

**ALUMNO:** Simon Azul Sanchez Vottero; **DNI:** 39.173.576

**REPOSITORIO:** [https://github.com/simonbleu/Practica\\_Profesionalizante\\_I---TSCDIA](https://github.com/simonbleu/Practica_Profesionalizante_I---TSCDIA)

**CARRERA:** Tecnicatura Superior en Ciencia de datos e Inteligencia Artificial

**COMISIÓN:** A; **DOCENTE:** Dianela Accietto

---

## **A) Investigación Profunda del Caso**

En primer lugar debería haberse analizado el proyecto en sí mismo, ya que el presupuesto de desarrollo, sin contar con el mantenimiento del software ni mucho menos las pérdidas incurridas, podría potencialmente haber cubierto empleados, capacitaciones, marketing y/o sistemas quizá más rígidos pero igualmente útiles. Casos como la gestión de turnos, no requieren un sistema tan complejo

En cuanto a las causas del estrepitoso fallo del caso en particular, se observan fallas organizativas (análisis, organización e interacción de equipos y comités profesionales, presupuesto, toma de decisiones como el escalado prematuro, con poca capacitación y retroalimentación con clientes y admisión y helpdesk.), técnicas (problemas de integración con sistemas existentes, de seguridad con la base de datos, de carga sin contingencias aparentes como “timeouts”, de prompting y tuning del modelo) y humanas (falta de supervisión, de pruebas de usabilidad en especial en adultos mayores, de retroalimentación, y de personalización que eviten un tratamiento impersonal ante situaciones delicadas)

En el plano normativo, la implementación debería haberse alineado con la Ley 25.326 de Protección de Datos Personales[1] (vulnerados con el uso aparentemente irrestricto de información sensible de la LLM) y a la Resolución 282/2020 de Telemedicina en Argentina[2] (si bien es una ley más reciente, claramente no cumple con los estándares exigidos) y tampoco se alinea con los estándares internacionales ISO/IEC 27001[3][4] e ISO/IEC 27701[5] (protección de información sensible y personal) ni IEEE 7010[6] (estándar de desempeño ético de la IA). La ausencia de cumplimiento regulatorio no sólo implica riesgos legales, sino que aumenta la probabilidad de fallos críticos.

Entre las pérdidas observadas, esta el costo nominal del proyecto en sí y que puede afectar futuros proyectos o reestructuración del existente, el desgaste del personal ante la mayor carga laboral la cual implica un mayor gasto ya sea en personal de respaldo o nuevo personal a entrenarse en caso de renuncias y el costo de reputación afectando la cantidad de clientes y su predisposición. También existe un costo de oportunidad perdido, ya que el fallo de la empresa puede incentivar a la competencia a capitalizar en la experiencia y ofrecer un mejor servicio, limitando la cuota de mercado a captar en un futuro intento.

Se destacan entre sus stakeholders a los pacientes, y en especial aquellos de edad avanzada, que buscan un acceso rápido y claro, con un lenguaje más humano y la opción de hablar con uno; De la administración, cuya prioridad debería ser aumentar la seguridad del paciente y reducir la carga en su personal, manteniéndose dentro de plazos y presupuestos razonables, y siempre de manera ética y bajo estándares normativos; Del personal de la clínica, interesados en la eficacia y fiabilidad del programa con respecto a la clasificación y pre-diagnóstico; Y del departamento de IT, cuyas prioridades serían la seguridad e integridad de los datos, el rendimiento, la integración, la escalabilidad y portabilidad, la trazabilidad, etc para poder mantener el programa correctamente.

Los mayores puntos de inflexión observados son:

- Haber ignorado los errores técnicos y fallos de integración con el MVP y la beta, decidiendo en cambio el escalado inmediato
- No activar contingencias automáticas o mediante supervisión para derivar pacientes ante el primer incidente observado
- Mantener la escala y enfoque durante meses aun con todos los fallos en vez de segmentar, delimitar y monitorear al programa tras la alta tasa de quejas

Y por lo tanto concluyó que el eslabón más débil y causante del escalado del fallo ha sido el sector gerencial, seguido, de existir como tal, de los departamentos de testing y el PO, potencialmente por no haber transmitido o identificado debidamente las falencias del proyecto a la gerencia. Aun ante una hipotética inevitabilidad del fallo del proyecto, el impacto negativo fue absolutamente evitable.

