

**Tema: Desarrollo e Implementación de un Chatbot Conversacional para el Ingreso y Consulta de Datos en Bases de Datos SQL en Pequeños Negocios**

**Alumna: Cynthia Marcela Villagra**

**Carrera: Tecnicatura Superior en Ciencias de Datos e Inteligencia Artificial**

**Materia: Proyecto Integrador**

**Profesor: Erick Ravelo**

**2do Cuatrimestre 2025**

# **ÍNDICE**

[**ÍNDICE 1**](#_tamqbv3pmglr)

[**1. Análisis de la Problemática 4**](#_mkmm0ac83wov)

[1.1 Planteamiento del Problema 4](#_x6oj1u2rypbb)

[1.2 Planteamiento de la Solución 4](#_tsdweiipvdcl)

[1.3 Hipótesis 5](#_p2o0jeuy3z5q)

[1.4 Objetivos Generales y Específicos 5](#_xzask3gpyzrz)

[1.4.1 Objetivo General 5](#_o29f9blug1ej)

[1.4.2 Objetivos Específicos 5](#_elc4wu7kgfxu)

[1.5 Marco Contextual 5](#_1zueb11gubwp)

[1.5.1 Ampliación del Problema 5](#_j8k1mod2ugb7)

[1.5.2 Revisión del Estado del Arte 6](#_m6d7r8reuxz8)

[1.5.3 Justificación del Problema 6](#_h0iwp52yc17n)

[**2. Descripción Técnica del Proyecto 7**](#_fd780om7j9pg)

[2.2 Descripción Técnica del Proyecto 7](#_jnrn3e6s84qi)

[2.3 Arquitectura del Proyecto 7](#_la9cfm6yc7ul)

[Arquitectura Seleccionada: Cliente-Servidor 7](#_5fe0kmrgw03b)

[Componentes Principales 7](#_twovwq4l2zqk)

[Chatbot basado en OpenAI: 7](#_g0rsqnub8iwj)

[API REST en entorno PHP: 7](#_i8sis9be2h4p)

[Base de Datos SQL (MySQL): 7](#_pm1g9ubh9z6g)

[2.4 Tecnologías y Herramientas 8](#_sgypf8i5zhhk)

[Lenguajes de Programación: 8](#_scpwgzu2ytol)

[Frameworks y Librerías según el entorno: 8](#_gstmvbhu5nq)

[Para API REST en PHP (a futuro): 8](#_9b3ggjisa159)

[Para consumo de API REST en HTML, CSS y JavaScript: 8](#_tdx3e96zucjg)

[Para Lógica de Servidor: 8](#_vnzv1gk668s9)

[Bases de Datos: 8](#_efhynda42aab)

[Herramientas de Desarrollo y Pruebas: 8](#_yhssaai5qto4)

[2.5 Componentes Técnicos 8](#_jux1e6p5lhzb)

[Procesamiento de Datos 8](#_7mkoro5ocpu)

[Modelos de Machine Learning 9](#_btdipfxbo50)

[Interfaz de Usuario (UI) 9](#_28bl259nmhwa)

[Seguridad 9](#_wqp6uuoewgd6)

[2.6 Limitaciones de las Tecnologías y Herramientas 9](#_orv9nganv82p)

[Rendimiento: 9](#_r8tytb94tre0)

[Escalabilidad: 9](#_mhjklkyrfj15)

[Compatibilidad: 9](#_rijwpgp7v3x)

[Curva de Aprendizaje: 10](#_m36yi6g4bzk8)

[Costo: 10](#_bjofsdr1fig3)

[**3. Metodología de Trabajo 10**](#_xnggm7xlb3nn)

[3.1 Elección de la Metodología Ágil 10](#_agw51r8sj0zu)

[3.2 Descripción del Proceso Ágil 10](#_y2x2touvzjw5)

[3.3 Gestión del Progreso y Herramientas 11](#_rqq5ngymdcmf)

[3.4 Adaptabilidad y Mejora Continua 11](#_2ge6j1s9q1n5)

[**4. Diagramado del proyecto 12**](#_quprrmxjnviy)

[4.1. Diagrama de Gantt 12](#_g2gat9js6ozs)

[4.2. Flujograma funcional de la herramienta 14](#_6muab8k8omv3)

[4.2.1 Flujograma 15](#_njtnzenstm6z)

[**5. Análisis financiero 16**](#_jf9tlxbnen0d)

[5.1 Matriz Van (en u$s) 16](#_4ylofqk8dbli)

[5.2 Conclusión del Análisis Financiero 16](#_larw5w1m386m)

[**6. Conclusiones 17**](#_df8dusxc8l9a)

[6.1 Síntesis de Resultados 17](#_gzpnc738ukof)

[6.2 Evaluación de Hipótesis 17](#_jgvqh7s3wgqz)

[6.3 Evaluación de Objetivos Cumplidos 18](#_8o5yrxfjvtg)

[6.4 Recomendaciones para Futuros Proyectos 19](#_tvayogct8enp)

[6.5 Limitaciones del Proyecto 19](#_ri6wf8kbzeu1)

[7. Bibliografía consultada 20](#_v1k890i7vc9n)

[**8. Anexos 22**](#_4x1mzraymshc)

[8.A. Anexo A. Experimento de medición de tiempos y validación del ahorro 22](#_q2q7re8oyp8y)

[8.A.1. Objetivo del experimento 22](#_f2cvz3r6thi4)

[8.A.2. Diseño del experimento 22](#_2s7onhhvbx1w)

[8.A.3. Resultados obtenidos 23](#_hx09byrui728)

[8.A.4. Conclusión del experimento 23](#_hn91q8a0pbd6)

[8.B. Anexo B. Cálculo del ahorro económico y análisis financiero 24](#_2h7b0125gpa7)

[8.B.1. Parámetros base 24](#_5ygtofi1ocew)

[8.B.2. Costos involucrados 24](#_chb0wmc877l2)

[8.B.2.1 Justificación de costo de realizar un solo registro 24](#_gtorhuyp9xw)

[8.B.2.2 Costos involucrados (Tabla) 26](#_p0odw538ld6k)

[8.B.3. Beneficio estimado 26](#_btx8x0ib5f9s)

