



**"AÑO DE LA ESPERANZA Y EL
FORTALECIMIENTO DE LA DEMOCRACIA"**

**UNIVERSIDAD NACIONAL SAN LUIS
GONZAGA**



FACULTAD DE CIENCIAS – ESCUELA DE ESTADISTICA

TEMA:



✓ **CURSO:**
ANALISIS Y DISEÑO DE SISTEMA

✓ **DOCENTE:**
LANDEO ALFARO ELMER LEONIDAS

INTEGRANTES DEL GRUPO:

- APAZA QUISPE ABRAHAM
- AQUIJE MENDOZA WILFREDO
- FLORES ESTALLA ALEJANDRO
- TORRES MAGALLANES MARIA
- LOPEZ ASTOYAUARI RONADO

ICA - 2026

ÍNDICE GENERAL

INTRODUCCIÓN	3
OBJETIVOS	5
• Objetivo General	5
• Objetivos Específicos	5
1. JUSTIFICACIÓN, ALCANCES Y LIMITACIONES	6
• 1.1 Justificación	6
• 1.2 Alcances	6
• 1.3 Limitaciones	7
2. CONCEPTOS BÁSICOS	8
• 2.1 SARS-CoV-2 y su impacto en los sistemas de salud	8
• 2.2 Vigilancia epidemiológica como proceso sistemático	8
• 2.3 Datos clínicos y estructuración de variables	9
• 2.4 Indicadores epidemiológicos y análisis estadístico	10
• 2.5 Sistemas de información y digitalización en salud	10
• 2.6 Lenguaje R como herramienta para la gestión y análisis de datos	11
3. APLICACIÓN DEL SISTEMA EN R	12
• 3.1 Limitaciones del sistema	13
• 3.2 Mejoras futuras del sistema	13
4. HERRAMIENTAS Y MÉTODOS	14
• 4.1 Herramientas utilizadas	14
• 4.2 Métodos de desarrollo	15
• 4.3 Enfoque metodológico	15
5. EJEMPLO PRÁCTICO	16
• 5.1 Escenario, Proceso de registro y Análisis	16
6. CONCLUSIONES	17
7. RECOMENDACIONES	17

INTRODUCCION

La pandemia causada por el SARS-CoV-2 representó uno de los mayores desafíos sanitarios, sociales y tecnológicos del siglo XXI. Desde su identificación en 2019 y su posterior declaración como pandemia en marzo de 2020 por la Organización Mundial de la Salud, los sistemas de salud enfrentaron la necesidad urgente de mejorar los mecanismos de vigilancia epidemiológica, monitoreo clínico y gestión de información.

Uno de los principales problemas detectados durante la crisis sanitaria fue la fragmentación y desorganización de los datos clínicos en distintos niveles del sistema sanitario. La falta de registros digitales estructurados dificultó la toma de decisiones oportunas, el análisis estadístico confiable y la planificación de recursos hospitalarios. En muchos contextos, la información se encontraba dispersa en formularios físicos o bases de datos no integradas, lo que generaba retrasos en la consolidación de reportes y limitaciones en el análisis en tiempo real.

Desde un enfoque multiescalar, la pandemia puede analizarse en distintos niveles interconectados:

- Nivel biológico: dinámica del virus dentro del organismo, respuesta inmunológica y evolución clínica.
- Nivel individual: registro clínico de cada paciente, síntomas, antecedentes y evolución.
- Nivel poblacional: propagación comunitaria, tasas de contagio e incidencia.
- Nivel territorial: movilidad, redes de contacto y distribución geográfica de los casos.

En este contexto, el registro estructurado de datos individuales constituye la unidad básica que alimenta modelos epidemiológicos y sistemas de análisis más complejos. Sin datos organizados y correctamente sistematizados, resulta imposible generar indicadores confiables, estimar tasas de transmisión o proyectar escenarios futuros.

La digitalización y sistematización de información se convierte así en un elemento estratégico para fortalecer la gestión sanitaria. Los sistemas digitales permiten:

- Almacenamiento seguro y ordenado de información.
- Acceso rápido a historiales clínicos.
- Procesamiento estadístico automatizado.
- Generación de reportes y visualizaciones para la toma de decisiones.