## B) ¿Qué Pudo Prevenirse?

- Validación técnica: El proyecto debería haber captado inmediatamente las incompatibilidades con el HIS tras la implementación del MVP y nunca debería haber continuado tras una beta fallida. Tanto el testeo técnico como la validación por parte de un comité médico han sido deficientes, ausentes o ignorados.
- Pilotos escalonados: El proyecto pasó de una beta (fallida) directamente a un despliegue completo, cuando una buena práctica hubiera sido integrarlo progresivamente junto a un exhaustivo control de calidad. Idealmente habría variación y rotación con múltiples pilotos basados en retroalimentación
- Derivación: Aun si el modelo utilizado fuera incapaz de tomar la decisión de manera acertada, y de hecho aun si fuera capaz de ello, debería haber un protocolo automático de salvaguardas que derivaran la conversación a un representante humano, por ejemplo tras cierta cantidad de interacciones, palabras claves o minutos (lo que suceda primero). Como mínimo, tras un “trigger” como tal debería asignarse a un supervisor.
- Privacidad: Los datos confidenciales no solo deberían estar encriptados ante ataques externos, sino también del mismísimo modelo de IA, la cual debería trabajar con un cierto rango de información, intercedido por una base intermedia que encapsule esta información y así evitar filtraciones accidentales y alucinaciones ofensivas.
- Respaldo: Ya sea que se utilice un sistema de “cola” o asincronía, debería haber un sistema secundario que “desagote” al principal, tomando entre medio de la interacción con el usuario pasos mucho más estructurados cuando no sea necesaria mayor sofisticación o cuando no haya suficientes recursos. Un ejemplo claro de esto es la toma de datos, que podría utilizar una lista de mensajes y formularios “enlatados”.

## C) ¿Qué NO Pudo Prevenirse?

- Limitaciones de las LLM's: Por impresionantes y sofisticadas que puedan ser, los modelos de lenguajes carecen de cualquier noción real de contexto y se limitan a utilizar probabilidad para predecir texto. Aun de haber contando con memoria persistente y una personalización mucho más efectiva, las LLM son propensas a “alucinar” respuestas, por lo tanto, debería haberse utilizado el modelo como una capa de “naturalización” para un sistema mucho más estructurado, derivando las complejidades directamente a un humano y escalando paulatinamente.

- Resistencia generacional: Parte de la población, y en especial aquella de edad avanzada, siempre demostrará un mayor recelo ante nuevas tecnologías y en especial aquellas sin contacto humano – y no es único al contexto dado, incluso se puede observar en las cajas de autoservicio del supermercado. Este factor en ocasiones está ligado a una reticencia al cambio, o aspectos emocionales, o una mayor curva de aprendizaje.
- Presupuesto limitado: Debido a la escala y el tiempo de desarrollo, el presupuesto claramente ha tenido que “tomar atajos”, ya sea en el tamaño del equipo, su idoneidad, o el alcance de las herramientas utilizadas. Como tal, considero que el proyecto fue demasiado ambicioso.
- Impacto reputacional: Con un mismo producto dado, si bien se puede influenciar la recepción con técnicas de marketing, es imposible controlar la reacción de los clientes y las interacciones entre ellos, ni prevenir una explotación mediática de la indignación tras cruzar un umbral crítico de quejas y exposición. A ese punto debería haberse cambiado la estrategia a una de control de daños.
- Limitaciones del ecosistema tecnológico: Uno no tiene control sobre el estado de las APIs y la infraestructura a la que el hospital y los pacientes tienen acceso, lo cual puede afectar el servicio.

#### **D) Cómo Trabajarías en el Futuro**

1. Análisis (a priori):
  - 1.1. De requerimientos: Un comité integral con personal técnico, hospitalario y POs debería identificar en primer lugar cuales son las necesidades
  - 1.2. De viabilidad: Tanto técnica, como cultural, de seguridad y presupuestaria (económica y temporal). Si alguna falla, debe volverse al punto anterior y reducir los requerimientos.
  - 1.3. De mercado: La teoría y la práctica no siempre van de la mano. Debe analizarse al cliente real y a la competencia para evitar gastos y riesgos innecesarios.
  - 1.4. De riesgos: Una matriz de riesgos me parece algo fundamental para identificar el rumbo del desarrollo del proyecto
2. Desarrollo y despliegue:
  - 2.1. Escalonado: El software debería ser lanzado por partes, con funciones y público limitado, variaciones nacidas tanto del diseño como la retroalimentación, y con supervisión humana. También debería existir un proceso en paralelo (no visible al paciente) al supervisor humano para contrastar respuestas y mejorar el modelo o incluso el sistema.
  - 2.2. Salvaguardas: Establecer criterios mínimos para pasar a la siguiente fase del programa, y a la vez establecer los “triggers” que deriven automáticamente en un supervisor humano
  - 2.3. Limitaciones: La LLM debe estar “anclada” a ciertos parámetros para evitar mayores desviaciones y reducir alucinaciones, y solo tener acceso indirecto a información sensible
  - 2.4. Modularización: El software en cuestión debería poder ser escalado, truncado, trasladado o mutado sin mayores problemas en caso de cambios en las demandas, en la tecnología, en la gestión del hospital, en los sistemas

