# **Informe Técnico**

**Proyecto:** Asistente Inteligente para la Tienda Online Level-Up Gamer

## **1. Introducción**

Level-Up Gamer es una tienda online chilena especializada en productos para gamers: consolas, computadores, accesorios, merchandising y servicio técnico. Opera únicamente en línea, despachando a todo Chile, y busca consolidarse como la tienda gamer líder del país, ofreciendo un programa de fidelización y fomentando una comunidad activa.

Actualmente, la tienda enfrenta desafíos relacionados con la atención al cliente, debido al aumento de consultas sobre disponibilidad de productos, políticas de envío, garantías y beneficios del programa de fidelización. Esto genera tiempos de respuesta lentos y limita la autogestión de los clientes.

La solución propuesta consiste en un **asistente inteligente** basado en **LLM (Large Language Model)** y **RAG (Retrieval-Augmented Generation)**, implementado con **Streamlit y LangChain**. Este asistente permite ofrecer respuestas rápidas, confiables, trazables y evaluables mediante métricas automáticas de fidelidad y relevancia.

## **2. Análisis del Caso**

**Problemas identificados:**

* Altos tiempos de respuesta ante consultas frecuentes (>24h).
* Abandono de carritos estimado en 30%.
* Bajo aprovechamiento del programa de fidelización y participación en la comunidad.
* Consultas recurrentes sobre servicio técnico, garantías y estado de pedidos.

**Impacto:**

* Pérdida de oportunidades de venta.
* Sobrecarga del equipo de soporte en consultas repetitivas.
* Reducción de la satisfacción del cliente y fidelización.

**Objetivos específicos:**

1. Reducir el tiempo de respuesta de consultas a menos de 1 minuto.
2. Automatizar al menos el 70% de preguntas frecuentes sobre productos, despacho, garantías y fidelización.
3. Incrementar en un 15% la tasa de conversión mediante integración con el programa de fidelización.
4. Mantener un 100% de trazabilidad de fuentes en cada respuesta.

**Datos disponibles para el proyecto:**

* Catálogo de productos (categorías, precios, descripciones).
* Políticas de despacho, devoluciones y garantías.
* Información sobre programa de fidelización y comunidad gamer.
* Contenido educativo y blogs relacionados con gaming.

## **3. Diseño de Prompts y Contextos**

Los prompts y documentos se gestionan de forma dinámica en Streamlit y LangChain.

* **Prompts principales:**
  + **Consulta de catálogo:** mostrar productos con nombre, precio y breve descripción, incluyendo fuente.
  + **Fidelización y comunidad:** explicar puntos y beneficios de LevelUp, con referencias.
  + **Servicio técnico y garantías:** detallar procedimientos y tiempos de atención.
  + **Evaluación de respuestas:** prompts automáticos que calculan fidelidad, relevancia y precisión del contexto.
* **Gestión de documentos:**
  + Textos de catálogo, políticas, blogs y casos de servicio técnico.
  + Agregados, editables y eliminables directamente desde la interfaz de Streamlit.
  + Embeddings generados con **LangChain** usando text-embedding-3-small.

## **4. Arquitectura de la Solución**

**Componentes implementados:**

* **Interfaz web (Streamlit):** permite realizar consultas, administrar documentos y visualizar métricas.
* **Client LLM (OpenAI API):** genera respuestas basadas en contexto recuperado.
* **Vector DB virtual:** embeddings almacenados en memoria (np.array) para búsqueda semántica.
* **Retriever híbrido:** combina similitud semántica (cosine similarity) y coincidencia de palabras clave.
* **Post-proceso:** evaluación de fidelidad, relevancia y precisión del contexto mediante prompts de LLM.
* **Logging:** registro de interacciones y métricas para exportación a CSV o LangSmith.

**Flujo simplificado:**

1. Usuario realiza consulta → Streamlit recibe input
2. Embeddings generan representación semántica → Búsqueda híbrida en documentos
3. Contexto relevante → LLM genera respuesta
4. Evaluación automática → Métricas de fidelidad, relevancia y precisión
5. Respuesta al usuario + log de métricas

## 

## **5. Implementación y Pruebas**

**Prototipo funcional:**

* Indexación de 10–12 documentos iniciales de catálogo, políticas y casos de servicio técnico.
* Consultas de prueba sobre:  
  + Disponibilidad y características de productos.
  + Procedimientos de devolución y despacho.
  + Beneficios de fidelización y participación en la comunidad.
* Métricas automatizadas:  
  + **Fidelidad:** 1–10, verifica si la respuesta se basa en contexto.
  + **Relevancia:** 1–10, evalúa pertinencia de la respuesta.
  + **Precisión del contexto:** proporción de documentos relevantes recuperados.
* Exportación de métricas y logs en **JSON o CSV**.

## **6. Justificación Técnica**

* **Confiabilidad:** todas las respuestas se fundamentan en documentos internos, con trazabilidad completa.
* **Escalabilidad:** incorporación rápida de nuevos productos o políticas a través de la interfaz de documentos.
* **Evaluación continua:** métricas automáticas permiten medir desempeño del asistente y ajustar modelos o documentos.
* **Reducción de carga operativa:** soporte humano solo interviene en casos no cubiertos por el asistente.

El sistema permite evolucionar hacia:

* Integración con sistemas de recomendación personalizados.
* Automatización de consultas en múltiples canales (WhatsApp, redes sociales).
* Potenciación de la comunidad gamer y engagement con el programa de fidelización.

## **7. Conclusiones**

La implementación del asistente inteligente basado en **LLM + RAG**, gestionado con Streamlit y LangChain, permitirá:

* Reducir tiempos de respuesta y mejorar la autogestión de clientes.
* Automatizar consultas frecuentes sobre productos, despacho, garantías y fidelización.
* Evaluar sistemáticamente la calidad de las respuestas con métricas de fidelidad y relevancia.
* Escalar la solución a todo el catálogo, servicios técnicos y nuevos canales digitales.