[8.B.4. Fuentes consultadas 26](#_ko7n7s5ve7y7)

## 

# **1. Análisis de la Problemática**

## **1.1 Planteamiento del Problema**

En la actualidad, muchos pequeños negocios enfrentan dificultades para gestionar sus bases de datos de manera eficiente. Estas dificultades incluyen:

* **Errores de transcripción y redundancia:** Al registrar datos manualmente, es común que se cometan errores o se ingresen datos duplicados, lo que afecta la calidad de la información almacenada.
* **Falta de conocimientos técnicos:** Pequeños emprendedores y dueños de negocios no siempre cuentan con la capacitación necesaria para manejar bases de datos SQL o realizar consultas de manera efectiva.
* **Pérdida de tiempo:** La gestión manual de datos requiere pasos adicionales, como anotar información en papel y luego ingresarla en la base de datos, lo que incrementa la posibilidad de errores y duplica el trabajo.
* **Dificultad en la actualización de registros:** Sin un sistema adecuado, mantener actualizados los datos puede ser un proceso complicado y propenso a errores.

Se necesita una solución automatizada que permita ingresar y consultar datos de manera sencilla y eficiente, eliminando la dependencia de procesos manuales y reduciendo los errores en la gestión de información. Esta solución debe ser accesible para pequeños negocios, sin requerir conocimientos avanzados en bases de datos o programación.

## **1.2 Planteamiento de la Solución**

Se propone una solución basada en interacciones conversacionales mediante un chatbot impulsado por ChatGPT, que permita ingresar y consultar datos en bases de datos SQL de manera sencilla y precisa.

El sistema permitirá a los usuarios interactuar en lenguaje natural con un chatbot, quien interpretará las solicitudes y realizará las acciones correspondientes en la base de datos.

La solución ofrece:

* **Ingreso de datos conversacional:** Los usuarios podrán dictar o escribir información en lenguaje natural sin necesidad de utilizar formularios.
* **Validación automática:** Antes de ingresar un nuevo dato, el sistema realizará diversas validaciones para garantizar la calidad de la información almacenada. Se verificará si el dato ya existe en la base de datos, analizará similitudes con registros previos para evitar duplicaciones, validará formatos de datos y detectará información incompleta.
* **Corrección de errores en tiempo real:** Se identificarán datos inconsistentes o duplicados para minimizar errores.
* **Consultas en lenguaje natural (a futuro):** Si el usuario lo requiere, podrá realizar consultas sobre la base de datos sin necesidad de escribir consultas SQL manualmente.

## **1.3 Hipótesis**

1. La automatización del ingreso de datos reducirá los errores de transcripción en al menos un 30%, en comparación con el ingreso manual de datos en pequeños negocios.
2. Los pequeños negocios sin conocimientos en bases de datos reducirán el tiempo de ingreso y consulta de datos en al menos un 40% mediante la implementación del chatbot conversacional.
3. La implementación del chatbot reducirá en un 50% el tiempo que los dueños de pequeños negocios dedican a la administración de datos, en comparación con métodos manuales.

## **1.4 Objetivos Generales y Específicos**

### **1.4.1 Objetivo General**

Desarrollar una herramienta conversacional basada en OpenAI que permita a pequeños negocios ingresar y consultar datos en su base de datos SQL sin conocimientos técnicos, reduciendo errores y mejorando la eficiencia.

### **1.4.2 Objetivos Específicos**

* Implementar la conexión entre OpenAI y bases de datos SQL.
* Desarrollar un sistema de validación para evitar duplicaciones y errores.
* Permitir la consulta de datos en lenguaje natural si el cliente lo requiere.
* Crear un flujo de interacción intuitivo para usuarios sin experiencia en bases de datos.
* Probar la solución y medir su efectividad mediante la reducción de errores, la disminución del tiempo de ingreso y consulta de información, y la optimización del tiempo de administración de datos en pequeños negocios.

## **1.5 Marco Contextual**

### **1.5.1 Ampliación del Problema**

La gestión de datos en pequeños negocios representa un desafío significativo debido a la falta de herramientas accesibles y automatizadas. Los métodos actuales presentan diversas limitaciones, entre ellas:

* **Dependencia de procesos manuales**, lo que aumenta la probabilidad de errores de transcripción y redundancia en los datos.
* **Falta de conocimientos técnicos** en bases de datos por parte de los dueños de negocios.
* **Uso ineficiente del tiempo**, ya que los datos suelen registrarse primero en papel o aplicaciones básicas, para luego ser transcritos manualmente.
* **Dificultad para mantener la información actualizada**, lo que puede generar inconsistencias en los registros de clientes, inventarios y transacciones.

### **1.5.2 Revisión del Estado del Arte**

Investigaciones previas han explorado la aplicación de chatbots en la automatización de procesos, principalmente en el ámbito del servicio al cliente y el comercio electrónico. Sin embargo, pocos modelos han sido diseñados específicamente para la interacción con bases de datos SQL, dejando un vacío en la optimización del manejo de información en pequeños negocios.

### **1.5.3 Justificación del Problema**

Esta solución es necesaria debido a:

* **Accesibilidad:** No todos los pequeños negocios pueden costear un sistema de gestión avanzado o contratar un administrador de bases de datos.
* **Automatización:** Evita pasos innecesarios como escribir datos en papel y luego ingresarlos manualmente.
* **Eficiencia:** Reduce el tiempo dedicado a tareas administrativas.
* **Menor tasa de errores:** La validación en tiempo real evita inconsistencias en la base de datos.
* **Ahorro de costos:** Evita la necesidad de contratar personal especializado en gestión de datos.

# **2. Descripción Técnica del Proyecto**

## **2.2 Descripción Técnica del Proyecto**

El proyecto consiste en el desarrollo de un sistema conversacional basado en ChatGPT que permite el ingreso y consulta de datos en bases de datos SQL sin necesidad de conocimientos técnicos avanzados. La solución se basa en una **API REST**, que actúa como intermediario entre el chatbot y la base de datos, facilitando la comunicación y garantizando la validación de la información antes de su almacenamiento.

