02sK5mbO5-qRcFA6PbtXCY-dKbHHjS6zJY8XQKuix8uCtk1zgTYQuiG-aN3-zlwPUxsYio7mVJfK8fjFKIB5\\_MDveDx4GR31R-NhFL16Oe4fD171-D2Ph\\_2jaDQcDqLj4WA0GAYOjx97\\_Htw3z8aRcMxPVE\\_or-i8bjHMRUk2ovqgg\\_3\\_OMfye-Ufg](https://mermaid.live/edit#pako:eNgNVc1uEzEQfhXL51BCCN1kD0hpg1CIFgoRVEK5THcnG1OvvbW9oaXqw_AAnLhxzYsx9v5kmx9gD8nG-ebvm2_GDzzRKfKYW7wtUSU4FZAZyOeK0QOJ04Z9siUYoaujAowTiShAOXa1BGcnRcHAssnIWft7F6IGyoP815U2Nwupv7E3KhMKd7HnOhOJR7_TqbYsRXa-_kVH4M134dMTj73U1mUGZx_O6WAX9L5ANTnzwPrNpztDsxlJvgxAZckSRMh5kqFYuDnbhX9ljf8ibP32GQNvM_govVefNZHPXr9umlqJhdX6J7AcIYWvNbr5l4BUssc4g1kLYiDZfV4vtb6p8AQiaOAtZh8xEdfI7llhdIJ2y3UAEXh6ErNTrWwpHbAUXCBaskQKX22FnZ4827id4qpEuUIm1EKbHBKx_qW6xYF07A01rcSVZqddP924hwr3bb4mmxUqkUJlh9Kid_kRM2Gd0ewSVXrAr6_nsi743uu2BCm-16X9RxxqJVgvxtKwmfF1LkwaketdfpaqYzbQkVRClONY_pLDCEI90AnJjU2H3Udsa7QZo5efLRFJfSNnhndNUR5SQg5dFJdaNecdR46CT7ScnAkk5rQCpfQskqqsxbKV7SaFt6iQMjBoixKtg30BdeGIsttSoKHWrn9b2i8pMFTsBGzoN-IPJyL37HZ7dCjblvMrzFNhcq2jPam2pJ7yKobgppfP0OvejZpr8JIP-mjfyqDQyGJHz9ffosZILgiKHaik5r_3qk9EmkpZ1noQQR7Z2HSTMBzOs0QSlaO4EIH339pVwDTKAY0W5Y5KNhirRPrAgwt56DJJ4b_rrR2EpZcl-tNhQuhunxXW25ncvelsLM_65W72aFbcMgM93iONPoipWvwwR_PuVtijnMe02sK5mbO5-qRcFA6PbtXCY-dKbHHjS6zJY8XQKuix8uCtk1zgTYQuiG-aN3-zlwPUxsYio7mVJfK8fjFKIB5_MDveDx4GR31R-NhFL16Oe4fD171-D2Ph_2jaDQcDqLj4WA0GAYOjx97_Htw3z8aRcMxPVE_or-i8bjHMRUk2ovqgg_3_OMfye-Ufg)

## 3. Plataforma

El sistema es una **aplicación de backend basada en flujos de trabajo**, ejecutada sobre la plataforma de automatización **n8n** (auto-hospedada o versión Cloud).

Se expone al usuario final a través de la **API de WhatsApp Business**.

## 4. Requisitos de Rendimiento

**Tiempo de Respuesta:** El objetivo es responder al usuario final en menos de 5 segundos para consultas típicas. *Nota: El rendimiento real dependerá críticamente de la latencia de las APIs externas (WhatsApp, OpenAI, Qdrant) y del rendimiento del servidor que hospeda n8n.*

**Concurrencia:** El sistema debe ser capaz de manejar múltiples conversaciones concurrentes. La capacidad específica dependerá de la configuración de **workers** de n8n y los recursos del servidor host.



