IMMUNE
TECHNOLOGY INSTITUTE

Memoria de Capstone Project

Plataforma Integrada de Análisis y Consulta Inteligente para la Gestión de Datos Académicos en IMMUNE

IA & DATA SCIENCE FOR BUSINESS
OCT 24/DIC 25

Sergio Martín Álvarez
Pablo Zafra-Polo Borrego

IMMUNE Technology Institute
Paseo de la Castellana 89,
28046 Madrid

ÍNDICE

1. INTRODUCCIÓN:

Este proyecto se plantea como un punto de partida sólido para que IMMUNE pueda disponer de una infraestructura centralizada de análisis de datos. Dado que no contábamos con los datos reales del instituto —más allá de los resultados de encuestas de satisfacción que recibíamos como alumnos— fue necesario generar *datasets sintéticos completos y coherentes* que permitieran diseñar la solución end-to-end. La idea es que, en el futuro, estos datos sintéticos puedan sustituirse directamente por los datos reales del centro sin necesidad de rehacer el pipeline.

El objetivo fundamental es crear un *centro de control de datos* que facilite la comprensión y explotación de la información generada por la escuela: interacción de los alumnos, resultados de satisfacción, métricas de uso de la web y comportamiento de los usuarios que visitan la página, completan un formulario o se matriculan. Sobre esta base, se desarrollan también estrategias de *clustering* orientadas a mejorar la segmentación y optimización de las campañas de marketing, permitiendo identificar patrones por edad, IP, nacionalidad o procedencia profesional. Esto ofrece a la escuela una visión más clara de qué perfiles están mostrando interés y qué perfiles acaban convirtiendo, generando así información valiosa para la toma de decisiones.

El proyecto se articula en varios bloques: construcción de datasets sintéticos, creación de dashboards interactivos, desarrollo de un agente RAG entrenado con ADK para consultar los feedbacks de los alumnos, y aplicación de técnicas de clustering sobre los datos de interacción web y formularios completados. Finalmente, se plantea como línea futura el desarrollo de un modelo predictivo capaz de estimar la probabilidad de que un usuario con ciertas características llegue a matricularse.

En conjunto, la memoria describe la motivación del proyecto, los objetivos perseguidos, la metodología empleada y la estructura final de la solución, situando el trabajo dentro del marco de proyectos de análisis y explotación de datos aplicados al ámbito académico.

2. ESTADO DEL ARTE:

En los últimos años, la adopción de modelos de lenguaje y sistemas de recuperación aumentada (RAG) se ha consolidado como una solución eficaz para integrar conocimiento estructurado y no estructurado dentro de organizaciones educativas. Diversos trabajos han mostrado cómo los agentes conversacionales basados en IA pueden facilitar la consulta de grandes volúmenes de datos, automatizar respuestas y mejorar la accesibilidad a la información institucional. En este contexto, el ecosistema Azure AI Developer Kit (ADK) se ha posicionado como una herramienta relevante para construir agentes personalizados que combinan bases vectoriales, flujos de inferencia y conexión directa con bases de datos académicas o administrativas.

El uso de ADK permite implementar arquitecturas donde los datos internos —como feedback de alumnos, documentación institucional o métricas de interacción— pueden indexarse y consultarse de forma natural mediante lenguaje conversacional. Estudios recientes sobre plataformas RAG aplicadas al sector educativo destacan su potencial para centralizar el conocimiento, mejorar la calidad de las consultas y reducir el tiempo de acceso a la información crítica.

En paralelo, la analítica educativa ha incorporado de forma creciente técnicas de *machine learning* orientadas a segmentación, recomendación y detección de patrones de comportamiento. Entre estas técnicas, los métodos de clustering son especialmente relevantes para identificar perfiles de usuarios y comprender cómo interactúan con una plataforma formativa. Algoritmos como *K-Means*, *DBSCAN* o *Gaussian Mixture Models* se emplean habitualmente para detectar grupos latentes a partir de variables demográficas, patrones de navegación, tasas de conversión o progresión académica. La literatura señala que estos enfoques permiten optimizar estrategias de marketing, mejorar la captación de estudiantes y diseñar intervenciones más personalizadas.

Este proyecto se sitúa en la intersección de ambas líneas: integra un agente conversacional RAG basado en ADK para la consulta de datos de feedback y, al mismo tiempo, aplica técnicas de clustering para analizar el comportamiento de los usuarios que visitan la web de IMMUNE, completan formularios o acaban matriculándose. La contribución principal reside en combinar estos componentes dentro de una arquitectura unificada y fácilmente desplegable, diseñada para sustituir datos sintéticos por datos reales sin modificar el pipeline subyacente.

3. DESARROLLO:

El proyecto se articula en cuatro componentes principales: la generación de datos sintéticos, el desarrollo del dashboard en Power BI, la construcción del agente conversacional basado en ADK y la aplicación de técnicas de clustering orientadas a análisis y futura predicción.

Cada bloque responde a una parte concreta del objetivo general: disponer de una plataforma unificada que permita a IMMUNE analizar sus datos académicos y operativos de forma centralizada.

3.1 Generación de datos sintéticos

Dado que el acceso a los datos reales del instituto estaba limitado, el primer paso fue la creación de un conjunto de datasets sintéticos que reprodujeran la estructura, lógica interna y patrones esperables de los datos institucionales. Estos datasets permiten diseñar y validar un pipeline completo que, en el futuro, podrá funcionar de manera idéntica sustituyendo únicamente las fuentes sintéticas por datos reales.

Los principales archivos generados fueron:

- **Immune_metricas.csv**: métricas de interacción de usuarios en la web.
- **Formularios.csv**: datos de usuarios que completan o no el proceso de registro.
- **Cursos_Immune.csv**: catálogo de cursos, utilizado para relaciones y asignaciones internas.
- **Feedbacks.csv**: dataset sintético de encuestas de satisfacción, el más complejo del sistema.

