

**UNIVERSIDAD NACIONAL DE CAAGUAZÚ
FACULTAD DE CIENCIAS Y TECNOLOGÍAS
CARRERA DE INGENIERÍA EN INFORMATICA**



PROYECTO FINAL DE GRADO

**EVALUACIÓN DESCRIPTIVA DE LA OFERTA
ACADÉMICA Y EGRESO EN CARRERAS
DEL AREA DE SALUD: UNA APROXIMACIÓN
DESDE DATA ANALYTICS (2012-2024)**

CARLOS RODRIGUEZ

TUTOR: PROF. MG. HECTOR ESTIGARRIBIA

CORONEL OVIEDO, JUNIO DE 2025



Usted es libre de:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato
- **Adaptar** — remezclar, transformar y construir a partir del material

Bajo los siguientes términos:

- **Atribución** — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- **NoComercial** — Usted no puede hacer uso del material con [propósitos comerciales](#).



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

DERECHO DE AUTOR

Quien suscribe, Carlos Federico Rodríguez Fernández, autor del trabajo de investigación titulado **“Evaluación Descriptiva de la Oferta Académica y Egreso en Carreras del Área de Salud: una Aproximación desde Data Analytics (2012-2024)”**, declara que voluntariamente cede a título gratuito en forma pura y simple ilimitada e irrevocablemente a favor de la Facultad de Ciencias y Tecnologías – UNCA, el derecho de autor de contenido patrimonial, que le corresponde sobre el trabajo de referencia. Conforme a lo anteriormente expresado, esta sesión le otorga a la FCyT la Facultad de comunicar la obra divulgarla, publicarla y reproducirla en soportes analógicos o digitales en la oportunidad que así lo estime conveniente. La FCyT deberá indicar qué autoría o creación del trabajo corresponde a mi persona y hará referencia al autor y a las personas que hayan colaborado en la realización del presente trabajo de investigación.

En la ciudad de Coronel Oviedo a los, del mes de del 2025

.....

Carlos Federico Rodríguez Fernandez



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

PÁGINA DE APROBACIÓN

Trabajo de fin de grado para la obtención del Título de Ingeniero Electricista, aprobado en representación de la Facultad Ciencias y Tecnología de la Universidad Nacional de Caaguazú, por el Tribunal Examinador constituido por los siguientes profesores y con la siguiente nota final:

CALIFICACIÓN FINAL: _____

ACTA N°: _____

FECHA : _____

Prof. Ing.

Prof. Ing.

Prof. Ing.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

DEDICATORIA

A mis padres, quienes han sido el pilar fundamental de mi vida, este trabajo final es un reflejo de todo lo que me han enseñado: el valor del esfuerzo, la importancia de la honestidad y el poder de los sueños. Dedico este logro a ustedes, porque sin su guía silenciosa y su amor incondicional, no sería quien soy hoy.

A mis hermanas Macarena y Adriana, compañeras de vida y confidentes eternas, el triunfo también es de ustedes, porque han sido parte de cada risa, cada desafío y cada celebración. Su cariño y complicidad hacen que todo valga la pena, y este camino no tendría sentido sin compartirlo con ustedes.

A mi novia Jazmín, quien ha sido mi refugio y mi inspiración en cada paso de este proceso. Este trabajo lleva tu esencia, porque me has mostrado que el amor verdadero es el motor que nos impulsa a ser mejores.

A mis primos, quienes siempre han creído en mí más de lo que yo mismo he creído. Les dedico este trabajo como un recordatorio de que los sueños se cumplen cuando uno apuesta por ellos con todo. Espero que sigan apostando por sus propios caminos, porque sé que grandes cosas les esperan. Este logro también es de ustedes.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

AGRADECIMIENTOS

A Dios, por ser mi guía y sostén en cada paso de este camino, por darme la fortaleza para enfrentar los desafíos, la claridad para encontrar soluciones y la gratitud para valorar cada aprendizaje.

A mi tutor, Héctor Estigarribia, por su invaluable acompañamiento, sabiduría y dedicación a lo largo de este proceso. Su orientación constante y su capacidad para inspirarme a dar lo mejor de mí han sido fundamentales para la realización de este trabajo. Le estaré siempre agradecido por creer en mí y por compartir sus conocimientos con tanta generosidad.

A todos aquellos que, de manera directa o indirecta, contribuyeron a mi formación personal y académica. A mis profesores, compañeros y amigos, cuyas enseñanzas, consejos y experiencias compartidas dejaron una huella imborrable en mi camino. También agradezco a quienes me brindaron su apoyo desde la distancia, con palabras de ánimo o simplemente con su compañía en los momentos más inesperados.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

RESUMEN

El presente estudio aborda la problemática de la fragmentación informativa sobre la oferta académica y el egreso en carreras del área de salud en Paraguay, caracterizada por la ausencia de estándares unificados y la dispersión de datos entre diferentes instituciones reguladoras, lo que dificulta la comprensión holística del sistema formativo. El objetivo principal es analizar descriptivamente la oferta académica y el egreso en carreras del área de salud mediante técnicas de Data Analytics aplicadas a datos abiertos del sistema de educación superior. Metodológicamente, se desarrolla un enfoque cuantitativo no experimental basado en el análisis de datos secundarios provenientes del Registro Nacional de Carreras (RNC) y el Registro Nacional de Títulos de Educación Superior (RTES). Se implementa un pipeline de procesamiento que incluye: descarga y preparación de datasets, filtrado de datos relacionados con salud, agrupamiento temático de carreras, completado del campo sexo mediante inferencia de género, incorporación del tipo de gestión institucional, limpieza de formatos y análisis mediante tablas dinámicas. Los resultados evidencian un sistema de formación sanitaria caracterizado por una marcada privatización (89.2% de los títulos provienen del sector privado), una significativa feminización (75.7% de los títulos otorgados a mujeres), y una concentración geográfica en tres departamentos que acumulan el 49.1% de la oferta académica. Se identifica la predominancia de la enfermería (36.9% de los títulos) sobre medicina (20.6%), estableciendo una proporción de 2.3 enfermeros por cada médico. El análisis revela también importantes disparidades entre instituciones, como el caso del Instituto Superior en Ciencias de la Salud Juan Pablo II, que lidera la oferta académica, pero presenta una producción modesta de egresados. Las universidades emergen como el eje central del sistema, otorgando el 73.9% de los títulos pese a representar solo el 39.3% de las instituciones. Estos hallazgos proporcionan información crítica para la toma de decisiones en política educativa y planificación de recursos humanos en salud.

Palabras claves: Educación superior, ciencias de la salud, análisis de datos, oferta académica, tasas de egreso, Paraguay.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

ABSTRACT

The present study addresses the problem of information fragmentation on academic offerings and graduation in health careers in Paraguay, characterized by the absence of unified standards and data dispersion among different regulatory institutions, which hinders a holistic understanding of the educational system. The main objective is to descriptively analyze the academic offerings and graduation in health careers through Data Analytics techniques applied to open data from the higher education system.

Methodologically, a non-experimental quantitative approach is developed based on the analysis of secondary data from the Registro Nacional de Carreras (RNC) and the Registro Nacional de Títulos de Educación Superior (RTES). A processing pipeline is implemented that includes: downloading and preparing datasets, filtering health-related data, thematic grouping of careers, completing the sex field through gender inference, incorporating the type of institutional management, cleaning formats, and analysis through pivot tables. The results show a health training system characterized by marked privatization (89.2% of degrees come from the private sector), significant feminization (75.7% of degrees awarded to women), and geographic concentration in three departments that accumulate 49.1% of academic offerings. The predominance of nursing (36.9% of degrees) over medicine (20.6%) is identified, establishing a ratio of 2.3 nurses for each doctor. The analysis also reveals important disparities between institutions, such as the case of the Higher Institute of Health Sciences Juan Pablo II, which leads academic offerings but presents a modest production of graduates. Universities emerge as the central axis of the system, awarding 73.9% of degrees despite representing only 39.3% of institutions. These findings provide critical information for decision-making in educational policy and planning of human resources in health.

Keywords: Higher education, health sciences, data analysis, academic programs, graduation rates, Paraguay.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

INDICE

DERECHO DE AUTOR.....	3
INTRODUCCION	12
<i>ANTECEDENTES DEL PROYECTO</i>	13
<i>PLANTEAMIENTO DEL PROBLEMA</i>	14
<i>DELIMITACIÓN Y ALCANCE</i>	15
<i>JUSTIFICACIÓN</i>	16
<i>OBJETIVOS</i>	16
<i>Objetivo general</i>	16
<i>Objetivos específicos</i>	16
METODOLOGÍA	17
<i>ENFOQUE DE INVESTIGACIÓN</i>	17
<i>ALCANCE DE INVESTIGACIÓN</i>	17
<i>TIPO DE INVESTIGACIÓN</i>	18
<i>TÉCNICA DE INVESTIGACIÓN</i>	19
<i>INSTRUMENTOS DE INVESTIGACION</i>	20
<i>FASES METODOLÓGICAS</i>	21
<i>Proceso de Descarga y Preparación de los Datos</i>	22
<i>Filtrado de Datos Relacionados con el Área de Salud</i>	22
<i>Agrupamiento de Carreras por Temática</i>	23
<i>Completado del Campo Sexo</i>	24
<i>Incorporación del Campo Tipo de Gestión</i>	26
<i>Limpieza de Formatos</i>	26
<i>Fase de Análisis y Visualización: Creación de Tablas Dinámicas y Gráficos</i>	27
<i>Fase de desarrollo del Dashboard Interactivo</i>	28
RESULTADOS Y ANALISIS.....	30
<i>ANÁLISIS DE LA OFERTA</i>	30
<i>ANÁLISIS DE LOS EGRESADOS</i>	35
<i>ANÁLISIS DEL DASHBOARD</i>	44
REFERENCIAS BIBLIOGRÁFICAS.....	50
ANEXOS	51



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

INDICE DE FIGURAS

Figura 1. Dinámica de titulación (2012-2024).....	36
Figura 2. Comportamiento de titulación por género (2012-2024).	36
Figura 3. Cantidad de titulados por grupo de carrera (2012-2024).....	38
Figura 4. Cantidad de titulados por tipo de gestión.	40
Figura 5. Dashboard académico del área de salud.	40
Figura 6. Dashboard académico del área de salud con filtros aplicados.....	40



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

INDICE DE TABLAS

Tabla 1. Número de ofertas de títulos de carreras por tipo institución y nivel de titulación.....	31
Tabla 2. Número de ofertas de títulos de carreras por tipo de gestión y nivel de titulación.	32
Tabla 3. Número de instituciones por tipo de institución.	33
Tabla 4. Número de ofertas de títulos de carreras por institución.....	34
Tabla 5. Número de ofertas de títulos de carreras por Departamento y Campo Detallado.....	35
Tabla 6. Número de títulos por Tipo de Institución y Sexo.	37
Tabla 7. Top 20 de las carreras con mayor número de títulos registrados.....	40
Tabla 8. Número de títulos de Institutos Superiores discriminado por nombre.....	41
Tabla 9. Número de títulos de Universidades discriminado por nombre.....	42
Tabla 10. Número de títulos de Institutos Técnicos Superiores discriminado por nombre	43

INTRODUCCION

Los avances en el análisis de datos, presenta una oportunidad significativa para obtener información valiosa sobre la dinámica de la oferta académica y las tendencias de egreso, particularmente dentro del área las carreras relacionadas con la salud.

Este proyecto se centra en una evaluación descriptiva de la oferta académica y el egreso en carreras de salud en Paraguay, empleando un enfoque basado en datos para comprender de manera integral el panorama actual de la educación en dicho sector.

La investigación tiene una relevancia directa para la Agenda 2030 para el Desarrollo Sostenible de las Naciones Unidas, contribuyendo significativamente al ODS 3: Salud y Bienestar, que tiene como objetivo garantizar vidas saludables y promover el bienestar para todos en todas las edades, una comprensión profunda de la oferta académica y las tasas de egreso en carreras de salud es fundamental para fortalecer la fuerza laboral de la salud, una meta clave del presente objetivo. Además, una oferta adecuada de profesionales de la salud calificados es esencial para lograr la cobertura universal de salud (Meta 3.8), garantizando el acceso a servicios de atención médica esenciales de calidad para todos.

La calidad y la cantidad de profesionales de la salud producidos por el sistema educativo paraguayo influyen directamente en la capacidad de la nación para abordar las necesidades de salud de su población y avanzar hacia el ODS 3. Una evaluación basada en datos puede identificar eficazmente las áreas dentro del proceso educativo que requieren fortalecimiento para satisfacer estas demandas.

Además, el proyecto se alinea con el ODS 4: Educación de Calidad, que enfatiza la importancia de garantizar una educación inclusiva y equitativa de calidad y promover oportunidades de aprendizaje permanente para todos. Al evaluar la oferta académica en ciencias de la salud, este estudio revela posibles disparidades en la disponibilidad y los tipos de programas de salud en diferentes regiones o instituciones dentro de Paraguay que pueden afectar el acceso equitativo a una educación de calidad en este campo crucial, obstaculizando el logro de resultados de aprendizaje relevantes y efectivos (Meta 4.1) y la igualdad de acceso a la educación técnica, profesional y superior de calidad (Meta 4.3) [1].

En última instancia, esta investigación es crucial para informar decisiones políticas basadas en evidencia destinadas a fortalecer tanto el sector de la salud como el sistema de educación superior en Paraguay, contribuyendo así a la realización más amplia de los Objetivos de Desarrollo Sostenible.

ANTECEDENTES DEL PROYECTO

El trabajo titulado *"Registro de la educación en Paraguay: Estudio descriptivo de los registros de títulos de la Educación Superior en Paraguay, corte longitudinal 2012 al 2020"* establece un precedente metodológico para el análisis de datos educativos nacionales, investigación donde se analizaron 327.714 registros de títulos registrados entre 2012 y 2020, provenientes de 330 instituciones educativas, incluyendo instituciones de educación superior, institutos técnicos y otras entidades formativas [2]. Los resultados evidenciaron una mayor tasa de titulación entre mujeres en comparación con varones, con predominancia en las áreas de salud y educación, por el contrario, las carreras de ingeniería y ciencias exactas, sectores estratégicos de alta remuneración, presentaron bajas tasas de titulación, con escasa participación femenina, factor que contribuye negativamente a la equidad en ingresos laborales. Si bien en dicho trabajo se consideraron los egresos totales por área de conocimiento, no se realizó un análisis específico sobre la cantidad de carreras ofertadas en cada área, aspecto diferencial que el presente proyecto aborda, enfocándose particularmente en el ámbito de las carreras de salud.

Dentro del panorama de la educación superior paraguaya, se destaca un enfoque específico en la variedad de carreras relacionadas con la salud disponibles, estas incluyen carreras como medicina, enfermería, odontología, farmacia y diversas profesiones de la salud aliadas.

Tal y como se presenta en *"Estudiantes brasileiros de medicina en Presidente Franco (PY): motivaciones y tensión de un flujo universitario transfronterizo"* existe una marcada dinámica regional transfronteriza en la educación médica, caracterizada por un significativo flujo de estudiantes brasileños que cursan medicina en Paraguay lo que revela patrones migratorios educativos con implicaciones para ambos sistemas de salud nacionales [3]. Las motivaciones identificadas para este desplazamiento académico transfronterizo comprenden diversos factores socioeconómicos y estructurales, entre ellos: mayor accesibilidad y menor costo de la educación médica paraguaya, así como procesos de admisión menos restrictivos en comparación con las instituciones brasileñas, este fenómeno constituye un indicador relevante de las asimetrías existentes entre los sistemas educativos regionales y sus efectos en la formación de profesionales de la salud, esto se con investigaciones que documentan que la proximidad geográfica representa uno de los principales factores determinantes, permitiendo a los estudiantes mantener su residencia en Brasil mientras cursan estudios en Paraguay, beneficiándose simultáneamente del menor costo de vida en las ciudades paraguayas fronterizas [4], esta importante afluencia de estudiantes de medicina brasileños sugiere una demanda de educación médica que las instituciones paraguayas están atendiendo. Sin embargo, este fenómeno también plantea importantes interrogantes sobre la capacidad y los recursos de las instituciones educativas paraguayas y las consecuencias a largo plazo para la calidad

de la educación médica proporcionada y el posible impacto en el sistema nacional de salud paraguayo.

Las tendencias generales en las tasas de egreso dentro de la educación superior paraguaya se pueden obtener de la literatura existente, como el estudio previamente mencionado sobre el registro nacional de títulos [2]. Si bien los datos específicos sobre las tasas de egreso dentro de los campos relacionados con la salud pueden ser limitados, investigaciones centradas en otras disciplinas como aquellas relacionadas a las TICs, proporcionan información valiosa sobre las metodologías para analizar las tasas de egreso en el contexto paraguayo [5]. Aunque este estudio en particular se centra en los programas de ingeniería, su enfoque para examinar la progresión y la finalización de los estudios de los estudiantes se adapta para investigar las tasas de egreso en los programas de carreras de salud en Paraguay.

Estos estudios también resaltan tanto las oportunidades como los desafíos asociados con el empleo de datos educativos abiertos en Paraguay, particularmente en lo que respecta a la calidad y estandarización de los datos. La aplicación existente de datos abiertos para analizar la oferta académica y el egreso en campos distintos de la salud demuestra la viabilidad y los beneficios potenciales de adoptar un enfoque similar basado en datos para investigar las carreras de salud.

Una visión general del mercado laboral para los profesionales de la salud en Paraguay, basada en la información de fuentes específicas (como informes, estadísticas, bases de datos del mercado laboral, estudios del MEC, del Ministerio de Salud o de organismos internacionales), es esencial para contextualizar la oferta académica y las tasas de egreso.

PLANTEAMIENTO DEL PROBLEMA

El análisis comprensivo del sistema de educación superior paraguayo, específicamente en el área de salud, enfrenta importantes desafíos metodológicos y estructurales. Si bien existe un registro nacional de títulos en el portal de datos abiertos del Ministerio de Educación [6] que permite cierta aproximación al fenómeno educativo, la información se encuentra fragmentada en múltiples datasets administrados por diferentes instituciones gubernamentales. Por un lado, se dispone de datos sobre títulos registrados, mientras que por otro existen registros de las carreras y datos parciales sobre egresados. Esta fragmentación representa un obstáculo significativo para la comprensión holística del panorama formativo en el área de salud y dificulta la toma de decisiones basadas en evidencia.

Paralelamente, el fenómeno transfronterizo ha generado una dinámica particular en la formación de profesionales médicos, con un flujo constante de estudiantes brasileños que buscan titulación en Paraguay. Este proceso ha intensificado las preocupaciones sobre una posible superpoblación de médicos, particularmente en regiones fronterizas, situación que, según diversos medios periodísticos y declaraciones de asociaciones profesionales,

podría estar vinculada a deficiencias en los mecanismos de control de calidad educativa [7] [8]. La convergencia entre la alta oferta académica, el creciente número de egresados y los cuestionamientos sobre estándares formativos constituye un problema multidimensional que requiere análisis basados en datos.

Desde la perspectiva del análisis de datos, la problemática se complejiza debido a la ausencia de estándares unificados entre las diferentes fuentes de información, la información sobre carreras universitarias en Paraguay se encuentra dispersa entre ANEAES, CONES y MEC [9], sin criterios homogéneos para su catalogación y actualización. Esta situación genera inconsistencias en los datos, dificultando la trazabilidad de indicadores críticos como tasas de egreso, tiempo promedio de titulación o distribución geográfica de los profesionales formados.

DELIMITACIÓN Y ALCANCE

La presente investigación se enfoca específicamente en el análisis de la oferta y las tasas de egreso en carreras relacionadas con el campo de la salud dentro del sistema de educación superior paraguayo. Sin embargo, debido a la escasez de datos directos sobre este tema, el estudio se basa en la adaptación metodológica de investigaciones previas realizadas en otras disciplinas, como el análisis de las tasas de egreso en carreras de ingeniería en la Facultad de Ciencias y Tecnologías (FCYT) de la Universidad Nacional de Caaguazú (UNCA).

El alcance temporal de la investigación abarca datos disponibles desde el 2012 hasta el año 2024, geográficamente, el estudio se centra en instituciones de educación superior ubicadas en Paraguay, sin profundizar en comparaciones internacionales.

Además, se excluyen aspectos relacionados con la calidad de los egresados o su inserción laboral, ya que el objetivo principal es comprender las dinámicas de permanencia y egreso académico. Por último, aunque se mencionan factores socioeconómicos, estos no son analizados en detalle debido a la falta de datos específicos para las carreras de salud.

Para garantizar un enfoque preciso y relevante, se excluyen aquellas carreras que, aunque están vinculadas al sector médico o de la salud en un sentido amplio, no corresponden directamente al ámbito de la práctica médica o asistencial. Por ejemplo, se descartan programas académicos como administración en salud, derecho médico y veterinaria.

De esta manera, el análisis se limita a carreras que tienen como objetivo principal la formación de profesionales capacitados para brindar atención directa a pacientes humanos, como medicina, enfermería, odontología, fisioterapia, nutrición, farmacia y otras disciplinas afines. Esta delimitación permite centrar la investigación en los desafíos específicos que enfrentan los estudiantes de estas áreas, que suelen estar más influenciados por factores prácticos, éticos y emocionales propios de la atención médica.

JUSTIFICACIÓN

En Paraguay, donde los desafíos relacionados con el acceso a servicios de salud y la calidad de atención médica siguen siendo prioridades nacionales, contar con información precisa sobre la oferta educativa y la producción de recursos humanos calificados en este campo es esencial para planificar políticas públicas efectivas y garantizar el fortalecimiento del sistema sanitario.

La oferta académica en Medicina no solo refleja la capacidad institucional de las universidades para formar profesionales, sino también su alineación con las necesidades del mercado laboral y las demandas de salud pública. Además, conocer la cantidad de títulos otorgados en el sector permite evaluar si la producción de médicos está en consonancia con las necesidades del país o si existen desequilibrios que requieran ajustes en la planificación educativa.

Este estudio se justifica también por su contribución a la transparencia y accesibilidad de la información, al utilizar bases de datos abiertas del sistema de educación superior, se busca organizar y presentar datos dispersos sobre la oferta académica y la cantidad de egresados en Medicina de manera clara y estructurada. Esta sistematización no solo facilita la comprensión del panorama actual, sino que también sienta las bases para futuras investigaciones que profundicen en aspectos específicos del sector.

Al identificar la distribución geográfica de las carreras de Medicina, se pueden detectar regiones donde existe una menor oferta educativa y, consecuentemente, una menor producción de profesionales médicos, lo que puede orientar la creación de nuevos programas en áreas subatendidas o el fortalecimiento de los ya existentes.

OBJETIVOS

Objetivo general

Analizar descriptivamente la oferta académica y egreso en carreras de Medicina del Paraguay mediante técnicas de Data Analytics aplicadas a datos abiertos del sistema de educación superior.

Objetivos específicos

1. Diseñar e implementar un pipeline de procesamiento y limpieza de datos.
2. Aplicar métodos estadísticos y técnicas de análisis descriptivo para caracterizar los patrones actuales en la oferta académica y las tasas de egreso de las carreras del área de Salud.

3. Desarrollar gráficos y visualizaciones interactivas que integren los resultados obtenidos durante el análisis descriptivo

METODOLOGÍA

ENFOQUE DE INVESTIGACIÓN

El estudio se enmarcó en un enfoque cuantitativo, el cual se caracteriza por la recopilación y análisis de datos numéricos para identificar patrones, tendencias y relaciones entre variables. Este enfoque fue seleccionado debido a su capacidad para proporcionar resultados objetivos, medibles y generalizables [10], lo que resulta especialmente útil en investigaciones que buscan describir y analizar fenómenos a gran escala, como es el caso de la oferta educativa en el área de la salud en Paraguay.

La naturaleza cuantitativa del estudio permitió trabajar con grandes volúmenes de datos provenientes de fuentes oficiales, como el Registro Nacional de Carreras (RNC) y el Registro Nacional de Títulos de Educación Superior (RTES), garantizando una base sólida para el análisis estadístico y la generación de conclusiones robustas.

El enfoque cuantitativo se fundamentó en el análisis de datos secundarios, los cuales fueron procesados y transformados mediante técnicas de programación y herramientas estadísticas. La utilización de los datos permitió acceder a información histórica y actualizada sobre las carreras y títulos otorgados en el área de la salud, lo que facilitó la identificación de tendencias temporales y la comparación entre diferentes categorías, como género, tipo de gestión (pública o privada) y grupos temáticos de carreras. Este enfoque no solo optimizó los recursos disponibles, sino que también aseguró la replicabilidad del estudio, ya que los datos utilizados son de acceso público y están estandarizados por el Ministerio de Educación y Cultura (MEC) [11].

Además, el enfoque cuantitativo permitió la aplicación de técnicas de análisis estadístico y la creación de visualizaciones que facilitaron la interpretación de los resultados.

Mediante el uso de herramientas como Python para el procesamiento de datos y Excel para la creación de tablas dinámicas y gráficos, se logró transformar los datos crudos en información significativa. Estas técnicas incluyeron el filtrado de datos, la normalización de textos, la inferencia de género y el agrupamiento temático de carreras, lo que permitió abordar de manera sistemática las preguntas de investigación planteadas.

ALCANCE DE INVESTIGACIÓN

El alcance de esta investigación fue descriptivo, ya que el objetivo principal fue caracterizar y analizar la oferta educativa y el comportamiento de egreso en el área de la

salud en Paraguay, a partir de datos oficiales proporcionados por el Ministerio de Educación y Cultura (MEC).

Un estudio descriptivo se enfoca en detallar las características de un fenómeno o población específica, sin buscar establecer relaciones causales o explicativas [10], en este caso, el estudio permitió describir de manera sistemática y detallada aspectos como la distribución de carreras y títulos otorgados, la participación de instituciones públicas y privadas, las tendencias temporales y las diferencias por género, entre otros.

El alcance descriptivo se justificó por la necesidad de contar con información actualizada y organizada sobre las carreras y títulos relacionados con la salud en Paraguay. A través del análisis de datos secundarios provenientes del Registro Nacional de Carreras (RNC) y el Registro Nacional de Títulos de Educación Superior (RTES), se logró identificar y categorizar las carreras según su campo amplio, nivel de titulación y modalidad de asistencia, entre otros aspectos. Además, se describieron las características de los egresados, como su distribución por género y la institución de la que provienen.

El estudio también incluyó una descripción del comportamiento temporal en la emisión de títulos, lo que permitió identificar cambios en la oferta educativa a lo largo del tiempo. Por ejemplo, se pudo observar si ciertas carreras han ganado o perdido popularidad, o si ha habido un aumento en la participación de instituciones privadas en la formación de profesionales de la salud. Asimismo, se describieron las diferencias en la distribución de títulos por género, lo que aportó información valiosa sobre la participación de hombres y mujeres en diferentes áreas de la salud.