**Rendimiento de Inicialización:** El tiempo para inicializar/actualizar la base de conocimientos dependerá del volumen de documentos en Google Drive, la velocidad de la API de OpenAI para embeddings y la capacidad de escritura de Qdrant.

## 5. Lenguajes de Programación

**Lógica Personalizada:** JavaScript (dentro de los nodos `Function` y `Code` de n8n).

**Definición del Flujo:** JSON (estructura nativa de los workflows de n8n).

**Consultas a Base de Datos:** SQL (dialecto PostgreSQL utilizado en los nodos Postgres).

## 6. Herramientas de Desarrollo

**Principal:** Editor de n8n (interfaz web para diseño y depuración de flujos).

**Edición de Código:** Potencialmente un IDE externo (ej. Visual Studio Code) para escribir/validar el JavaScript de los nodos `Function`.

**Gestión de Base de Datos:** Un cliente de PostgreSQL (ej. pgAdmin, DBeaver) para diseño inicial, inspección de datos y depuración.

**Control de Versiones (Recomendado):** Git y una plataforma como GitHub o GitLab para versionar el archivo JSON del workflow y cualquier script auxiliar.

**Pruebas API (Opcional):** Herramientas como Postman para probar interacciones directas con Qdrant HTTP API o simular webhooks durante el desarrollo.

## 7. Hardware

**Requerimientos de Usuario (finales):**

- Un dispositivo (smartphone, tablet, computadora) con la aplicación WhatsApp instalada y funcional.
- Conexión a Internet activa.

**Requerimientos del Sistema (Hosting n8n, DB, Qdrant):**

- Servidor(es): Uno o más servidores (físicos, VMs o contenedores) para alojar el motor n8n, la base de datos PostgreSQL y la base de datos vectorial Qdrant (si son auto-hospedados).

- **Recursos:** CPU, RAM y almacenamiento en disco adecuados para la carga esperada. La cantidad específica dependerá del volumen de mensajes, la concurrencia y la complejidad del flujo. Se requiere un punto de partida razonable (ej. múltiples núcleos de CPU, varios GB de RAM) y monitorizar para ajustar.
- **Red:** Conectividad de red fiable y de baja latencia a Internet para interactuar con las APIs externas (WhatsApp, OpenAI, Google Drive).

### **Expectativas de los Usuarios:**

- **Rapidez:** Esperan respuestas casi inmediatas a sus mensajes de texto (< 5-10 segundos idealmente).
- **Precisión y Relevancia:** Las respuestas deben ser correctas, basadas en la información de la empresa (base de conocimientos) y relevantes para su consulta.
- **Conversación Natural:** Esperan poder escribir de forma natural y que el sistema entienda la intención detrás de sus mensajes.
- **Consistencia:** El asistente debe recordar quiénes son (una vez registrados) y mantener el contexto de la conversación (gestionado por el módulo de memoria de Langchain).
- **Claridad:** Los mensajes del asistente (bienvenida, confirmación, respuestas) deben ser claros, concisos y fáciles de entender.
- **Resolución Eficaz:** El objetivo final es resolver su consulta o completar su solicitud (ej. registro) de manera eficiente.
- **Escalada Transparente:** Si se requiere un humano, esperan ser informados y que la transición sea lo más fluida posible (el JSON actual solo identifica la necesidad).

### **Características más Importantes:**

- Disponibilidad 24/7 (implícita al ser automatizado).
- Comprensión del Lenguaje Natural.
- Acceso rápido a información de la base de conocimientos.
- Proceso de registro/onboarding simple para nuevos usuarios.
- Persistencia de la identidad del cliente.

## **Requerimientos del Sistema (requisitos técnicos del sistema en el cual se ejecutará)**

### **1. Software:**

- Motor n8n (versión compatible con los nodos utilizados, ej.  $\geq 1.0$ ).
- Node.js (versión compatible con n8n, si es auto-hospedado).
- Servidor de Base de Datos PostgreSQL (versión compatible, ej.  $\geq 12$ ).
- Servidor de Base de Datos Vectorial Qdrant (versión compatible).
- Credenciales/Acceso a: API de WhatsApp Business, API de OpenAI, API de Google Drive, API de Qdrant.
- Navegador Web moderno (para el Editor n8n).