Generación del dataset Feedbacks.csv

Este dataset contiene más de 1.200 encuestas sintéticas correspondientes al periodo 2023–2025. Su construcción no fue aleatoria: se diseñó un sistema que replicara comportamientos reales observados en encuestas académicas, siguiendo criterios de coherencia lógica, correlación interna y consistencia narrativa en los comentarios.

Estructura y variables

Incluye identificadores únicos (Id_encuesta, id_usuario, Id_curso), información contextual (tipo_clase, fecha) y un conjunto de valoraciones numéricas en escala 1–5 sobre profesor, metodología, contenidos y soporte. También incorpora variables categóricas con respuestas del tipo "Pésimo" a "Genial", similares a las devueltas por Google Forms en los cuestionarios reales de la escuela.

Relaciones internas

Los valores no se generan de forma independiente; se asigna a cada encuesta un *perfil de alumno*:

- *Superfans* (30%): predominio de valoraciones 4–5.
- *Críticos* (10%): predominio de valoraciones 1–2.
- *Equilibrados* (60%): distribución más realista y moderada.

A partir de estas valoraciones, la variable satisfaccion_general se calcula mediante la media de los campos clave más un pequeño ruido aleatorio. Esto imita el comportamiento estadístico real de encuestas de satisfacción, donde las variables suelen presentar alta correlación entre sí (efecto halo).

Además, se aplican condiciones lógicas: por ejemplo, la calidad de conexión solo se genera para cursos en modalidad online o híbrida.

Comentarios coherentes y únicos

Se implementó un sistema avanzado de asignación de comentarios para evitar incoherencias típicas de datasets sintéticos. Cada comentario está etiquetado con metadatos (polaridad, rango de satisfacción, temática) y se selecciona únicamente si encaja con las puntuaciones de esa encuesta. De este modo:

- No existen contradicciones entre el texto y las valoraciones numéricas.
- No se repite ningún comentario.
- El tono y la polaridad reflejan con fidelidad el nivel de satisfacción del alumno.

Este enfoque garantiza que el dataset sea útil tanto para análisis descriptivo como para modelos futuros de NLP, clasificación o predicción.

Generación del dataset Formularios.csv

El dataset de *formularios* constituye la base para entender el comportamiento y las características demográficas y profesionales de los usuarios que muestran interés por la oferta formativa de IMMUNE. Su función principal es describir el perfil de los leads: personas que han visitado la web, han iniciado un registro y han completado un formulario con sus datos, independientemente de si finalmente se matriculan o no. Este dataset será posteriormente clave para los análisis de clustering y para el diseño del futuro modelo predictivo de conversión.

Estructura general

El archivo formularios.xlsx consolida la información relevante de estos usuarios, generando un total de 700 registros sintéticos basados en patrones realistas. Cada registro está conectado mediante la primary key id_usuario, compartida con el resto de datasets del proyecto para garantizar coherencia e integridad relacional.

Las principales variables incluidas son:

- Identificador
    - *id_usuario* (formato U0001...): clave primaria que permite enlazar el formulario con métricas de navegación y con datos de satisfacción cuando procede.
- Variables demográficas
    - Edad
    - Género
    - País
    - Ciudad
- Variables profesionales
    - Titulación académica
    - Experiencia laboral
    - Sector laboral
- Intereses y motivaciones
    - Área de interés formativo
    - Motivo principal de la formación (mejora profesional, cambio de sector, actualización de habilidades, etc.)

Lógica de generación y relaciones internas

La creación de este dataset no se basa en selección aleatoria, sino en un conjunto de reglas que buscan reproducir distribuciones verosímiles y patrones coherentes con el comportamiento esperado de los leads de una escuela tecnológica.

1. **Coherencia entre edad y experiencia laboral**
   Se aplica una regla estricta:

   Años de experiencia < Edad − 18
   Garantiza que no aparezcan registros imposibles (ej. personas de 22 años con 10 años de experiencia).

2. **Distribución demográfica controlada**
   Se fuerza una distribución de edades representativa del público típico de formaciones tecnológicas:

   - 60% entre 18 y 30 años
   - 30% entre 30 y 40 años
   - 10% mayores de 40
     Esto permite que el clustering posterior tenga una señal demográfica clara.

3. **Mapeos geográficos realistas**
   Las ciudades se asignan en función del país correspondiente, evitando combinaciones incoherentes (ej. “Bogotá – España”).

4. **Mapeos profesionales dependientes del área de interés**
   El sector laboral se ajusta según el área de interés del usuario.
   Ejemplos:

   - Interés en Data Science → mayor probabilidad de venir de ingeniería, informática o consultoría.
   - Interés en Ciberseguridad → perfiles técnicos o administrativos con motivación de reconversión.
   - Interés en UX/UI → sectores creativos o marketing digital.

De este modo, las combinaciones de variables reflejan perfiles típicos del alumnado potencial de IMMUNE, permitiendo un análisis posterior más sólido y evitando resultados artificiales o estadísticamente inconsistentes.

Creación del dataset de Immune_metricas.csv

El dataset Immune_metricas.csv representa el componente digital del proyecto: simula la actividad de usuarios que visitan la web de IMMUNE y registra cómo interactúan con ella. Su función es actuar como un *log web sintético*, imprescindible para analizar patrones de navegación, reconstruir el embudo de conversión y alimentar los modelos de clustering orientados a comportamiento digital.

El archivo contiene aproximadamente 5.000 registros generados entre 2024 y finales de 2025, diseñados para reproducir tanto visitas únicas como retornos de un mismo usuario. De este modo se pueden identificar comportamientos repetitivos, fuentes de tráfico relevantes y señales que podrían influir en una futura matriculación.