## **2.3 Arquitectura del Proyecto**

El sistema se basa en una **arquitectura cliente-servidor**, donde el cliente envía solicitudes a una **API REST** que procesa la información y la almacena en una base de datos SQL. Este modelo permite **escalabilidad, modularidad y compatibilidad** con distintos entornos.

### **Arquitectura Seleccionada: Cliente-Servidor**

* **Cliente:** Usuario que interactúa con el chatbot.
* **Servidor:** Recibe las solicitudes del chatbot, procesa la información y actualiza la base de datos SQL.

### **Componentes Principales**

#### **Chatbot basado en OpenAI:**

* Procesar las solicitudes del usuario en lenguaje natural.
* Generar consultas SQL basadas en la intención del usuario.
* Enviar solicitudes a la API REST para la ejecución de operaciones en la base de datos.

#### **API REST en entorno PHP:**

* Exponer endpoints para la comunicación con la base de datos.
* Manejar la autenticación y control de acceso a los datos.
* Implementar validaciones para evitar duplicaciones e inconsistencias.
* Responder en formato JSON para facilitar la integración con distintos clientes.

#### **Base de Datos SQL (MySQL):**

* Almacenar los datos estructurados según el esquema del sistema existente.
* Aplicar restricciones de integridad y validaciones de entrada de datos.
* Permitir consultas optimizadas para mejorar el rendimiento del sistema.

## **2.4 Tecnologías y Herramientas**

### **Lenguajes de Programación:**

* **PHP** en caso de requerirse una API REST en este entorno.
* **SQL** para la manipulación de datos en MySQL.
* **JavaScript** para consumir la API REST desde un frontend basado en HTML, CSS y JS.
* **Python** para Lógica del servidor.

### **Frameworks y Librerías según el entorno:**

#### **Para API REST en PHP (a futuro):**

* **JWT Authentication** → Para autenticación segura con tokens en PHP.

#### **Para consumo de API REST en HTML, CSS y JavaScript:**

* **Fetch API** → Para realizar solicitudes a la API REST desde JavaScript.

#### **Para Lógica de Servidor:**

* **Fast API** → Para realizar solicitudes a la API REST desde JavaScript.

### **Bases de Datos:**

* **MySQL** para almacenamiento en servidores externos.

### **Herramientas de Desarrollo y Pruebas:**

* **XAMPP** → servidor local con Apache + MySQL para pruebas en máquina de desarrollo.
* **Postman** → Para pruebas y documentación de la API REST.
* **Git** → Para control de versiones.
* **Windsurf**→ Como entorno de desarrollo principal.
* **PhpMyAdmin** → Para administrar bases de datos MySQL en servidores con PHP.

## **2.5 Componentes Técnicos**

### **Procesamiento de Datos**

* **Conversión en tiempo real de lenguaje natural a SQL:** ChatGPT interpreta las solicitudes del usuario y genera consultas SQL estructuradas.
* **Validación de datos antes del ingreso:** Se verifica que los registros no sean duplicados y cumplan con las restricciones establecidas en la base de datos.
* **Optimización de respuestas:** La API REST maneja las solicitudes de manera eficiente, reduciendo la latencia en la ejecución de consultas SQL.

### **Modelos de Machine Learning**

* **Modelo utilizado:** ChatGPT (basado en arquitecturas de Transformers).
* **Uso dentro del sistema:** Configuración y optimización del modelo para interpretar solicitudes en lenguaje natural y convertirlas en consultas SQL estructuradas.
* **Método de ajuste:** No se realiza entrenamiento propio, sino que se personaliza el comportamiento del chatbot mediante **prompts optimizados y configuración en OpenAI**.
* **Requisitos computacionales:** Procesamiento en la nube a través de la API de OpenAI, sin necesidad de infraestructura local para ejecutar modelos de Machine Learning.

### **Interfaz de Usuario (UI)**

* Se desarrolla un frontend con php/html/ js en este proyecto.
* La API REST es independiente de la interfaz de usuario, permitiendo que cualquier frontend PHP, HTML/JS pueda consumir sus endpoints sin modificaciones.

### **Seguridad**

Si bien en esta etapa del proyecto (MVP) no se implementaron todas las medidas de seguridad, se definen a continuación las prácticas futuras previstas para la versión funcional completa:

* **Protección contra inyección SQL**: Las consultas a la base de datos se realizarán usando parámetros seguros, evitando concatenar texto directamente. Esto reduce el riesgo de inyección SQL.
* **Gestión de permisos**: Los endpoints del chatbot estarán protegidos mediante claves secretas o tokens.
* **Cifrado de datos sensibles**: Se planea asegurar toda la comunicación entre PHP, MCP y la base de datos mediante HTTPS y conexiones cifradas (TLS/SSL).

## **2.6 Limitaciones de las Tecnologías y Herramientas**

### **Rendimiento:**

* La generación de consultas SQL por ChatGPT puede causar ligeros retrasos en comparación con consultas SQL escritas manualmente.

### **Escalabilidad:**

* SQLite3 no es ideal para bases de datos grandes.

### **Compatibilidad:**

* La API REST está diseñada para bases de datos SQL; no es compatible con NoSQL.

### **Curva de Aprendizaje:**

* Usuarios sin conocimientos previos en entornos de desarrollo y APIs pueden necesitar **capacitación básica para activar el servidor e interpretar respuestas del chatbot y estructurar consultas en lenguaje natural de manera efectiva**.

### **Costo:**

* El uso de la API de OpenAI para la interpretación de lenguaje natural tiene costos asociados, ya que se basa en un modelo de pago por uso. Dependiendo del volumen de consultas procesadas por el chatbot, los costos pueden aumentar, por lo que es necesario optimizar su uso para evitar gastos innecesarios.
* El hosting donde se alojará la solución tambien representa un costo asociado a considerar por cliente.

# **3. Metodología de Trabajo**

## **3.1 Elección de la Metodología Ágil**

Se eligió una metodología ágil **Kanban individual**, ya que el proyecto es desarrollado por una sola persona. Es la más adecuada porque:

* No se requiere coordinación entre múltiples roles como en Scrum.
* Permite visualizar tareas en flujo continuo sin planificación por sprints.
* Favorece la flexibilidad y la priorización diaria.