### **2. Hardware:**

- Servidor(es) con capacidad de CPU, RAM y Disco adecuada para la carga de n8n, PostgreSQL y Qdrant. (Ver sección Hardware -> Requerimientos del Sistema).
- Conexión de Red estable y con suficiente ancho de banda.

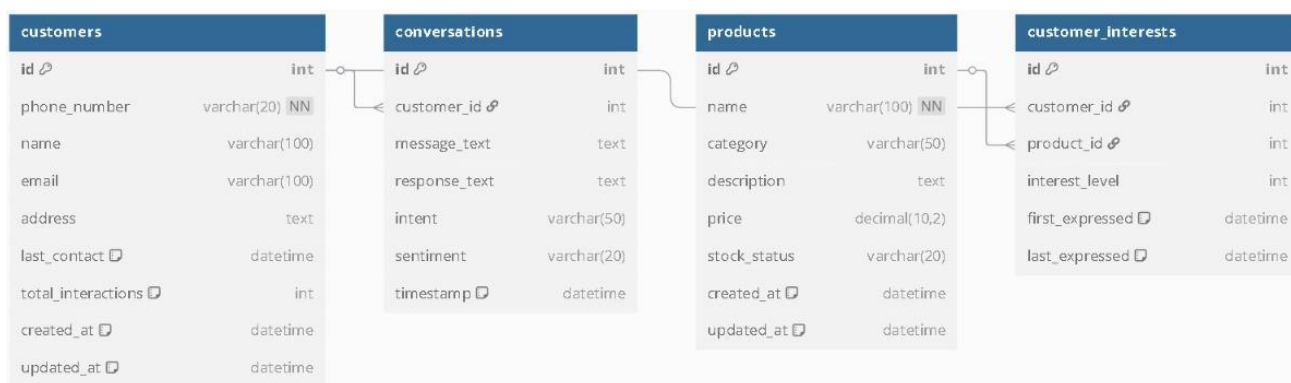
### **3. Sistema Operativo:**

- Compatible con los requisitos de n8n, PostgreSQL y Qdrant.  
Comúnmente:
  - Distribuciones de Linux (Debian, Ubuntu, CentOS, etc.) - Preferido para producción.
  - Windows Server (Posible, verificar compatibilidad de todas las herramientas).
  - macOS (Principalmente para desarrollo).
  - Entornos contenerizados (Docker).

## **Interfases Propuestas**

- Panel N8N para administradores
- WhatsApp para usuarios finales
- API de conexión con IA

## Base de Datos del Proyecto



## Descripción General

La base de datos está diseñada para soportar el sistema de asistente virtual automatizado en WhatsApp. Esta estructura permitirá registrar usuarios, almacenar conversaciones, administrar productos, y relacionar intereses de clientes.

## Entidades Principales

- **Customers:** Información de clientes (nombre, teléfono, email, etc.)
- **Conversations:** Registro de mensajes enviados y recibidos, junto con su análisis (intención y sentimiento).
- **Products:** Catálogo de productos disponibles para consulta y venta.
- **Customer\_Interests:** Asociación de productos con clientes interesados, con niveles de interés.

## Propósito de la Base de Datos

Esta estructura de base de datos permite gestionar de forma eficiente la interacción cliente-asistente, así como analizar intereses de productos y conversaciones para optimizar la experiencia de usuario y las estrategias comerciales.