Estructura general

Las variables incluidas reflejan los elementos esenciales de un sistema de analítica web:

- Variables de sesión
    - *usuario_temp*: identificador temporal único por sesión
    - *IP_usuario*: dirección IP asignada a la sesión
    - *fecha_hora*: marca temporal completa
- Variables del usuario
    - *id_usuario*: aparece cuando el usuario está registrado; permanece vacío en visitas anónimas
- Contexto de navegación
    - *origen_plataforma*: LinkedIn, Google, tráfico directo, redes sociales.
    - *dispositivo*: desktop, móvil o tablet
    - *localizacion*: país y ciudad estimados
- Comportamiento digital
    - *tiempo_en_pagina*
    - *programa_oferta_click*: curso o producto formativo visitado
    - *matriculado*: indicador final del embudo de conversión

Lógica de generación y relaciones internas

El dataset no se genera de forma aleatoria. Cada registro responde a una serie de reglas diseñadas para reproducir el funcionamiento de un entorno de captación real en una escuela tecnológica.

1. **Integración con el dataset de Formularios**
   El script importa la lista de *id_usuario* procedente de formularios.xlsx.
   - Si un visitante tiene *id_usuario*, su localización debe coincidir con la registrada en su formulario.
   - Esto garantiza consistencia entre el comportamiento digital y el perfil demográfico/profesional.

2. **Simulación del embudo de conversión**
   Solo un usuario identificado (*id_usuario* ≠ vacío) puede figurar como *matriculado*.

   La probabilidad base de conversión es del 32%, pero se ajusta mediante reglas de negocio:
   - Tráfico desde LinkedIn → mayor probabilidad de matriculación.
   - Dispositivo Desktop → comportamiento más propenso a registro y conversión.
     Este enfoque permite analizar qué canales generan perfiles de mayor valor.

3. **Modelado del tiempo en página**
   Para reproducir comportamientos reales:
   - Usuarios registrados → sesiones más largas (entre 2 y 15 minutos).
   - Usuarios anónimos → sesiones breves (10 segundos a 3 minutos).
     Esta diferencia es crucial para los análisis de clustering orientados a engagement.

4. **Coherencia en el uso de la IP**
   Un mismo *id_usuario* se conecta desde la misma IP o desde un conjunto reducido de IPs coherentes, simulando uso desde casa/oficina.
   Esto permite detectar visitas repetidas y evaluar la persistencia del interés.

5. **Distribución cronológica realista**
   Las fechas y horas siguen patrones típicos del tráfico académico:
    - Mayor actividad en días laborales.
    - Horarios de mañana y tarde como picos principales.
    - Distribución continua a lo largo de los 24 meses simulados.

Relación con el resto del pipeline

Este dataset actúa como el primer nivel del flujo lógico del proyecto:

Métricas Web → Registro → Formularios → Matriculación → Feedbacks

La coherencia entre estas capas permite:

- analizar la progresión del usuario desde su primera visita,
- identificar patrones que explican qué perfiles acaban matriculándose,
- alimentar el clustering con señales de comportamiento digital,
- y preparar la infraestructura para un futuro modelo predictivo.

Creación del dataset de Cursos_immune.csv

El catálogo de cursos constituye la **tabla de dimensiones principal** sobre la que se articula todo el ecosistema de datos del proyecto. Define los productos educativos que IMMUNE ofrece y establece las relaciones necesarias para enlazar tanto los datasets de métricas web como los formularios de registro y las encuestas de satisfacción. Sin esta tabla, el pipeline no tendría un punto de referencia común para identificar qué curso consulta, visita o valora un usuario.

El directorio contiene dos versiones del catálogo:

- **cursos_immune.xlsx**: versión simplificada utilizada por los scripts técnicos encargados de generar los datos sintéticos.
- **cursos_immune_agente.xlsx**: versión enriquecida utilizada por el agente conversacional ADK, que incorpora información comercial más extensa.

Estructura general

Las variables incluidas describen los atributos estáticos de cada producto formativo:

- **Identificador**
  - *id_curso* (formato Cxxxx): clave primaria y punto de unión entre todos los datasets (Feedbacks, Métricas, Formularios).
- **Descriptivas**
  - *nombre*: título comercial del curso.
- **Clasificación académica**
  - *tipo_de_programa*: Bootcamp, Máster, Executive, Curso intensivo, etc.
  - *sector*: categorías como Data Science, Ciberseguridad, Desarrollo Web, Cloud, IA, etc.
- **Características logísticas y económicas**
  - *modalidad*: Online, Presencial o Híbrido.
  - *inicio*: fechas previstas de convocatoria.
  - *precio*: coste económico.
  - *duracion*: duración estimada del programa.

Lógica interna y relaciones clave

1. **Punto único de verdad (Single Source of Truth)**
   Este dataset define qué *id_curso* son válidos.
   - Si un ID aparece en *Feedbacks.csv* o *Immune_metricas.csv* y **no figura** en este catálogo, supone un error de integridad referencial.
   - De este modo se asegura consistencia en todo el pipeline.
2. **Impacto directo en la generación de feedbacks**
   La columna *modalidad* condiciona la creación de datos sintéticos:
   - Cursos Online → se generan métricas como *calidad_conexion* o *audio*.
   - Cursos Presenciales → esas métricas permanecen vacías.
     Esto evita inconsistencias y reproduce el comportamiento real de los formularios de la escuela.
3. **Rol central en el agente ADK**
   El archivo **cursos_immune_agente.xlsx** actúa como la base de conocimiento del chatbot.
   Cuando un usuario pregunta, por ejemplo:

   "¿Qué curso de Python tenéis que sea económico?"
   el agente filtra este dataset utilizando atributos como *nombre* y *precio*.