TIPO DE INVESTIGACIÓN

Esta investigación se clasifica como aplicada y no experimental, basada en el análisis de datos secundarios el fin principal fue generar información útil y práctica para la toma de decisiones en el ámbito de la educación superior y las políticas públicas relacionadas con la formación de profesionales de la salud en Paraguay. Este estudio se enfocó en resolver un problema concreto: la falta de un diagnóstico actualizado y detallado sobre la oferta educativa en el área de la salud.

El diseño de la investigación es no experimental, ya que no se manipuló ni se intervino en las variables estudiadas [10]. En su lugar, se analizaron datos existentes provenientes de fuentes oficiales, como el Registro Nacional de Carreras (RNC) y el Registro Nacional de Títulos de Educación Superior (RTES), proporcionados por el Ministerio de Educación y Cultura (MEC), con esto se logró observar y describir los fenómenos educativos en su contexto natural, sin alterar las condiciones en las que ocurrieron [10]. Al ser un estudio no experimental, se evitó la introducción de sesgos asociados con la manipulación de variables, lo que garantizó la objetividad y la validez de los resultados.

TÉCNICA DE INVESTIGACIÓN

El presente estudio se fundamenta metodológicamente en el trabajo “*Análisis de oferta y egreso en carreras TIC en la educación superior en Paraguay, basado en el estudio de datos abiertos*” [5], adoptó un enfoque cuantitativo basado en el análisis de datos secundarios provenientes de dos conjuntos de datos oficiales del Ministerio de Educación y Cultura (MEC): el Registro Nacional de Carreras (RNC) y el Registro Nacional de Títulos de Educación Superior (RTES). Estos datasets son generados y mantenidos por el MEC como parte de su sistema de información sobre la educación superior en Paraguay.

La técnica de investigación utilizada en este estudio fue el análisis de datos secundarios, que consiste en el procesamiento y examen de información preexistente recolectada por otras entidades. Esta técnica fue seleccionada debido a su capacidad para proporcionar una base de datos robusta y confiable, lo que permitió realizar un análisis exhaustivo sin la necesidad de recolectar información primaria.

El análisis de datos secundarios implicó un proceso riguroso de limpieza, transformación y organización de la información. Esto incluyó la identificación y corrección de inconsistencias, la normalización de formatos y la eliminación de datos duplicados o incompletos. Además, se aplicaron técnicas de programación para automatizar tareas repetitivas, como el filtrado de registros y la inferencia de género a partir de nombres.

Otra técnica clave fue la clasificación y agrupamiento temático de las carreras relacionadas con la salud. Mediante el uso de algoritmos programados en Python, se implementó un proceso de agrupamiento basado en palabras clave y similitud semántica a fin de superar la heterogeneidad en la denominación de las carreras y facilitar la interpretación de los resultados.

Finalmente, la técnica de investigación incluyó la visualización de datos como un componente esencial para comunicar los hallazgos. Se utilizaron herramientas como Microsoft Excel para crear tablas dinámicas y gráficos que resumieran los patrones y tendencias identificados. Posteriormente, se desarrolló un dashboard interactivo utilizando tecnologías web (HTML, CSS y JavaScript) que permitió la exploración dinámica de los datos mediante filtros por año, tipo de institución, carrera, género y tipo de gestión.

El dashboard incorporó diversas representaciones gráficas, incluyendo diagramas circulares para distribuciones proporcionales, gráficos de líneas para tendencias temporales, y visualizaciones de barras para comparativos entre categorías de tal forma a transformar la experiencia analítica al permitir a los usuarios interactuar directamente con

la información, profundizando en áreas específicas de interés y descubriendo relaciones que podrían no ser evidentes en un informe estático.

Esta aproximación multifacética a la visualización de datos resultó fundamental para presentar los resultados de manera clara y accesible para diferentes audiencias, desde autoridades educativas hasta investigadores y profesionales de la salud, ofreciendo una herramienta valiosa para la toma de decisiones basadas en evidencia en el sector académico de salud.

INSTRUMENTOS DE INVESTIGACION

Los instrumentos de investigación utilizados en este estudio fueron principalmente bases de datos estructuradas proporcionadas por el Ministerio de Educación y Cultura (MEC) de Paraguay. Estas bases de datos, el Registro Nacional de Carreras (RNC) y el Registro Nacional de Títulos de Educación Superior (RTES), contenían información detallada sobre las carreras ofertadas y los títulos otorgados en el área de la salud. Estos instrumentos fueron seleccionados por su confiabilidad y representatividad, ya que son generados y mantenidos por una institución oficial.

Registro Nacional de Carreras (RNC)

Este dataset contiene información sobre carreras de grado y programas de posgrado de universidades e institutos superiores, tanto del sector público como privado, que se encuentran legalmente habilitados en base al análisis de instrumentos legales presentados por las instituciones al Consejo Nacional de Educación Superior (CONES) vía declaración jurada. Incluye carreras creadas en el marco de diversas leyes vigentes desde 1889 hasta la fecha y cuenta con los siguientes campos:

- Nombre de la Institución
- Código de la Institución
- Tipo de Institución
- Tipo de Gestión
- Código y nombre del Departamento
- Código y nombre del Distrito
- Código del Establecimiento
- Denominación de la Carrera
- Clasificación según el Campo Amplio, Específico y Detallado de la Clasificación Internacional Normalizada de Educación (CINE-P 2013) de la UNESCO.
- Nivel de Titulación
- Modalidad de Asistencia, entre otros.

Registro Nacional de Títulos de Educación Superior (RTES):

Este dataset contiene información oficial sobre egresados discriminados por año, mes, documento de identidad, nombre completo, carrera, título obtenido, número de resolución, tipo de institución, institución, sexo, y otros datos relevantes. Los registros incluyen títulos de grado y postgrado registrados en el MEC a partir de 2012.

Otro instrumento clave fue el software de programación, específicamente Python, que se utilizó para el procesamiento y análisis de datos. Python permitió la implementación de algoritmos para filtrar, normalizar y agrupar los datos, así como para inferir el género de los egresados a partir de sus nombres

Además, se utilizaron hojas de cálculo (como Microsoft Excel) como instrumentos para la organización y visualización de los datos. Excel permitió la creación de tablas dinámicas y gráficos que resumieron los patrones y tendencias identificados en el análisis. Este instrumento fue especialmente útil para presentar los resultados de manera clara y accesible, facilitando la interpretación de los hallazgos.

Finalmente, se emplearon bibliotecas especializadas en Python, como Polars y gender-guesser, como instrumentos adicionales para el análisis de datos. Estas bibliotecas permitieron realizar operaciones avanzadas, como la manipulación de grandes conjuntos de datos y la inferencia de género, lo que enriqueció el análisis y aseguró la precisión de los resultados.

FASES METODOLÓGICAS

Aunque técnicamente era posible realizar todo el procesamiento de datos en un único módulo o script, sin la necesidad de exportar múltiples archivos CSV intermedios, se optó por un enfoque modular y por guardar los resultados de cada fase en archivos separados. Esta decisión se basó en la necesidad de garantizar la transparencia, la trazabilidad y la facilidad de revisión en cada etapa del proceso.

Al dividir el análisis en fases claramente definidas y exportar los resultados intermedios, se pudo observar y validar el aporte específico de cada proceso, lo que contribuyó a la robustez y confiabilidad del estudio.

La modularidad permitió aislar y resolver problemas de manera más eficiente. Al trabajar con grandes volúmenes de datos, es común enfrentar errores o inconsistencias que pueden ser difíciles de identificar si todo el procesamiento se realiza en un solo paso. Al dividir el proceso en fases independientes, como el filtrado de datos, el agrupamiento temático y la inferencia de género, fue posible detectar y corregir errores en cada etapa sin afectar el resto del análisis. Además, esta estructura facilitó la replicabilidad del estudio, ya que cada fase puede ser ejecutada y validada de manera independiente.

La exportación de resultados intermedios en archivos CSV también permitió documentar el flujo de trabajo y proporcionar una visión clara de cómo los datos se transformaron en cada fase.

Esta práctica no solo mejoró la calidad del análisis, sino que también facilitó la colaboración con otros investigadores, quienes podían revisar y validar los resultados en cualquier etapa del proceso.

Finalmente, este enfoque modular y la exportación de resultados intermedios tuvieron un impacto positivo en la eficiencia a largo plazo. Aunque inicialmente requirió más tiempo y esfuerzo, la estructura modular permitió reutilizar y adaptar partes del código para futuros proyectos. Además, los archivos CSV intermedios sirvieron como respaldos que podrían ser utilizados en caso de errores o para realizar análisis adicionales sin necesidad de repetir todo el proceso desde cero.

Proceso de Descarga y Preparación de los Datos

Dado que el portal del MEC permite filtrar los datos por campos específicos, como la clasificación del campo amplio en el RNC, se identificó que el enlace para descargar datos filtrados no funcionaba correctamente. Por lo tanto, se optó por descargar ambos datasets en su totalidad, sin aplicar filtros previos. Esta decisión fue motivada por la variabilidad en la denominación de las carreras, especialmente en áreas como la salud, donde una misma área de conocimiento puede estar representada por múltiples nombres (por ejemplo, "Medicina" y "Cirugía"). Filtrar antes de la descarga no se consideró una opción viable debido a la heterogeneidad en la nomenclatura de las carreras.

Filtrado de Datos Relacionados con el Área de Salud

Una vez descargados los datasets completos (RNC y RTES), se procedió a filtrar los registros relevantes para el estudio, específicamente aquellos relacionados con el área de salud. Este proceso se realizó mediante algoritmos programados en Python utilizando las bibliotecas Polars y JSON, así como herramientas estándares como CSV.

- Filtrado de Títulos del RTES

Para identificar los títulos relacionados con el área de salud, se implementó un script en Python que aplicó un enfoque basado en palabras clave a través de los siguientes pasos:

- Se creó una lista de palabras clave asociadas con el área de salud, como "SALUD", "MEDIC", "ENFERM", "ODONTO", "FARMAC", entre otras.
- Simultáneamente, se definió una lista de palabras excluidas para evitar la inclusión de carreras no relacionadas directamente con la salud, como "VETERINARIA", "ADMINISTRA", "EDUCAC", y "TURISMO".
- El archivo JSON correspondiente al RTES (**registros_titulos.json**) fue cargado y convertido en un DataFrame utilizando la biblioteca Polars, lo que

permitió manipular los datos de manera eficiente y realizar operaciones de filtrado avanzado.

- Se desarrolló una función lógica (**filtrar_carrera_salud**) que evaluaba si el nombre de una carrera contenía alguna de las palabras clave definidas y no incluía ninguna de las palabras excluidas. La misma se aplicó a cada registro del DataFrame utilizando el método. **apply()**, lo que permitió identificar y retener únicamente los registros relacionados con el área de salud.
- Para facilitar el análisis posterior y evitar problemas de formato, todas las comas presentes en los datos textuales fueron reemplazadas por guiones utilizando el método **.str.replace_all()** de Polars.
- Los registros filtrados fueron exportados a un archivo CSV (**titulosSalud.csv**) para su uso en etapas posteriores del análisis.
- Además, se generaron resúmenes descriptivos, como el número total de registros filtrados y ejemplos de las carreras únicas encontradas, para validar la efectividad del proceso.

- Filtrado de Carreras del RNC

El filtrado de carreras del RNC se realizó mediante un enfoque similar pero adaptado a la estructura específica de este dataset. Los pasos principales fueron:

- El archivo JSON correspondiente al RNC (**carreras.json**) fue cargado y validado para asegurar que contenía una lista de registros.
- Se seleccionaron únicamente los registros cuyo campo **clasificacion_campo_amplio** tenía el valor "Salud y bienestar". Este campo está basado en la Clasificación Internacional Normalizada de Educación de la UNESCO.
- Los registros filtrados se exportaron a un archivo CSV (**carrerasSalud.csv**) para su posterior análisis. Además, se generaron ejemplos descriptivos de las carreras filtradas, mostrando detalles como la denominación de la carrera, el título otorgado, la institución correspondiente y la clasificación detallada.
- El proceso de filtrado fue validado mediante la revisión manual de una muestra aleatoria de los registros resultantes, asegurando que los criterios de inclusión y exclusión fueran aplicados correctamente. Además, se verificó que los archivos CSV generados no contuvieran errores de formato o inconsistencias.

Agrupamiento de Carreras por Temática

Una vez filtrados los títulos y las carreras relacionadas con el área de salud, se identificó que muchas de estas carreras compartían temáticas similares, pero presentaban variaciones significativas en su denominación. Este fenómeno dificultaba la caracterización y análisis de los datos debido a la heterogeneidad en la nomenclatura, por tanto, a fin de abordar este desafío, se implementó un proceso de agrupamiento temático

mediante un algoritmo programado en Python permitiendo una mejor organización y descripción de los datos lo que facilitó la interpretación de los resultados y proporcionó una visión más estructurada del panorama de carreras relacionadas con la salud.

- Se crearon 10 grupos temáticos basados en áreas específicas dentro del campo de la salud. Cada grupo fue definido mediante una lista exhaustiva de palabras clave asociadas con las temáticas correspondientes.
 - Medicina General y Especialidades Médicas
 - Cirugía y Especialidades Quirúrgicas
 - Enfermería y Cuidados Intensivos
 - Cardiología y Ciencias del Corazón
 - Ginecología - Obstetricia y Salud Reproductiva
 - Psicología y Psiquiatría
 - Odontología y Especialidades Dentales
 - Ciencias Biomédicas y Laboratorio
 - Imagenología y Diagnóstico
 - Nutrición - Farmacia y Otras Ciencias de la Salud
- Para garantizar la consistencia en el proceso de asignación de grupos, se implementó una función de normalización de texto (**normalizar_texto**) cuyo objetivo fue la eliminación de acentos y caracteres especiales, la conversión de todo el texto a minúsculas, la remoción de símbolos y espacios adicionales. Con ello se aseguró que las comparaciones entre las denominaciones de las carreras y las palabras clave fueran insensibles a diferencias tipográficas o estilísticas.
- Se desarrolló una función (**asignar_grupo**) que evaluaba cada carrera en función de las palabras clave definidas para cada grupo. El proceso incluyó coincidencias exactas entre la denominación de la carrera y las palabras clave, coincidencias parciales, priorizando términos más específicos o largos, la asignación de un puntaje a cada grupo en función de la relevancia de las coincidencias encontradas. Si no se encontraron coincidencias significativas, la carrera fue clasificada en la categoría "Otros".
- El archivo CSV resultante de la Fase 2 (**titulosSalud.csv**) fue cargado utilizando el módulo estándar csv de Python, se verificó la existencia de la columna carrera, necesaria para el agrupamiento, se añadió una nueva columna (**carreraGrupo**) al archivo, donde se registró el grupo temático asignado a cada carrera.
- Los datos procesados fueron exportados a un nuevo archivo CSV (**titulosSalud_agrupado.csv**), que incluía la columna adicional con el grupo asignado. Además, se generaron estadísticas descriptivas para validar la distribución de las carreras entre los grupos temáticos.

Completado del Campo Sexo

Una vez agrupadas las carreras por temáticas específicas, se identificó que varios registros en el dataset presentaban valores vacíos o nulos en el campo sexo. Para abordar este problema, se implementó un proceso automatizado basado en la inferencia de género a partir del nombre completo de los egresados, utilizando la biblioteca **gender-guesser**.

El objetivo principal de esta fase fue completar los valores faltantes en el campo sexo, minimizando la cantidad de registros incompletos para garantizar la integridad de los datos y facilitar análisis posteriores desagregados por género.

- Se desarrolló una función (**normalizar_texto**) para estandarizar los nombres completos antes del procesamiento. Con esta función se eliminaron los acentos y caracteres especiales, y el texto se convirtió a minúscula.
- Se implementó una función (**obtener_primer_nombre**) para extraer el primer nombre del campo **nombre_completo**, ya que la inferencia de género se basa principalmente en el primer nombre. Los resultados posibles incluyeron:
 - HOMBRE: Si el género detectado es male o mostly_male.
 - MUJER: Si el género detectado es female o mostly_female.
 - No determinado: Si el género es unknown o andy (andrógino).
- En casos donde no fue posible determinar el género con el primer nombre, se intentó inferirlo utilizando el segundo nombre del campo **nombre_completo**.
- El archivo CSV resultante de la Fase 3 (**titulosSalud_agrupado.csv**) fue cargado utilizando el módulo estándar csv de Python.
- Se verificó la existencia de las columnas necesarias (**nombre_completo y sexo**) para garantizar la viabilidad del proceso.
- Se recorrieron todos los registros para identificar aquellos con el campo sexo vacío o nulo. Para cada uno de estos registros, se aplicó el proceso de inferencia descrito anteriormente.
- Durante el procesamiento, se generaron estadísticas para evaluar la efectividad del método, como el número total de registros con el campo sexo vacío, los registros completados automáticamente, los registros que no pudieron ser determinados automáticamente. Finalmente, se calculó la distribución final de géneros en el dataset, mostrando el porcentaje de hombres, mujeres y registros no determinados.

A pesar de la automatización, algunos registros permanecieron sin asignación de género debido a limitaciones inherentes al método de inferencia (por ejemplo, nombres ambiguos o atípicos). Estos registros fueron completados manualmente para asegurar que no quedaran valores faltantes en el campo sexo.

- Los datos procesados fueron exportados a un nuevo archivo CSV (**titulosSalud_completo.csv**), que incluía el campo sexo actualizado. Al finalizar esta fase, se logró reducir el número de registros con valores faltantes en el campo sexo, mejorando la calidad y utilidad del dataset para análisis

posteriores. Los registros residuales que no pudieron ser completados automáticamente fueron abordados manualmente, garantizando la integridad de los datos.

Incorporación del Campo Tipo de Gestión

Una vez completado el campo sexo, se procedió a incorporar un nuevo campo al dataset: **tipo_gestion**. Este campo tiene como objetivo clasificar las instituciones educativas según su gestión administrativa (pública o privada).

El campo fue obtenido mediante la integración de datos entre dos fuentes: El archivo **titulosSalud_completo.csv**, que contiene información sobre los títulos otorgados por diversas instituciones y el archivo **carrerasSalud.csv**, que incluye detalles sobre las carreras ofertadas por instituciones reconocidas por el Consejo Nacional de Educación Superior (CONES), ambos ya filtrados según la metodología anteriormente descripta.

- Para garantizar la consistencia en la comparación de nombres de instituciones, se implementó una función de normalización intensiva (**normalizar_texto**). Con ello se eliminaron los acentos y caracteres especiales, se convirtió el texto a minúscula, pero además se removieron las palabras comunes que no aportan significado (por ejemplo, "Universidad", "Instituto", "Facultad").
- Se crearon dos mapas auxiliares para facilitar la asignación del tipo de gestión un mapa de instituciones que relacionó los nombres originales de las instituciones con su tipo de gestión (pública o privada) y un mapa de nombres normalizados que asoció las versiones normalizadas de los nombres de las instituciones con sus nombres originales.
- El proceso de asignación siguió tres niveles de coincidencia, la exacta, la normalizada y la de contención, siendo esta última producto de la verificación del nombre de la institución contenido dentro de otro nombre registrado.
- Los datos procesados fueron exportados a un nuevo archivo CSV (**titulos_con_gestion.csv**), que incluyó el campo **tipo_gestion** actualizado.

Limpieza de Formatos

Dicha fase consistió en la limpieza de formatos del dataset para garantizar la consistencia y calidad de los datos. Este paso fue crucial para eliminar errores estructurales, corregir caracteres especiales mal codificados y estandarizar campos clave como fecha resolución, además de realizar ajustes finales en el agrupamiento de carreras basado en similitud semántica.

- El campo **fecha_resolucion** presentaba inconsistencias en su formato, con fechas expresadas en diferentes patrones (por ejemplo, yyyy-mm-dd, dd/mm/yyyy, entre otros).

- Se desarrolló una función (**normalizar_fecha**) para estandarizar todas las fechas al formato dd/mm/yyyy. Esta función identificó patrones comunes utilizando expresiones regulares, convirtió los formatos alternativos (como yyyy-mm-dd o dd-mm-yyyy) al formato estándar y validó la estructura de fechas para asegurar su coherencia antes de guardarlas.
- Para abordar inconsistencias en la denominación de carreras dentro del campo **carreraGrupo**, se implementó un proceso de agrupamiento basado en similitud semántica. Se creó un diccionario de correcciones específicas para los grupos de carreras, se utilizó una función (**corregir_carrera_grupo**) para detectar coincidencias parciales y corregir nombres truncados o mal escritos.
- Los nombres de instituciones también presentaban inconsistencias, incluyendo variantes mal codificadas o abreviaturas incorrectas.
- Se desarrolló una función (**corregir_institucion**) que utilizó una lista oficial de instituciones para normalizar los nombres. Esto incluyó las coincidencias exactas con nombres oficiales, aquellas basadas en prefijos o palabras clave significativas y la corrección de nombres truncados o mal escritos utilizando técnicas de comparación de cadenas.
- El archivo CSV resultante de la fase anterior (**titulos_con_gestion.csv**) fue cargado utilizando métodos robustos que permitieron manejar diferentes codificaciones (UTF-8 y Latin-1).
- Los datos procesados fueron exportados a un nuevo archivo CSV (**titulos_final.csv**), que incluía los campos actualizados.

Fase de Análisis y Visualización: Creación de Tablas Dinámicas y Gráficos

Tras completar el proceso de limpieza y estandarización del dataset, se procedió al análisis de los datos utilizando Microsoft Excel como herramienta principal, dicha elección se debió a la versatilidad y facilidad de uso que ofrece Excel para la creación de tablas dinámicas y la presentación visual de los resultados.

Las tablas dinámicas fueron fundamentales para explorar patrones clave en los datos, permitiendo desglosar la información en función de variables como año, género, institución y tipo de gestión. Estas tablas facilitaron la identificación de tendencias temporales, diferencias entre géneros y la distribución de títulos otorgados por instituciones públicas y privadas. Además, se utilizaron para analizar la evolución de las carreras relacionadas con el área de salud, proporcionando una visión clara de cómo ha cambiado la oferta educativa a lo largo del tiempo.

Para la presentación de los resultados, Excel se utilizó para generar gráficos y tablas resumen que destacaran los hallazgos más relevantes. Estos elementos visuales permitieron comunicar de manera efectiva las conclusiones del estudio, facilitando su interpretación por parte de diferentes audiencias.

Las tablas dinámicas fueron una herramienta fundamental para explorar los datos de manera estructurada y eficiente, a través de estas se desglosó la información en función de múltiples variables, como año de egreso, género, institución educativa, tipo de gestión (pública o privada), grupo temático de carreras y nivel de titulación.

El archivo CSV final fue importado a Excel, asegurando que todos los campos estuvieran correctamente formateados, se verificó la integridad de los datos, asegurando que no hubiera valores faltantes o inconsistentes en las columnas clave.

Se crearon múltiples tablas dinámicas para analizar diferentes aspectos de los datos. Cada tabla dinámica se configuró con filas, columnas y valores específicos, dependiendo del análisis requerido.

Una vez configuradas las tablas dinámicas, se procedió a la creación de gráficos y visualizaciones que permitieran comunicar los hallazgos de manera clara y efectiva. Los gráficos se diseñaron para resaltar patrones, tendencias y diferencias significativas en los datos.

Se utilizaron gráficos de barras para comparar la distribución de títulos entre diferentes categorías, como género, tipo de gestión y grupos temáticos. Los gráficos de líneas fueron ideales para representar tendencias temporales. Por ejemplo, se creó un gráfico de líneas que mostraba la evolución del número de títulos otorgados en Medicina General y Enfermería a lo largo de los años.

Fase de desarrollo del Dashboard Interactivo

Tras la creación de tablas dinámicas y visualizaciones estáticas en Excel, se procedió al diseño e implementación de un dashboard interactivo utilizando tecnologías web (HTML, CSS y JavaScript) que permitiera una exploración dinámica y en profundidad de los datos del área de salud, esquema conceptualizado como una herramienta central para la visualización y el análisis, facilitando la interacción directa con los datos por parte de los usuarios.

El proceso de desarrollo comenzó con la estructuración del diseño, dividiendo la interfaz en dos secciones principales: una barra lateral dedicada a los filtros interactivos y un área principal para la presentación de indicadores clave y visualizaciones. Se implementó un sistema de filtros interconectados que permite al usuario realizar selecciones por rangos de años, tipo de institución, carrera, tipo de gestión y género, posibilitando un análisis multidimensional de los datos.

Para las visualizaciones dinámicas, se utilizó la biblioteca Chart.js que permitió crear representaciones gráficas interactivas y responsivas, además se presentaron elementos visuales estratégicamente seleccionados: gráficos circulares para la distribución por tipo de institución y género, gráficos de líneas para la evolución temporal de títulos, gráficos

de barras para la distribución por género según año, visualizaciones horizontales para el ranking de top 10 carreras e instituciones, y un gráfico de tendencia mensual para detectar patrones estacionales.

Se prestó especial atención a la arquitectura técnica del dashboard, implementando un sistema de carga y procesamiento de datos que permite trabajar eficientemente con el conjunto completo de registros. Esto incluyó la creación de funciones para la lectura del archivo Excel, validación de la estructura de datos, transformación y agregación de la información según los filtros aplicados, y actualización dinámica de todas las visualizaciones sin necesidad de recargar la página.

La interfaz de usuario fue diseñada priorizando la experiencia del usuario, con un enfoque en la accesibilidad y facilidad de interpretación, se incluyeron tarjetas de estadísticas generales que muestran métricas clave como el total de títulos, distribución por género y número de carreras únicas, proporcionando un contexto inmediato al usuario antes de profundizar en las visualizaciones detalladas. Adicionalmente, se implementó un sistema de respuesta adaptativa que optimiza la presentación del dashboard en diferentes dispositivos y tamaños de pantalla.

Por último, se incorporaron mecanismos de validación y manejo de errores para garantizar la robustez del dashboard ante situaciones imprevistas, como problemas en la carga de datos o selecciones de filtros que pudieran resultar en conjuntos vacíos. Esto incluyó la implementación de mensajes informativos para el usuario y opciones de recuperación, asegurando una experiencia fluida durante el análisis de los datos del sector académico de salud.

RESULTADOS Y ANALISIS

ANÁLISIS DE LA OFERTA

En el contexto general del sistema educativo superior paraguayo, es notable que del total de 27,715 programas formativos registrados, 4,532 corresponden específicamente al campo amplio de "Salud y Bienestar", lo que representa un 16.4% de la oferta académica total del país. Esta significativa proporción subraya la relevancia del sector salud en el panorama formativo nacional, posicionándolo como uno de los campos disciplinares con mayor presencia en el sistema educativo.