# DISEÑO DE LA BASE DE DATOS

## Especificaciones:

### 1. Arquitectura

**Modelo:** Arquitectura orientada a eventos y microservicios orquestada por n8n.

**Componente Central:** Flujo de trabajo n8n ("V4-Improved") que actúa como backend y orquestador.

**Activador:** Evento de mensaje entrante desde la API de WhatsApp (n8n-nodes-base.whatsappTrigger).

**Lógica de Negocio:** Implementada mediante una secuencia de nodos n8n (Switch, If, Function, nodos de Base de Datos, nodos de Langchain).

**Persistencia de Datos:** Base de datos PostgreSQL para almacenar información de clientes (customers), historial de conversaciones (conversations), catálogo de productos (products) e intereses de clientes (customer\_interests).

**Base de Conocimientos:** Repositorio de documentos en Google Drive, indexado en una base de datos vectorial Qdrant para búsqueda semántica.

#### Inteligencia Artificial:

Modelo de lenguaje OpenAI (GPT-4o) para análisis de mensajes (intención, sentimiento) y generación de respuestas conversacionales (@n8n/n8n-nodes-langchain.openAi, @n8n/n8n-nodes-langchain.lmChatOpenAi).

Modelo de embeddings OpenAI para indexación y búsqueda en Qdrant (@n8n/n8n-nodes-langchain.embeddingsOpenAi).

Orquestación AI avanzada mediante Langchain dentro de n8n (@n8n/n8n-nodes-langchain.agent, @n8n/n8n-nodes-langchain.memoryBufferWindow, @n8n/n8n-nodes-langchain.toolVectorStore).

**Interfaces Externas:** API de WhatsApp Business, API de OpenAI, Base de Datos PostgreSQL, Vector Store Qdrant (vía HTTP y/o Langchain), API de Google Drive.

## **Flujos Separados:**

Flujo principal reactivo a mensajes de WhatsApp.

Flujo(s) de inicialización manual para configurar la base de datos relacional y la base de datos vectorial (Qdrant) a partir de Google Drive.

## 2. Usabilidad

**Interacción Principal:** A través de mensajes de texto en la interfaz nativa de WhatsApp del usuario final.

**Feedback al Usuario:** Respuestas automáticas contextuales: bienvenida a nuevos usuarios, solicitud de información, confirmación de registro, respuestas a consultas generadas por IA, mensaje para tipos de mensaje no soportados.

**Manejo de Errores (Usuario):** Se informa al usuario si envía un mensaje no textual. No se detalla explícitamente cómo se manejan otros errores de cara al usuario (ej. si la IA falla).

**Onboarding:** Flujo guiado para nuevos usuarios solicitando información básica (nombre, email, dirección) en el segundo mensaje.

**Lenguaje Natural:** Diseñado para procesar y responder a consultas en lenguaje natural.

**Escalado a Humanos:** Previsto (Human Intervention Check, Notify Human Agent), pero la notificación real y el proceso de handover no están implementados en el JSON (son placeholders).

### a) Interfaz de Usuario

**Usuario Final:** La interfaz de chat de la aplicación WhatsApp (móvil o escritorio). El sistema no provee una interfaz propia para el usuario final.

**Administrador/Desarrollador:** La interfaz de usuario del editor de n8n para diseñar, modificar, monitorear y depurar el flujo de trabajo. También para ejecutar manualmente los flujos de inicialización.

b) Compatibilidad con Dispositivos Móviles y de Escritorio

**Usuario Final:** Alta compatibilidad, ya que depende de la aplicación WhatsApp, disponible en la mayoría de dispositivos móviles (iOS, Android) y plataformas de escritorio (Windows, macOS, Web).

**Backend (n8n):** Se ejecuta en el entorno del servidor donde se hospeda n8n (Linux, Docker, etc.), no directamente en el dispositivo del usuario final.

c) Que se espera a 5 años y como enfrentar la escalabilidad.

**Expectativa a 5 años:** Se espera un aumento significativo en el volumen de conversaciones, la cantidad de clientes registrados, el tamaño de la base de conocimientos y potencialmente la complejidad de las interacciones y las integraciones requeridas (CRM, ERP, etc.). El sistema debe mantener tiempos de respuesta bajos y alta disponibilidad.