   Además:
   - Permite clasificar programas por sector o modalidad.
   - Sirve de soporte para recomendaciones y búsquedas conversacionales.
4. **Sistema de fallback**
   El agente está diseñado para leer esta información desde un Google Sheet actualizado en tiempo real.
   - Si hay problemas de conectividad, el sistema recurre automáticamente al archivo local **cursos_immune_agente.xlsx**.
   - Esto garantiza disponibilidad continua de la información comercial

Importancia dentro del ecosistema

La dualidad entre catálogo técnico y catálogo enriquecido permite equilibrar dos necesidades:

- **Data Engineering**: scripts ligeros, centrados en relaciones y modalidades.
- **Agente conversacional**: contexto detallado para responder preguntas reales de usuarios.

Este diseño modular facilita la sustitución futura de los datos sintéticos por datos reales sin romper el pipeline y asegura que todos los componentes del sistema —métricas, formularios, feedbacks y agente RAG— compartan una referencia única y consistente.

3.2 Dashboard y métricas en Power BI

Tras realizar un análisis exploratorio de los datos generados (revisión de valores categóricos, análisis de outliers, inclusión de nuevas columnas), obtuvimos los ficheros “Immune_metricas_PBI”, “formularios_PBI” y “Feedbacks_PBI” con los que procedimos a hacer las visualizaciones en PBI que dieran respuesta a las preguntas:

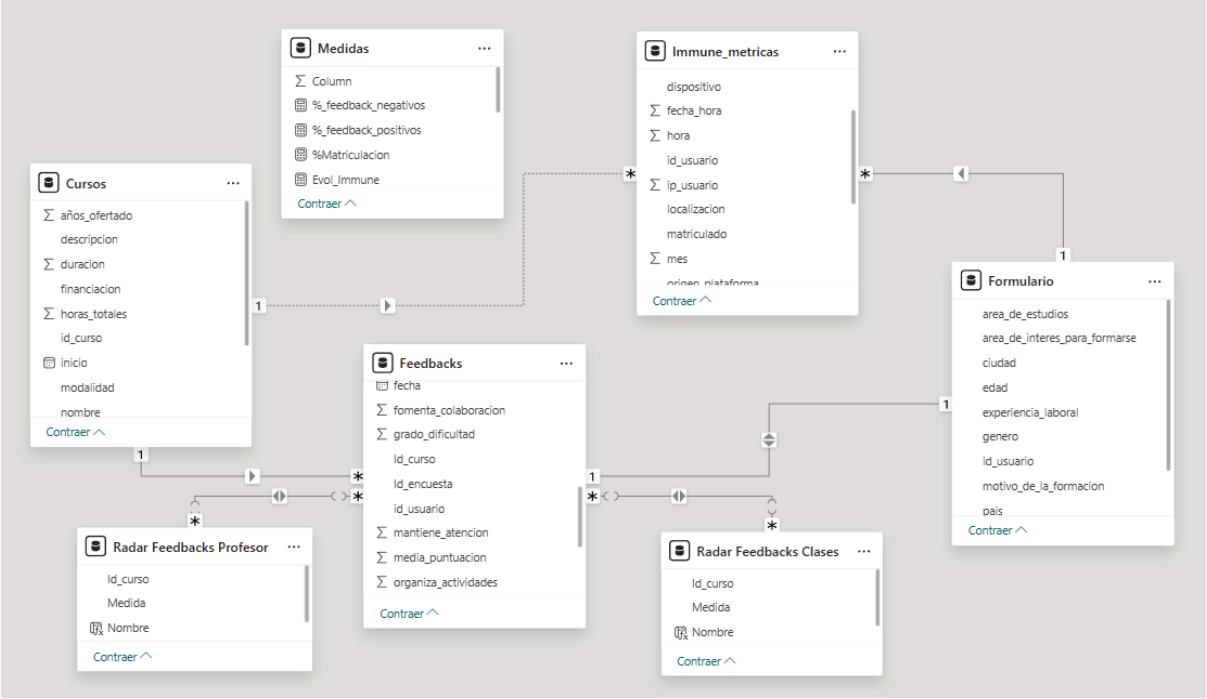
- ¿Cómo se distribuyen las visitas a la página de Immune en el tiempo?¿Y las matriculaciones?
- ¿Qué influencia tienen los canales por donde acceden los interesados?
- ¿Desde qué países llegan interesados por los programas?
- ¿Qué programas son los más y menos visitados y qué proporción de matriculados tienen?
- ¿Qué valoraciones obtenemos de nuestros programas y cómo evolucionan en el tiempo?
- ¿Las valoraciones son positivas (puntuación superior a 3,5 sobre 5) o negativas?
- ¿Cómo se distribuyen estas categorías entre los distintos tipos de programas (bootcamp, máster, cursos, etc.) y modalidades?
- ¿Las puntuaciones obtenidas se deben al profesor o a las clases?

Para ello hemos creado un **modelo en estrella como estructura principal**, enriquecido con algunas relaciones del tipo copo de nieve y tablas puente para análisis específicos. Está formado por:

- Las tablas “Immune_metricas_PBI”, “formularios_PBI” y “Feedbacks_PBI” ya mencionadas.
- La tabla “cursos_immune” que obtuvimos con web scrapping.
- Las tablas Radar Feedbacks de Clases y Profesor obtenidas a partir de la tabla de Feedbacks para poder hacer KPIs concretos. Agrupa las medias de cada categoría medida en las encuestas y las agrupamos en dos tablas dependiendo de si afectan al profesor o a las clases.
- Tabla con todas las medidas creadas para las visualizaciones.