La oferta académica total en el área de salud asciende a 4,532 programas formativos, distribuidos entre cuatro tipos principales de instituciones educativas. Las Universidades concentran la mayor proporción con 2,067 ofertas (45.6% del total), seguidas por los Institutos Técnicos Superiores con 1,737 ofertas (38.3%), los Institutos Superiores con 682 ofertas (15.0%), y finalmente los Institutos de Formación Docente con 46 ofertas (1.0%).

Tabla 1. Número de ofertas de títulos de carreras por tipo institución y nivel de titulación.

Tipo Institución/Nivel de Titulación	Oferta Académica Salud
Instituto de Formación Docente	46
Educación de Pre-Grado - Educación Técnica Superior (Formación inicial)	6
NO DEFINIDO	40
Instituto Superior	682
Educación de Grado	74
Educación de Posgrado (Especialización)	18
Educación de Posgrado (Maestría)	7
Educación de Pre-Grado	3
Educación de Pre-grado - Educación Técnica Superior (Capacitaciones)	1
Educación de Pre-Grado - Educación Técnica Superior (Formación inicial)	579
Instituto Técnico Superior	1737
Educación de Posgrado (Especialización)	2
Educación de Pre-Grado	8
Educación de Pre-grado - Educación Técnica Superior (Capacitaciones)	5
Educación de Pre-Grado - Educación Técnica Superior (Formación inicial)	1715
NO DEFINIDO	7
Universidad	2067
Educación de Grado	848
Educación de Posgrado (Doctorado)	23
Educación de Posgrado (Especialización)	418
Educación de Posgrado (Maestría)	78
Educación de Pre-Grado	3
Educación de Pre-grado - Educación Técnica Superior (Capacitaciones)	1
Educación de Pre-Grado - Educación Técnica Superior (Formación inicial)	696
Total General	4532

Respecto a los niveles de titulación, se observa un claro predominio de los programas de Educación Técnica Superior en la modalidad de formación inicial, que representan 2,990 ofertas (66.0% del total). Este tipo de formación constituye el núcleo principal de la oferta en Institutos Técnicos Superiores (1,715 programas) e Institutos Superiores (579 programas), siendo también significativa en Universidades (696 programas).

Las ofertas de Educación de Grado suman 922 programas (20.3% del total), concentrándose principalmente en las Universidades (848 programas). Los programas de posgrado en sus diferentes modalidades (Especialización, Maestría y Doctorado) representan 546 ofertas (12.0% del total), destacándose las Especializaciones con 438 programas, seguidas por las Maestrías (85 programas) y los Doctorados (23 programas), estos últimos ofrecidos exclusivamente por Universidades.

Del total de 4,532 ofertas académicas en el área de salud, se observa un marcado predominio del sector privado, que concentra 4,106 ofertas (90.6% del total). En contraste, el sector oficial representa solo 375 ofertas (8.3%), mientras que las instituciones de gestión privada subvencionada aportan apenas 51 ofertas (1.1%).

En el sector oficial (375 ofertas), las Universidades son las principales proveedoras con 333 programas (88.8% de la oferta oficial), seguidas por los Institutos de Formación Docente con 40 programas (10.7%) y una presencia mínima de Institutos Técnicos Superiores, con solo 2 programas (0.5%).

El sector privado (4,106 ofertas) presenta una distribución más equilibrada entre diferentes tipos de instituciones: las Universidades ofrecen 1,734 programas (42.2% de la oferta privada), los Institutos Técnicos Superiores aportan 1,684 programas (41.0%), los Institutos Superiores contribuyen con 682 programas (16.6%), y los Institutos de Formación Docente con solo 6 programas (0.1%).

En cuanto al sector privado subvencionado (51 ofertas), este está representado exclusivamente por Institutos Técnicos Superiores, sin participación de otros tipos de instituciones educativas.

El sistema de educación superior paraguayo cuenta con un total de 122 instituciones que ofrecen formación en el área de salud, distribuidas en cuatro categorías: 63 Institutos Técnicos Superiores (51.6%), 48 Universidades (39.3%), 9 Institutos Superiores (7.4%) y 2 Institutos de Formación Docente (1.6%).

Tabla 2. Número de ofertas de títulos de carreras por tipo de gestión y nivel de titulación.

Tipo de Gestión/Tipo Institución	Oferta Académica Salud
Oficial	375
Instituto de Formación Docente	40
Instituto Técnico Superior	2
Universidad	333
Privado	4106
Instituto de Formación Docente	6
Instituto Superior	682
Instituto Técnico Superior	1684
Universidad	1734
Privado Subvencionado	51
Instituto Técnico Superior	51
Total General	4532

Es importante destacar que esta cifra de 122 instituciones es el resultado de un proceso de limpieza y estandarización de datos. Inicialmente, se habían identificado 140 instituciones diferentes en el conjunto de datos sin procesar. Sin embargo, al realizar un análisis más detallado, se detectó que varias de estas entradas correspondían a la misma institución, pero aparecían como entidades separadas debido a pequeñas diferencias semánticas en su registro.

Tabla 3. Número de instituciones por tipo de institución.

Tipo Institución	Nro. Instituciones
Instituto de Formación Docente	2
Instituto Superior	9
Instituto Técnico Superior	63
Universidad	48
Total General	122

Tomando en consideración las 20 instituciones con mayor oferta académica en salud, se observa que el Instituto Superior en Ciencias de la Salud Juan Pablo II lidera claramente la lista con 395 programas, seguido por la Universidad Autónoma del Sur con 356 programas y la Universidad Privada del Guairá con 266 programas.

Un hallazgo significativo es la marcada presencia de institutos frente a universidades en esta clasificación. De las 20 instituciones listadas, 14 corresponden a institutos (70%) mientras que solo 6 son universidades (30%).

Esta tendencia es particularmente notable considerando que, en el análisis global del sistema previamente realizado, se identificaron 48 universidades y 74 institutos (sumando todas las categorías de institutos). Esto sugiere una mayor concentración de la oferta académica en salud en un número reducido de institutos especializados, mientras que la oferta universitaria parece estar más distribuida entre un mayor número de instituciones.

La descripción de la institución con mayor oferta se concentra significativamente en cinco áreas principales que representan el 78.7% del total de su oferta. Farmacia encabeza claramente esta distribución con 110 programas (27.8%), seguida por Radiología con 64 programas (16.2%), Masaje Terapéutico con 55 programas (13.9%), Enfermería con 51 programas (12.9%) e Instrumentación Quirúrgica con 32 programas (8.1%). Le sigue de cerca Laboratorio Clínico con 31 programas (7.8%), mientras que las restantes 14 áreas formativas se reparten apenas 55 programas (13.9% del total).

Tabla 4. Número de ofertas de títulos de carreras por institución

Nombre de la Institución	Nro. Ofertas
INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD JUAN PABLO II	395
UNIVERSIDAD AUTÓNOMA DEL SUR	356
UNIVERSIDAD PRIVADA DEL GUAIRA	266
UNIVERSIDAD NACIONAL DE ASUNCIÓN	197
UNIVERSIDAD CATÓLICA NUESTRA SEÑORA DE LA ASUNCIÓN	183
INSTITUTO SUPERIOR PROFESIONAL AVANZADO	136
INSTITUTO TÉCNICO SUPERIOR SAN IGNACIO DE LOYOLA	134
UNIVERSIDAD LA PAZ	134
INSTITUTO TÉCNICO SUPERIOR ACADEMO	133
UNIVERSIDAD ADVENTISTA DEL PARAGUAY	114
INSTITUTO SUPERIOR SAN PATRICIO DE IRLANDA DEL NORTE	92
INSTITUTO TÉCNICO SUPERIOR PROFESIONAL DE EDUCACIÓN	91
INSTITUTO TÉCNICO SUPERIOR MARÍA AUXILIADORA	79
INSTITUTO TÉCNICO SUPERIOR DEL SABER	76
UNIVERSIDAD POLITÉCNICA Y ARTÍSTICA DEL PARAGUAY	75
INSTITUTO TÉCNICO SUPERIOR JESÚS DE NAZARET	66
INSTITUTO TÉCNICO SUPERIOR GAMMA	59
CENTRO EDUCATIVO SUPERIOR EN SALUD	56
INSTITUTO TÉCNICO SUPERIOR NUESTRA SEÑORA DE FÁTIMA	56
INSTITUTO TÉCNICO SUPERIOR SAN ROQUE GONZÁLEZ DE SANTA CRUZ	54

Esta distribución revela una clara especialización institucional hacia áreas técnicas y de apoyo clínico dentro del campo de la salud, particularmente en farmacia y radiología que juntas representan el 44% de su oferta total. Es notable la baja presencia de programas en áreas como psicología (solo 4 programas sumando Psicología Clínica y Psicología general) o nutrición (5 programas).

Resulta interesante observar que existen algunas superposiciones o posibles duplicaciones en la clasificación de carreras, como "KINESIOLOGÍA Y FISIOTERAPIA" (4 programas) y "FISIOTERAPIA Y KINESIOLOGÍA" (3 programas), así como "PODOLOGÍA" que aparece dos veces con 5 y 4 programas respectivamente. Esto podría reflejar las dificultades en la estandarización de datos mencionadas en la problemática de investigación.

La predominancia de programas de farmacia (110) podría estar relacionada con la creciente demanda de técnicos y profesionales farmacéuticos tanto en el mercado paraguayo como potencialmente en el brasileño, considerando el fenómeno transfronterizo mencionado anteriormente en la problemática de investigación.

Del total de 4,532 programas formativos en salud, se observa una marcada concentración en determinados departamentos. Capital (Asunción) lidera claramente con 947 programas (20.9% del total), seguida por Central con 868 programas (19.2%), y en tercer lugar Alto Paraná con 409 programas (9.0%). Estos tres departamentos concentran casi la mitad (49.1%) de toda la oferta académica en salud del país, lo que refleja un importante desequilibrio territorial en la distribución de oportunidades formativas.

En cuanto a los campos detallados, Medicina muestra la mayor presencia con 1,239 programas (27.3% del total), seguido por Enfermería y partería con 924 programas (20.4%) y Farmacia con 897 programas (19.8%). Estos tres campos representan más de dos tercios (67.5%) de la oferta total en salud. En contraste, campos como Asistencia a la infancia y servicios para jóvenes (4 programas), Medicina tradicional y complementaria (4 programas) y Trabajo social y orientación (37 programas) muestran una presencia mínima.

La distribución de los campos formativos varía significativamente entre departamentos. En Capital, Medicina domina con 272 programas, mientras que en Central también predomina Medicina con 317 programas. En Alto Paraná, Medicina lidera igualmente con 124 programas. Sin embargo, en Itapúa se observa una mayor concentración en Farmacia (115 programas), mientras que en Boquerón predomina Enfermería y partería (115 programas).

Tabla 5. Número de ofertas de títulos de carreras por Departamento y Campo Detallado

DEPARTAMENTO/ CAMPO DETALLADO	Asistencia a la infancia y servicios para jóvenes	Enfermería y partería	Farmacia	Medicina	Medicina y terapia tradicional y complementaria	Odontología	Tecnología de diagnóstico y tratamiento médico	Terapia y rehabilitación	Trabajo social y orientación	Total/ Departamento
Alto Paraná	*	60	40	124	*	39	71	75	*	409
Amambay	*	11	3	24	*	8	8	3	*	57
Boqueron	*	115	1	1	*	*	1	1	*	119
Caaguazú	*	43	32	91	1	18	51	60	*	296
Caazapá	*	12	36	11	*	*	38	5	1	103
Canindeyú	*	29	7	12	*	6	5	3	*	62
Capital	2	116	213	272	2	156	70	111	5	947
Central	1	140	192	317	*	17	85	104	12	868
Concepción	*	26	16	35	*	7	8	10	4	106
Cordillera	*	45	31	50	*	4	20	14	1	165
Guairá	*	12	18	112	1	14	13	6	*	176
Itapúa	1	88	115	68	*	10	56	26	5	369
Misiones	*	15	42	27	*	*	13	7	5	109
Ñeembucú	*	7	10	5	*	7	1	11	1	42
Paraguarí	*	98	71	27	*	40	58	70	2	366
Presidente Hayes	*	5	1	3	*	*	1	1	*	11
San Pedro	*	102	69	60	*	6	72	17	1	327
Total General	4	924	897	1239	4	332	571	524	37	4532

ANÁLISIS DE LOS EGRESADOS

Del total de 604,722 títulos otorgados en el sistema educativo superior paraguayo, 128,228 corresponden a carreras del área de salud relacionadas con la formación de profesionales capacitados para brindar atención directa a pacientes humanos, lo que representa el 21.2% del total de egresados del país.

Comportamiento ascendente de titulados (2012-2024) con caída durante pandemia

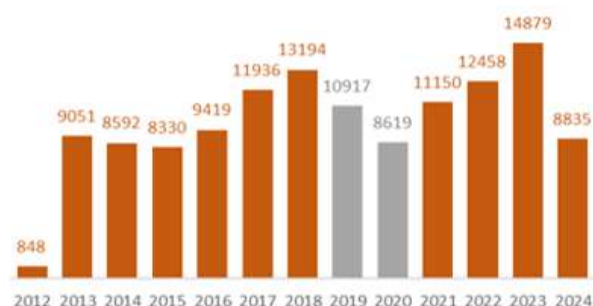


Figura 1. Dinámica de titulación (2012-2024)

El análisis de estos datos revela una marcada predominancia femenina constante en la obtención de títulos del área de salud a lo largo de todo el período estudiado (2012-2024), con un porcentaje del 75.7% correspondiente a mujeres frente al 24.3% de hombres.

En 2012, año inicial del registro, se observa una diferencia ya significativa con 670 títulos otorgados a mujeres frente a 178 a hombres, estableciendo una proporción de 3.8 mujeres por cada hombre. Esta brecha se amplía notablemente en 2013, cuando se registra un salto cuantitativo importante en ambos géneros (7,267 mujeres y 1,784 hombres), manteniendo una proporción aproximada de 4:1.

Clara Supremacía Femenina en Carreras de Salud 3 de cada 4 egresados son mujeres, manteniendo una tendencia histórica de dominancia en el sector



Figura 2. Comportamiento de titulación por género (2012-2024).

La tendencia general muestra un crecimiento sostenido, aunque con fluctuaciones para ambos géneros hasta 2018, año que marca un pico intermedio (10,207 mujeres y 2,987 hombres). Posteriormente, se observa una caída en 2019 y una reducción más pronunciada en 2020, coincidiendo probablemente con el impacto de la pandemia de COVID-19. Específicamente, en 2020 se registra una disminución significativa hasta

6,411 títulos para mujeres y 2,208 para hombres, representando una caída del 21.5% y 19.8% respectivamente respecto al año anterior.

A partir de 2021 se evidencia una recuperación que culmina en 2023 con los valores máximos de toda la serie histórica: 10,868 títulos otorgados a mujeres y 4,011 a hombres. Este año representa el punto más alto tanto en términos absolutos como en la proporción de participación masculina, que alcanza el 27% del total. Sin embargo, en 2024 se observa una caída pronunciada para ambos géneros, retornando a niveles similares a los de 2015, lo que podría indicar un proceso de ajuste del sistema educativo o cambios en las políticas de admisión y egreso.

A lo largo de todo el período analizado, las mujeres han obtenido sistemáticamente entre el 71% y el 79% del total de títulos en el área de salud, confirmando una feminización persistente de este campo profesional en Paraguay. Esta tendencia es consistente con patrones observados en otros países latinoamericanos y podría estar relacionada con factores socioculturales asociados a los roles de cuidado tradicionalmente asignados a las mujeres, así como con transformaciones en el acceso femenino a la educación superior.

Del total de 128,228 títulos otorgados en el área de salud, las Universidades concentran la mayor proporción con 94,809 títulos (73.9%), seguidas por los Institutos Superiores con 25,601 títulos (20.0%), los Institutos Técnicos Superiores con 7,161 títulos (5.6%), y finalmente los Institutos de Formación Docente con apenas 657 títulos (0.5%).

Tabla 6. Número de títulos por Tipo de Institución y Sexo.

Tipo Intitucion/Sexo	Nro. Titulos
INSTITUTO DE FORMACIÓN DOCENTE	657
HOMBRE	104
MUJER	553
INSTITUTO SUPERIOR	25601
HOMBRE	5892
MUJER	19709
INSTITUTO TÉCNICO SUPERIOR	7161
HOMBRE	1574
MUJER	5587
UNIVERSIDAD	94809
HOMBRE	23607
MUJER	71202
Total General	128228

La marcada predominancia femenina observada anteriormente (75.7% del total) se mantiene consistente a través de todos los tipos de instituciones, aunque con algunas variaciones en la intensidad de esta feminización. En los Institutos de Formación Docente, las mujeres representan el 84.2% de los títulos otorgados (553 de 657), estableciendo la mayor proporción femenina entre todas las categorías institucionales. En las Universidades, las mujeres obtuvieron el 75.1% de los títulos (71,202 de 94,809), mientras que en los Institutos Superiores alcanzaron el 77.0% (19,709 de 25,601) y en los Institutos Técnicos Superiores el 78.0% (5,587 de 7,161).

Resulta interesante contrastar estos datos de egreso con la información previamente analizada sobre la oferta académica. Mientras que las Universidades representan el 45.6% de la oferta académica en salud (2,067 de 4,532 programas), generan el 73.9% de los títulos otorgados. Por el contrario, los Institutos Técnicos Superiores constituyen el 38.3% de la oferta (1,737 programas), pero solo producen el 5.6% de los títulos. Esta disparidad sugiere importantes diferencias en la eficiencia terminal o en la escala de las matriculaciones entre distintos tipos de instituciones.

La participación femenina es ligeramente superior en instituciones no universitarias (Institutos Superiores, Técnicos Superiores y de Formación Docente) con un promedio ponderado de 77.3%, frente al 75.1% en Universidades. Esta pequeña diferencia podría estar relacionada con la composición de la oferta académica en cada tipo de institución, particularmente considerando que ciertos campos tradicionalmente más masculinizados como medicina especializada o cirugía suelen concentrarse en el ámbito universitario.

En la distribución por grupos de carreras, Enfermería y Cuidados Intensivos lidera notablemente con 47,370 títulos (36.9% del total en salud), seguido por Medicina General y Especialidades Médicas con 26,548 títulos (20.7%) y Nutrición - Farmacia y Otras Ciencias de la Salud con 25,167 títulos (19.6%). Estos tres grupos concentran el 77.1% de todos los títulos otorgados en el área de salud.

En un nivel intermedio se encuentran Psicología y Psiquiatría con 10,982 títulos (8.6%), Imagenología y Diagnóstico con 8,690 títulos (6.8%) y Odontología y Especialidades Dentales con 5,122 títulos (4.0%). Los grupos con menor número de títulos incluyen Ginecología, Obstetricia y Salud Reproductiva (2,755 títulos), Cirugía y Especialidades Quirúrgicas (767 títulos), Ciencias Farmacéuticas y Bioquímicas (648 títulos), por último, Cardiología y Ciencias del Corazón (179 títulos)



Figura 3. Cantidad de titulados por grupo de carrera (2012-2024).

Esta distribución revela un patrón interesante en la formación de profesionales de la salud en Paraguay, con una clara predominancia de enfermería sobre medicina y otras especialidades. Esta configuración podría responder tanto a características estructurales

del sistema educativo como a necesidades específicas del mercado laboral sanitario, la notable diferencia entre el número de títulos en áreas generalistas como enfermería y medicina frente a especialidades más específicas como cardiología o cirugía refleja la estructura piramidal típica de los sistemas de salud, con una base amplia de profesionales de atención primaria y una cúspide más reducida de especialistas.

La distribución de títulos otorgados en el área de salud según el tipo de gestión institucional muestra un notable paralelismo con los patrones observados previamente en la oferta académica. Los datos revelan que, del total de 128,228 títulos otorgados, el 89.2% (114,412 títulos) proviene de instituciones privadas, mientras que solo el 9.6% (12,263 títulos) corresponde a instituciones oficiales, con un remanente de 1.2% (1,553 títulos) procedente de instituciones privadas clausuradas.

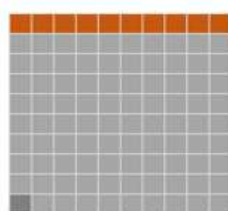
Esta marcada predominancia del sector privado en la emisión de títulos refleja casi exactamente la distribución de la oferta académica analizada anteriormente, donde se identificó que el 90.6% de los programas formativos en salud (4,106 de 4,532) pertenecían al sector privado, frente a solo 8.3% (375 programas) del sector oficial. Esta correspondencia casi perfecta entre los porcentajes de oferta y de títulos otorgados sugiere una eficiencia terminal similar entre ambos sectores, algo que no se observaba al analizar la distribución por tipo de institución (universidades versus institutos).

La coherencia entre ambas dimensiones (oferta y egreso) confirma la consolidación estructural de un sistema de formación en salud predominantemente privatizado en Paraguay. Este alineamiento entre oferta y producción de egresados indica que no existen diferencias significativas en la capacidad de retención o de titulación entre instituciones públicas y privadas, a pesar de las posibles diferencias en los procesos de selección, aranceles y mecanismos de apoyo estudiantil.

Es relevante destacar que este alto grado de privatización tanto en la oferta académica como en la emisión de títulos contrasta con la estructura de los sistemas sanitarios en la región, que suelen mantener un importante componente público. Esta situación plantea interrogantes sobre la accesibilidad a la formación en salud, la alineación entre los perfiles de egreso y las necesidades del sistema sanitario público, y los mecanismos de regulación y aseguramiento de la calidad en un sistema tan fuertemente inclinado hacia el sector privado.

La presencia de títulos emitidos por instituciones privadas clausuradas (1,553) añade una dimensión adicional al análisis, sugiriendo posibles vulnerabilidades en los mecanismos de supervisión y control de calidad

Predominio de la Educación Privada en la Formación Profesional
Solo el 10% de egresados proviene de instituciones públicas



Tipo Gestion	Nro.Titulos
Oficial	12263
Privado	114412
Privado-Clausurado	1553
Total General	128228

Figura 4. Cantidad de titulados por tipo de gestión.

Entre las carreras analizadas, Enfermería destaca notablemente como la principal fuente de profesionales de la salud, con 46,269 títulos otorgados, lo que representa más de un tercio del total de títulos en el sector. Esta predominancia de Enfermería refleja tanto la estructura piramidal típica de los sistemas sanitarios, con una amplia base de profesionales de atención primaria, como las tendencias globales en la configuración de los equipos de salud.

Medicina aparece en segundo lugar con 20,393 títulos, menos de la mitad de los otorgados en Enfermería. Esta proporción aproximada de 2.3 enfermeros por cada médico se alinea con las recomendaciones internacionales sobre la composición óptima de los equipos sanitarios, aunque con un ligero sesgo hacia la enfermería.

Tabla 7. Top 20 de las carreras con mayor número de títulos registrados

Carrera	Nro.Titulo
ENFERMERIA	46269
MEDICINA	20393
FARMACIA	8169
PSICOLOGIA	7382
RADIOLOGIA	7033
ODONTOLOGIA	4727
NUTRICION	3900
KINESIOLOGIA Y FISIOTERAPIA	2884
FISIOTERAPIA Y KINESIOLOGIA	2610
QUIMICA Y FARMACIA	2094
SALUD PUBLICA	1797
MEDICINA FAMILIAR Y COMUNITARIA	1624
PSICOLOGIA CLINICA	1615
OBSTETRICIA	1589
MEDICINA Y CIRUGIA	1410
RADIOLOGIA Y BIOIMAGENES	1184
FONOAUDIOLOGIA	762
MEDICINA INTERNA	758
PEDIATRIA	570
NUTRICION HUMANA	486

Las carreras relacionadas con el medicamento y productos farmacéuticos suman aproximadamente 10,263 títulos (combinando Farmacia y Química y Farmacia), posicionándose como el tercer gran bloque formativo. Las áreas de salud mental

(Psicología y Psicología Clínica) acumulan 8,997 títulos, mientras que el diagnóstico por imágenes (Radiología y Radiología y Bioimágenes) suma 8,217 títulos.

Es interesante observar que carreras tradicionalmente percibidas como prestigiosas, como Medicina y Cirugía (1,410 títulos), Medicina Interna (758 títulos) o Pediatría (570 títulos), ocupan posiciones relativamente bajas en la lista, lo que podría reflejar tanto la especialización posterior a la formación médica general como posibles barreras de acceso a estas especializaciones.

También resulta significativa la presencia de enfoques complementarios dentro de una misma disciplina, como se evidencia en la coexistencia de "Kinesiología y Fisioterapia" (2,884 títulos) y "Fisioterapia y Kinesiología" (2,610 títulos) como carreras aparentemente distintas. Esta duplicación podría ser indicio de las dificultades en la estandarización de denominaciones mencionadas en la problemática de investigación.

La identificación específica de "Medicina Familiar y Comunitaria" (1,624 títulos) como una carrera con entidad propia refleja la influencia de los modelos de atención basados en la atención primaria de la salud, mientras que la presencia de "Salud Pública" (1,797 títulos) indica cierto desarrollo de la formación en enfoques poblacionales y preventivos.

En el ámbito de los institutos superiores, que otorgaron 25,601 títulos (20.0% del total), el Instituto Superior Profesional Avanzado lidera claramente con 7,516 títulos (29.4% de esta categoría), seguido por el Instituto Superior en Ciencias de la Salud San Patricio de Irlanda del Norte con 5,769 títulos (22.5%) y el Instituto Superior en Ciencias de la Salud Santa Rosa Mística con 3,594 títulos (14.0%).

Tabla 8. Número de títulos de Institutos Superiores discriminado por nombre

Tipo Institucion/Institucion	Nro.Títulos
INSTITUTO DE FORMACIÓN DOCENTE	657
INSTITUTO DE FORMACIÓN DOCENTE LA PAZ	657
INSTITUTO SUPERIOR	25601
CENTRO EDUCATIVO SUPERIOR EN SALUD	2122
INSTITUTO NACIONAL DE SALUD	227
INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD JUAN PABLO II	1643
INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SAN AGUSTÍN	572
INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SAN PATRICIO DE IRLANDA DEL NORTE	5769
INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SANTA ROSA MÍSTICA	3594
INSTITUTO SUPERIOR INTERREGIONAL EN CIENCIAS DE LA SALUD	2125
INSTITUTO SUPERIOR LATINOAMERICANO	1018
INSTITUTO SUPERIOR PROFESIONAL AVANZADO	7516
INSTITUTO SUPERIOR SAN AGUSTÍN	466
INSTITUTO SUPERIOR SANTA LIBRADA	535
INSTITUTO TÉCNICO SUPERIOR MARÍA AUXILIADORA	14

Entre las universidades, que en conjunto otorgaron 94,809 títulos (73.9% del total), se destacan particularmente la Universidad Católica Nuestra Señora de la Asunción con 10,052 títulos (10.6% de los títulos universitarios), la Universidad del Norte con 7,648 títulos (8.1%), la Universidad Técnica de Comercialización y Desarrollo con 7,177 títulos (7.6%), la Universidad San Lorenzo con 5,707 títulos (6.0%) y la Universidad Nacional

de Asunción con 5,275 títulos (5.6%). Estas cinco universidades concentran aproximadamente el 37.9% de todos los títulos universitarios en salud.