**Estrategias de Escalabilidad:**

**n8n:**

Utilizar el modo queue y escalar horizontalmente añadiendo más workers de n8n.

Optimizar el flujo: refactorizar nodos Function complejos, minimizar operaciones bloqueantes.

Dividir el flujo en micro-workflows más pequeños si la complejidad aumenta.

Considerar n8n Enterprise para características avanzadas de escalabilidad y gestión.

**Base de Datos (PostgreSQL):**

Optimizar consultas y asegurar índices adecuados (ya incluidos en el script de inicialización).

Implementar connection pooling (ej. PgBouncer) para manejar más conexiones concurrentes desde n8n.

Escalar verticalmente (más recursos al servidor DB) u horizontalmente (read replicas para consultas, sharding si es necesario en un futuro muy lejano).

Mantenimiento regular (VACUUM, ANALYZE).

### **Vector Store (Qdrant):**

Escalar la instancia/cluster de Qdrant (vertical u horizontalmente) según las recomendaciones del proveedor o los recursos de auto-hospedaje.

Optimizar la configuración de la colección y los parámetros de indexación.

### **APIs Externas (OpenAI, WhatsApp):**

Monitorear el uso y los límites de las APIs.

Implementar rate limiting y backoff/retry logic en las llamadas desde n8n.

Optimizar el uso de tokens de OpenAI (prompts más eficientes, modelos adecuados).

**Base de Conocimientos (Google Drive):** Gestionar el volumen de documentos. Optimizar el proceso de re-indexación en Qdrant para que sea eficiente.

**Procesamiento Asíncrono:** Para tareas no críticas que puedan surgir (analíticas complejas, notificaciones diferidas), derivarlas a colas de mensajes y workflows separados para no impactar la latencia de respuesta al usuario.

## **3. Timeline y backlog esperado**

### **Hitos Clave Alcanzados**

1. Integración básica con WhatsApp y manejo de tipos de mensaje.
2. Creación/Consulta/Actualización de Clientes en PostgreSQL.
3. Flujo de bienvenida y onboarding para nuevos clientes.
4. Lógica para procesar y almacenar información de registro de clientes existentes.
5. Infraestructura para inicialización de Base de Datos (Postgres y Qdrant desde GDrive).
6. Implementación del Agente AI conversacional con Memoria y Herramienta de Base de Conocimientos (Langchain).
7. Logging estructurado de conversaciones (bienvenida, registro, interacciones AI).



8. Mecanismo básico para identificar necesidad de intervención humana.
9. Placeholder para registro de intereses de producto.

**Backlog Esperado (Tareas Pendientes/Próximas):**

1. **Pruebas exhaustivas:** Pruebas unitarias (nodos individuales), pruebas de integración (flujos completos), pruebas de carga.

**2. Refinamiento del Flujo:**

Reemplazar valores hardcoded por variables/credenciales.

Implementar manejo de errores robusto (Try/Catch, gestión de fallos de API/DB).

Revisar/confirmar la necesidad del nodo Extract Customer Details post-IA.

**3. Implementación de Placeholders:**

Desarrollar la notificación real a agentes humanos (Notify Human Agent).

Implementar la lógica de negocio para registrar intereses de producto (Update Product Interests).

4. **Handover a Agente Humano:** Definir e implementar el proceso y posible interfaz para que un agente tome control de una conversación.
5. **Gestión de Conocimiento:** Establecer un proceso claro para actualizar documentos en Google Drive y ejecutar el flujo de re-indexación de Qdrant periódicamente.
6. **Monitorización y Analítica:** Configurar monitorización de rendimiento del workflow, latencia de API, uso de recursos. Desarrollar reportes/dashboards basados en los datos de las tablas conversations y customers.
7. **Despliegue a Producción:** Preparar entorno, gestión segura de credenciales (API Keys, DB passwords), configurar URL de Qdrant real.
8. **Considerar FRs Adicionales (del SRS original si aplican):** Implementar consulta de bolsa de trabajo (RF-5) si se requiere. Desarrollar panel admin si es necesario (RF-1, RF-2, RF-3, RF-7).