Estas tablas se relacionan entre sí de la siguiente forma:

- Immune_metricas[programa_oferta_click] ↔ Cursos[id_curso]
- Immune_metricas[id_usuario] ↔ Cursos[id_usuario]
- Feedbacks[id_curso] ↔ Cursos[id_curso]
- Feedbacks[id_usuario] ↔ Formulario[id_usuario]
- Feedbacks[id_curso] ↔ Radar Feedbacks Profesor[id_curso]
- Feedbacks[id_curso] ↔ Radar Feedbacks Clases[id_curso]



Nuestro PBI consta de 6 pestañas:

- **Feedbacks:** Muestra etiquetas sobre el número de cursos ofertados, alumnos registrados, encuestas realizadas, porcentajes de encuestas positivas y negativas y puntuación media de las encuestas. Entre sus visualizaciones podemos ver un listado de los programas con métricas sobre sus valoraciones y varias gráficas para analizar el desglose de las valoraciones por modalidad, tipo de programa, grupos de edad y evolución en el tiempo.
- **KPIs Feedbacks:** Esta pestaña utiliza las tablas de Radar Feedbacks Profesor y Radar Feedbacks Clases para mostrar las medias de las variables a puntuar en las encuestas.
- **Immune:** Entre sus visualizaciones podemos ver la distribución de visitas a la página y matriculaciones en el tiempo, el efecto de los canales por los que llegan los interesados, el motivo por el que se matriculan, un mapa con los países registrados y comparativas con los programas más vistos.
- Adicionalmente contamos con pestañas de **detalle** que generan visualizaciones específicas al pasar el ratón por alguna de las visualizaciones anteriores.

3.3 Desarrollo del Agente Conversacional ADK

El agente conversacional constituye uno de los pilares del proyecto, ya que permite consultar de manera inteligente tanto el catálogo de cursos como los datos de satisfacción de los alumnos. Su propósito es actuar como un asistente académico capaz de responder a preguntas complejas, analizar feedbacks y generar recomendaciones basadas en datos estructurados y modelos generativos.

El desarrollo del sistema implicó un proceso iterativo de exploración de arquitecturas, pruebas de herramientas y ajustes en los datasets para garantizar compatibilidad con la lógica del agente. Tras evaluar distintas estrategias (agentes secuenciales, agentes en bucle, herramientas personalizadas, etc.), se llegó a una arquitectura final sólida, estable y alineada con los objetivos funcionales del proyecto.

Arquitectura final del agente

El diseño final se basa en un patrón **Single Orchestrator with Tool-Use**, en el que un único agente central gestiona toda la lógica conversacional, apoyándose en un conjunto de herramientas deterministas implementadas en Python. Esta decisión sustituye a un enfoque previo basado en cadenas de agentes, que resultó rígido, difícil de depurar y propenso a bucles.

Los elementos principales de la arquitectura son:

- **Agente único (“Immune Agent”)**, encargado de interpretar las consultas del usuario y orquestar el uso de herramientas.
- **Modelo base:** *gemini-2.5-flash*, elegido por su equilibrio entre velocidad y capacidad de razonamiento.
- **Herramientas (FunctionTools):** funciones Python encapsuladas que permiten realizar filtrados, cálculos estadísticos, análisis cualitativos y control de flujo.
- **Infraestructura:** Python 3.11+ y Pandas para procesado de datos; Google GenAI SDK para inferencia.

Este enfoque combina la estabilidad de la lógica determinista con la flexibilidad del razonamiento generativo.

Gestión de datos y pipeline interno

Para evitar inconsistencias y garantizar que el agente opere siempre sobre datos verificables, se implementó un sistema híbrido de carga:

- **Catálogo de cursos (cursos_immune_agente.xlsx):**
    - Intento prioritario de obtener datos desde Google Sheets en tiempo real.
    - Si falla, se utiliza el archivo local como fuente segura.
    - Se normalizan columnas clave y se controla la codificación.
- **Feedbacks (Feedbacks.csv):**
    - Carga desde Google Drive API cuando es posible.
    - Fallback a la versión local si no hay credenciales.

Durante el preprocesamiento se aplican:

- conversión estricta de columnas numéricas,
- indexación por id_curso para acelerar análisis,
- validación de integridad de las relaciones entre datasets.

Herramientas personalizadas (Custom Tools)

El valor funcional del agente se apoya en herramientas especializadas que ejecutan tareas concretas de forma determinista. Destacan cuatro grupos:

A. Herramientas de exploración de datos

- **get_unique_values(column_name):** permite al agente consultar valores reales (sectores, tipos de programa, modalidades).
  Evita que invente categorías inexistentes.
- **filtrar_cursos(query, modalidad, precio_max):** motor de búsqueda que aplica filtros multicriterio sobre el catálogo.

Herramientas de análisis avanzado

La herramienta principal es **consultar_analista**, que encapsula la lógica de análisis de un curso:

1. Identifica el id_curso correspondiente al nombre consultado.
2. Extrae todos los feedbacks del curso.
3. Calcula métricas cuantitativas (satisfacción, profesor, contenidos).
4. Selecciona una muestra representativa de comentarios.
5. Llama a un modelo generativo secundario para sintetizar:
    - fortalezas,
    - debilidades,
    - áreas de mejora.

Esta herramienta sustituyó a un agente autónomo inicial, evitando problemas de serialización y mejorando la estabilidad.

C. Herramientas de métricas globales

- **metrics_tool(analysis_type):** genera rankings, comparativas globales y análisis a nivel escuela mediante *groupby* y cálculos agregados.

D. Herramientas de control de flujo

- **fallback_tool:** devuelve respuestas aleatorias predefinidas ante consultas ambiguas, evitando loops y estabilizando la conversación.

Ingeniería del prompt

La personalidad y disciplina del agente se ajustaron mediante un prompt de sistema cuidadosamente diseñado. Entre los elementos más importantes:

- Identidad explícita: el agente se presenta siempre como *Immune Agent*.
- Flujo guiado: se obliga a ofrecer al usuario un menú inicial (A/B/C/D) y avanzar por pasos.
- Reglas estrictas:
    - prohibido inventar nombres de cursos,
    - siempre mostrar el resultado completo de las tools,
    - evitar resúmenes si la herramienta devuelve un análisis detallado.

Este control estricto evita desvíos, mantiene la coherencia y garantiza reproducibilidad.