## **3.2 Descripción del Proceso Ágil**

**Sprints/Ciclos:**No se trabaja con sprints fijos. Las tareas se gestionan de forma continua según prioridad y disponibilidad, en un esquema de flujo permanente.

**Backlog de Producto y Sprint Backlog:**Se utiliza un único **backlog general** con todas las tareas divididas en:

* Tareas pendientes
* En proceso
* Completadas

Las tareas se priorizan según dependencias y entregas próximas. No se usa backlog por sprint.

## **3.3 Gestión del Progreso y Herramientas**

**Herramienta Ágil:**Se utiliza **ClickUp** para organizar el tablero Kanban. Ahí se documentan todas las tareas, comentarios, fechas estimadas y entregas.

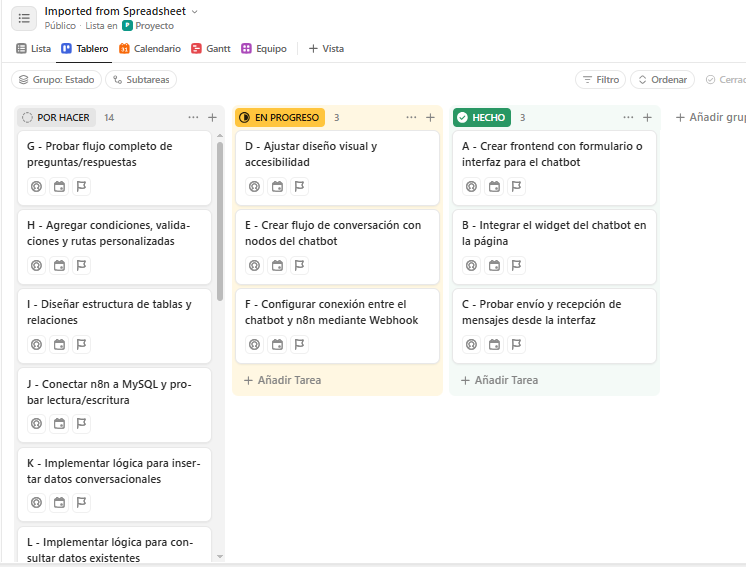
**Indicadores de Progreso:** El progreso se mide observando:

* Tareas completadas vs. planificadas por semana
* Estado visual del tablero (avance de columnas)
* Comparación entre la duración estimada (Gantt) y la duración real

## **3.4 Adaptabilidad y Mejora Continua**

**Plan de Adaptación:** En caso de imprevistos o cambios en la prioridad:

* Se reordena el backlog según urgencia o nuevas tareas surgidas
* Se ajustan fechas y dependencias en el Gantt
* Se registran desvíos para mejorar la estimación en entregas futuras



# **4. Diagramado del proyecto**

## **4.1. Diagrama de Gantt**

Se elaboró un diagrama de Gantt que detalla todas las tareas del proyecto, agrupadas en bloques funcionales:

* Diseño de la interfaz web
* Integración con MCP
* Conexión con la base de datos MySQL
* Pruebas y ajustes
* Entregas teóricas

Cada tarea tiene asignada una duración estimada, una o más predecesoras, y se puede visualizar el orden de ejecución. El cronograma permite también superposición entre tareas, respetando las dependencias lógicas y plazos de entrega. No se registraron diferencias entre el tiempo estimado y el tiempo real de ejecución, ya que las tareas fueron cronometradas durante su desarrollo efectivo.

| Grupo | Letra | Tarea | Duración estimada (días) | Tareas Predecesoras |
| --- | --- | --- | --- | --- |
| Diseño de la Interfaz Web | A | Crear frontend con formulario o interfaz para el chatbot | 2 |  |
| Diseño de la Interfaz Web | B | Integrar el widget del chatbot en la página | 2 | A |
| Diseño de la Interfaz Web | C | Probar envío y recepción de mensajes desde la interfaz | 3 | B |
| Diseño de la Interfaz Web | D | Ajustar diseño visual y accesibilidad | 1 | C |
| Integración con MCP | E | Crear flujo de conversación del chatbot | 2 |  |
| Integración con MCP | F | Configurar conexión entre el chatbot y la web | 1 | E |
| Integración con MCP | G | Probar flujo completo de preguntas/respuestas | 2 | F |
| Integración con MCP | H | Agregar condiciones y validaciones | 2 | G |
| Base de Datos MySQL | I | Diseñar estructura de tablas y relaciones | 2 |  |
| Base de Datos MySQL | J | Conectar el backend a MySQL y probar lectura/escritura | 2 | I |
| Base de Datos MySQL | K | Implementar lógica para insertar datos conversacionales | 2 | J |
| Base de Datos MySQL | L | Implementar lógica para consultar datos existentes | 2 | K |
| Pruebas y Ajustes Finales | M | Validar funcionamiento completo del sistema | 2 | C,H,L |
| Pruebas y Ajustes Finales | N | Ajustar errores detectados y refinar respuestas del bot | 2 | M |
| Pruebas y Ajustes Finales | O | Preparar demo grabada o en vivo del sistema funcionando | 1 | M |
| Pruebas y Ajustes Finales | P | Documentar funcionalidades principales | 2 | K |
| Entregas Teóricas | Q | Entrega 1: Análisis del problema y descripción técnica | 6 |  |
| Entregas Teóricas | R | Entrega 2: Metodología ágil (Scrum/Kanban) y diagramas | 2 | Q |
| Entregas Teóricas | S | Entrega 3: Análisis financiero + conclusiones | 1 | P, R |
| Entregas Teóricas | T | Entrega 4: Presentación (PPT + discurso) | 1 | S,P |



| Letra | SEMANA 1 | | | SEMANA 2 | | | SEMANA 3 | | | SEMANA 4 | | | SEMANA 5 | | | SEMANA 6 | | | SEMANA 7 | | | SEMANA 8 | | | SEMANA 9 | | | SEMANA 10 | | | SEMANA 11 | | | SEMANA 12 | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Diseño de la Interfaz Web | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A | a |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  |  | b |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  |  | c |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  | d |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | Integración con MCP | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  | e |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F |  |  |  |  |  |  |  |  |  |  | f |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| G |  |  |  |  |  |  |  |  |  |  |  | g |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| H |  |  |  |  |  |  |  |  |  |  |  |  |  | h |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Base de Datos MySQL | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | i |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| J |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | j |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| K |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | k |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | l |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pruebas y Ajustes Finales | | | | | | |  |  |  |  |  |  |
| M |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | m |  |  |  |  |  |  |  |  |  |  |  |  |
| N |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | n |  |  |  |  |  |  |  |  |  |  |
| O |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | o |  |  |  |  |  |  |  |  |
| P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | p |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Entregas Teóricas | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |
| Q | q |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | r |  |  |  |  |  |
| S |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | s |  |  |  |
| T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | t |  |  |