#### 4. Diccionario de la base de datos

##### 1. Tabla: customers

| NOMBRE DEL CAMPO   | TIPO      | LONGITUD | DESCRIPCIÓN   |
|--------------------|-----------|----------|---|
| id                 | SERIAL    | N/A      | Identificador único autoincremental del cliente (PK)    |
| phone_number       | VARCHAR   | 20       | Número de teléfono del cliente (UNIQUE, NOT NULL)       |
| name               | VARCHAR   | 100      | Nombre del cliente                                      |
| email              | VARCHAR   | 100      | Correo electrónico del cliente                          |
| address            | VARCHAR   | 255      | Dirección del cliente                                   |
| last_contact       | TIMESTAMP | N/A      | Fecha y hora del último contacto (DEFAULT NOW())        |
| total_interactions | INTEGER   | N/A      | Contador de interacciones del cliente (DEFAULT 1)       |
| created_at         | TIMESTAMP | N/A      | Fecha y hora de creación del registro (DEFAULT NOW())   |
| updated_at         | TIMESTAMP | N/A      | Fecha y hora de la última actualización (DEFAULT NOW()) |

##### 2. Tabla: conversations

| NOMBRE DEL CAMPO | TIPO      | LONGITUD | DESCRIPCIÓN  |
|------------------|-----------|----------|--|
| id               | SERIAL    | N/A      | Identificador único autoincremental de la conversación (PK)  |
| customer_id      | VARCHAR   | 20       | ID del cliente asociado (FK references customers.id)         |
| message_text     | VARCHAR   | 100      | Texto del mensaje enviado por el cliente                     |
| response_text    | VARCHAR   | 100      | Texto de la respuesta enviada por el asistente               |
| intent           | VARCHAR   | 255      | Intención detectada en el mensaje del cliente                |
| sentiment        | VARCHAR   | 100      | Sentimiento detectado en el mensaje del cliente              |
| timestamp        | TIMESTAMP | N/A      | Fecha y hora del registro de la conversación (DEFAULT NOW()) |

### 3. Tabla: products

| NOMBRE DEL CAMPO | TIPO      | LONGITUD | DESCRIPCIÓN   |
|------------------|-----------|----------|---|
| id               | SERIAL    | N/A      | Identificador único autoincremental del producto (PK) |
| name             | VARCHAR   | 20       | Nombre del producto (NOT NULL)                        |
| category         | VARCHAR   | 100      | Categoría del producto                                |
| description      | VARCHAR   | 100      | Descripción del producto                              |
| price            | DECIMAL   | 255      | Precio del producto                                   |
| stock_status     | VARCHAR   | 100      | Estado del stock (ej. 'in stock', 'out of stock')     |
| created_at       | TIMESTAMP | N/A      | Fecha y hora de creación del registro (DEFAULT NOW()) |

### 4. Tabla: customer\_interests

| NOMBRE DEL CAMPO | TIPO      | LONGITUD | DESCRIPCIÓN   |
|------------------|-----------|----------|---|
| id               | SERIAL    | N/A      | Identificador único autoincremental del interés (PK)            |
| customer_id      | INTEGER   | N/A      | ID del cliente asociado (FK references customers.id)            |
| product_id       | INTEGER   | N/A      | ID del producto asociado (FK references products.id)            |
| interest_level   | INTEGER   | N/A      | Nivel de interés (ej. 1-5)                                      |
| first_expressed  | TIMESTAMP | N/A      | Fecha y hora de la primera expresión de interés (DEFAULT NOW()) |
| last_expressed   | TIMESTAMP | N/A      | Fecha y hora de la última expresión de interés (DEFAULT NOW())  |

### 5. Diseño de la base de datos