Retos técnicos y soluciones adoptadas

Durante el desarrollo surgieron diversas limitaciones técnicas que obligaron a revisar la arquitectura:

- **Errores de descubrimiento del agente:**
  solucionados estableciendo correctamente la estructura del paquete y los archivos de inicialización.
- **Problemas de serialización (Pickle):**
  la comunicación entre agentes externos generaba errores; la solución fue integrar la lógica analítica dentro de una única tool.
- **Bucles conversacionales:**
  mitigados refinando el prompt e introduciendo el sistema de fallbacks.
- **Modelos no disponibles:**
  se testearon distintas versiones hasta estandarizar en *gemini-2.5-flash*, más estable.

Resultado final

El resultado es un agente conversacional robusto, modular y útil para la toma de decisiones académicas. Combina:

- **Pandas** → precisión en cálculos y filtrado.
- **Gemini** → análisis cualitativo y razonamiento sobre comentarios reales.

El sistema no solo facilita la búsqueda de cursos, sino que ofrece informes automáticos de satisfacción, identifica puntos fuertes y débiles por programa y permite comparar métricas a nivel global o segmentado. **Es una herramienta que IMMUNE podría incorporar directamente —una vez sustituidos los datos sintéticos por los reales— para mejorar su capacidad de análisis interno y soporte al usuario.**

3.4 Clustering

En esta sección nos propusimos desarrollar **dos modelos de clustering con K-Means** a partir de los datos de navegación de Immune y formularios de alta que los usuarios habían completado al solicitar más información sobre los programas, con el objetivo de **identificar patrones de comportamiento y perfiles de usuarios**, y analizar **cómo estos perfiles se relacionan con la matriculación**.

Estableciendo una relación 1:1 entre ip_usuario e id_usuario (suponemos que un usuario siempre va a acceder a la página desde la misma ip y que no hay más de un usuario que use la misma ip) y combinando las tablas “Immune_metricas_PBI”, “formularios_PBI” y “Feedbacks_PBI”, generamos los datasets tabla_resumen_ip y tabla_usuarios.

Preparación del modelo

Una vez preparados los dataset, se ha aplicado el siguiente **flujo de trabajo**:

- Análisis de correlaciones mediante mapa de calor.
- Estandarización de variables.
- Reducción dimensional con PCA a 2 componentes para visualización.
- Selección del número óptimo de clusters usando:
  - Método del codo
  - Silhouette score
- Aplicamos un modelo de KMEANS con el número de cluster elegido
- Representamos el dataset y los centroides de cada cluster para analizar características
- Comparamos los cluster obtenidos con los datos de matriculación para analizar dependencias.

Clustering por IP (tabla_resumen_ip)

Objetivo: Agrupar las conexiones a la web por ip_usuario, asumiendo una relación 1:1 entre IP e id_usuario, para **obtener perfiles de navegación agregados** y reducir los 5000 registros originales a perfiles únicos.

Transformaciones realizadas

- Se completan los **id_usuario nulos** usando la relación IP–usuario cuando existe.
- Se incorpora información de **sector** desde cursos_immune.
- Se identifican y contabilizan **sesiones sin clic** en programa.
- Se eliminan sesiones sin programa clicado para el análisis principal.

Se construye la tabla resumen por ip_usuario con las siguientes **variables creadas por ip**:

- **Volumen de actividad:**
  - total de visitas
  - visitas por año (2024 / 2025)
  - intervalo temporal de actividad
- **Intensidad:**
  - tiempo medio en página
  - visitas por mes (media y máximo)
- **Comportamiento:**
  - visitas entre semana vs fin de semana
  - número de programas y sectores distintos consultados
  - origen del tráfico (Google, Ads, Instagram, LinkedIn)
- **Comparativas frente a la media global de los programas:** Generamos una tabla que calcula métricas por programa sobre la media de visitas, tiempo en la página, distribución de visitas entre semana y en fines de semana y media de visitas mensual para establecer unas medias globales y comparar con las métricas por ip.
- **Número de sesiones sin programa consultado**

El modelo detectó cuatro perfiles de navegación:

- **Cluster 0:** Usuarios con pocas visitas y sesiones muy cortas. Navegan sobre todo en fines de semana y no consultan programas.

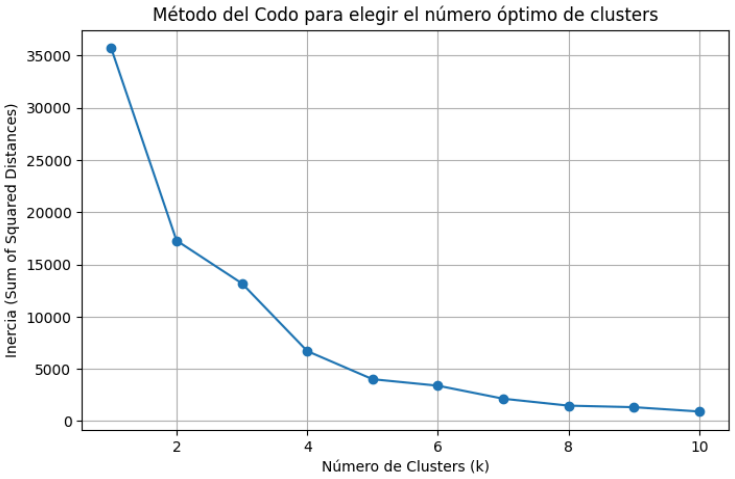
- **Cluster 1:** Usuarios muy activos: visitan la web durante varios meses, consultan muchos programas y sectores y acceden desde múltiples fuentes.

- **Cluster 2:** Usuarios con actividad moderada pero con mayor tiempo en página; a veces navegan sin clicar programas.