## 

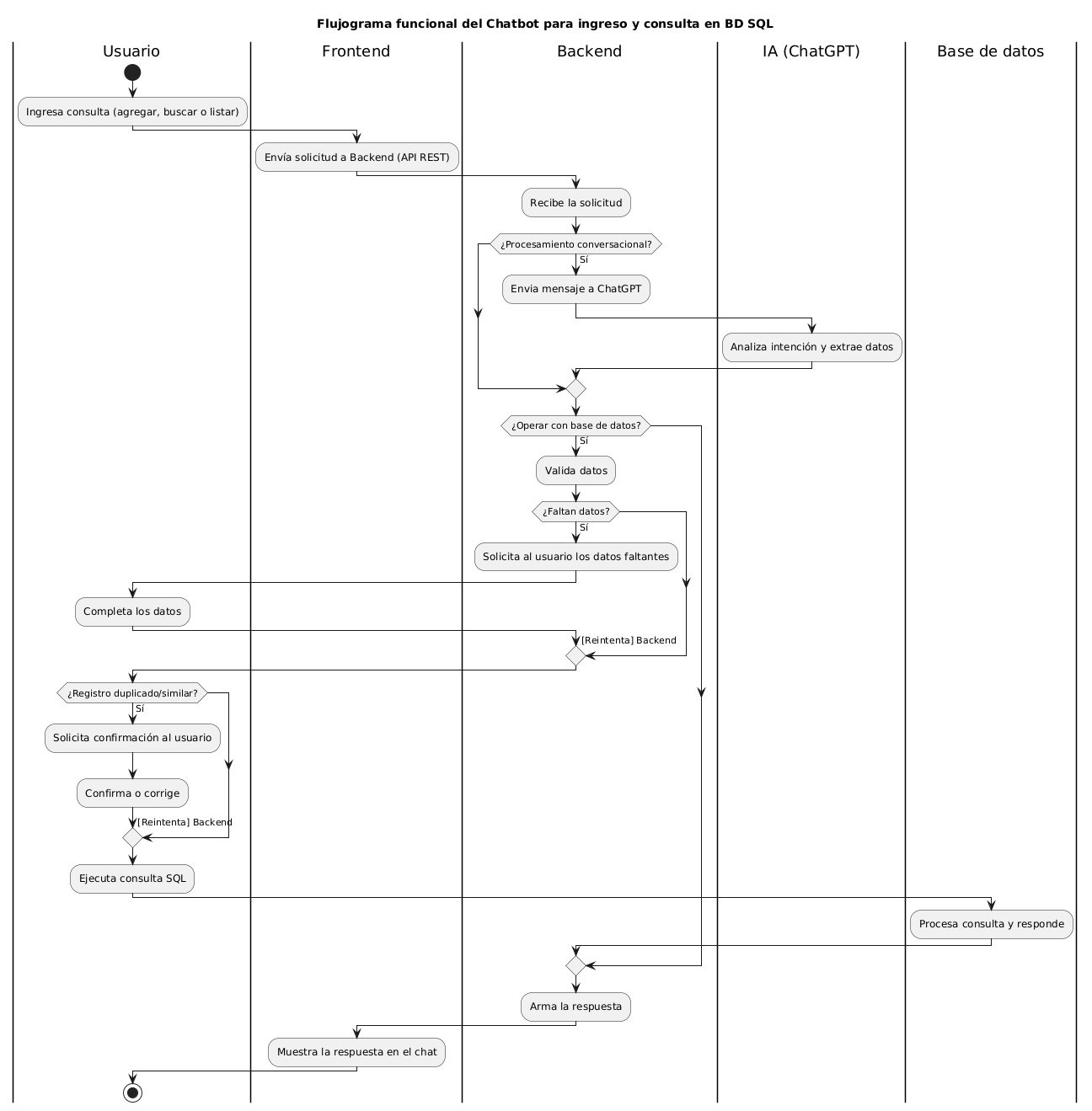
## **4.2. Flujograma funcional de la herramienta**

El flujograma que representa el funcionamiento de la herramienta incluye las siguientes etapas:

1. **El usuario ingresa una consulta desde la interfaz web**, ya sea para agregar, buscar o listar datos.
2. **El frontend envía la solicitud al backend** a través de la API REST.
3. **El backend recibe y analiza la solicitud**. Si corresponde, el mensaje es procesado por el módulo conversacional (IA) para interpretar la intención y extraer los datos relevantes.
4. **Si la acción requiere operar con la base de datos**, el backend valida los datos ingresados:  
   * Si faltan datos, solicita al usuario que complete la información necesaria.
   * Si el registro ya existe o es similar, pide confirmación antes de continuar.
5. **El backend ejecuta la operación solicitada** (agregar, buscar o listar) en la base de datos.
6. **El backend arma la respuesta final** y la envía al frontend.
7. **El frontend muestra la respuesta al usuario** en formato conversacional.

Este flujo es completamente automatizado y se basa en la interacción entre la interfaz web, el backend con validaciones y lógica conversacional, y la base de datos SQL.