Tabla 9. Número de títulos de Universidades discriminado por nombre

Tipo Institucion/Institucion	Nro.Titulos
UNIVERSIDAD	94809
UNIVERSIDAD CATÓLICA NUESTRA SEÑORA DE LA ASUNCIÓN	10052
UNIVERSIDAD DEL NORTE	7648
UNIVERSIDAD TÉCNICA DE COMERCIALIZACIÓN Y DESARROLLO	7177
UNIVERSIDAD SAN LORENZO	5707
UNIVERSIDAD NACIONAL DE ASUNCIÓN	5275
UNIVERSIDAD AUTÓNOMA DEL SUR	5266
UNIVERSIDAD POLITÉCNICA Y ARTÍSTICA DEL PARAGUAY	4586
UNIVERSIDAD MARÍA AUXILIADORA	3732
UNIVERSIDAD CENTRO MÉDICO BAUTISTA	3693
UNIVERSIDAD INTERNACIONAL TRES FRONTERAS	3266
UNIVERSIDAD PRIVADA DEL ESTE	3084
UNIVERSIDAD PRIVADA DEL GUAIRA	3063
UNIVERSIDAD DEL PACÍFICO	2591
UNIVERSIDAD NACIONAL DEL ESTE	2288
UNIVERSIDAD AUTÓNOMA DEL PARAGUAY	2234
UNIVERSIDAD TECNOLÓGICA INTERCONTINENTAL	1975
UNIVERSIDAD LA PAZ	1847
UNIVERSIDAD SANTA CLARA DE ASÍS	1695
UNIVERSIDAD SUDAMERICANA	1586
UNIVERSIDAD NACIONAL DE ITAPÚA	1516
UNIVERSIDAD DE INTEGRACIÓN DE LAS AMÉRICAS	1483
CENTRO EDUCATIVO SUPERIOR EN SALUD	1415
UNIVERSIDAD AUTÓNOMA SAN SEBASTIÁN	1407
UNIVERSIDAD GRAN ASUNCIÓN	1385
UNIVERSIDAD NACIONAL DE CAAGUAZÚ	1125
UNIVERSIDAD DEL SOL	1059
UNIVERSIDAD MARÍA SERRANA	974
UNIVERSIDAD CENTRAL DEL PARAGUAY	967
UNIVERSIDAD METROPOLITANA DE ASUNCIÓN	874
UNIVERSIDAD NACIONAL DE VILLARRICA DEL ESPÍRITU SANTO	829
UNIVERSIDAD IBEROAMERICANA	763
UNIVERSIDAD NACIONAL DE CONCEPCIÓN	756
UNIVERSIDAD AUTÓNOMA DE ASUNCIÓN	718
UNIVERSIDAD DEL CHACO	474
UNIVERSIDAD AMERICANA	425
UNIVERSIDAD NACIONAL DE PILAR	414
UNIVERSIDAD COLUMBIA DEL PARAGUAY	240
UNIVERSIDAD EVANGÉLICA DEL PARAGUAY	207
UNIVERSIDAD AUTÓNOMA DE ENCARNACIÓN	176
UNIVERSIDAD NIHON GAKKO	145
UNIVERSIDAD HISPANO GUARANÍ PARA EL DESARROLLO HUMANO	132
UNIVERSIDAD INTERAMERICANA	132
UNIVERSIDAD AUTÓNOMA DE LUQUE	100
UNIVERSIDAD ADVENTISTA DEL PARAGUAY	95
UNIVERSIDAD LEONARDO DA VINCI	75
UNIVERSIDAD NACIONAL DE CANINDEYÚ	60
UNIVERSIDAD DEL CONO SUR DE LAS AMÉRICAS	59
UNIVERSIDAD DE DESARROLLO SUSTENTABLE	25
UNIVERSIDAD SAN CARLOS	14

Es relevante observar que la Universidad Autónoma del Sur, que en análisis previos se identificó como la segunda institución con mayor oferta académica en salud (356 programas), también figura entre las principales emisoras de títulos con 5,266, evidenciando una correlación entre su amplia oferta y su producción de egresados.

Tabla 10. Número de títulos de Institutos Técnicos Superiores discriminado por nombre

Tipo Institucion/Institucion	Nro.Títulos
INSTITUTO TÉCNICO SUPERIOR	7161
ESCUELA PARAGUAYA DE ENFERMERÍA Y CIENCIAS DE LA SALUD	12
INSTITUTO DE FORMACIÓN TÉCNICA SUPERIOR INFORTES	133
INSTITUTO SUPERIOR INTERREGIONAL EN CIENCIAS DE LA SALUD	582
INSTITUTO SUPERIOR PROFESIONAL AVANZADO	24
INSTITUTO TÉCNICO SUPERIOR ACADEMO	158
INSTITUTO TÉCNICO SUPERIOR ARANDÚ	110
INSTITUTO TÉCNICO SUPERIOR CARRERAS	227
INSTITUTO TÉCNICO SUPERIOR CENTRO PARAGUAYO DEL SABER	98
INSTITUTO TÉCNICO SUPERIOR COLONIAS UNIDAS	15
INSTITUTO TÉCNICO SUPERIOR DE AGENTES DE SALUD	78
INSTITUTO TÉCNICO SUPERIOR DE FORMACIÓN INTEGRAL PARAGUAYA	104
INSTITUTO TÉCNICO SUPERIOR DE SALUD YUTY	155
INSTITUTO TÉCNICO SUPERIOR DE SANIDAD MILITAR VÍCTOR PASTOR IDOYAGA AYALA	415
INSTITUTO TÉCNICO SUPERIOR DEL SABER	230
INSTITUTO TÉCNICO SUPERIOR DIVINO NIÑO JESÚS	420
INSTITUTO TÉCNICO SUPERIOR EL AMANECER	77
INSTITUTO TÉCNICO SUPERIOR EN CIENCIAS DE LA SALUD Y NATUROPATÍA VITAL	155
INSTITUTO TÉCNICO SUPERIOR EN SALUD	12
INSTITUTO TÉCNICO SUPERIOR EN SALUD SAN BUENAVENTURA	69
INSTITUTO TÉCNICO SUPERIOR EN SALUD SAN LORENZO	41
INSTITUTO TÉCNICO SUPERIOR EN SALUD SANTA ROSA	105
INSTITUTO TÉCNICO SUPERIOR EUROSUR	37
INSTITUTO TÉCNICO SUPERIOR GAMMA	289
INSTITUTO TÉCNICO SUPERIOR INFORTES	194
INSTITUTO TÉCNICO SUPERIOR JESÚS DE NAZARETH	213
INSTITUTO TÉCNICO SUPERIOR MARÍA AUXILIADORA	369
INSTITUTO TÉCNICO SUPERIOR MBO'E RAITY SAN IGNACIO - PARAGUARÍ	83
INSTITUTO TÉCNICO SUPERIOR MISIÓN DE AMISTAD	42
INSTITUTO TÉCNICO SUPERIOR MONSEÑOR EMILIO SOSA GAONA	70
INSTITUTO TÉCNICO SUPERIOR MONTERREY	105
INSTITUTO TÉCNICO SUPERIOR NUESTRA SEÑORA DE FÁTIMA	52
INSTITUTO TÉCNICO SUPERIOR PROFESIONAL DE EDUCACIÓN - IPES	111
INSTITUTO TÉCNICO SUPERIOR REGIONAL DE SALUD	48
INSTITUTO TÉCNICO SUPERIOR SALTO DEL GUAIRÁ•	5
INSTITUTO TÉCNICO SUPERIOR SAN BERNARDO	71
INSTITUTO TÉCNICO SUPERIOR SAN BLAS	470
INSTITUTO TÉCNICO SUPERIOR SAN ESTANISLAO	80
INSTITUTO TÉCNICO SUPERIOR SAN FRANCISCO	302
INSTITUTO TÉCNICO SUPERIOR SAN IGNACIO DE LOYOLA	81
INSTITUTO TÉCNICO SUPERIOR SAN JUAN BAUTISTA	129
INSTITUTO TÉCNICO SUPERIOR SAN MARCOS	48
INSTITUTO TÉCNICO SUPERIOR SAN ROQUE GONZÁLEZ DE SANTA CRUZ	196
INSTITUTO TÉCNICO SUPERIOR SANTA ELENA	155
INSTITUTO TÉCNICO SUPERIOR VILLARRICA	61
INSTITUTO TÉCNICO SUPERIOR VIRGEN DE CAACUPÉ	576
INSTITUTO TÉCNICO SUPERIOR VIRGEN DE LA PAZ	34
INSTITUTO TÉCNICO SUPERIOR VITAL	120

Los institutos técnicos superiores, que en conjunto otorgaron 7,161 títulos (5.6% del total), muestran una distribución más fragmentada entre numerosas instituciones, donde las principales son el Instituto Técnico Superior de Sanidad Militar Víctor Pastor Idoyaga Ayala (415 títulos), el Instituto Técnico Superior Divino Niño Jesús (420 títulos) y el Instituto Técnico Superior María Auxiliadora (369 títulos).

Un hallazgo particularmente significativo es el caso del Instituto Superior en Ciencias de la Salud Juan Pablo II. A pesar de haberse identificado previamente como la institución con la mayor oferta académica en salud, con 395 programas formativos, su producción de títulos (1,643) es relativamente modesta en comparación con su amplia oferta. Esta disparidad entre cantidad de programas ofrecidos y títulos efectivamente otorgados sugiere una baja eficiencia terminal o una sobredimensión de su oferta académica en relación con su capacidad real de formación. Esta institución representa apenas el 6.4% de los títulos otorgados por institutos superiores, a pesar de su liderazgo en la oferta de programas.

ANÁLISIS DEL DASHBOARD

El dashboard desarrollado presenta una estructura bipartita claramente definida, compuesta por una barra lateral izquierda de navegación y un área principal de visualización de datos.

La barra lateral incluye el logo institucional del área de salud, seguido por el título "Área de Salud - Dashboard Académico" que establece el contexto, y un menú de navegación con opciones como "Inicio", "Filtros", "Reporte", "Análisis" y "Configuración", esta estructura facilita la navegación intuitiva y mantiene los controles accesibles en todo momento para el usuario.

El área de filtros representa uno de los componentes interactivos clave de la barra lateral, ofreciendo controles para segmentar los datos según múltiples dimensiones: rango de años ("Desde" y "Hasta"), tipo de institución, carrera, tipo de gestión y género diseñados con elementos interactivos desplegables que permiten una selección precisa y un botón "Aplicar Filtros" que actualiza todas las visualizaciones simultáneamente. Esta capacidad de filtrado dinámico transforma el dashboard de una herramienta estática de visualización a un sistema analítico interactivo que permite exploraciones personalizadas.

La zona principal del dashboard está encabezada por cuatro tarjetas de estadísticas clave que presentan métricas fundamentales: total de 26,548 títulos registrados, distribución por género (15,799 mujeres y 10,749 hombres) y 53 carreras únicas identificadas.

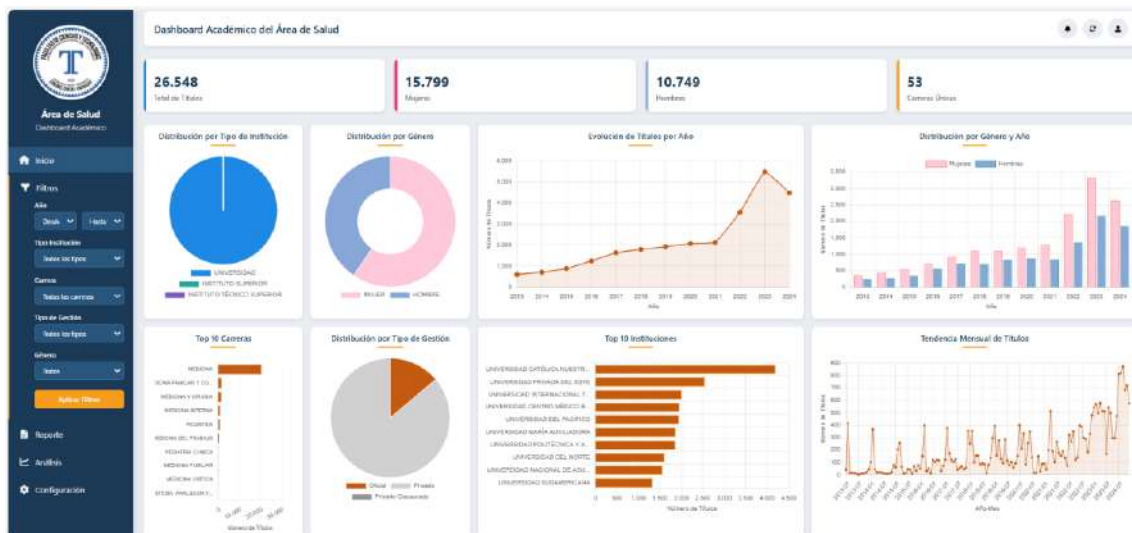


Figura 5. Dashboard académico del área de salud.

El área central del dashboard se organiza en una cuadrícula de visualizaciones que combina gráficos de diferentes tipos, cada uno diseñado para responder a preguntas analíticas específicas. Esta distribución en cuadrícula permite aprovechar eficientemente el espacio disponible mientras mantiene una jerarquía visual clara, los gráficos están organizados en dos filas principales con una combinación equilibrada de visualizaciones de tamaño estándar y visualizaciones expandidas que ocupan dos posiciones para representar datos más complejos.

La primera fila contiene cuatro visualizaciones fundamentales: dos gráficos circulares que representan distribuciones proporcionales (tipo de institución y género) y dos gráficos más complejos sobre evolución temporal y comparativa por género. Esta combinación permite al usuario obtener rápidamente una comprensión de la composición general de los datos y sus tendencias a lo largo del tiempo.

La segunda fila contiene visualizaciones más específicas y detalladas: un gráfico de barras horizontales para el top 10 de carreras, un gráfico circular para la distribución por tipo de gestión, una visualización de barras horizontales para el top 10 de instituciones, y un gráfico de línea para la tendencia mensual. Estos elementos visuales permiten un análisis más granular y específico sobre aspectos particulares de los datos, complementando la visión general proporcionada por la primera fila de visualizaciones.

El dashboard se distingue por su alto nivel de interactividad, que va más allá de los filtros en la barra lateral, cada visualización incorpora elementos interactivos que se activan al pasar el cursor sobre los componentes gráficos, mostrando información detallada mediante tooltips contextuales. Estos tooltips presentan los valores exactos y etiquetas descriptivas, permitiendo una lectura precisa de los datos sin sobrecargar visualmente los gráficos con etiquetas permanentes.

Los gráficos de series temporales (evolución por año y tendencia mensual) incorporan una funcionalidad de zoom que permite al usuario enfocarse en períodos específicos, facilitando la identificación de patrones estacionales o anomalías puntuales. Adicionalmente, las leyendas de los gráficos son interactivas, permitiendo mostrar u ocultar series de datos específicas, lo que resulta particularmente útil en visualizaciones como la distribución por género y año, donde el usuario puede elegir ver solamente los datos de un género específico para análisis comparativos más claros.

La consistencia en el diseño de interacciones a lo largo del dashboard crea una experiencia de usuario fluida e intuitiva. Los elementos visuales responden de manera uniforme a las interacciones, con efectos de hover sutiles que indican la disponibilidad de información adicional, y transiciones animadas suaves cuando los gráficos se actualizan en respuesta a cambios en los filtros aplicados.

El dashboard implementa una paleta de colores cuidadosamente seleccionada que combina tonos institucionales con un sistema cromático funcional, el esquema general utiliza un fondo claro para el área principal, contrastando con una barra lateral en azul oscuro que proporciona un marcado contraste visual y establece una clara separación de áreas funcionales.

Para las visualizaciones, se utiliza una combinación coherente de colores que mantiene la consistencia a través de los diferentes gráficos, los gráficos de distribución por género emplean consistentemente rosa para mujeres y azul claro para hombres; la distribución por tipo de institución utiliza tonos de azul para universidad y púrpura para institutos; mientras que la distribución por tipo de gestión emplea marrón para oficial y gris para privado. Esta consistencia cromática permite al usuario establecer rápidamente asociaciones visuales entre diferentes gráficos que representan las mismas categorías.

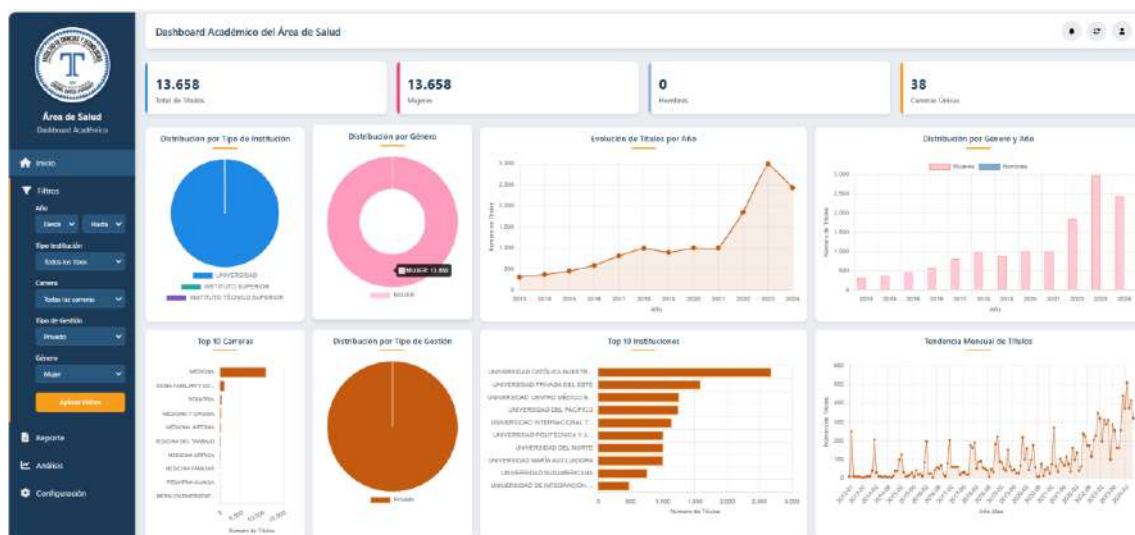


Figura 5. Dashboard académico del área de Salud con filtros aplicados.

Las tarjetas de estadísticas y contenedores de gráficos utilizan un fondo blanco con sutiles efectos de sombra que crean una sensación de elevación y separación del fondo, implementando principios de diseño material. Cada tarjeta de estadística incorpora además un indicador de color en su borde izquierdo que codifica la categoría de información, facilitando la asociación visual y la rápida identificación de métricas relacionadas.

Desde el punto de vista técnico, el dashboard está implementado utilizando tecnologías web estándar (HTML, CSS y JavaScript) con la biblioteca Chart.js para la generación de visualizaciones. Esta elección tecnológica permite una gran flexibilidad en la personalización de gráficos mientras mantiene un rendimiento óptimo incluso con conjuntos de datos extensos.

La arquitectura técnica del dashboard implementa una separación clara entre la capa de datos, la lógica de negocio y la presentación visual. Los datos se cargan desde un archivo Excel utilizando la biblioteca SheetJS, procesados mediante funciones JavaScript que incluyen operaciones de agregación y filtrado, y finalmente visualizados a través de componentes gráficos. Esta separación facilita el mantenimiento y la extensibilidad del dashboard.

El diseño es completamente responsivo, adaptándose a diferentes tamaños de pantalla mediante el uso de CSS Grid y Flexbox para la disposición de elementos. En pantallas más pequeñas, la barra lateral se contrae a un menú de iconos para maximizar el espacio disponible para las visualizaciones, mientras que los gráficos que normalmente ocupan dos posiciones pasan a ocupar una sola columna. Esta adaptabilidad garantiza que el dashboard sea utilizable en diversos dispositivos, desde computadoras de escritorio hasta tablets.

CONCLUSIONES Y RECOMENDACIONES

Basado en el análisis integral de los datos procesados sobre la oferta académica y el egreso en carreras del área de salud en Paraguay, se pueden establecer las siguientes conclusiones:

La implementación del pipeline de procesamiento y limpieza de datos permitió identificar y corregir inconsistencias significativas en los registros institucionales, particularmente en la denominación de entidades educativas, así mismo se tuvo que utilizar librerías para estimar el género de los egresados ya que había un alto número de registros sin dicho campo. La reducción de 140 a 122 instituciones tras el proceso de estandarización evidencia la magnitud de los problemas de calidad de datos en el sistema de información de educación superior paraguayo, confirmando la problemática metodológica planteada inicialmente.

Un hallazgo crítico durante el proceso de integración de datos fue la ausencia de estándares unificados tanto en la nomenclatura de los campos como en el formato de los registros mismos. Se identificaron variaciones significativas en la denominación de campos equivalentes entre diferentes repositorios, así como inconsistencias en la codificación de variables categóricas. Esta falta de estandarización no solo complicó la fase de integración de datos, sino que incrementó significativamente el riesgo de interpretaciones erróneas y duplicaciones en los análisis subsiguientes.

El análisis descriptivo revela un sistema de formación en salud caracterizado por una marcada privatización, donde aproximadamente el 90% tanto de la oferta académica como de los títulos otorgados corresponden a instituciones de gestión privada. Esta coincidencia entre los porcentajes de oferta y títulos sugiere una eficiencia terminal similar entre sectores, aunque no entre tipos de instituciones.

Se constata una significativa feminización del sector, con 75.7% de los títulos otorgados a mujeres, tendencia que se mantiene constante a lo largo del período analizado y a través de todos los tipos de instituciones, siendo ligeramente más pronunciada en instituciones no universitarias.

La distribución geográfica de la oferta académica muestra una notable concentración en tres departamentos (Capital, Central y Alto Paraná), que acumulan casi la mitad (49.1%) de todos los programas formativos, evidenciando importantes desequilibrios territoriales que podrían impactar en la accesibilidad a la formación y en la distribución de profesionales sanitarios.

Se identifica una discordancia significativa entre la cantidad de programas ofrecidos y la producción efectiva de egresados en algunas instituciones. El caso emblemático es el Instituto Superior en Ciencias de la Salud Juan Pablo II, que lidera la oferta académica

con 395 programas, pero solo ha otorgado 1,643 títulos, evidenciando posibles ineficiencias o sobredimensionamientos en su planificación académica.

En términos de campos formativos, Enfermería emerge como la carrera dominante con 46,269 títulos otorgados, estableciendo una proporción de 2.3 enfermeros por cada médico (20,393 títulos). Esta distribución se alinea con recomendaciones internacionales sobre composición de equipos sanitarios, aunque con particularidades locales como la alta producción en áreas técnicas como Farmacia y Radiología.

Las universidades representan el elemento central del sistema formativo en salud, otorgando el 73.9% de todos los títulos pese a constituir solo el 39.3% de las instituciones y ofrecer el 45.6% de los programas. Esta mayor eficiencia terminal universitaria contrasta con la baja productividad relativa de los institutos técnicos superiores, que representan el 51.6% de las instituciones, pero solo producen el 5.6% de los egresados.

La evolución temporal (2012-2024) muestra un crecimiento general en la formación de profesionales de la salud, con fluctuaciones significativas, incluyendo una marcada caída en 2020 (probablemente asociada a la pandemia de COVID-19) y un pico en 2023. Esta tendencia creciente, sumada al predominio de la formación privada, plantea interrogantes sobre la capacidad del sistema sanitario para absorber a estos profesionales.

El análisis integrado de estos hallazgos proporciona una visión comprehensiva de la configuración actual del sistema de formación en salud paraguayo, caracterizado por altos niveles de privatización, feminización, concentración geográfica e institucional, y ciertos desequilibrios entre oferta académica y producción efectiva de egresados. Estas características estructurales deben ser consideradas en el diseño de políticas públicas orientadas tanto a la regulación del sistema educativo como a la planificación estratégica de recursos humanos en salud.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Naciones Unidas, La Agenda 2030 y los Objetivos de Desarrollo Sostenible Una oportunidad para América Latina y el Caribe, Santiago: CEPAL, 2018.
- [2] Á. Ferreira, «Estudio descriptivo de los registros de títulos de la Educación Superior en Paraguay, corte longitudinal 2012 al 2020,» *Revista De Investigación Científica y Tecnológica*, vol. 5, pp. 40-53, 2024.
- [3] M. Webber, «Estudiantes brasileiros de medicina en Presidente Franco (PY): motivaciones y tensión de un flujo universitario transfronterizo,» 2018.
- [4] J. Prettel, Relaciones Brasil-Paraguay: El caso de la carrera de Medicina en la Triple Frontera, 2021: UNILA, Foz de Yguazú.
- [5] H. Estigarribia, «Análisis de la oferta académica y egreso de carreras TIC en la educación superior en Paraguay, basado en el estudio de datos abiertos,» *ACADEMO Revista De Investigación En Ciencias Sociales Y Humanidades*, vol. 11, pp. 201-208, 2024.
- [6] Ministerio de Educación, «Registro Nacional de Títulos,» [En línea]. Available: https://datos.mec.gov.py/data/registros_titulos. [Último acceso: 15 Enero 2025].
- [7] ABC Color, «Medicina: campos de práctica en la frontera estarían saturados, concluye informe,» 07 Noviembre 2023. [En línea]. Available: <https://www.abc.com.py/politica/2023/11/07/un-verdadero-carnaval-con-las-facultades-de-medicina/>. [Último acceso: 2025 Enero 21].
- [8] Agencia de Información Paraguaya, «Habilitación de 39 carreras en forma irregular en 18 filiales de universidad privada es revocada por la Corte,» 13 Agosto 2024. [En línea]. Available: <https://www.ip.gov.py/ip/2024/08/13/habilitacion-de-39-carreras-en-forma-irregular-en-18-filiales-de-universidad-privada-es-revocada-por-la-corte/>. [Último acceso: 2025 Marzo 23].
- [9] H. Estigarribia y A. Rojas, «Estadísticas sobre los egresados de la carrera de medicina en Paraguay y su inserción al sistema de salud brasileño,» *UCOM Scientia*, pp. 107-124, 2023.
- [10] R. Hernández, C. Fernández y M. Baptista, Metodología de la Investigación, México D.F: Mc Graw Hill, 2003.
- [11] Ministerio de Educación, «Datos Abiertos del Mec,» MEC, [En línea]. Available: <https://datos.mec.gov.py/>. [Último acceso: 12 Febrero 2025].