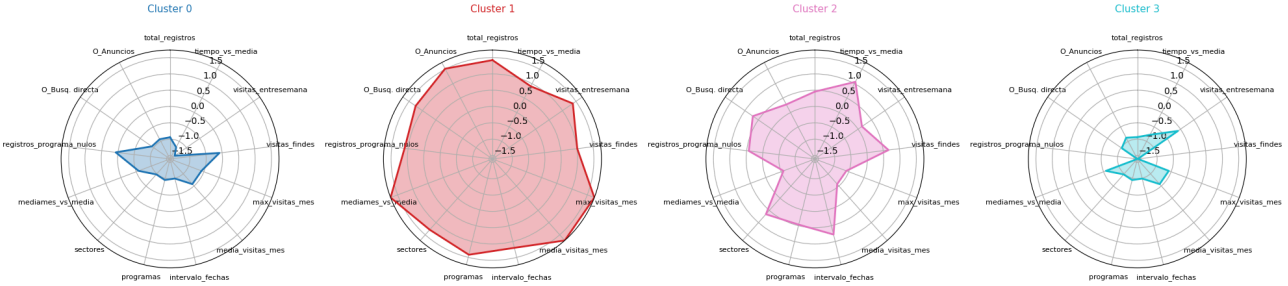
- **Cluster 3:** Usuarios con pocas visitas, concentradas en fines de semana, influenciados principalmente por anuncios y centrados en un solo programa.

Gráficas que argumentan las decisiones

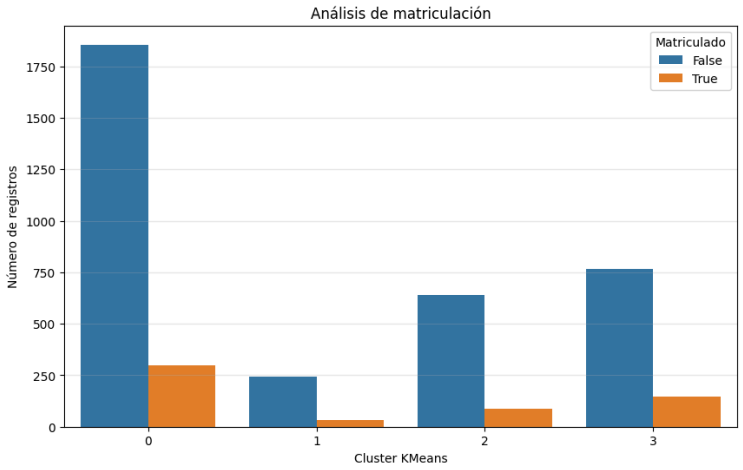
- ### Gráfica del método del codo



- ### Representación de los Centroides



- ### Efecto de los cluster en la matriculación



Clustering por usuario (tabla_usuario)

Objetivo: Crear un segundo modelo de clustering a nivel de **id_usuario**, incorporando comportamiento de navegación e información declarada en el formulario como área de estudios, experiencia laboral, áreas de interés para formarse, motivo de la formación, etc. pudiendo identificar así **perfiles de alumnos** y analizar su relación con la matriculación.

Transformaciones realizadas

- Unificación y recategorización de áreas de estudio, áreas de interés y sector laboral
- Creación de categorías macro (Tecnología, Ciencia, Humanidades, etc.).
- One-Hot Encoding de variables categóricas.
- Creación de variables derivadas como tipo de alumno (nacional / internacional)
- Eliminación de variables redundantes o altamente correlacionadas.

Transformaciones en métricas de navegación

- Filtrado a usuarios con id_usuario informado.
- Agrupación por usuario para obtener:
    - número de visitas
    - tiempo medio en página
    - visitas mensuales
    - programas y sectores consultados
    - comportamiento semanal

Se unifica esta información con el formulario para crear el **dataset final de usuarios**.

El número de variables obligó a reducir las clasificaciones para facilitar el análisis de los resultados lo que supuso crear nuevas columnas que agruparan las existentes en categorías que mostraran la relación con las áreas de Immune y las que no.

El modelo detectó tres perfiles claros:

- **Cluster 0:** Jóvenes sin experiencia laboral, con estudios medios, que buscan su primer empleo. Muy activos y con alta exploración de programas.

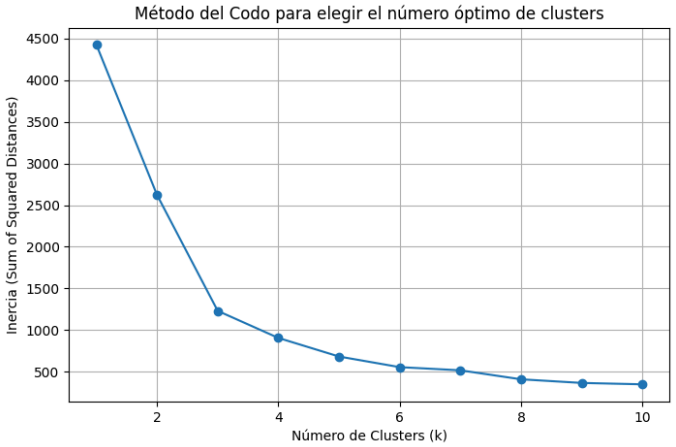
- **Cluster 1:** Usuarios con titulación básica y experiencia laboral en sectores distintos a Immune. Consultan pocos programas, orientados a conocimiento o promoción interna.

- **Cluster 2:** Profesionales senior con Máster o Doctorado y amplia experiencia. Navegan con un objetivo formativo concreto.