- secundarios/auxiliares y bases de datos, etc. Para ello se requiere planear y desarrollar las contingencias con previsión.
- 2.5. Capacitación: Tanto del personal para aumentar su efectividad con la herramienta, como de alfabetización de pacientes para aumentar la adopción y reducir preventivamente quejas relacionadas al uso.
3. Testing
- 3.1. De gestión interna: Considero importante establecer simulacros de presión que permitan al equipo conocerse entre sí y saber qué esperar del otro. Estos escenarios hipotéticos no deberían llevar demasiado tiempo, pero pueden resultar invaluable
- 3.2. De integración: Primordial que todas las partes del sistema interactúan entre sí de la manera esperada y de manera uniforme. En especial con las BBDD del hospital – Siempre con datos sintéticos para no exponer pacientes
- 3.3. De rendimiento: El programa debe funcionar de manera aceptable aun bajo “presión”, y deben establecerse contingencias funcionales ante cualquier eventualidad
- 3.4. De seguridad: Tanto para cumplir estándares internos como normativos, las vulnerabilidades del programa, de existir, deben ser identificadas con celeridad.
- 3.5. De usabilidad: En particular con aquellos grupos más propensos a tener problemas con el sistema, como la gente mayor o personas que requieran alguna nueva configuración de accesibilidad en la UI
4. Análisis (a posteriori):
- 4.1. Auditorías: Periódicamente, generando informes sobre el desempeño de la aplicación respecto a los criterios funcionales, de seguridad y éticos determinados previamente.
- 4.2. Métricas y retroalimentación: La evolución del programa debe ser constante y dirigida por aquellos utilizándolo diariamente (pacientes y profesionales), y por lo tanto se debe recabar información implícita y explícitamente de manera constante.

## **E) Lecciones Aprendidas Personales**

- Ética y seguridad ante todo: No deben tomar atajos que vulneren ni a los pacientes ni al hospital. Contemplación y supervisión, ambas son indispensables.
- La IA no siempre es la respuesta: Es una herramienta poderosa y versátil, pero no infalible ni falta de controversia. En este caso hubiera sido más efectivo un uso más circunspecto, y siempre con un humano dentro del circuito
- Integrar gestión y trazabilidad: Debe haber siempre una comunicación lo suficientemente abierta en donde los datos obtenidos del desarrollo y despliegue de la aplicación no sean estáticos si no que influyan en la toma de decisiones.
- Siempre hay una persona detrás del cliente: A veces pueden obviarse factores humanos como la falta de alfabetización digital y de acceso a la tecnología, discapacidades y condiciones que requieran consideraciones especiales de accesibilidad y sensibilidades que puedan causar una respuesta emocional negativa.
- Nunca sacrificar validación por velocidad: El costo de fallos como el ejemplificado son mucho mayores al costo de oportunidad perdido por un despliegue más tardío.

## **F) Recomendaciones Estratégicas**

La clínica requiere un “rebranding” inmediato (potencialmente de manera literal – nombre – en el caso más extremo, para evitar asociaciones indeseadas) con un enfoque de marketing proactivo en donde se intente reconstruir la confianza mediante atención personalizada y transparencia, haciendo énfasis en un storytelling positivo y certificaciones externas. De considerarse necesario y viable, establecer alianzas estratégicas con otras clínicas para reducir el tiempo de traspaso de una a la otra; Dependiendo del costo de mantenimiento y reestructuración del proyecto inicial, debería destinarse el mismo para uso interno o no crítico (turnero, administración, redacción y resúmenes para el profesional, etc) y con control mucho más milimétrico y despliegue acotado.

A la vez, si se considera recomendable continuar con el proyecto, los dos mayores cambios a posteriori que pueden y deberían aplicarse son los de validación (médica, POs y testing) y el cambio de alcance del proyecto por uno que goce de mayor tolerancia al fallo.

Es recomendable capacitar al personal para que pueda comprender el funcionamiento de la aplicación, no solo su uso, “auditar” al programa (y al hospital, de parte organismos reguladores que garanticen estándares éticos) de manera rutinaria para identificar potenciales problemas y falsos positivos/negativos, y establecer un sistema de métricas relevantes, como lo es la satisfacción, duración, eficacia (con múltiples parámetros como las derivaciones, naturaleza de la consulta y correcta identificación así como las desviaciones, etc)

También se recomienda ahondar en casos similares en donde la aplicación de software similar ha fallado – como lo fue el caso de IBM Watson Health[7][8], plataforma de IA de IBM destinada a la oncología que enfrentó críticas por ofrecer recomendaciones de tratamiento inconsistentes y, en algunos casos, peligrosas. Esto se atribuyó a una falta de validación clínica rigurosa y a la dependencia de datos no representativos.– y otras de éxito – como Aidoc[9], una plataforma de IA que ayuda a los radiólogos a identificar hemorragias cerebrales y otras condiciones críticas en imágenes médicas, reduciendo significativamente el tiempo de diagnóstico.