ANEXOS

1. Código Fuente para extracción, tratamiento, y carga de datos

1.1. Estructura para seleccionar títulos relacionados al área de salud

```
import polars as pl
import json

def main():
    # Definir las palabras clave para filtrar
    palabras_clave_salud = [
        "SALUD", "MEDIC", "ENFERM", "ODONTO", "FARMAC", "NUTRICI",
        "FISIOTER", "KINESIO", "PSICO", "FONOAUD", "OBSTET", "CARDIO",
        "CIRUGI", "ANESTESI", "ENDOCRI", "GASTRO", "GINECO", "HEMATO",
        "NEURO", "ONCOLOG", "PEDIATR", "RADIOLOG", "TRAUMATO", "CLINICA",
        "TERAPIA", "SANITARIA", "HOSPITALAR"
    ]

    palabras_excluidas = [
        "VETERINARIA", "ANIMAL", "ZOOTECN", "AGROPECU", "ADMINISTRA",
        "AUDITO", "GESTION", "GERENCIA",
        "DIDACTICA", "EDUCAC", "LEGISLA", "PROFESOR", "METODOLOGIA",
        "GASTRONOM", "ARQUITECTURA", "TURISMO", "HOTELER", "PREPARACION
FISICA", "DEPORT", "PSICOPEDAGOGIA", "EMPRES"
    ]

    try:
        # Leer el archivo JSON
        print("Leyendo archivo JSON...")
        with open('registros_titulos.json', 'r', encoding='utf-8') as file:
            data = json.load(file)

        # Convertir a DataFrame de Polars
        df = pl.DataFrame(data)

        print(f"Total de registros cargados: {len(df)}")

        # Función para comprobar si una carrera está relacionada con salud
        def filtrar_carrera_salud(carrera):
            if not carrera:
                return False

            carrera_upper = carrera.upper()

            # Comprobar si contiene alguna palabra clave
```

```

        tiene_palabra_clave = any(clave in carrera_upper for clave in
palabras_clave_salud)

        # Comprobar que no tenga palabras excluidas
        sin_palabras_excluidas = not any(excluida in carrera_upper for
excluida in palabras_excluidas)

        return tiene_palabra_clave and sin_palabras_excluidas

# Aplicar el filtro usando `.apply()`
df_salud = df.filter(
    df["carrera"].apply(filtrar_carrera_salud)
)

print(f"Registros filtrados del área de salud: {len(df_salud)}")

# Reemplazar comas por guiones en todas las columnas de texto
df_salud_modificado = df_salud.with_columns(
    pl.col(pl.Utf8).str.replace_all(",", "-") # Reemplaza todas las
comas por guiones en columnas de texto
)

# Guardar el resultado en CSV
df_salud_modificado.write_csv("titulosSalud.csv")
print("Archivo CSV 'titulosSalud.csv' creado exitosamente con comas
reemplazadas por guiones")

# Mostrar un resumen de los primeros registros
print("\nPrimeros 5 registros del DataFrame filtrado:")
print(df_salud_modificado.head(5))

# Mostrar las carreras únicas encontradas
carreras_unicas = df_salud_modificado.select("carrera").unique()
print(f"\nSe encontraron {len(carreras_unicas)} carreras únicas del
área de salud")
print("Ejemplos de carreras encontradas:")
print(carreras_unicas.head(10))

return df_salud_modificado

except FileNotFoundError:
    print("Error: No se encontró el archivo 'registros_titulos.json'")
    return None
except Exception as e:
    print(f"Error inesperado: {str(e)}")
    return None

```

```
if __name__ == "__main__":
    df_resultado = main()
```

1.2. Estructura para agrupar carreras relacionados al área de salud

```
import csv
import re
import unicodedata
import random

def normalizar_texto(texto):
    """Normaliza el texto eliminando acentos y caracteres especiales"""
    if not isinstance(texto, str):
        texto = str(texto)
    # Normalizar NFD y eliminar diacríticos
    texto = unicodedata.normalize('NFD', texto)
    texto = ''.join(c for c in texto if not unicodedata.combining(c))
    # Convertir a minúsculas
    texto = texto.lower()
    # Eliminar caracteres especiales y símbolos
    texto = re.sub(r'[^w\s]', '', texto)
    # Eliminar espacios múltiples
    texto = re.sub(r'\s+', ' ', texto).strip()
    return texto

def crear_grupos_carreras():
    """Define los 10 grupos de carreras con todas las palabras clave necesarias"""
    grupos = {
        "Medicina General y Especialidades Médicas": [
            # Términos generales
            "medicina", "medico", "clinica", "ciencias medicas",
            # Especialidades médicas específicas
            "anestesia", "anestesiologia", "analgesia", "reanimacion",
            "pediatria", "pediatrica", "geriatria", "gerontologia",
            "neurologia", "neurociencias", "cognicion", "neuropsiquiatria",
            "endocrinologia", "metabolismo", "infectologia",
            "bacteriologia", "gastroenterologia", "emergencia"
        ],
        "Hematología e Inmunología": [
            "hematologia", "autoinmunes", "inmunologia", "alergologia",
            "medicina interna", "medicina familiar", "medicina legal",
            "medicina critica",
            "medicina del trabajo", "medicina perinatal", "forense",
            "medicina estetica", "oncologia", "gastroenterologia"
        ],
        "Urgencias y Emergencias": [
            # Términos de urgencia y emergencia
            "emergencias medicas", "emergentologia", "urgencias", "urgencias
            pediatricas",
            # Términos específicos
            "adolescencia", "salud integral", "perinatal"
```


],

```
"Cirugía y Especialidades Quirúrgicas": [  
  # Términos generales  
  "cirugia", "cirujano", "quirurgica", "quirurgico",  
  # Especialidades quirúrgicas  
  "cirugia general", "cirugia pediatrica", "cirugia infantil",  
  "cirugia vascular",  
  "cirugia cardiaca", "cardiocirugia", "neurocirugia", "cirugia  
  ortopedica",  
  "cirugia traumatologica", "cirugia plastica", "cirugia  
  reconstructiva",  
  "cirugia oncologica", "cirugia colon", "cirugia recto", "cirugia  
  ano",  
  "cirugia del colon recto y ano", "cirugia maxilofacial", "cirugia  
  de trauma",  
  # Términos relacionados  
  "traumatologia", "ortopedia", "quemaduras", "maxilofacial",  
  "retina medica", "retina quirurgica", "retina", "oftalmologia"  
],
```

```
"Enfermería y Cuidados Intensivos": [  
  # Términos generales  
  "enfermeria", "enfermero", "cuidados", "cuidado",  
  # Especialidades de enfermería  
  "enfermeria pediatrica", "enfermeria neonatal", "enfermeria  
  geriatria",  
  "enfermeria en cuidados intensivos", "enfermeria en emergencia",  
  "enfermeria en medicina critica", "enfermeria en nefrologia",  
  "enfermeria oncologica", "enfermeria psiquiatrica", "enfermeria y  
  obstetricia",  
  # Términos de cuidados intensivos  
  "cuidados intensivos", "terapia intensiva", "ciudades intensivos  
  pediatricos",  
  "cuidados intensivos cardiovasculares", "uci",  
  # Términos de emergencia  
  "emergencias y urgencias", "emergenciasmedicas", "pre  
  hospitalaria"  
],
```

```
"Cardiología y Ciencias del Corazón": [  
  # Términos generales  
  "cardiologia", "cardiaco", "corazon", "cardiovascular",  
  # Especialidades cardiológicas  
  "cardiologia clinica", "cardiologia pediatrica", "cardiologia  
  intervencionista",
```

"ecocardiografia", "ecocardiografia pediátrica", "hemodinamia cardiovascular",

Términos relacionados

"hipertension arterial", "riesgo cardiovascular", "arritmias"

],

"Ginecología - Obstetricia y Salud Reproductiva": [

Términos generales

"ginecologia", "gineco", "obstetricia", "obstetrico", "materno", "reproduccion",

Términos específicos

"gineco obstetricia", "ginecoobstetrica", "clinica ginecoobstetrica",

"salud materno infantil", "endocrinologia ginecologica", "endoscopia ginecologica",

"citologia ginecologica", "ecografia obstetrica", "ecografia ginecologica",

Términos relacionados

"fertilidad", "salud reproductiva", "salud sexual", "reproduccion asistida"

],

"Psicología y Psiquiatría": [

Términos generales

"psicologia", "psicologo", "psiquiatria", "psiquiatra", "psico", "mental",

Especialidades psicológicas

"psicologia clinica", "psicologia social", "psicologia juridica", "psicologia forense", "psicologia laboral", "psicologia organizacional",

"psicologia oncologica", "psicologia cognitiva", "neuropsicologia",

Términos de psiquiatría

"psiquiatria clinica", "psiquiatria infantil", "psiquiatria juvenil",

Términos de terapia

"psicoterapia", "psicoterapias", "terapia cognitiva", "conductual",

"terapia cognitivo conductual", "rogeriano", "consultoria psicologica",

Términos específicos

"salud mental", "psicopedagogia", "psicomotricidad", "neuropsicopatologia",

"neuropsicopatologia infantil", "programacion neurolinguistica"

],

"Odontología y Especialidades Dentales": [

```

# Términos generales
"odontologia", "dental", "dent", "bucal",
# Especialidades odontológicas
"odontopediatria", "endodoncia", "odontologia estetica",
"odontologia legal",
"ortodoncia", "periodoncia", "cirugia dental", "cirugia
dentoalveolar",
"cirugia bucal", "maxilo facial", "maxilofacial", "buc maxilo
facial",
# Términos específicos
"ciencias odontologicas", "operatoria dental", "rehabilitacion
oral",
"protesis dental", "estomatologia"
],

"Ciencias Biomédicas y Laboratorio": [
# Términos generales
"biomedica", "biologica", "biologia", "laboratorio",
"bioquimica",
# Disciplinas específicas
"ciencias biomedicas", "ciencias biologicas", "ciencias
quimicas",
"ciencias quimico biologicas", "bioquimica clinica",
"bacteriologia clinica",
"microscopia clinica", "microbiologia", "parasitaria",
"parasitarias",
"hematologia", "medicina transfusional", "hemoterapia", "hemato",
# Términos relacionados
"investigacion en salud", "fisiologia", "fisica medica",
"analisis clinico",
"molecular", "farmacologia molecular", "quimico farmaceutico",
"molecula"
],

"Imagenología y Diagnóstico": [
# Términos generales
"radiologia", "radiologica", "imagen", "imagenes", "diagnostico",
# Especialidades
"radiologia e imagen", "radiologia e imagenes", "radiologia e
imagenologia",
"imagenologia", "imagenologia medica", "medicina en imagenes",
"medicina por imagenes",
"diagnostico por imagenes", "diagnostico por imagen",
"bioimagenes",
# Términos específicos
"imagenologia con enfasis en pre hospitalaria", "imagen medica",
"produccion de bioimagenes"

```

```

    ],

    "Nutrición - Farmacia y Otras Ciencias de la Salud": [
        # Nutrición
        "nutricion", "nutricionista", "dietetica", "dieta", "nutricion
clinica",
        "dietetica clinica", "soporte nutricional", "alimentacion",
        # Farmacia
        "farmacia", "farmaceutica", "farmaceutico", "quimica
farmaceutica",
        "quimica y farmacia", "ciencias farmaceuticas",
        # Rehabilitación y terapias
        "fisioterapia", "kinesiologia", "fisiatria", "rehabilitacion",
        "terapia fisica", "fonoaudiologia", "terapia ocupacional",
"logopedia",
        # Salud pública y gestión
        "salud publica", "epidemiologia", "salud comunitaria", "salud
familiar",
        "salud colectiva", "salud", "gestion", "gestion sanitaria",
        "direccion y gestion", "organizaciones sanitarias", "sanitaria",
        # Otros campos específicos
        "ingenieria sanitaria", "ambiental", "ingenieria sanitaria y
ambiental",
        "ciencias de la salud", "ciencias naturales", "pedagogia",
        "pedagogia con enfasis en ciencias de la naturaleza y la salud",
        "docencia", "deporte", "salud deportiva"
    ]
}

return grupos

def asignar_grupo(carrera, grupos):
    """Asigna un grupo a una carrera por coincidencia de términos"""
    carrera_norm = normalizar_texto(carrera)

    # Inicializar puntuación para cada grupo
    puntuaciones = {grupo: 0 for grupo in grupos}

    # Calcular puntuación para cada grupo
    for grupo, keywords in grupos.items():
        for keyword in keywords:
            keyword_norm = normalizar_texto(keyword)
            if keyword_norm in carrera_norm:
                # La puntuación es proporcional a la longitud de la palabra
clave
                # y a qué tan específica es la coincidencia
                if keyword_norm == carrera_norm: # Coincidencia exacta
                    puntuaciones[grupo] += len(keyword_norm) * 3

```

```

        elif carrera_norm.startswith(keyword_norm + " ") or
carrera_norm.endswith(" " + keyword_norm): # Palabra completa al inicio o
final
            puntuaciones[grupo] += len(keyword_norm) * 2
        else: # Coincidencia parcial
            puntuaciones[grupo] += len(keyword_norm)

# Encontrar el grupo con la mayor puntuación
mejor_grupo = max(puntuaciones.items(), key=lambda x: x[1])

# Si hay al menos una coincidencia, devolver el mejor grupo
if mejor_grupo[1] > 0:
    return mejor_grupo[0]

# Si no hay coincidencias, asignar a "Otros"
return "Otros"

def cargar_csv(archivo):
    """Carga un archivo CSV usando el módulo csv estándar"""
    data = []
    with open(archivo, 'r', encoding='utf-8') as f:
        reader = csv.DictReader(f)
        for row in reader:
            data.append(row)
    return data

def guardar_csv(datos, archivo, headers):
    """Guarda los datos en un archivo CSV"""
    with open(archivo, 'w', newline='', encoding='utf-8') as f:
        writer = csv.DictWriter(f, fieldnames=headers)
        writer.writeheader()
        writer.writerows(datos)

def procesar_csv(archivo_entrada, archivo_salida):
    """Procesa el CSV añadiendo la columna de grupo de carrera"""
    try:
        # Cargar el CSV
        print(f"Cargando archivo: {archivo_entrada}")
        datos = cargar_csv(archivo_entrada)

        # Verificar que exista la columna 'carrera'
        if not datos or 'carrera' not in datos[0]:
            print(f"Error: La columna 'carrera' no existe en el archivo
{archivo_entrada}")
            return False

```

```

        print(f"Archivo cargado correctamente. Total de registros:
{len(datos)}")

    # Crear grupos
    grupos = crear_grupos_carreras()

    # Imprimir los grupos para verificación
    print("\nGrupos definidos:")
    for grupo in grupos.keys():
        print(f"    - {grupo}")

    # Añadir la columna carreraGrupo
    print("\nClasificando carreras en grupos...")
    for row in datos:
        # Asignar grupo a cada carrera
        if 'carrera' in row:
            row['carreraGrupo'] = asignar_grupo(str(row['carrera']),
grupos)

    # Crear los encabezados para el CSV (manteniendo el orden original y
añadiendo carreraGrupo)
    headers = list(datos[0].keys())

    # Guardar el resultado
    guardar_csv(datos, archivo_salida, headers)
    print(f"Archivo procesado guardado como: {archivo_salida}")

    # Contar ocurrencias de cada grupo
    grupo_counts = {}
    for row in datos:
        grupo = row['carreraGrupo']
        if grupo in grupo_counts:
            grupo_counts[grupo] += 1
        else:
            grupo_counts[grupo] = 1

    # Mostrar estadísticas de los grupos
    print("\nEstadísticas de grupos de carreras:")
    for grupo, count in sorted(grupo_counts.items(), key=lambda x: x[1],
reverse=True):
        print(f"{grupo}: {count} registros")

    # Mostrar algunos ejemplos para verificar
    print("\nEjemplos de asignación de grupos:")
    muestra = random.sample(datos, min(10, len(datos)))
    for row in muestra:

```

```

        print(f"Carrera: {row['carrera']} -> Grupo:
{row['carreraGrupo']}")

    except Exception as e:
        print(f"Error al procesar el archivo: {str(e)}")
        import traceback
        traceback.print_exc()
        return False

def main():
    archivo_entrada = 'titulosSalud.csv'
    archivo_salida = 'titulosSalud_agrupado.csv'

    print("Script para agrupar carreras de salud")
    print("-----")
    print(f"Procesando archivo: {archivo_entrada}")

    procesar_csv(archivo_entrada, archivo_salida)

    print("\nProceso completado.")

if __name__ == "__main__":
    main()

```

1.3. Estructura para completar géneros de registros en función a nombre

```

import csv
import gender_guesser.detector as gender
import unicodedata
import re

def normalizar_texto(texto):
    """Normaliza el texto eliminando acentos y caracteres especiales"""
    if not isinstance(texto, str):
        texto = str(texto)
    # Normalizar NFD y eliminar diacríticos
    texto = unicodedata.normalize('NFD', texto)
    texto = ''.join(c for c in texto if not unicodedata.combining(c))
    # Convertir a minúsculas
    texto = texto.lower()
    return texto

def obtener_primer_nombre(nombre_completo):
    """Extrae el primer nombre del nombre completo"""
    if not nombre_completo or not isinstance(nombre_completo, str):
        return ""

```

```

# Normalizar y dividir el nombre
nombre_normalizado = normalizar_texto(nombre_completo)
partes = re.split(r'\s+', nombre_normalizado)

# Retornar el primer elemento si existe
if partes:
    return partes[0]
return ""

def inferir_sexo(nombre_completo, detector):
    """Infiere el sexo a partir del nombre completo"""
    primer_nombre = obtener_primer_nombre(nombre_completo)

    if not primer_nombre:
        return None

    # Usar gender-guesser para inferir el género
    genero = detector.get_gender(primer_nombre)

    # Mapear resultados a HOMBRE o MUJER
    if genero in ['male', 'mostly_male']:
        return "HOMBRE"
    elif genero in ['female', 'mostly_female']:
        return "MUJER"
    else:
        # Para 'unknown' o 'andy' (andrógino) intentamos con el segundo
nombre
        partes = re.split(r'\s+', normalizar_texto(nombre_completo))
        if len(partes) > 1:
            genero_segundo = detector.get_gender(partes[1])
            if genero_segundo in ['male', 'mostly_male']:
                return "HOMBRE"
            elif genero_segundo in ['female', 'mostly_female']:
                return "MUJER"

    # Si no podemos determinar, retornamos None
    return None

def cargar_csv(archivo):
    """Carga un archivo CSV usando el módulo csv estándar"""
    data = []
    with open(archivo, 'r', encoding='utf-8') as f:
        reader = csv.DictReader(f)
        for row in reader:
            data.append(row)
    return data

```



```

def guardar_csv(datos, archivo, headers):
    """Guarda los datos en un archivo CSV"""
    with open(archivo, 'w', newline='', encoding='utf-8') as f:
        writer = csv.DictWriter(f, fieldnames=headers)
        writer.writeheader()
        writer.writerows(datos)

def completar_sexo_en_csv(archivo_entrada, archivo_salida):
    """Completa el campo sexo en vacíos usando el nombre_completo"""
    try:
        # Cargar el CSV
        print(f"Cargando archivo: {archivo_entrada}")
        datos = cargar_csv(archivo_entrada)

        # Verificar que existan las columnas necesarias
        if not datos or 'nombre_completo' not in datos[0] or 'sexo' not in
datos[0]:
            print(f"Error: Faltan columnas requeridas en el archivo
{archivo_entrada}")
            return False

        print(f"Archivo cargado correctamente. Total de registros:
{len(datos)}")

        # Inicializar detector de género
        detector = gender.Detector(case_sensitive=False)

        # Estadísticas
        total_vacios = 0
        completados = 0
        no_determinados = 0

        # Completar campo sexo
        print("\nCompletando valores vacíos en campo 'sexo'...")
        for row in datos:
            # Verificar si el campo sexo está vacío o nulo
            if 'sexo' not in row or not row['sexo'] or row['sexo'].strip() ==
'':
                total_vacios += 1

            # Inferir sexo a partir del nombre
            sexo_inferido = inferir_sexo(row['nombre_completo'],
detector)

            if sexo_inferido:
                row['sexo'] = sexo_inferido

```

```

        completados += 1
    else:
        no_determinados += 1

# Estadísticas finales de valores completados
print(f"Valores vacíos encontrados: {total_vacios}")
print(f"Valores completados: {completados}")
print(f"Valores que no se pudieron determinar: {no_determinados}")

# Guardar el resultado
guardar_csv(datos, archivo_salida, list(datos[0].keys()))
print(f"Archivo procesado guardado como: {archivo_salida}")

# Mostrar distribución final de géneros
hombres = sum(1 for row in datos if row.get('sexo') == 'HOMBRE')
mujeres = sum(1 for row in datos if row.get('sexo') == 'MUJER')
otros = sum(1 for row in datos if row.get('sexo') not in ['HOMBRE',
'MUJER'] or not row.get('sexo'))

print("\nDistribución final de géneros:")
print(f"HOMBRE: {hombres} registros ({hombres/len(datos)*100:.2f}%)")
print(f"MUJER: {mujeres} registros ({mujeres/len(datos)*100:.2f}%)")
print(f"No determinado: {otros} registros
({otros/len(datos)*100:.2f}%)")

return True

except Exception as e:
    print(f"Error al procesar el archivo: {str(e)}")
    import traceback
    traceback.print_exc()
    return False

def main():
    archivo_entrada = 'titulosSalud_agrupado.csv'
    archivo_salida = 'titulosSalud_completo.csv'

    print("Script para completar datos de sexo")
    print("-----")
    print(f"Procesando archivo: {archivo_entrada}")

    completar_sexo_en_csv(archivo_entrada, archivo_salida)

    print("\nProceso completado.")

if __name__ == "__main__":
    main()

```

1.4. Estructura para limpiar registros de carreras con sintaxis errónea

```
import csv
import unicodedata
import re
import datetime
import os
from difflib import get_close_matches

def cargar_lista_instituciones():
    """Carga la lista de instituciones oficiales y crea un mapeo para la
    normalización"""
    # Lista de instituciones base oficiales
    instituciones_oficiales = [
        "UNIVERSIDAD GRAN ASUNCIÓN",
        "CENTRO EDUCATIVO SUPERIOR EN SALUD",
        "ESCUELA PARAGUAYA DE ENFERMERÍA Y CIENCIAS DE LA SALUD",
        "INSTITUTO DE FORMACIÓN DOCENTE LA PAZ",
        "INSTITUTO DE FORMACIÓN TÉCNICA SUPERIOR INFORTES",
        "INSTITUTO NACIONAL DE SALUD",
        "INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SANTA ROSA MÍSTICA",
        "INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD JUAN PABLO II",
        "INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SAN AGUSTÍN",
        "INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SAN PATRICIO DE IRLANDA
DEL NORTE",
        "INSTITUTO SUPERIOR INTERREGIONAL EN CIENCIAS DE LA SALUD",
        "INSTITUTO SUPERIOR LATINOAMERICANO",
        "INSTITUTO SUPERIOR PROFESIONAL AVANZADO",
        "INSTITUTO SUPERIOR SAN AGUSTÍN",
        "INSTITUTO SUPERIOR SANTA LIBRADA",
        "INSTITUTO TÉCNICO SUPERIOR INFORTES",
        "INSTITUTO TÉCNICO SUPERIOR SANTA ANA",
        "INSTITUTO TÉCNICO SUPERIOR ACADEMO",
        "INSTITUTO TÉCNICO SUPERIOR ARANDÚ",
        "INSTITUTO TÉCNICO SUPERIOR CARRERAS",
        "INSTITUTO TÉCNICO SUPERIOR CEPROCAL",
        "INSTITUTO TÉCNICO SUPERIOR CENTRO PARAGUAYO DEL SABER",
        "INSTITUTO TÉCNICO SUPERIOR COLONIAS UNIDAS",
        "INSTITUTO TÉCNICO SUPERIOR DE AGENTES DE SALUD",
        "INSTITUTO TÉCNICO SUPERIOR DE FORMACIÓN INTEGRAL PARAGUAYA",
        "INSTITUTO TÉCNICO SUPERIOR DE SALUD YUTY",
        "INSTITUTO TÉCNICO SUPERIOR DE SANIDAD MILITAR VÍCTOR PASTOR IDOYAGA
AYALA",
        "INSTITUTO TÉCNICO SUPERIOR DEL SABER",
        "INSTITUTO TÉCNICO SUPERIOR DIVINO NIÑO JESÚS",
        "INSTITUTO TÉCNICO SUPERIOR EL AMANECER",
```