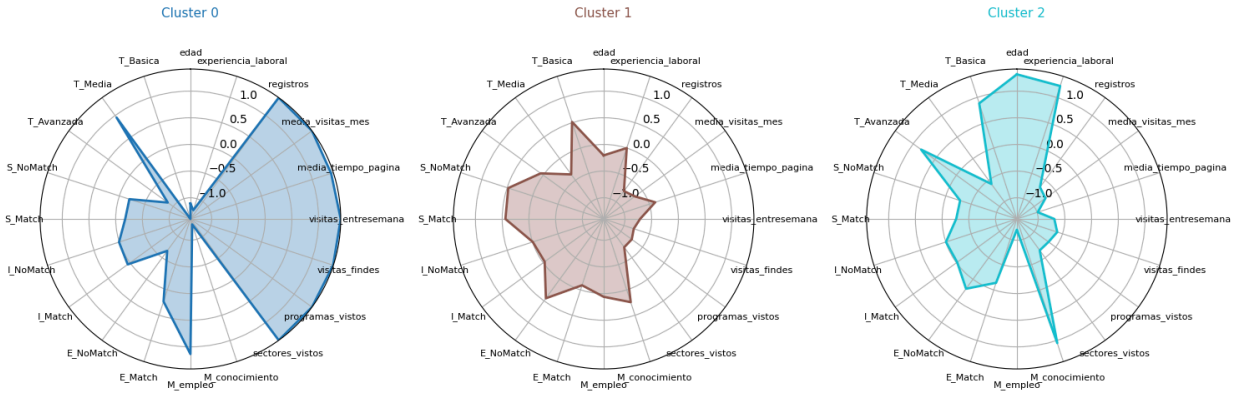
Nuestro análisis de número de cluster sugirió otra opción con 5 grupos que queda documentada en el GitHub.

Gráficas que argumentan las decisiones

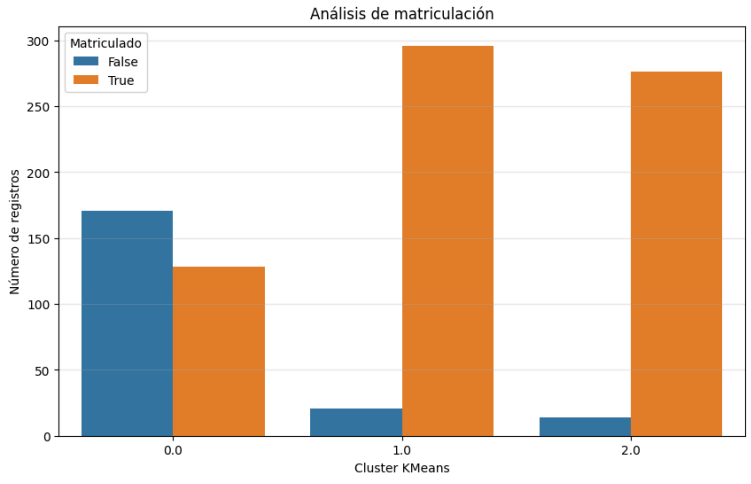
- Gráfica del método del codo



- Representación de los Centroides



- Efecto de los cluster en la matriculación



5. CONCLUSIONES:

Logros Principales del Proyecto

1. Diseño y construcción de un ecosistema completo de datos
   Se ha desarrollado una arquitectura integral capaz de centralizar información procedente de distintos procesos académicos y operativos: métricas web, formularios de leads, catálogo de cursos y encuestas de satisfacción. Aunque los datos son sintéticos, el pipeline está preparado para recibir datos reales sin necesidad de rehacer la solución.

2. Generación de datasets sintéticos realistas y coherentes
   Se crearon más de 7.000 registros simulando comportamiento de usuarios, perfiles formativos, interacciones web y evaluaciones académicas. Estos datos reproducen patrones lógicos, relaciones internas, distribuciones demográficas y reglas de negocio, permitiendo realizar análisis sólidos y tests realistas sobre el pipeline.

3. Desarrollo de un agente conversacional avanzado con ADK
   El proyecto logró implementar un agente robusto capaz de:

   - Buscar cursos por múltiples criterios,
   - Consultar métricas globales y segmentadas,
   - Analizar feedbacks reales mediante generación de informes automáticos,
   - Manejar ambigüedades y flujos conversacionales complejos,
   - Operar sobre datos deterministas y aplicar razonamiento cualitativo.
     Esto convierte al agente en una herramienta de consulta académica que aporta valor real a la toma de decisiones.

4. Construcción de un dashboard interactivo en Power BI
   Aunque pendiente de detallar en la sección técnica, se logró consolidar un panel que integra:
   - métricas de tráfico web,
   - comportamiento de leads,
   - distribución de perfiles por sectores, edades y motivaciones,
   - resultados globales de satisfacción.
     Este dashboard permite visualizar el embudo de captación y detectar patrones clave.
5. Modelado del comportamiento digital de los usuarios
   El dataset de métricas web reproduce sesiones, fuentes de tráfico, tiempos de permanencia, visitas repetidas y tasas de matriculación simuladas. Esto permite analizar:
   - qué canales son más efectivos,
   - qué perfiles muestran mayor interés,
   - cómo evoluciona el engagement desde la primera visita hasta la posible conversión.
6. Aplicación de técnicas de clustering sobre leads y comportamiento
   Se generaron las condiciones para identificar segmentos de usuarios basados en edad, sector, origen, tiempo en página, área de interés o historia de interacción. Este componente prepara el terreno para modelos predictivos futuros orientados a estimar la probabilidad de matrícula.
7. Arquitectura modular y escalable
   Todo el sistema está diseñado para ser intercambiable: cambiando simplemente los datos sintéticos por datos reales, el pipeline sigue funcionando. Esto permite al instituto adoptar la solución directamente y ampliarla con nuevas fuentes de datos.

6. REFERENCIAS BIBLIOGRÁFICAS

1. Kaggle – 5-Day Agents Course

Kaggle. (s.f.). *5 Day Agents – Learn Guide*. Recuperado de https://www.kaggle.com/learn-guide/5-day-agents

Autor corporativo: Kaggle

---

2. Google AI Studio – API Keys

Google. (s.f.). *API Keys – Google AI Studio*. Recuperado de https://aistudio.google.com/app/api-keys

Autor corporativo: Google.

---

3. GitHub – Agent Development Kit (ADK) Crash Course

Hancock, B. (s.f.). *Agent Development Kit Crash Course* [Repositorio GitHub]. GitHub. Recuperado de https://github.com/bhancockio/agent-development-kit-crash-course/tree/main