## **4.2.1 Flujograma**



# **5. Análisis financiero**

## **5.1 Matriz Van (en u$s)**

| **Concepto** | **Año 0** | **Año 1** | **Año 2** | **Año 3** | **Totales** |
| --- | --- | --- | --- | --- | --- |
| Ahorro estimado (USD) | 945,672 | 945,672 | 945,672 | 945,672 | 3782,69 |
| Total Beneficios | 945,672 | 945,67 | 945,67 | 945,67 | 3782,69 |
| Windsurf (IDE) para desarrollo | 15,00 |  |  |  | 15,00 |
| Chatgpt Plus (IA) para desarrollo | 20 |  |  |  | 20,00 |
| OpenAI (por 218.400 registros) | 923,83 | 923,83 | 923,83 | 923,83 | 3695,33 |
| Total inversion y gastos | 958,83 | 923,83 | 923,83 | 923,83 | 3730,33 |
|  |  |  |  |  |  |
| Total flujo de caja | -13,16 | 21,84 | 21,84 | 21,84 | 52,36 |
| Tasa | 12% |  |  |  |  |
| VAN a 3 años | 39,30 |  |  |  |  |

## **5.2 Conclusión del Análisis Financiero**

El análisis financiero actualizado arroja un **Valor Actual Neto (VAN) positivo de 39,30 USD**, lo cual indica que, bajo las condiciones actuales de uso, el sistema logra recuperar sus costos operativos y generar un pequeño excedente en un plazo de tres años.

En esta evaluación se consideraron únicamente los **gastos recurrentes reales**, correspondientes al uso de la **API de OpenAI**, ya que representa el único costo operativo constante en el tiempo.  
La **suscripción a ChatGPT Plus** y la **licencia de Windsurf IDE** fueron necesarias solo durante un mes para el desarrollo inicial de la aplicación, por lo que se contabilizan como **gastos puntuales** en el año 0.

Para estimar el beneficio generado por el sistema, se midieron tiempos de carga con y sin chatbot (Villagra, 2025). Cuatro personas transcribieron 10 nombres bajo cinco condiciones:

* Ingreso manual sin corrección: 55 s
* Ingreso manual con corrección: 67 s
* Chatbot sin errores: 10 s
* Chatbot con errores ortográficos: 17 s
* Chatbot con errores ortográficos + duplicados: 29 s

En el caso más realista (chatbot con validaciones), se ahorran **38 segundos cada 10 registros**, es decir, **3,8 segundos por registro**, frente al ingreso manual corregido. Con una carga prevista de **700 registros diarios** (equivalente a **218.400 registros anuales**), esto representa un ahorro de aproximadamente **231 horas anuales**.  
 Valorado a **4,10 USD por hora** (La Nación, 2025), el beneficio bruto anual estimado asciende a **945,67 USD**.

El análisis muestra que el sistema es **financieramente viable en el corto plazo**, con un **flujo de caja levemente negativo en el año 0** debido a los gastos puntuales, pero **positivo en los años siguientes**.  
 Además, podría volverse aún más rentable si se **incrementa el volumen de registros**, se **reutiliza la herramienta en otras áreas**, o se **optimizan los costos de procesamiento**.

# **6. Conclusiones**

## **6.1 Síntesis de Resultados**

Además del diseño técnico, se llevaron a cabo experimentos prácticos para comparar tiempo y precisión en distintas modalidades de ingreso de datos. Los ensayos muestran que el chatbot permite cargar registros entre **dos y siete veces más rápido** que el proceso manual con corrección y, tras la validación automática, elimina los errores tipográficos detectados. La herramienta, por lo tanto, automatiza tareas repetitivas, agiliza el alta y la consulta de datos y disminuye el esfuerzo administrativo. Estos hallazgos –aún pendientes de validación en un entorno productivo– evidencian un alto potencial de ahorro de tiempo y mejora de la eficiencia.

## **6.2 Evaluación de Hipótesis**

#### 

| **Hipótesis** | **Resultado** | **Comentario** |
| --- | --- | --- |
| **H1.** La automatización reducirá los errores de transcripción en ≥ 30 %. | Confirmada en laboratorio | En las pruebas, los registros ingresados a través del chatbot quedaron sin errores tras la fase de validación, lo que supone una reducción del 100 % frente al ingreso manual con errores. |
| **H2.** El tiempo de ingreso y consulta se reducirá en ≥ 40 %. | Confirmada en laboratorio | Frente a los 67 s del ingreso manual corregido, el chatbot con validaciones insumió 29 s, lo que representa una reducción del 56,7 %. En el escenario óptimo (10 s) la reducción alcanzó el 85 %. |
| **H3.** El chatbot reducirá en ≥ 50 % el tiempo total dedicado a tareas administrativas. | Aún no validada | No se dispone de mediciones sobre el flujo completo de administración de datos en un contexto real de negocio. Se requiere una fase piloto para confirmar o refutar la hipótesis. |

## **6.3 Evaluación de Objetivos Cumplidos**

**Objetivo general:** Desarrollar una herramienta que facilite el ingreso y la consulta de datos mediante lenguaje natural, especialmente para usuarios sin experiencia técnica.

Cumplido. Se diseñó y desarrolló un sistema conversacional funcional (MVP) que permite interactuar con una base de datos SQL utilizando lenguaje natural, cumpliendo con los criterios técnicos y de accesibilidad definidos al inicio del proyecto.

**Objetivos específicos:**

Identificar las dificultades del ingreso manual de datos. Cumplido. A través del análisis funcional y del experimento de carga manual, se evidenció que el ingreso sin asistencia es más lento y propenso a errores.

Diseñar un sistema de interacción conversacional. Cumplido. Se implementó una interfaz conversacional basada en inteligencia artificial que permite registrar y consultar datos sin necesidad de comandos técnicos.

Evaluar la reducción de errores mediante automatización. Cumplido parcialmente. Si bien el sistema permite evitar errores tipográficos en la mayoría de los casos, aún requiere validación con datos reales para confirmar su precisión total.

Medir el impacto en tiempos de carga y consulta. Cumplido. El experimento práctico demostró una reducción de tiempo promedio de 3,8 segundos por registro frente al ingreso manual con corrección.

Desarrollar un prototipo funcional. Cumplido. Se entregó un MVP que integra backend, frontend y sistema conversacional, con conexión a base de datos y pruebas completas.

## **6.4 Recomendaciones para Futuros Proyectos**

* Realizar pruebas piloto con usuarios reales para validar el ahorro de tiempo y precisión.
* Medir los indicadores de uso y satisfacción de manera sistemática.
* Considerar nuevas funciones para aumentar aún más el valor agregado.