"INSTITUTO TÉCNICO SUPERIOR EN CIENCIAS DE LA SALUD Y NATUROPATÍA
VITAL",
"INSTITUTO TÉCNICO SUPERIOR EN SALUD",
"INSTITUTO TÉCNICO SUPERIOR EN SALUD SAN BUENAVENTURA",
"INSTITUTO TÉCNICO SUPERIOR EN SALUD SAN LORENZO",
"INSTITUTO TÉCNICO SUPERIOR EN SALUD SANTA ROSA",
"INSTITUTO TÉCNICO SUPERIOR EUROSUR",
"INSTITUTO TÉCNICO SUPERIOR GAMMA",
"INSTITUTO TÉCNICO SUPERIOR JESÚS DE NAZARET",
"INSTITUTO TÉCNICO SUPERIOR MARÍA AUXILIADORA",
"INSTITUTO TÉCNICO SUPERIOR MBO'E RAITY SAN IGNACIO - PARAGUARÍ",
"INSTITUTO TÉCNICO SUPERIOR MISIÓN DE AMISTAD",
"INSTITUTO TÉCNICO SUPERIOR MONSEÑOR EMILIO SOSA GAONA",
"INSTITUTO TÉCNICO SUPERIOR MONTERREY",
"INSTITUTO TÉCNICO SUPERIOR NUESTRA SEÑORA DE FÁTIMA",
"INSTITUTO TÉCNICO SUPERIOR PROFESIONAL DE EDUCACIÓN - IPES",
"INSTITUTO TÉCNICO SUPERIOR REGIONAL DE SALUD",
"INSTITUTO TÉCNICO SUPERIOR SALTO DEL GUAIRÁ",
"INSTITUTO TÉCNICO SUPERIOR SAN BERNARDO",
"INSTITUTO TÉCNICO SUPERIOR SAN BLAS",
"INSTITUTO TÉCNICO SUPERIOR SAN ESTANISLAO",
"INSTITUTO TÉCNICO SUPERIOR SAN FRANCISCO",
"INSTITUTO TÉCNICO SUPERIOR SAN IGNACIO DE LOYOLA",
"INSTITUTO TÉCNICO SUPERIOR SAN JUAN BAUTISTA",
"INSTITUTO TÉCNICO SUPERIOR SAN MARCOS",
"INSTITUTO TÉCNICO SUPERIOR SAN ROQUE GONZÁLEZ DE SANTA CRUZ",
"INSTITUTO TÉCNICO SUPERIOR SANTA ELENA",
"INSTITUTO TÉCNICO SUPERIOR VILLARRICA",
"INSTITUTO TÉCNICO SUPERIOR VIRGEN DE CAACUPÉ",
"INSTITUTO TÉCNICO SUPERIOR VIRGEN DE LA PAZ",
"INSTITUTO TÉCNICO SUPERIOR VITAL",
"UNIVERSIDAD HISPANO GUARANÍ PARA EL DESARROLLO HUMANO",
"UNIVERSIDAD ADVENTISTA DEL PARAGUAY",
"UNIVERSIDAD AMERICANA",
"UNIVERSIDAD AUTÓNOMA DE ASUNCIÓN",
"UNIVERSIDAD AUTÓNOMA DE ENCARNACIÓN",
"UNIVERSIDAD AUTÓNOMA DE LUQUE",
"UNIVERSIDAD AUTÓNOMA DEL PARAGUAY",
"UNIVERSIDAD AUTÓNOMA DEL SUR",
"UNIVERSIDAD AUTÓNOMA SAN SEBASTIÁN",
"UNIVERSIDAD CATÓLICA NUESTRA SEÑORA DE LA ASUNCIÓN",
"UNIVERSIDAD CENTRAL DEL PARAGUAY",
"UNIVERSIDAD CENTRO MÉDICO BAUTISTA",
"UNIVERSIDAD COLUMBIA DEL PARAGUAY",
"UNIVERSIDAD DE DESARROLLO SUSTENTABLE",
"UNIVERSIDAD DE INTEGRACIÓN DE LAS AMÉRICAS",
"UNIVERSIDAD DEL CHACO",

```

"UNIVERSIDAD DEL CONO SUR DE LAS AMÉRICAS",
"UNIVERSIDAD DEL NORTE",
"UNIVERSIDAD DEL PACÍFICO",
"UNIVERSIDAD DEL SOL",
"UNIVERSIDAD EVANGÉLICA DEL PARAGUAY",
"UNIVERSIDAD IBEROAMERICANA",
"UNIVERSIDAD INTERAMERICANA",
"UNIVERSIDAD INTERNACIONAL TRES FRONTERAS",
"UNIVERSIDAD LA PAZ",
"UNIVERSIDAD LEONARDO DA VINCI",
"UNIVERSIDAD MARÍA AUXILIADORA",
"UNIVERSIDAD MARÍA SERRANA",
"UNIVERSIDAD METROPOLITANA DE ASUNCIÓN",
"UNIVERSIDAD NACIONAL DE ASUNCIÓN",
"UNIVERSIDAD NACIONAL DE CAAGUAZÚ",
"UNIVERSIDAD NACIONAL DE CANINDEYÚ",
"UNIVERSIDAD NACIONAL DE CONCEPCIÓN",
"UNIVERSIDAD NACIONAL DE ITAPÚA",
"UNIVERSIDAD NACIONAL DE PILAR",
"UNIVERSIDAD NACIONAL DE VILLARRICA DEL ESPÍRITU SANTO",
"UNIVERSIDAD NACIONAL DEL ESTE",
"UNIVERSIDAD NIHON GAKKO",
"UNIVERSIDAD POLITÉCNICA Y ARTÍSTICA DEL PARAGUAY",
"UNIVERSIDAD PRIVADA DEL ESTE",
"UNIVERSIDAD PRIVADA DEL GUAIRA",
"UNIVERSIDAD SAN CARLOS",
"UNIVERSIDAD SAN LORENZO",
"UNIVERSIDAD SANTA CLARA DE ASÍS",
"UNIVERSIDAD SUDAMERICANA",
"UNIVERSIDAD TÉCNICA DE COMERCIALIZACIÓN Y DESARROLLO",
"UNIVERSIDAD TECNOLÓGICA INTERCONTINENTAL" # Adicional que aparece
en tu lista
]

```

```

# Diccionario de mapeo para nombres similares o variantes
mapeo_variantes = {
    'UNIVERSIDAD MARIA AUXILIADORA': "UNIVERSIDAD MARÍA AUXILIADORA",
    'UNIVERSIDAD NACIONAL DE ASUNCIÃ"N - CIENCIAS MEDICAS': "UNIVERSIDAD
NACIONAL DE ASUNCIÓN",
    'UNIVERSIDAD NACIONAL DE ASUNCIÃ"N - FILOSOFÃA': "UNIVERSIDAD
NACIONAL DE ASUNCIÓN",
    'UNIVERSIDAD NACIONAL DE ASUNCIÃ"N- ODONTOLOGÃA': "UNIVERSIDAD
NACIONAL DE ASUNCIÓN",
    'UNIVERSIDAD NACIONAL DE ASUNCIÃ"N - POLITÃ%CNICA': "UNIVERSIDAD
NACIONAL DE ASUNCIÓN",
    'UNIVERSIDAD NACIONAL DE ASUNCIÃ"N - CAACUPE': "UNIVERSIDAD NACIONAL
DE ASUNCIÓN",

```

'UNIVERSIDAD NACIONAL DE ASUNCIÓN' - CONCEPCIÓN': "UNIVERSIDAD NACIONAL DE ASUNCIÓN",

'UNIVERSIDAD NACIONAL DE ASUNCIÓN' - OVIEDO': "UNIVERSIDAD NACIONAL DE ASUNCIÓN",

'UNIVERSIDAD NACIONAL DE ASUNCIÓN' - QUIINDY': "UNIVERSIDAD NACIONAL DE ASUNCIÓN",

'UNIVERSIDAD NACIONAL DE ASUNCIÓN' - QUIMICA': "UNIVERSIDAD NACIONAL DE ASUNCIÓN",

'UNIVERSIDAD NACIONAL DE ASUNCIÓN' - SAN ESTANISLAO': "UNIVERSIDAD NACIONAL DE ASUNCIÓN",

'UNIVERSIDAD NACIONAL DE ASUNCIÓN' - SAN PEDRO DEL YCUAMANDYJU': "UNIVERSIDAD NACIONAL DE ASUNCIÓN",

'UNIVERSIDAD NACIONAL DE ASUNCIÓN' - FACULTAD DE ENFERMERÍA Y OBSTETRICIA': "UNIVERSIDAD NACIONAL DE ASUNCIÓN",

'UNIVERSIDAD PRIVADA DEL GUAIRA SEDE ENCARNACION': "UNIVERSIDAD PRIVADA DEL GUAIRA",

'UNIVERSIDAD PRIVADA DEL GUAIRA SEDE SAN JUAN NEPOMUCENO': "UNIVERSIDAD PRIVADA DEL GUAIRA",

'UNIVERSIDAD PRIVADA MARÍA SERRANA': "UNIVERSIDAD MARÍA SERRANA",

'UNIVERSIDAD SAN LORENZO UNISAL': "UNIVERSIDAD SAN LORENZO",

'UNIVERSIDAD LEONARDO DA VINCI - CNEL. OVIEDO': "UNIVERSIDAD LEONARDO DA VINCI",

'UNIVERSIDAD LEONARDO DA VINCI - LOMA PLATA': "UNIVERSIDAD LEONARDO DA VINCI",

'UNIVERSIDAD NACIONAL DE ITAPUA': "UNIVERSIDAD NACIONAL DE ITAPÚA",

'UNIVERSIDAD NACIONAL DE CAAGUAZU': "UNIVERSIDAD NACIONAL DE CAAGUAZÚ",

'UNIVERSIDAD NACIONAL DE CANINDEYU': "UNIVERSIDAD NACIONAL DE CANINDEYÚ",

'UNIVERSIDAD NACIONAL DE VILLARRICA DEL ESPIRITU SANTO': "UNIVERSIDAD NACIONAL DE VILLARRICA DEL ESPÍRITU SANTO",

'UNIVERSIDAD NACIONAL DEL ESTE': "UNIVERSIDAD NACIONAL DEL ESTE",

'UNIVERSIDAD SANTA CLARA DE ASÍS': "UNIVERSIDAD SANTA CLARA DE ASÍS",

'UNIVERSIDAD POLITÉCNICA Y ARTÍSTICA': "UNIVERSIDAD POLITÉCNICA Y ARTÍSTICA DEL PARAGUAY",

'UNIVERSIDAD POLITÉCNICA Y ARTÍSTICA DEL PARAGUAY': "UNIVERSIDAD POLITÉCNICA Y ARTÍSTICA DEL PARAGUAY",

'UNIVERSIDAD PRIVADA DEL ESTE': "UNIVERSIDAD PRIVADA DEL ESTE",

'UNIVERSIDAD DE LA INTEGRACIÓN DE LAS AMÉRICAS': "UNIVERSIDAD DE INTEGRACIÓN DE LAS AMÉRICAS",

'UNIVERSIDAD AUTÓNOMA SAN SEBASTIAN': "UNIVERSIDAD AUTÓNOMA SAN SEBASTIÁN",

'UNIVERSIDAD HISPANO GUARANÍ PARA EL DESARROLLO HUMANO': "UNIVERSIDAD HISPANO GUARANÍ PARA EL DESARROLLO HUMANO",

'UNIVERSIDAD \ "HISPANO-GUARANÍ\ " PARA EL DESARROLLO HUMANO': "UNIVERSIDAD HISPANO GUARANÍ PARA EL DESARROLLO HUMANO",

```

        'INSTITUTO TÉCNICO SUPERIOR SAN FRANCISCO SOLANO': "INSTITUTO
TÉCNICO SUPERIOR SAN FRANCISCO",
        'INSTITUTO TÉCNICO SUPERIOR JESÚS DE NAZARETH': "INSTITUTO TÉCNICO
SUPERIOR JESÚS DE NAZARET",
        'INSTITUTO TECNICO SUPERIOR DE SANIDAD MILITAR - GRAL DE SAN DR
VICTOR PASTOR IDOYAGA AYALA': "INSTITUTO TÉCNICO SUPERIOR DE SANIDAD MILITAR
VÍCTOR PASTOR IDOYAGA AYALA",
        'INSTITUTO TÉCNICO SUPERIOR CENTRO PARAGUAYO DE PRODUCTIVIDAD Y
CALIDAD - CEPROCA': "INSTITUTO TÉCNICO SUPERIOR CEPROCAL",
        'INSTITUTO TÉCNICO SUPERIOR DE FORMACIÓN TÉCNICA SUPERIOR-
INFORTES': "INSTITUTO TÉCNICO SUPERIOR INFORTES",
        'INSTITUTO SUPERIOR PROFESIONAL AVANZADO - ISPA': "INSTITUTO SUPERIOR
PROFESIONAL AVANZADO",
        'INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD "SANTA ROSA MÁSTICA':
"INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SANTA ROSA MÍSTICA",
        'INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD "SANTA ROSA MÁSTICA\''':
"INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SANTA ROSA MÍSTICA",
        'INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SANTA ROSA MÁSTICA':
"INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SANTA ROSA MÍSTICA",
        'INSTITUTO SUPERIOR EN CIENCIAS DE LA SALUD SAN AGUSTIN': "INSTITUTO
SUPERIOR EN CIENCIAS DE LA SALUD SAN AGUSTÍN",
        'INSTITUTO SUPERIOR SAN AGUSTÁN': "INSTITUTO SUPERIOR SAN AGUSTÍN",
        'ESCUELA PARAGUAYA DE ENFERMERIA Y CIENCIAS DE LA SALUD': "ESCUELA
PARAGUAYA DE ENFERMERÍA Y CIENCIAS DE LA SALUD",
        'INSTITUTO DE FORMACIÓN DOCENTE LA PAZ': "INSTITUTO DE FORMACIÓN
DOCENTE LA PAZ",
        'UNIVERSIDAD GRAN ASUNCIÓN': "UNIVERSIDAD GRAN ASUNCIÓN"
    }
}

```

```

# Crear versiones normalizadas para comparación
instituciones_normalizadas = {}

# Primero agregar las instituciones oficiales
for inst in instituciones_oficiales:
    inst_norm = normalizar_texto_simple(inst)
    instituciones_normalizadas[inst_norm] = inst

# Luego agregar las variantes mapeadas
for variante, oficial in mapeo_variantes.items():
    # Corregir caracteres especiales en la variante antes de normalizar
    variante_corregida = corregir_caracteres_especiales(variante)
    variante_norm = normalizar_texto_simple(variante_corregida)
    instituciones_normalizadas[variante_norm] = oficial

return instituciones_oficiales, instituciones_normalizadas,
mapeo_variantes

```

```

def normalizar_texto_simple(texto):
    """Normaliza el texto para comparaciones eliminando acentos, espacios
    extras y convirtiendo a minúsculas"""
    if not isinstance(texto, str):
        texto = str(texto)

    # Eliminar acentos
    texto = unicodedata.normalize('NFD', texto)
    texto = ''.join(c for c in texto if not unicodedata.combining(c))

    # Convertir a minúsculas
    texto = texto.lower()

    # Eliminar espacios múltiples
    texto = re.sub(r'\s+', ' ', texto).strip()

    return texto

def corregir_institucion(institucion, instituciones_oficiales,
instituciones_normalizadas, mapeo_variantes=None):
    """Corrige el nombre de la institución según la lista oficial y mapeo de
    variantes"""
    if not isinstance(institucion, str) or not institucion.strip():
        return institucion

    # Corregir caracteres especiales primero
    institucion_corregida = corregir_caracteres_especiales(institucion)

    # 1. Verificar si existe en el mapeo directo de variantes (coincidencia
    exacta)
    if mapeo_variantes and institucion_corregida in mapeo_variantes:
        return mapeo_variantes[institucion_corregida]

    # 2. Verificar coincidencia exacta en las instituciones oficiales
    if institucion_corregida in instituciones_oficiales:
        return institucion_corregida

    # 3. Normalizar y buscar en el diccionario de instituciones normalizadas
    institucion_norm = normalizar_texto_simple(institucion_corregida)
    if institucion_norm in instituciones_normalizadas:
        return instituciones_normalizadas[institucion_norm]

    # 4. Buscar por prefijo exacto (para casos como "UNIVERSIDAD X - FACULTAD
    Y")
    # Separar por guiones, comas, dos puntos, etc.
    partes = re.split(r'[-,/:"\']', institucion_corregida)
    base_institucion = partes[0].strip()

```



```

    for inst_oficial in instituciones_oficiales:
        if inst_oficial.startswith(base_institucion) or
base_institucion.startswith(inst_oficial):
            return inst_oficial

# 5. Para universidades, tratar de extraer el nombre base
if "UNIVERSIDAD" in institucion_corregida:
    # Buscar todas las instituciones oficiales que contienen
"UNIVERSIDAD"
    unis_oficiales = [inst for inst in instituciones_oficiales if
"UNIVERSIDAD" in inst]

    for uni_oficial in unis_oficiales:
        # Si el nombre de la universidad oficial está contenido en el
texto
        # o si el texto empiece con el mismo prefijo
        palabras_uni = set(uni_oficial.split()[:3]) # Primeras 3
palabras
        palabras_inst = set(institucion_corregida.split()[:4]) #
Primeras 4 palabras

        # Si hay al menos 3 palabras en común al inicio
        if len(palabras_uni & palabras_inst) >= 3:
            return uni_oficial

# 6. Para institutos, similar al caso de universidades
if "INSTITUTO" in institucion_corregida:
    institutos_oficiales = [inst for inst in instituciones_oficiales if
"INSTITUTO" in inst]

    for inst_oficial in institutos_oficiales:
        # Primeras palabras significativas
        palabras_inst_of = set(inst_oficial.split()[:4])
        palabras_inst_in = set(institucion_corregida.split()[:5])

        # Si hay al menos 3 palabras en común al inicio
        if len(palabras_inst_of & palabras_inst_in) >= 3:
            return inst_oficial

# 7. Última opción: buscar la institución con más palabras en común
palabras_inst = set(institucion_corregida.split())
mejor_coincidencia = None
max_palabras_comunes = 2 # Mínimo 3 palabras en común

for inst_oficial in instituciones_oficiales:
    palabras_oficial = set(inst_oficial.split())

```

```

num_palabras_comunes = len(palabras_inst & palabras_oficial)

if num_palabras_comunes > max_palabras_comunes:
    max_palabras_comunes = num_palabras_comunes
    mejor_coincidencia = inst_oficial

# Devolver la mejor coincidencia o el original si no hay suficiente
similitud
return mejor_coincidencia if mejor_coincidencia else
institucion_corregida

def corregir_caracteres_especiales(texto):
    """Corrige los caracteres especiales con codificación incorrecta"""
    if not isinstance(texto, str):
        return texto

    # Corregir combinaciones específicas de caracteres mal codificados
    correcciones = {
        'Ã|': 'á', 'Ã@': 'é', 'Ã': 'í', 'Ã³': 'ó', 'Ãº': 'ú',
        'Ã\x81': 'Á', 'Ã%': 'É', 'Ã\x8d': 'Í', 'Ã"': 'Ó', 'Ãš': 'Ú',
        'Ã±': 'ñ', 'Ã¼': 'ü', 'Ãæ': 'Ü',
        'Ã¸': 'â', 'Ãª': 'ê', 'Ã®': 'î', 'Ã´': 'ô', 'Ã»': 'û',
        'Ã\x82': 'Â', 'ÃŠ': 'Ê', 'ÃŽ': 'Î', 'Ã"': 'Ô', 'Ã¸': 'Û',
        'Ã§': 'ç', 'Ã±': 'ç'
    }

    for mal_codificado, correcto in correcciones.items():
        texto = texto.replace(mal_codificado, correcto)

    return texto

def corregir_carrera_grupo(texto):
    """Función específica para corregir el campo carreraGrupo"""
    if not isinstance(texto, str):
        return texto

    # Correcciones específicas para los grupos de carreras
    correcciones_grupos = {
        'Enfermería y Cuidados Intensivos': 'Enfermería y Cuidados
Intensivos',
        'Psicología y Psiquiatría': 'Psicología y Psiquiatría',
        'Cardiología y Ciencias del Corazón': 'Cardiología y Ciencias del
Corazón',
        'Odontología y Especialidades Dentales': 'Odontología y
Especialidades Dentales',
        'Imagenología y Diagnóstico': 'Imagenología y Diagnóstico',

```

```

        'GinecologÃa, Obstetricia y Salud Reproductiva': 'Ginecología,
Obstetricia y Salud Reproductiva',
        'RehabilitaciÃ³n y Terapia': 'Rehabilitación y Terapia',
        'CirugÃa y Especialidades QuirÃ³rgicas': 'Cirugía y Especialidades
Quirúrgicas',
        'Medicina General y Especialidades MÃ©dicas': 'Medicina General y
Especialidades Médicas',
        'Ciencias FarmacÃ©uticas y BioquÃmicas': 'Ciencias Farmacéuticas y
Bioquímicas',
        'Salud PÃblica y EpidemiologÃa': 'Salud Pública y Epidemiología'
    }

```

```

    # Intentar reemplazo directo primero
    if texto in correcciones_grupos:
        return correcciones_grupos[texto]

    # Aplicar correcciones generales de caracteres especiales
    texto_corregido = corregir_caracteres_especiales(texto)

    # Si la corrección general no funcionó, intentar detectar parcialmente
    for mal_codificado, correcto in correcciones_grupos.items():
        # Verificar si hay coincidencia parcial y corregir
        if texto.startswith(mal_codificado[:5]) or
texto.endswith(mal_codificado[-5:]):
            return correcto

```

```

    return texto_corregido

```

```

def normalizar_fecha(fecha_texto):
    """Normaliza el formato de fecha a dd/mm/yyyy"""
    if not fecha_texto or not isinstance(fecha_texto, str):
        return fecha_texto

```

```

    fecha_texto = fecha_texto.strip()

```

```

    # Identificar formato tipo "yyyy-mm-dd" o variantes
    patron_iso = r'(\d{4})[/-](\d{1,2})[/-](\d{1,2})'
    match_iso = re.match(patron_iso, fecha_texto)

```

```

    if match_iso:
        anio, mes, dia = match_iso.groups()
        # Asegurar que mes y día tengan 2 dígitos
        mes = mes.zfill(2)
        dia = dia.zfill(2)
        return f"{dia}/{mes}/{anio}"

```

```

    # Identificar formato tipo "dd/mm/yyyy" o variantes

```

```

patron_normal = r'(\d{1,2})[/-](\d{1,2})[/-](\d{4})'
match_normal = re.match(patron_normal, fecha_texto)

if match_normal:
    dia, mes, anio = match_normal.groups()
    # Asegurar que mes y día tengan 2 dígitos
    mes = mes.zfill(2)
    dia = dia.zfill(2)
    return f"{dia}/{mes}/{anio}"

# Verificar si es una fecha válida antes de devolver (tratar de
convertir)
try:
    # Intentar diferentes formatos comunes
    formatos = [
        '%d/%m/%Y', '%Y-%m-%d', '%d-%m-%Y',
        '%Y/%m/%d', '%m/%d/%Y', '%d%m%Y'
    ]

    for formato in formatos:
        try:
            fecha_obj = datetime.datetime.strptime(fecha_texto, formato)
            return fecha_obj.strftime('%d/%m/%Y')
        except ValueError:
            continue
except:
    pass

# Si no se pudo convertir, retornar el texto original
return fecha_texto

def cargar_csv(ruta_archivo):
    """Carga un archivo CSV de la manera más segura posible"""
    datos = []
    campos = []

    try:
        # Intentar primero con UTF-8
        with open(ruta_archivo, 'r', encoding='utf-8', errors='replace') as
f:
            # Leer la primera línea para obtener los encabezados
            primera_linea = f.readline().strip()
            campos = [c.strip() for c in primera_linea.split(',')]

            # Volver al inicio del archivo
            f.seek(0)

```

```

        # Leer los datos con DictReader
        reader = csv.DictReader(f)
        for row in reader:
            datos.append(row)

    return datos, campos
except Exception as e:
    print(f"Error al cargar con UTF-8, intentando con latin-1: {str(e)}")
    try:
        # Intentar con latin-1 si UTF-8 falla
        with open(ruta_archivo, 'r', encoding='latin-1',
errors='replace') as f:
            reader = csv.DictReader(f)
            campos = reader.fieldnames
            for row in reader:
                datos.append(row)
            return datos, campos
    except Exception as e:
        print(f"Error al cargar el archivo {ruta_archivo}: {str(e)}")
        return [], []

def guardar_csv(datos, campos, ruta_archivo):
    """Guarda los datos en un archivo CSV"""
    try:
        with open(ruta_archivo, 'w', newline='', encoding='utf-8') as f:
            writer = csv.DictWriter(f, fieldnames=campos)
            writer.writeheader()
            writer.writerows(datos)
        return True
    except Exception as e:
        print(f"Error al guardar el archivo {ruta_archivo}: {str(e)}")
        import traceback
        traceback.print_exc()
        return False

def limpiar_csv(archivo_entrada, archivo_salida):
    """Limpia el CSV corrigiendo fechas, caracteres especiales e
instituciones"""
    try:
        # Cargar lista de instituciones oficiales
        instituciones_oficiales, instituciones_normalizadas, mapeo_variantes
= cargar_lista_instituciones()
        print(f"Lista de instituciones oficiales cargada:
{len(instituciones_oficiales)} instituciones")

        # Cargar el CSV
        print(f"Cargando archivo: {archivo_entrada}")

```

```

datos, campos = cargar_csv(archivo_entrada)

if not datos:
    print("No se pudieron cargar los datos")
    return False

print(f"Archivo cargado correctamente. Total de registros:
{len(datos)}")

# Número de registros procesados por tipo
fechas_corregidas = 0
textos_corregidos = 0
carrera_grupo_corregidas = 0
instituciones_corregidas = 0

# Procesar cada registro
for i, registro in enumerate(datos):
    # Mostrar progreso
    if i % 10000 == 0 and i > 0:
        print(f"Procesados {i} registros...")

    # 1. Corregir la fecha de resolución
    if 'fecha_resolucion' in registro:
        fecha_original = registro['fecha_resolucion']
        fecha_normalizada = normalizar_fecha(fecha_original)

        if fecha_original != fecha_normalizada:
            registro['fecha_resolucion'] = fecha_normalizada
            fechas_corregidas += 1

    # 2. Corregir específicamente el campo carreraGrupo
    if 'carreraGrupo' in registro:
        valor_original = registro['carreraGrupo']
        valor_corregido = corregir_carrera_grupo(valor_original)

        if valor_original != valor_corregido:
            registro['carreraGrupo'] = valor_corregido
            carrera_grupo_corregidas += 1

    # 3. Corregir el campo institución
    if 'institucion' in registro:
        valor_original = registro['institucion']
        valor_corregido = corregir_institucion(valor_original,
instituciones_oficiales, instituciones_normalizadas, mapeo_variantes)

        if valor_original != valor_corregido:
            registro['institucion'] = valor_corregido

```

```

        instituciones_corregidas += 1

# 4. Corregir caracteres especiales en todos los demás campos de
texto
    for campo, valor in registro.items():
        if campo not in ['carreraGrupo', 'institucion'] and
isinstance(valor, str):
            valor_corregido = corregir_caracteres_especiales(valor)
            if valor != valor_corregido:
                registro[campo] = valor_corregido
                textos_corregidos += 1

# Guardar el resultado
print(f"Guardando resultado en: {archivo_salida}")
if guardar_csv(datos, campos, archivo_salida):
    print("Archivo guardado exitosamente")
    print(f"Se corrigieron {fechas_corregidas} fechas")
    print(f"Se corrigieron {carrera_grupo_corregidas} campos de
carreraGrupo")
    print(f"Se corrigieron {instituciones_corregidas} campos de
institución")
    print(f"Se corrigieron caracteres especiales en
{textos_corregidos} otros campos")

# Mostrar ejemplos de correcciones en carreraGrupo
print("\nEjemplos de campos carreraGrupo corregidos:")
ejemplos_mostrados = 0
grupos_unicos = {}

for registro in datos:
    if 'carreraGrupo' in registro and registro['carreraGrupo']:
        grupo = registro['carreraGrupo']
        if grupo not in grupos_unicos:
            grupos_unicos[grupo] = True
            print(f"- {grupo}")
            ejemplos_mostrados += 1
            if ejemplos_mostrados >= 10:
                break

# Mostrar ejemplos de instituciones corregidas
print("\nEjemplos de instituciones corregidas:")
ejemplos_mostrados = 0
instituciones_unicas = {}

for registro in datos:
    if 'institucion' in registro and registro['institucion']:
        institucion = registro['institucion']

```

```

        if institucion not in instituciones_unicas:
            instituciones_unicas[institucion] = True
            print(f"- {institucion}")
            ejemplos_mostrados += 1
            if ejemplos_mostrados >= 10:
                break

# Mostrar ejemplos de registros procesados
print("\nEjemplos de registros completos procesados:")
for i, registro in enumerate(datos[:3]): # Mostrar solo 3
registros
    print(f"\nRegistro {i+1}:")
    if 'carrera' in registro:
        print(f"Carrera: {registro['carrera']}")
    if 'carreraGrupo' in registro:
        print(f"Grupo de carrera: {registro['carreraGrupo']}")
    if 'institucion' in registro:
        print(f"Institución: {registro['institucion']}")
    if 'fecha_resolucion' in registro:
        print(f"Fecha resolución:
{registro['fecha_resolucion']}")

    return True
else:
    print("Error al guardar el archivo")
    return False

except Exception as e:
    print(f"Error en el proceso: {str(e)}")
    import traceback
    traceback.print_exc()
    return False

def main():
    archivo_entrada = 'titulos_con_gestion.csv'
    archivo_salida = 'titulos_final.csv'

    print("Script para limpiar formato de fechas, caracteres especiales e
instituciones")
    print("-----")
    print("-----")
    # Verificar que el archivo existe
    if not os.path.exists(archivo_entrada):
        print(f"Error: No se encontró el archivo {archivo_entrada}")
        return

    limpiar_csv(archivo_entrada, archivo_salida)

```



```

print("\nProceso completado.")

if __name__ == "__main__":
    main()

```

1.5. Estructura para añadir campo de tipo de gestión a títulos

```

import csv
import unicodedata
import os
import re

def normalizar_texto(texto):
    """Normaliza el texto de manera más intensiva para mejorar
    coincidencias"""
    if not isinstance(texto, str):
        texto = str(texto)
    # Normalizar NFD y eliminar diacríticos
    texto = unicodedata.normalize('NFD', texto)
    texto = ''.join(c for c in texto if not unicodedata.combining(c))
    # Convertir a minúsculas
    texto = texto.lower()
    # Eliminar espacios extras
    texto = " ".join(texto.split())
    # Eliminar símbolos y puntuación
    texto = re.sub(r'^\w\s]', '', texto)
    # Reemplazar caracteres problemáticos
    texto = texto.replace('ã', 'a')
    texto = texto.replace('ö', 'o')
    texto = texto.replace('ë', 'e')
    texto = texto.replace('ï', 'i')
    texto = texto.replace('ü', 'u')
    # Eliminar palabras comunes que a veces se omiten
    texto = re.sub(r'\bde\b|\bdel\b|\bla\b|\blas\b|\blos\b|\by\b', ' ',
    texto)
    # Eliminar espacios extras nuevamente después de las sustituciones
    texto = " ".join(texto.split())
    return texto

def obtener_palabras_clave(texto):
    """Extrae palabras clave significativas del nombre de la institución"""
    if not texto:
        return []

    texto_norm = normalizar_texto(texto)

    # Palabras a ignorar

```

```

    ignorar = ['de', 'del', 'la', 'las', 'los', 'y', 'e', 'en', 'con',
'para', 'por',
               'instituto', 'universidad', 'facultad', 'escuela', 'centro',
'colegio',
               'superior', 'tecnico', 'nacional', 'regional',
'departamental', 'privado',
               'publico', 'oficial']

    palabras = texto_norm.split()
    palabras_clave = [p for p in palabras if len(p) > 3 and p not in ignorar]

    return palabras_clave

def esta_contenido(texto1, texto2):
    """Verifica si un texto está contenido en otro o viceversa"""
    if not texto1 or not texto2:
        return False

    texto1_norm = normalizar_texto(texto1)
    texto2_norm = normalizar_texto(texto2)

    # Verificar si uno está contenido en el otro (después de normalizar)
    return texto1_norm in texto2_norm or texto2_norm in texto1_norm

def cargar_csv(ruta_archivo):
    """Carga un archivo CSV usando métodos de bajo nivel para evitar
    problemas"""
    datos = []
    try:
        with open(ruta_archivo, 'r', encoding='utf-8') as f:
            # Leer las cabeceras
            primera_linea = f.readline().strip()
            nombres_campos = primera_linea.split(',')
            nombres_campos = [campo.strip() for campo in nombres_campos]

            # Leer los datos línea por línea
            for linea in f:
                valores = linea.strip().split(',')
                # Asegurar que tengamos suficientes valores
                while len(valores) < len(nombres_campos):
                    valores.append('')
                # Truncar si hay demasiados valores
                if len(valores) > len(nombres_campos):
                    valores = valores[:len(nombres_campos)]

                # Crear un diccionario para esta fila
                fila = {}

```

```

        for i, nombre in enumerate(nombres_campos):
            if i < len(valores):
                fila[nombre] = valores[i]
            else:
                fila[nombre] = ''

        datos.append(fila)

    return datos, nombres_campos
except Exception as e:
    print(f"Error al cargar el archivo {ruta_archivo}: {str(e)}")
    return [], []

def guardar_csv_simple(datos, campos, ruta_archivo):
    """Guarda los datos en un archivo CSV de manera simple y directa"""
    try:
        with open(ruta_archivo, 'w', newline='', encoding='utf-8') as f:
            # Escribir cabecera
            f.write(','.join(campos) + '\n')

            # Escribir datos
            for fila in datos:
                valores = []
                for campo in campos:
                    # Obtener el valor o un string vacío
                    valor = str(fila.get(campo, ''))
                    # Encerrar en comillas si contiene comas
                    if ',' in valor:
                        valor = f'"{valor}"'
                    valores.append(valor)
                f.write(','.join(valores) + '\n')

        return True
    except Exception as e:
        print(f"Error al guardar el archivo {ruta_archivo}: {str(e)}")
        import traceback
        traceback.print_exc()
        return False

def mostrar_ejemplos_relaciones(titulos, mapa_nombres_originales,
                                mapa_instituciones):
    """Muestra ejemplos de relaciones entre instituciones"""
    print("\nEjemplos de relaciones entre nombres de instituciones:")
    ejemplos_mostrados = 0

    for i, titulo in enumerate(titulos):
        if ejemplos_mostrados >= 5:

```

```

        break

    if 'institucion' not in titulo or not titulo['institucion']:
        continue

    nombre_institucion = titulo['institucion']

    # Buscar en mapas
    encontrado = False
    tipo_relacion = ""
    institucion_relacionada = ""

    # Comprobar contención
    for nombre_norm, nombre_original in mapa_nombres_originales.items():
        if esta_contenido(nombre_institucion, nombre_original):
            encontrado = True
            tipo_relacion = "Contención"
            institucion_relacionada = nombre_original
            break

    if encontrado:
        print(f"\nInstitucion en títulos: {nombre_institucion}")
        print(f"Institución relacionada: {institucion_relacionada}")
        print(f"Tipo de relación: {tipo_relacion}")
        print(f"Tipo gestión asignado:
{mapa_instituciones.get(normalizar_texto(institucion_relacionada), 'No
disponible'}}")
        ejemplos_mostrados += 1

def agregar_tipo_gestion(archivo_titulos, archivo_carreras, archivo_salida):
    """
    Agrega la columna tipo_gestion al archivo de títulos, usando contención
    entre los nombres de institución para las coincidencias
    """
    try:
        # Cargar ambos archivos
        print(f"Cargando archivo de títulos: {archivo_titulos}")
        titulos, campos_titulos = cargar_csv(archivo_titulos)

        if not titulos:
            print("No se pudieron cargar los datos de títulos")
            return False

        print(f"Registros cargados en títulos: {len(titulos)}")

        print(f"Cargando archivo de carreras: {archivo_carreras}")
        carreras, _ = cargar_csv(archivo_carreras)

```

```

if not carreras:
    print("No se pudieron cargar los datos de carreras")
    return False

print(f"Registros cargados en carreras: {len(carreras)}")

# Verificar campos requeridos
if 'institucion' not in campos_titulos:
    print("Error: Falta el campo 'institucion' en el archivo de
títulos")
    return False

    if 'nombre_institucion' not in carreras[0] or 'tipo_gestion' not in
carreras[0]:
        print("Error: Faltan campos requeridos en el archivo de
carreras")
        return False

# Crear mapas de instituciones
mapa_instituciones = {} # Mapa de nombres originales a tipo_gestion
mapa_nombres_normalizados = {} # Mapa de nombres normalizados a
nombres originales

# Poblar los mapas
for carrera in carreras:
    if 'nombre_institucion' in carrera and 'tipo_gestion' in carrera:
        nombre_original = carrera['nombre_institucion']
        if nombre_original and carrera['tipo_gestion']:
            # Guardar relación directa
            mapa_instituciones[nombre_original] =
carrera['tipo_gestion']
            # Guardar versión normalizada
            nombre_norm = normalizar_texto(nombre_original)
            mapa_nombres_normalizados[nombre_norm] = nombre_original

print(f"Se creó mapa de {len(mapa_instituciones)} instituciones")

# Añadir el nuevo campo si no existe
if 'tipo_gestion' not in campos_titulos:
    campos_titulos.append('tipo_gestion')

# Estadísticas
coincidencias_exactas = 0
coincidencias_contenido = 0
sin_coincidencia = 0

```

```

# Agregar el campo tipo_gestion a los títulos
for titulo in titulos:
    if 'institucion' not in titulo or not titulo['institucion']:
        titulo['tipo_gestion'] = ""
        sin_coincidencia += 1
        continue

    nombre_institucion = titulo['institucion']
    nombre_norm = normalizar_texto(nombre_institucion)

    # 1. Intentar coincidencia exacta primero
    if nombre_institucion in mapa_instituciones:
        titulo['tipo_gestion'] =
mapa_instituciones[nombre_institucion]
        coincidencias_exactas += 1
        continue

    # 2. Intentar coincidencia normalizada
    if nombre_norm in mapa_nombres_normalizados:
        nombre_original = mapa_nombres_normalizados[nombre_norm]
        titulo['tipo_gestion'] = mapa_instituciones[nombre_original]
        coincidencias_exactas += 1
        continue

    # 3. Verificar contención
    encontrado = False
    for nombre_original in mapa_instituciones.keys():
        if esta_contenido(nombre_institucion, nombre_original):
            titulo['tipo_gestion'] =
mapa_instituciones[nombre_original]
            coincidencias_contenido += 1
            encontrado = True
            break

    if not encontrado:
        titulo['tipo_gestion'] = ""
        sin_coincidencia += 1

# Mostrar estadísticas
print(f"\nCoincidencias exactas: {coincidencias_exactas}")
print(f"Coincidencias por contención: {coincidencias_contenido}")
print(f"Total coincidencias: {coincidencias_exactas +
coincidencias_contenido}")
print(f"Registros sin coincidencia: {sin_coincidencia}")

# Mostrar ejemplos de relaciones

```

```

        mostrar_ejemplos_relaciones(titulos, mapa_nombres_normalizados,
mapa_instituciones)

    # Guardar el resultado
    print(f"\nGuardando resultado en: {archivo_salida}")
    if guardar_csv_simple(titulos, campos_titulos, archivo_salida):
        print("Archivo guardado exitosamente")

    # Mostrar ejemplos del resultado final
    print("\nEjemplos de registros procesados:")
    ejemplos_mostrados = 0
    for titulo in titulos:
        if titulo.get('tipo_gestion'):
            print(f"\nInstitución: {titulo.get('institucion', '')}")
            print(f"Tipo Gestión: {titulo.get('tipo_gestion', '')}")
            ejemplos_mostrados += 1
            if ejemplos_mostrados >= 5:
                break

    return True
else:
    print("Error al guardar el archivo")
    return False

except Exception as e:
    print(f"Error en el proceso: {str(e)}")
    import traceback
    traceback.print_exc()
    return False

def main():
    archivo_titulos = 'titulosSalud_completo.csv'
    archivo_carreras = 'carrerasSalud.csv'
    archivo_salida = 'titulos_con_gestion.csv'

    print("Script para agregar tipo_gestion a titulos_limpios.csv")
    print("-----")

    # Verificar que los archivos existen
    if not os.path.exists(archivo_titulos):
        print(f"Error: No se encontró el archivo {archivo_titulos}")
        return

    if not os.path.exists(archivo_carreras):
        print(f"Error: No se encontró el archivo {archivo_carreras}")
        return

```

```

    agregar_tipo_gestion(archivo_titulos, archivo_carreras, archivo_salida)

    print("\nProceso completado.")

if __name__ == "__main__":
    main()

```

2. Código Fuente del Dashboard Académico del Área de Salud

2.1. Estructura General del Dashboard (index.html)

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard Académico del Área de Salud</title>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.18.5/xlsx.full.min.js"></s
cript>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.min.js">
</script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.9.1/chart.min.js"></sc
ript>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="loading-overlay" id="loadingOverlay">
        <div class="spinner"></div>
        <p class="loading-text">Cargando datos...</p>
    </div>

    <div class="dashboard-layout">
        <!-- Sidebar con filtros -->
        <div class="sidebar">
            <div class="profile-section">
                <div class="logo-container">
                    <!-- Logo personalizado con dimensiones específicas -->
                    

```



```

-->
        <!-- Imagen de respaldo en caso de que el logo no cargue
-->
        <div class="fallback-logo">
            <i class="fas fa-stethoscope"></i> <!-- Añadido un
icono de respaldo -->
        </div>
    </div>
    <h3>Área de Salud</h3>
    <p>Dashboard Académico</p>
</div>

<nav class="sidebar-nav">
    <ul>
        <li class="active">
            <a href="#"><i class="fas fa-home"></i> Inicio</a>
        </li>
        <li class="filter-section">
            <a href="#"><i class="fas fa-filter"></i> Filtros</a>
            <div class="filters-container">
                <div class="filter-group">
                    <label class="filter-label">Año</label>
                    <div class="range-filter">
                        <select id="yearFilterFrom">
                            <option value="all">Desde</option>
                        </select>
                        <select id="yearFilterTo">
                            <option value="all">Hasta</option>
                        </select>
                    </div>
                </div>
                <div class="filter-group">
                    <label class="filter-label">Tipo
Institución</label>

                    <select id="institutionTypeFilter">
                        <option value="all">Todos los
tipos</option>

                    </select>
                </div>
                <div class="filter-group">
                    <label class="filter-label">Carrera</label>
                    <select id="careerFilter">
                        <option value="all">Todas las
carreras</option>

                    </select>
                </div>
                <div class="filter-group">

```

```

        <label class="filter-label">Tipo de
Gestión</label>
        <select id="managementTypeFilter">
            <option value="all">Todos los
tipos</option>
        </select>
    </div>
    <div class="filter-group">
        <label class="filter-label">Género</label>
        <select id="genderFilter">
            <option value="all">Todos</option>
            <option value="MUJER">Mujer</option>
            <option value="HOMBRE">Hombre</option>
        </select>
    </div>
    <button id="applyFilters">Aplicar
Filtros</button>
    </div>
</li>
<li>
    <a href="#"><i class="fas fa-file-alt"></i>
Reporte</a>
    </li>
<li>
    <a href="#"><i class="fas fa-chart-line"></i>
Análisis</a>
    </li>
<li>
    <a href="#"><i class="fas fa-cog"></i>
Configuración</a>
    </li>
</ul>
</nav>
</div>

<!-- Contenido principal -->
<div class="main-content">
    <header class="dashboard-header">
        <h1>Dashboard Académico del Área de Salud</h1>
        <div class="header-controls">
            <button class="btn-icon"><i class="fas fa-
bell"></i></button>
            <button class="btn-icon"><i class="fas fa-sync-
alt"></i></button>
            <button class="btn-icon"><i class="fas fa-
user"></i></button>
        </div>

```

```

</header>

<!-- Tarjeta Estadística -->
<div class="stats-container" id="statsContainer">

</div>

<!-- Dashboard -->
<div class="dashboard" id="dashboard">
  <div class="chart-container">
    <h3 class="chart-title">Distribución por Tipo de
Institución</h3>
    <canvas id="institutionTypeChart"></canvas>
  </div>
  <div class="chart-container">
    <h3 class="chart-title">Distribución por Género</h3>
    <canvas id="genderChart"></canvas>
  </div>
  <div class="chart-container large-chart">
    <h3 class="chart-title">Evolución de Títulos por Año</h3>
    <canvas id="yearTrendChart"></canvas>
  </div>
  <div class="chart-container large-chart">
    <h3 class="chart-title">Distribución por Género y
Año</h3>
    <canvas id="genderYearChart"></canvas>
  </div>
  <div class="chart-container">
    <h3 class="chart-title">Top 10 Carreras</h3>
    <canvas id="careersChart"></canvas>
  </div>
  <div class="chart-container">
    <h3 class="chart-title">Distribución por Tipo de
Gestión</h3>
    <canvas id="managementTypeChart"></canvas>
  </div>
  <div class="chart-container large-chart">
    <h3 class="chart-title">Top 10 Instituciones</h3>
    <canvas id="institutionsChart"></canvas>
  </div>
  <div class="chart-container large-chart">
    <h3 class="chart-title">Tendencia Mensual de Títulos</h3>
    <canvas id="monthlyTrendChart"></canvas>
  </div>
</div>

<footer>

```

```

        <p>Evaluación Descriptiva de la Oferta Académica y Egreso en
Carreras del Area de Salud: Una Aproximación desde Data Analytics</p>
        <p>Carlos Rodriguez</p>
        <p>UNCA 2025</p>
    </footer>
</div>
</div>

<script src="script.js"></script>
<script>
    // Manejar el error de carga del logo
    document.addEventListener('DOMContentLoaded', function() {
        const logo = document.getElementById('sidebar-logo');

        if (logo) {
            logo.addEventListener('error', function() {
                // Agregar clase de error para mostrar el fallback
                this.classList.add('error');
                console.log('Error al cargar el logo. Mostrando imagen de
respaldo.');
```

2.2. Estilos Visuales del Dashboard (styles.css)

```

.range-filter {
    display: flex;
    gap: 10px;
}

.range-filter select {
    flex: 1;
}:root {
    --primary: #1b3a57;
    --primary-light: #29567d;
    --secondary: #f89c1c;
    --background: #eae0f2;
    --card-bg: #ffffff;
    --text: #333333;
    --text-light: #f1f1f1;
    --border: #e0e0e0;
    --shadow: rgba(0, 0, 0, 0.1);
    --chart-blue: #1e88e5;

```

```

    --chart-skyblue: rgba(136,167,217,255);
    --chart-pink: #ec407a;
    --sidebar-width: 250px;
}

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

body {
    background-color: var(--background);
    color: var(--text);
    overflow-x: hidden;
}

.dashboard-layout {
    display: flex;
    min-height: 100vh;
}

/* Barra Lateral*/
/* Estilos para el logo personalizado - ajustados */
.logo-container {
    width: 150px;
    height: 150px;
    border-radius: 50%;
    background-color: white;
    margin: 0 auto 15px;
    display: flex;
    align-items: center;
    justify-content: center;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
    overflow: hidden;
    position: relative;
}

#sidebar-logo {
    max-width: 100%;
    max-height: 100%;
    object-fit: contain;
    display: block;
}

.fallback-logo {

```

```

    position: absolute;
    top: 0;
    left: 0;

    display: flex;
    align-items: center;
    justify-content: center;
    opacity: 0;
    transition: opacity 0.3s;
    background-color: white;
}

/* Se muestra la imagen de respaldo si el logo no carga correctamente */
#sidebar-logo:not([src]),
#sidebar-logo[src=""],
#sidebar-logo[src="null"],
#sidebar-logo.error {
    display: none;
}

#sidebar-logo:not([src]) ~ .fallback-logo,
#sidebar-logo[src=""] ~ .fallback-logo,
#sidebar-logo[src="null"] ~ .fallback-logo,
#sidebar-logo.error ~ .fallback-logo {
    opacity: 1;
}

/* Ajuste adicional para mejorar la visualización del logo */
@media (max-width: 768px) {
    .logo-container {
        width: 120px;
        height: 120px;
    }

    .fallback-logo i {
        font-size: 100%;
    }
}

.sidebar {
    width: var(--sidebar-width);
    background-color: var(--primary);
    color: var(--text-light);
    box-shadow: 4px 0 10px rgba(0, 0, 0, 0.1);
    z-index: 10;
    overflow-y: auto;
    position: fixed;
    height: 100vh;
}

```

```
}

.profile-section {
  padding: 30px 20px;
  text-align: center;
  border-bottom: 1px solid rgba(255, 255, 255, 0.1);
}

.profile-section h3 {
  font-size: 18px;
  margin-bottom: 5px;
}

.profile-section p {
  font-size: 14px;
  opacity: 0.8;
}

.sidebar-nav ul {
  list-style: none;
  padding: 0;
}

.sidebar-nav li {
  margin-bottom: 5px;
}

.sidebar-nav a {
  display: flex;
  align-items: center;
  padding: 15px 20px;
  color: var(--text-light);
  text-decoration: none;
  transition: background-color 0.3s;
}

.sidebar-nav a:hover, .sidebar-nav li.active a {
  background-color: var(--primary-light);
}

.sidebar-nav i {
  margin-right: 10px;
  font-size: 18px;
  min-width: 24px;
  text-align: center;
}
```

```
.filter-section {
  border-left: 3px solid var(--secondary);
}

.filters-container {
  padding: 0 20px 20px 50px;
}

.filter-group {
  margin-bottom: 15px;
}

.filter-label {
  display: block;
  margin-bottom: 5px;
  font-weight: 600;
  font-size: 13px;
  color: var(--text-light);
}

.filters-container select {
  width: 100%;
  padding: 8px 12px;
  border: 1px solid rgba(255, 255, 255, 0.1);
  border-radius: 5px;
  background-color: var(--primary-light);
  color: white;
  font-size: 13px;
}

.filters-container button {
  width: 100%;
  padding: 10px 15px;
  background-color: var(--secondary);
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  font-weight: 600;
  margin-top: 5px;
  transition: background-color 0.3s;
}

.filters-container button:hover {
  background-color: #e6910a;
}
```



```
/* Contenido Principal */
.main-content {
  flex: 1;
  padding: 20px;
  margin-left: var(--sidebar-width);
  width: calc(100% - var(--sidebar-width));
}

.dashboard-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 25px;
  background-color: var(--card-bg);
  padding: 15px 20px;
  border-radius: 10px;
  box-shadow: 0 2px 5px var(--shadow);
}

.dashboard-header h1 {
  font-size: 20px;
  font-weight: 600;
  color: var(--primary);
}

.header-controls {
  display: flex;
  gap: 10px;
}

.btn-icon {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  background-color: var(--background);
  border: none;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  transition: background-color 0.3s;
}

.btn-icon:hover {
  background-color: var(--border);
}
```

```
/* Tarjeta Estadísticas*/
.stats-container {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
  margin-bottom: 25px;
}

.stat-card {
  background-color: var(--card-bg);
  border-radius: 10px;
  padding: 20px;
  box-shadow: 0 2px 5px var(--shadow);
  flex: 1;
  min-width: 180px;
  display: flex;
  flex-direction: column;
  justify-content: center;
  position: relative;
  overflow: hidden;
}

.stat-card::before {
  content: '';
  position: absolute;
  top: 0;
  left: 0;
  width: 5px;
  height: 100%;
  background-color: var(--chart-blue);
}

.stat-card:nth-child(2)::before {
  background-color: var(--chart-pink);
}

.stat-card:nth-child(3)::before {
  background-color: var(--chart-skyblue);
}

.stat-card:nth-child(4)::before {
  background-color: var(--secondary);
}

.stat-value {
  font-size: 28px;
  font-weight: 700;
}
```

```

        margin: 5px 0;
        color: var(--primary);
    }

    .stat-label {
        font-size: 14px;
        color: #757575;
        font-weight: 500;
    }

    /* Dashboard Graficos*/
    .dashboard {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
        gap: 15px;
        margin-bottom: 10px;
    }

    .chart-container {
        background-color: var(--card-bg);
        border-radius: 10px;
        padding: 12px;
        box-shadow: 0 2px 5px var(--shadow);
        transition: transform 0.3s, box-shadow 0.3s;
        height: auto;
        position: relative;
    }

    .chart-container:hover {
        transform: translateY(-5px);
        box-shadow: 0 6px 12px var(--shadow);
    }

    .chart-container canvas {
        min-height: 250px;
        max-height: 350px;
        overflow: hidden;
        max-height: 350px;
        width: 100% !important;
    }

    /* Estilos específicos para el gráfico Top 10 Carreras */
    .chart-container:has(#careersChart) {
        height: 350px;
        min-height: 350px;
        display: flex;
        flex-direction: column;
    }

```

```

}

.chart-container:has(#careersChart) canvas {
  flex: 1;
  min-height: 250px;
}

.chart-title {
  font-size: 16px;
  font-weight: 600;
  margin-bottom: 20px;
  color: var(--primary);
  text-align: center;
  position: relative;
  padding-bottom: 10px;
}

.chart-title::after {
  content: '';
  position: absolute;
  bottom: 0;
  left: 50%;
  transform: translateX(-50%);
  width: 50px;
  height: 3px;
  background-color: var(--secondary);
  border-radius: 3px;
}

.large-chart {
  grid-column: span 2;
}

/* Loading */
.loading-overlay {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(255, 255, 255, 0.9);
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  z-index: 1000;
}

```

```
.spinner {
  border: 4px solid rgba(0, 0, 0, 0.1);
  width: 50px;
  height: 50px;
  border-radius: 50%;
  border-left-color: var(--secondary);
  animation: spin 1s linear infinite;
}
```

```
.loading-text {
  margin-top: 15px;
  font-weight: 600;
  color: var(--primary);
}
```

```
@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
```

```
/* Footer */
footer {
  text-align: center;
  margin-top: 30px;
  padding-top: 20px;
  border-top: 1px solid var(--border);
  font-size: 14px;
  color: #757575;
}
```

```
/* Mensaje Error*/
.error-message {
  background-color: #f44336;
  color: white;
  padding: 20px;
  border-radius: 10px;
  margin: 20px 0;
  text-align: center;
  box-shadow: 0 2px 5px var(--shadow);
}
```

```
.error-message button {
  background-color: white;
  color: #f44336;
  border: none;
  padding: 10px 20px;
  margin-top: 15px;
}
```

```

        border-radius: 5px;
        cursor: pointer;
        font-weight: 600;
        transition: background-color 0.3s;
    }

    .error-message button:hover {
        background-color: #f1f1f1;
    }

    /* Carga Archivo */
    .file-input-container {
        text-align: center;
        padding: 30px;
        background-color: var(--card-bg);
        border-radius: 10px;
        margin-bottom: 20px;
        box-shadow: 0 2px 5px var(--shadow);
    }

    .file-input-container label {
        display: block;
        margin-bottom: 15px;
        font-weight: 600;
        color: var(--primary);
    }

    .file-input-container input[type="file"] {
        display: block;
        margin: 0 auto;
        padding: 15px;
        border: 2px dashed var(--border);
        border-radius: 10px;
        width: 100%;
        max-width: 500px;
        background-color: var(--background);
    }

    /*Diseño Responsive*/
    @media (max-width: 1200px) {
        .large-chart {
            grid-column: span 1;
        }
    }

    @media (max-width: 768px) {
        .sidebar {

```

```

        width: 70px;
        overflow: visible;
    }

    .sidebar:hover {
        width: var(--sidebar-width);
    }

    .profile-section h3, .profile-section p, .filter-label, .filters-
container {
        display: none;
    }