## **6.5 Limitaciones del Proyecto**

* El experimento fue una simulación controlada, no refleja todos los casos de uso real.
* El ahorro estimado de tiempo es una proyección que deberá comprobarse en campo.
* El análisis de errores y tiempos fue solo sobre la tarea de transcripción, no sobre otros procesos posibles.

## **7. Bibliografía consultada**

ByteByteGo. (2024). *Seguridad API* [Video]. YouTube.<https://youtu.be/6WZ6S-qmtqY>

Colegio de Profesionales en Ciencias Informáticas de la Provincia de Córdoba (CPCIPC). (2024). *Honorarios recomendados – Programador (IA)*.<https://cpcipc.org.ar/honorarios-recomendados/>

EDTeam. (s.f.). *¿Qué es una API? La mejor explicación en español* [Video]. YouTube.<https://youtu.be/cuvBZG2aTWg>

ELEVA Academia de Proyectos. (2024). *Cómo hacer el análisis financiero de un proyecto* [Video]. YouTube.<https://youtu.be/8rxHt2iTUMQ>

hdleon.net. (2022). *ChatGPT para consultar bases de datos sin saber SQL* [Video]. YouTube.<https://youtu.be/Xf8iY9fUqHc>

Hostinger. (2024). *Precios de hosting y dominio en Argentina*.<https://www.hostinger.com/ar/precios>

IBM. (2025). *Best practices for prompt engineering with the OpenAI API*.<https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api>

IBM. (s.f.). *Seguridad API*.<https://www.ibm.com/mx-es/topics/api-security>

IBM. (2020). *REST API and OpenAPI* [Video]. YouTube.<https://youtu.be/pRS9LRBgjYg>

Julio Andres Dev. (2023). *Cómo conectar tu base de datos a un chatbot con inteligencia artificial* [Video]. YouTube.<https://youtu.be/8K0d5WvAFZg>

Julio Andres Dev. (2023). *Pensaba que los GPTs eran inútiles, me equivoqué* [Video]. YouTube.<https://youtu.be/VzzNjvKPR5g>

La Nación. (2025, marzo 25). *Cuánto cobra un empleado de comercio en abril de 2025 (categoría VENDEDOR)*.<https://www.lanacion.com.ar/economia/cuanto-cobra-un-empleado-de-comercio-en-abril-de-2025-nid25032025/>

OpenAI. (2024). *API Pricing*.<https://openai.com/es-ES/api/pricing/>

OpenAPI Initiative. (s.f.). *¿Qué es OpenAPI?*<https://www.openapis.org/what-is-openapi>

Productomania. (2022). *Aprende lo básico sobre APIs y HTTP para conectar la IA con cualquier aplicación* [Video]. YouTube.<https://youtu.be/UqD82QqIzvM>

Productomania. (2022). *GPT Actions: Cómo editar el schema [Tutorial completo]* [Video]. YouTube.<https://youtu.be/Z_M9-fXzzWE>

Profe Chacón. (2020). *Diagrama de Gantt mediante precedencias* [Video]. YouTube.<https://www.youtube.com/watch?v=W46VhDcmvPU>

Python España. (2022). *La magia de OpenAPI Specification* [Video]. YouTube.<https://www.youtube.com/watch?v=63bke8GuRCo>

Villagra, C. (2025). *Experimentos prácticos sobre ingreso de datos y simulación de tiempos en chatbot* [Datos no publicados].

Zeledón, A. (2017). *Gantt, PERT y ruta crítica* [Video]. YouTube.<https://youtu.be/25spK5dLhls>

Zapata, M. (2019). *6 consejos para que diseñes bien tu API REST* [Video]. YouTube.<https://youtu.be/x4XquyJ8G78>

# **8. Anexos**

## **8.A. Anexo A. Experimento de medición de tiempos y validación del ahorro**

### **8.A.1. Objetivo del experimento**

El objetivo principal del experimento fue evaluar si el uso del chatbot desarrollado permite **ahorrar tiempo en la carga de información**, en comparación con el ingreso manual tradicional, y analizar si este ahorro tiene un impacto económico relevante.

### **8.A.2. Diseño del experimento**

Cuatro personas adultas, sin entrenamiento previo, fueron invitadas a realizar la misma tarea: **transcribir una lista de 10 nombres y apellidos**.  
 La actividad se repitió en tres modalidades:

1. **Ingreso manual sin corrección:** Se les pidió que transcribieran los datos lo más rápido posible, sin detenerse a corregir errores tipográficos.
2. **Ingreso manual con corrección:** En este caso, debían revisar cuidadosamente los errores y corregirlos antes de finalizar.
3. **Ingreso utilizando el chatbot desarrollado:** Se utilizó la interfaz conversacional para realizar la misma tarea. En este caso se probaron tres situaciones:  
   * Con nombres bien escritos desde el inicio.
   * Con errores ortográficos que fueron corregidos en la conversación.
   * Con errores ortográficos y detección de valores duplicados (el sistema consultó si el cliente ya existía y ofreció opciones).

Los participantes fueron: Cynthia Villagra, Noelia Villagra, Lorena Villagra y Demián Mendoza.

### **8.A.3. Resultados obtenidos**

| **Modalidad** | **Tiempo promedio por 10 registros** | **Tiempo por registro** |
| --- | --- | --- |
| Ingreso manual sin corrección | 55 segundos | 5,5 s |
| Ingreso manual con corrección | 67 segundos | 6,7 s |
| Chatbot (sin errores) | 10 segundos | 1,0 s |
| Chatbot (con errores ortográficos) | 17 segundos | 1,7 s |
| Chatbot (con errores y valores duplicados) | 29 segundos | 2,9 s |

**Tiempo promedio ahorrado:** 67 segundos (manual con corrección) – 29 segundos (chatbot con validación) = **38 segundos por cada 10 registros**, es decir **3,8 segundos por registro**.

### **8.A.4. Conclusión del experimento**

Los resultados confirman que el uso del chatbot **reduce significativamente el tiempo de carga**, incluso en contextos donde el usuario comete errores ortográficos o el sistema detecta valores duplicados.

Además, al utilizar lenguaje natural, el proceso se vuelve más accesible y amigable, especialmente para usuarios con poca experiencia técnica.