    .sidebar:hover .profile-section h3,
    .sidebar:hover .profile-section p,
    .sidebar:hover .filter-label,
    .sidebar:hover .filters-container {
        display: block;
    }

    .sidebar-nav span {
        display: none;
    }

    .sidebar:hover .sidebar-nav span {
        display: inline;
    }

    .main-content {
        padding: 8px;
        margin-left: 70px;
        ;
    }

    .dashboard {
        grid-template-columns: 1fr;
    }
}

@media (max-width: 576px) {
    .stats-container {
        flex-direction: column;
    }

    .dashboard-header {
        flex-direction: column;
        gap: 10px;
    }
}

```

```

        align-items: flex-start;
    }
}

```

2.3. Lógica de Procesamiento de Datos (script.js)

```

// Variables globales
let allData = [];
let filteredData = [];
let charts = {};
// Colores para gráficos
const colorPalette = [
    '#c45a10', '#d0d0d0', '#949494',
];
const colorPalette2 = [
    '#1e88e5', '#26a69a', '#7e57c2'
];
// Depuración: Mostrar mensajes de consola visibles en la interfaz si hay errores
window.onerror = function(message, source, lineno, colno, error) {
    const errorDiv = document.createElement('div');
    errorDiv.style.position = 'fixed';
    errorDiv.style.bottom = '10px';
    errorDiv.style.right = '10px';
    errorDiv.style.backgroundColor = '#f44336';
    errorDiv.style.color = 'white';
    errorDiv.style.padding = '10px';
    errorDiv.style.borderRadius = '5px';
    errorDiv.style.zIndex = '1000';
    errorDiv.innerHTML = `Error: ${message}<br>Línea:
${lineno}`;
    document.body.appendChild(errorDiv);

    setTimeout(() => {
        errorDiv.remove();
    }, 5000);

    return false;
};
// Función para cargar y procesar los datos del Excel (versión local)
async function loadData() {
    try {
        // Mostrar pantalla de carga
        document.getElementById('loadingOverlay').style.display = 'flex';
        console.log("Intentando cargar dashboard_chicho.xlsx...");
    }
}

```



```

// Cargar el archivo Excel usando fetch (versión local)
const response = await fetch('./data/dashboard_chicho.xlsx');
const excelData = await response.arrayBuffer();

console.log("Archivo Excel cargado correctamente, procesando
datos...");

// Procesar el archivo con SheetJS
const workbook = XLSX.read(new Uint8Array(excelData), {
  cellDates: true,
  cellNF: true,
  cellStyles: true
});

// Mostrar información sobre el libro Excel
console.log("Información del libro Excel:");
console.log("- Hojas disponibles:", workbook.SheetNames);
// Obtener datos de la primera hoja
const sheetName = workbook.SheetNames[0];
const sheet = workbook.Sheets[sheetName];
// Convertir a JSON con opciones para manejar fechas correctamente
allData = XLSX.utils.sheet_to_json(sheet, {
  raw: false,
  dateNF: 'yyyy-mm-dd',
  defval: '' // Valor por defecto para celdas vacías
});
// Validar la estructura de los datos
validateExcelData(allData);

console.log(`Datos cargados exitosamente: ${allData.length}
registros`);
console.log("Muestra de datos:", allData.slice(0, 2));

// Inicializar los filtros
initializeFilters();

// Aplicar filtros iniciales (mostrar todo)
filteredData = [...allData];

// Actualizar las estadísticas y gráficos
updateStats();
createCharts();

// Ocultar pantalla de carga
document.getElementById('loadingOverlay').style.display = 'none';
} catch (error) {
  console.error("Error cargando datos:", error);
}

```

```

// Mostrar un mensaje de error más descriptivo
let errorMessage = "Error al cargar los datos: ";

if (error.message) {
    errorMessage += error.message;
} else {
    errorMessage += "Verifique que el archivo Excel tenga el formato
correcto.";
}

// Crear un elemento de alerta en la página
const alertDiv = document.createElement('div');
alertDiv.className = "error-message";
alertDiv.innerHTML = `Error:</strong>
${errorMessage}<br>Asegúrate de que el archivo 'dashboard_chicho.xlsx' está
en la carpeta '/data'<br><button id="retryBtn">Reintentar</button>`;

// Reemplazar el contenido del dashboard con el mensaje de error
const dashboardEl = document.getElementById('dashboard');
dashboardEl.innerHTML = '';
dashboardEl.appendChild(alertDiv);

// Agregar opción para usar carga manual de archivo
const fileInputContainer = document.createElement('div');
fileInputContainer.className = "file-input-container";
fileInputContainer.innerHTML = `
    <label for="excelFile">Cargar archivo Excel manualmente:</label>
    <input type="file" id="excelFile" accept=".xlsx, .xls">
`;
dashboardEl.appendChild(fileInputContainer);

// Agregar evento para el input de archivo
document.getElementById('excelFile').addEventListener('change',
handleFileUpload);

// Agregar evento para reintentar
document.getElementById('retryBtn').addEventListener('click', () => {
    location.reload();
});

// Ocultar pantalla de carga
document.getElementById('loadingOverlay').style.display = 'none';
}
}

// Función para manejar la carga de archivos manual

```

```

function handleFileUpload(event) {
    const file = event.target.files[0];
    if (!file) return;

    // Mostrar pantalla de carga
    document.getElementById('loadingOverlay').style.display = 'flex';

    const reader = new FileReader();

    reader.onload = function(e) {
        try {
            const data = new Uint8Array(e.target.result);
            const workbook = XLSX.read(data, {
                cellDates: true,
                cellNF: true,
                cellStyles: true
            });

            // Obtener datos de la primera hoja
            const sheet = workbook.Sheets[workbook.SheetNames[0]];
            allData = XLSX.utils.sheet_to_json(sheet, {
                raw: false,
                dateNF: 'yyyy-mm-dd',
                defval: ''
            });

            // Validar la estructura de los datos
            validateExcelData(allData);

            // Inicializar los filtros
            initializeFilters();

            // Aplicar filtros iniciales (mostrar todo)
            filteredData = [...allData];

            // Restaurar el dashboard
            document.getElementById('dashboard').innerHTML = `
                <div class="chart-container">
                    <h3 class="chart-title">Distribución por Tipo de
Institución</h3>
                    <canvas id="institutionTypeChart"></canvas>
                </div>
                <div class="chart-container">
                    <h3 class="chart-title">Distribución por Género</h3>
                    <canvas id="genderChart"></canvas>
                </div>
                <div class="chart-container large-chart">

```

```

        <h3 class="chart-title">Evolución de Títulos por Año</h3>
        <canvas id="yearTrendChart"></canvas>
    </div>
    <div class="chart-container large-chart">
        <h3 class="chart-title">Distribución por Género y
Año</h3>

        <canvas id="genderYearChart"></canvas>
    </div>
    <div class="chart-container">
        <h3 class="chart-title">Top 10 Carreras</h3>
        <canvas id="careersChart"></canvas>
    </div>
    <div class="chart-container">
        <h3 class="chart-title">Distribución por Tipo de
Gestión</h3>

        <canvas id="managementTypeChart"></canvas>
    </div>
    <div class="chart-container large-chart">
        <h3 class="chart-title">Top 10 Instituciones</h3>
        <canvas id="institutionsChart"></canvas>
    </div>
    <div class="chart-container large-chart">
        <h3 class="chart-title">Tendencia Mensual de Títulos</h3>
        <canvas id="monthlyTrendChart"></canvas>
    </div>
    `;

    // Actualizar las estadísticas y gráficos
    updateStats();
    createCharts();

    // Ocultar pantalla de carga
    document.getElementById('loadingOverlay').style.display = 'none';
} catch (error) {
    console.error("Error al procesar el archivo:", error);
    alert("Error al procesar el archivo: " + error.message);
    document.getElementById('loadingOverlay').style.display = 'none';
}
};

reader.onerror = function(error) {
    console.error("Error al leer el archivo:", error);
    alert("Error al leer el archivo.");
    document.getElementById('loadingOverlay').style.display = 'none';
};

reader.readAsArrayBuffer(file);

```

```

}

// Función auxiliar para manejar casos donde los datos pueden ser
inconsistentes
function safeGetProperty(obj, prop, defaultValue = '') {
  if (!obj || obj[prop] === undefined || obj[prop] === null) {
    return defaultValue;
  }
  return obj[prop];
}

// Función para comprobar si la estructura de datos del Excel es la esperada
function validateExcelData(data) {
  if (!Array.isArray(data) || data.length === 0) {
    throw new Error("Los datos del Excel no son válidos o están vacíos");
  }

  // Verificar que al menos algunas propiedades clave estén presentes
  const requiredFields = ['anio', 'mes', 'carrera', 'sexo',
'tipo_institucion', 'institucion'];
  const sampleRow = data[0];

  const missingFields = requiredFields.filter(field => sampleRow[field] ===
undefined);
  if (missingFields.length > 0) {
    console.warn(`Advertencia: Faltan algunos campos esperados en los
datos: ${missingFields.join(', ')}`);
    alert(`Advertencia: El archivo Excel no tiene todos los campos
esperados. Algunas visualizaciones pueden no funcionar correctamente. Campos
faltantes: ${missingFields.join(', ')}`);
  }

  return true;
}

// Inicializar filtros
function initializeFilters() {
  console.log("Inicializando filtros con datos reales");

  // Filtros por año (desde y hasta)
  const yearFilterFrom = document.getElementById('yearFilterFrom');
  const yearFilterTo = document.getElementById('yearFilterTo');

  // Limpiar opciones existentes
  yearFilterFrom.innerHTML = '<option value="all">Desde</option>';
  yearFilterTo.innerHTML = '<option value="all">Hasta</option>';

```

```

// Obtener años ordenados
const years = [...new Set(allData.map(item => item.anio))].sort((a, b) =>
a - b);

// Llenar ambos selectores con los años disponibles
years.forEach(year => {
  // Para el filtro "desde"
  const optionFrom = document.createElement('option');
  optionFrom.value = year;
  optionFrom.textContent = year;
  yearFilterFrom.appendChild(optionFrom);

  // Para el filtro "hasta"
  const optionTo = document.createElement('option');
  optionTo.value = year;
  optionTo.textContent = year;
  yearFilterTo.appendChild(optionTo);
});

// Filtro por tipo de institución
const institutionTypeFilter =
document.getElementById('institutionTypeFilter');
institutionTypeFilter.innerHTML = '<option value="all">Todos los
tipos</option>'; // Limpiar opciones existentes
const institutionTypes = [...new Set(allData.map(item =>
item.tipo_institucion))].sort();
institutionTypes.forEach(type => {
  if (type) { // Verificar que no sea undefined o null
    const option = document.createElement('option');
    option.value = type;
    option.textContent = type;
    institutionTypeFilter.appendChild(option);
  }
});

// Filtro por carrera
const careerFilter = document.getElementById('careerFilter');
careerFilter.innerHTML = '<option value="all">Todas las
carreras</option>'; // Limpiar opciones existentes
const careers = [...new Set(allData.map(item => item.carrera))].sort();
careers.forEach(career => {
  if (career) { // Verificar que no sea undefined o null
    const option = document.createElement('option');
    option.value = career;
    option.textContent = career;
    careerFilter.appendChild(option);
  }
}

```

```

    });

    // Filtro por tipo de gestión
    const managementTypeFilter =
document.getElementById('managementTypeFilter');
    managementTypeFilter.innerHTML = '<option value="all">Todos los
tipos</option>'; // Limpiar opciones existentes
    const managementTypes = [...new Set(allData.map(item =>
item.tipo_gestion))].sort();
    managementTypes.forEach(type => {
        if (type) { // Verificar que no sea undefined o null
            const option = document.createElement('option');
            option.value = type;
            option.textContent = type;
            managementTypeFilter.appendChild(option);
        }
    });

    // Evento para aplicar filtros
    document.getElementById('applyFilters').addEventListener('click',
applyFilters);
}

// Aplicar filtros
function applyFilters() {
    const yearFilterFrom = document.getElementById('yearFilterFrom').value;
    const yearFilterTo = document.getElementById('yearFilterTo').value;
    const institutionTypeFilter =
document.getElementById('institutionTypeFilter').value;
    const careerFilter = document.getElementById('careerFilter').value;
    const managementTypeFilter =
document.getElementById('managementTypeFilter').value;
    const genderFilter = document.getElementById('genderFilter').value;

    // Mostrar los filtros seleccionados en la consola para depuración
    console.log("Filtros aplicados:", {
        añoDesde: yearFilterFrom,
        añoHasta: yearFilterTo,
        tipoInstitución: institutionTypeFilter,
        carrera: careerFilter,
        tipoGestión: managementTypeFilter,
        género: genderFilter
    });

    // Aplicar el filtrado de datos
    filteredData = allData.filter(item => {
        // Filtro de año con rango desde-hasta

```

```

        const yearMatch = (yearFilterFrom === 'all' || (item.anio &&
item.anio >= parseInt(yearFilterFrom))) &&
            (yearFilterTo === 'all' || (item.anio && item.anio <=
parseInt(yearFilterTo)));

        return yearMatch &&
            (institutionTypeFilter === 'all' || item.tipo_institucion ===
institutionTypeFilter) &&
            (careerFilter === 'all' || item.carrera === careerFilter) &&
            (managementTypeFilter === 'all' || item.tipo_gestion ===
managementTypeFilter) &&
            (genderFilter === 'all' || item.sexo === genderFilter);
    });

    console.log(`Datos filtrados: ${filteredData.length} registros de
${allData.length} totales`);

    // Actualizar estadísticas y gráficos
    updateStats();
    updateCharts();
}

// Actualizar estadísticas principales
function updateStats() {
    const statsContainer = document.getElementById('statsContainer');
    statsContainer.innerHTML = '';

    // Total de títulos
    const totalTitles = filteredData.length;

    // Total por género
    const genderCounts = {};
    filteredData.forEach(item => {
        genderCounts[item.sexo] = (genderCounts[item.sexo] || 0) + 1;
    });

    // Total por tipo de institución
    const institutionTypeCounts = {};
    filteredData.forEach(item => {
        institutionTypeCounts[item.tipo_institucion] =
(institutionTypeCounts[item.tipo_institucion] || 0) + 1;
    });

    // Total de carreras únicas
    const uniqueCareers = new Set(filteredData.map(item =>
item.carrera)).size;

```



```

    // Crear tarjetas de estadísticas
    createStatCard(statsContainer, 'Total de Títulos', totalTitles);
    createStatCard(statsContainer, 'Mujeres', genderCounts['MUJER'] || 0);
    createStatCard(statsContainer, 'Hombres', genderCounts['HOMBRE'] || 0);
    createStatCard(statsContainer, 'Carreras Únicas', uniqueCareers);
}

// Crear tarjeta de estadística
function createStatCard(container, label, value) {
    const card = document.createElement('div');
    card.className = 'stat-card';

    const valueEl = document.createElement('div');
    valueEl.className = 'stat-value';
    valueEl.textContent = value.toLocaleString();

    const labelEl = document.createElement('div');
    labelEl.className = 'stat-label';
    labelEl.textContent = label;

    card.appendChild(valueEl);
    card.appendChild(labelEl);
    container.appendChild(card);
}

// Crear todos los gráficos
function createCharts() {
    createInstitutionTypeChart();
    createGenderChart();
    createYearTrendChart();
    createGenderYearChart();
    createCareersChart();
    createManagementTypeChart();
    createInstitutionsChart();
    createMonthlyTrendChart();
}

// Actualizar todos los gráficos
function updateCharts() {
    // Destruir gráficos existentes para recrearlos
    Object.values(charts).forEach(chart => chart.destroy());
    charts = {};

    // Recrear todos los gráficos
    createCharts();
}

```

```

// Crear gráfico de tipo de institución
function createInstitutionTypeChart() {
    const ctx =
document.getElementById('institutionTypeChart').getContext('2d');

    // Contar por tipo de institución
    const counts = {};
    filteredData.forEach(item => {
        counts[item.tipo_institucion] = (counts[item.tipo_institucion] || 0)
+ 1;
    });

    const labels = Object.keys(counts);
    const data = Object.values(counts);

    charts.institutionType = new Chart(ctx, {
        type: 'pie',
        data: {
            labels: labels,
            datasets: [{
                data: data,
                backgroundColor: colorPalette2,
                borderWidth: 1
            }]
        },
        options: {
            responsive: true,
            plugins: {
                legend: {
                    position: 'bottom',
                }
            }
        }
    });
}

```

```

// Crear gráfico de género
function createGenderChart() {
    const ctx = document.getElementById('genderChart').getContext('2d');

    // Contar por género
    const counts = {};
    filteredData.forEach(item => {
        counts[item.sexo] = (counts[item.sexo] || 0) + 1;
    });

    const labels = Object.keys(counts);

```

```

const data = Object.values(counts);

charts.gender = new Chart(ctx, {
  type: 'doughnut',
  data: {
    labels: labels,
    datasets: [{
      data: data,
      backgroundColor: ['rgba(255,201,219,255)',
'rgba(136,167,217,255)'],
      borderWidth: 1
    }]
  },
  options: {
    responsive: true,
    plugins: {
      legend: {
        position: 'bottom',
      }
    }
  }
});
}

// Crear gráfico de tendencia por año
function createYearTrendChart() {
  const ctx = document.getElementById('yearTrendChart').getContext('2d');

  // Contar por año
  const counts = {};
  filteredData.forEach(item => {
    if (item.anio) { // Verificar que exista el año
      counts[item.anio] = (counts[item.anio] || 0) + 1;
    }
  });

  // Ordenar por año
  const sortedYears = Object.keys(counts).sort((a, b) => a - b);
  const data = sortedYears.map(year => counts[year]);

  charts.yearTrend = new Chart(ctx, {
    type: 'line',
    data: {
      labels: sortedYears,
      datasets: [{
        label: 'Número de Títulos',
        data: data,

```

```

        borderColor: '#c45a10',
        backgroundColor: 'rgba(196,90,16,0.1)',
        fill: true,
        tension: 0.1,
        borderWidth: 2,
        pointRadius: 4,
        pointBackgroundColor: '#c45a10'
    }]
},
options: {
    responsive: true,
    plugins: {
        legend: {
            display: false
        }
    },
    scales: {
        y: {
            beginAtZero: true,
            title: {
                display: true,
                text: 'Número de Títulos'
            }
        },
        x: {
            title: {
                display: true,
                text: 'Año'
            }
        }
    }
}
});
}

// Crear gráfico de género por año
function createGenderYearChart() {
    const ctx = document.getElementById('genderYearChart').getContext('2d');

    // Contar por año y género
    const countsByYearAndGender = {};
    filteredData.forEach(item => {
        if (!item.anio || !item.sexo) return; // Omitir registros sin año o
        género

        const year = item.anio;
        const gender = item.sexo;

```

```

    if (!countsByYearAndGender[year]) {
        countsByYearAndGender[year] = {};
    }

    countsByYearAndGender[year][gender] =
    (countsByYearAndGender[year][gender] || 0) + 1;
    });

    // Ordenar por año
    const sortedYears = Object.keys(countsByYearAndGender).sort((a, b) => a -
b);

    // Preparar datos para mujer y hombre
    const womenData = sortedYears.map(year =>
countsByYearAndGender[year]['MUJER'] || 0);
    const menData = sortedYears.map(year =>
countsByYearAndGender[year]['HOMBRE'] || 0);

    charts.genderYear = new Chart(ctx, {
        type: 'bar',
        data: {
            labels: sortedYears,
            datasets: [
                {
                    label: 'Mujeres',
                    data: womenData,
                    backgroundColor: 'rgba(255,201,219,255)',
                    borderColor: '#ff7043',
                    borderWidth: 1
                },
                {
                    label: 'Hombres',
                    data: menData,
                    backgroundColor: 'rgba(136,167,217,255)',
                    borderColor: '#26a69a',
                    borderWidth: 1
                }
            ]
        },
        options: {
            responsive: true,
            plugins: {
                legend: {
                    position: 'top',
                }
            }
        },
    },

```

```

        scales: {
            y: {
                beginAtZero: true,
                title: {
                    display: true,
                    text: 'Número de Títulos'
                }
            },
            x: {
                title: {
                    display: true,
                    text: 'Año'
                }
            }
        }
    });
}

```

```

function createCareersChart() {
    const ctx = document.getElementById('careersChart').getContext('2d');

    // Contar por carrera
    const counts = {};
    filteredData.forEach(item => {
        if (item.carrera) {
            counts[item.carrera] = (counts[item.carrera] || 0) + 1;
        }
    });

    // Ordenar y obtener top 10
    const sortedCareers = Object.entries(counts)
        .sort((a, b) => b[1] - a[1])
        .slice(0, 10);

    const labels = sortedCareers.map(item => item[0]);
    const data = sortedCareers.map(item => item[1]);

    // Acortar nombres largos de carreras para visualización
    const shortLabels = labels.map(label => {
        if (label.length > 25) {
            return label.substring(0, 22) + '...';
        }
        return label;
    });

    // Destruir el gráfico si ya existe

```

```

if (charts.careers) {
    charts.careers.destroy();
}

// Crear el gráfico horizontal de barras
charts.careers = new Chart(ctx, {
    type: 'bar',
    data: {
        labels: shortLabels,
        datasets: [{
            label: 'Número de Títulos',
            data: data,
            backgroundColor: '#c45a10',
            borderColor: '#c45a10',
            borderWidth: 1
        }]
    },
    options: {
        indexAxis: 'y',
        responsive: true,
        maintainAspectRatio: false,
        layout: {
            padding: {
                left: 10,
                right: 30,
                top: 0,
                bottom: 0
            }
        },
        plugins: {
            legend: {
                display: false
            },
            tooltip: {
                callbacks: {
                    title: function(tooltipItems) {
                        const index =
tooltipItems[0].dataIndex;
                        return labels[index];
                    }
                }
            }
        },
        scales: {
            x: {
                beginAtZero: true,
                title: {

```

```

        display: true,
        text: 'Número de Títulos',
        font: {
            size: 11
        }
    },
    grid: {
        display: true,
        color: "rgba(0, 0, 0, 0.05)"
    }
},
y: {
    ticks: {
        font: {
            size: 10
        },
        color: '#333'
    },
    grid: {
        display: false
    }
}
}
}
});
}

```

```

// Crear gráfico de tipo de gestión
function createManagementTypeChart() {
    const ctx =
document.getElementById('managementTypeChart').getContext('2d');

    // Contar por tipo de gestión
    const counts = {};
    filteredData.forEach(item => {
        if (item.tipo_gestion) {
            counts[item.tipo_gestion] = (counts[item.tipo_gestion] || 0) + 1;
        }
    });

    const labels = Object.keys(counts);
    const data = Object.values(counts);

    charts.managementType = new Chart(ctx, {
        type: 'pie',
        data: {
            labels: labels,

```



```

        datasets: [{
            data: data,
            backgroundColor: colorPalette,
            borderWidth: 1
        }]
    },
    options: {
        responsive: true,
        plugins: {
            legend: {
                position: 'bottom',
            }
        }
    }
});
}

// Crear gráfico de top instituciones
function createInstitutionsChart() {
    const ctx =
document.getElementById('institutionsChart').getContext('2d');

    // Contar por institución
    const counts = {};
    filteredData.forEach(item => {
        if (item.institucion) {
            counts[item.institucion] = (counts[item.institucion] || 0) + 1;
        }
    });

    // Ordenar y obtener top 10
    const sortedInstitutions = Object.entries(counts)
        .sort((a, b) => b[1] - a[1])
        .slice(0, 10);

    const labels = sortedInstitutions.map(item => item[0]);
    const data = sortedInstitutions.map(item => item[1]);

    // Acortar nombres largos de instituciones para visualización
    const shortLabels = labels.map(label => {
        if (label.length > 30) {
            return label.substring(0, 27) + '...';
        }
        return label;
    });

    charts.institutions = new Chart(ctx, {

```

```

    type: 'bar',
    data: {
      labels: shortLabels,
      datasets: [{
        label: 'Número de Títulos',
        data: data,
        backgroundColor: 'rgba(196,90,16,255)',
        borderColor: '#ffa726',
        borderWidth: 1
      }]
    },
    options: {
      indexAxis: 'y',
      responsive: true,
      plugins: {
        legend: {
          display: false
        },
        tooltip: {
          callbacks: {
            title: function(tooltipItems) {
              // Mostrar el nombre completo en el tooltip
              const index = tooltipItems[0].dataIndex;
              return labels[index];
            }
          }
        }
      },
      scales: {
        x: {
          beginAtZero: true,
          title: {
            display: true,
            text: 'Número de Títulos'
          }
        }
      }
    }
  });
}

```

```

// Crear gráfico de tendencia mensual
function createMonthlyTrendChart() {
  const ctx =
document.getElementById('monthlyTrendChart').getContext('2d');

```

```

  // Contar por año y mes

```

```

const countsByYearMonth = {};
filteredData.forEach(item => {
  if (!item.anio || !item.mes) return;

  const year = item.anio;
  const month = item.mes;
  const yearMonth = `${year}-${month.toString().padStart(2, '0')}`;

  countsByYearMonth[yearMonth] = (countsByYearMonth[yearMonth] || 0) +
1;
});

// Ordenar por año y mes
const sortedYearMonths = Object.keys(countsByYearMonth).sort();
const data = sortedYearMonths.map(yearMonth =>
countsByYearMonth[yearMonth]);

// Formatear etiquetas para mostrar solo cada 6 meses por legibilidad
const formattedLabels = sortedYearMonths.map((yearMonth, index) => {
  if (index % 6 === 0) {
    return yearMonth;
  }
  return '';
});

charts.monthlyTrend = new Chart(ctx, {
  type: 'line',
  data: {
    labels: formattedLabels,
    datasets: [{
      label: 'Número de Títulos por Mes',
      data: data,
      borderColor: '#c45a10',
      backgroundColor: 'rgba(196,90,16,0.1)',
      fill: true,
      tension: 0.1,
      borderWidth: 1,
      pointRadius: 2,
      pointBackgroundColor: '#c45a10'
    }]
  },
  options: {
    responsive: true,
    plugins: {
      legend: {
        display: false
      },
    },
  },
});

```

```

        tooltip: {
            callbacks: {
                title: function(tooltipItems) {
                    // Mostrar el año-mes en el tooltip
                    const index = tooltipItems[0].dataIndex;
                    const yearMonth = sortedYearMonths[index];
                    const [year, month] = yearMonth.split('-');
                    const monthNames = ['Enero', 'Febrero', 'Marzo',
'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre',
'Noviembre', 'Diciembre'];
                    return `${monthNames[parseInt(month) - 1]}
${year}`;
                }
            }
        },
        scales: {
            y: {
                beginAtZero: true,
                title: {
                    display: true,
                    text: 'Número de Títulos'
                }
            },
            x: {
                title: {
                    display: true,
                    text: 'Año-Mes'
                }
            }
        }
    });
});
// Cargar datos al iniciar
window.addEventListener('DOMContentLoaded', loadData);

```