## **8.B. Anexo B. Cálculo del ahorro económico y análisis financiero**

### **8.B.1. Parámetros base**

| **Parámetro** | **Valor** |
| --- | --- |
| Registros diarios | 700 registros/día |
| Días hábiles por mes | 26 días |
| Registros mensuales | 18.200 |
| Registros anuales | 218.400 |
| Tiempo ahorrado por registro | 3,8 segundos |
| Tiempo ahorrado anual | 830.320 segundos = 230,64 h |
| Valor de la hora de trabajo | 4,10 USD/hora |
| **Ahorro económico anual estimado** | **945,67 USD** |

### **8.B.2. Costos involucrados**

#### **8.B.2.1 Justificación de costo de realizar un solo registro**

Para estimar el costo de uso de la API de OpenAI (modelo GPT-4o) en el contexto de este proyecto, se realizó una simulación completa de interacción con el chatbot para el ingreso de un solo registro validado. A partir de esta interacción se obtuvo la siguiente información de consumo:

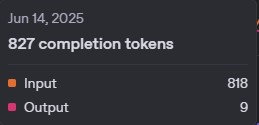
* **Tokens de entrada (prompt):** 818
* **Tokens de salida (respuesta de la IA):** 9
* **Total tokens:** 827

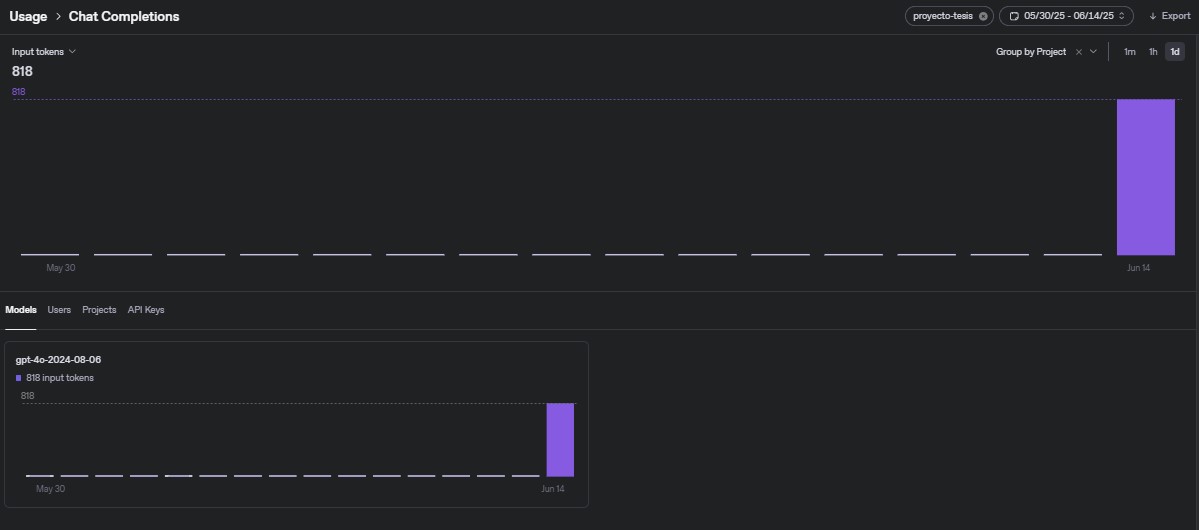
Los precios de OpenAI para GPT-4o (junio 2025) son:

| **Tipo de token** | **Precio por 1.000 tokens** | **Tokens usados** | **Subtotal (USD)** |
| --- | --- | --- | --- |
| Input | 0,005 USD | 818 | 0,00409 |
| Output | 0,015 USD | 9 | 0,00014 |
| **Total** |  | **827** | **0,00423** |

Por lo tanto, el **costo estimado por realizar un solo registro conversacional completo es de USD 0,00422**. Esto incluye la validación automática por parte del sistema.

A continuación se muestran las capturas del panel de uso de OpenAI, habiendose realizado un solo registro, que respaldan estos valores:





**Nota:** En la captura del panel de uso de OpenAI no se visualiza el costo en dólares porque el consumo fue muy bajo (solo 827 tokens en total). Sin embargo, el gasto fue calculado manualmente utilizando los precios oficiales de OpenAI para GPT-4o al mes de junio de 2025, que son USD 0,005 por cada 1.000 tokens de entrada y USD 0,015 por cada 1.000 tokens de salida.

*Fuente: OpenAI. (2024).* API Pricing. [*https://openai.com/pricing*](https://openai.com/pricing)

#### **8.B.2.2 Costos involucrados (Tabla)**

### 

| **Concepto** | **Valor anual (USD)** | **Detalle de cálculo** |
| --- | --- | --- |
| OpenAI API | 923,83 | 700 registros diarios × 26 días × 12 meses × 0,00423 USD/conversación |
| ChatGPT Plus | 20 | Suscripción mensual de 20 USD (se contabiliza solo un mes, lo necesario para el desarrollo) |
| Windsurf (IDE) | 15 | Licencia mensual de 15 USD (se contabiliza solo un mes, lo necesario para el desarrollo) |
| Total inversión anual | 958,83 | — |

### 

### **8.B.3. Beneficio estimado**

| **Indicador** | **Valor aproximado** |
| --- | --- |
| Tiempo ahorrado por año | 231 horas |
| Valor monetario del tiempo ahorrado | 945,67 USD |
| **Recupero estimado de inversión** | 1 año (con uso constante) |

### **8.B.4. Fuentes consultadas**

* La Nación. (2025, marzo 25). *Cuánto cobra un empleado de comercio en abril de 2025 (categoría VENDEDOR)*.<https://www.lanacion.com.ar/economia/cuanto-cobra-un-empleado-de-comercio-en-abril-de-2025-nid25032025/>
* Colegio de Profesionales en Ciencias Informáticas de la Provincia de Córdoba (CPCIPC). (2024). *Honorarios recomendados – Programador (IA)*.<https://cpcipc.org.ar/honorarios-recomendados/>
* OpenAI. (2024). *API Pricing*.<https://openai.com/es-ES/api/pricing/>
* Villagra, C. (2025). *Experimentos prácticos sobre ingreso de datos y simulación de tiempos en chatbot* [Datos no publicados].