En este sentido, las herramientas computacionales desempeñan un papel fundamental en la transformación digital del sector salud. El lenguaje R ha sido ampliamente utilizado en investigación científica debido a su capacidad para el análisis estadístico, modelado matemático y visualización de datos. Además, su naturaleza de software libre facilita su implementación en entornos académicos y profesionales.

El desarrollo de una aplicación orientada al registro digital de información clínica básica en R representa una contribución práctica alineada con la necesidad de modernización tecnológica en salud pública. Aunque se trata de un sistema básico, permite comprender la importancia de estructurar datos de manera organizada, estandarizada y reutilizable.

Asimismo, este tipo de aplicaciones puede servir como base para futuras ampliaciones que incluyan análisis automatizados, generación de gráficos epidemiológicos o integración con modelos matemáticos de propagación. De esta manera, se demuestra cómo una herramienta de registro individual puede formar parte de un sistema más amplio de vigilancia epidemiológica digital.

En conclusión, la pandemia evidenció que la gestión eficiente de la información no es un complemento del sistema sanitario, sino un componente esencial para su funcionamiento. La implementación de soluciones digitales, incluso a pequeña escala, constituye un paso importante hacia la construcción de sistemas de salud más resilientes, organizados y preparados ante futuras emergencias sanitarias.

OBJETIVOS

1) Objetivo General

Analizar la utilidad de un sistema de registro digital desarrollado en lenguaje R como herramienta de apoyo para la gestión y análisis de datos en el contexto del SARS-CoV-2, integrándolo dentro de un enfoque sistémico y multiescala.

Objetivos Específicos

- Describir la importancia de la gestión de datos en escenarios de emergencia sanitaria.
- Explicar el rol del registro estructurado de información individual dentro del modelo epidemiológico.
- Analizar las capacidades del lenguaje R para el almacenamiento y procesamiento de datos clínicos.
- Relacionar la aplicación desarrollada con la digitalización de sistemas de vigilancia epidemiológica.

1.1 Justificación

La pandemia evidenció que la disponibilidad de información organizada y accesible es un factor determinante en la toma de decisiones sanitarias. La ausencia de sistemas digitales adecuados puede generar retrasos en diagnósticos, errores administrativos y dificultades en la planificación de recursos.

El presente trabajo se justifica en la necesidad de fortalecer la cultura de digitalización en el ámbito sanitario, promoviendo herramientas accesibles que permitan registrar, organizar y analizar información básica de manera estructurada. El desarrollo de una aplicación en R no solo contribuye al aprendizaje académico en programación y análisis de datos, sino que también demuestra la aplicabilidad de soluciones tecnológicas en contextos reales de gestión sanitaria.

1.2 Alcances

El presente informe se centra en:

- El análisis conceptual del enfoque multiescala aplicado al SARS-CoV-2.
- El desarrollo y descripción de una aplicación básica de registro en lenguaje R.
- La relación entre digitalización de datos y gestión epidemiológica.

No incluye:

- Implementación hospitalaria real del sistema.
- Integración con bases de datos nacionales oficiales.
- Modelado matemático avanzado tipo SIR con simulaciones dinámicas complejas.

1.3 Limitaciones

Entre las principales limitaciones del trabajo se encuentran:

- La aplicación desarrollada corresponde a un sistema básico de registro y no a una plataforma clínica completa.
- No se realiza validación con datos reales de pacientes.
- No se incluye interoperabilidad con sistemas de salud existentes.

Sin embargo, el sistema constituye una base conceptual que podría ampliarse hacia modelos más complejos de análisis epidemiológico.

2) CONCEPTOS BÁSICOS

Para comprender el desarrollo de una aplicación de registro digital en lenguaje R aplicada al contexto sanitario, es necesario analizar los fundamentos teóricos que sustentan la epidemiología, la gestión de datos clínicos y la transformación digital en salud. A continuación, se presentan seis conceptos esenciales que estructuran este trabajo.

2.1 SARS-CoV-2 y su Impacto en los Sistemas de Salud

El SARS-CoV-2 es un virus de ARN responsable de la enfermedad COVID-19. Su rápida propagación a nivel mundial generó una crisis sanitaria sin precedentes, lo que llevó a que en marzo de 2020 la Organización Mundial de la Salud declarara oficialmente la pandemia.

Más allá del impacto clínico, el virus puso en evidencia debilidades estructurales en los sistemas de información sanitaria. Muchos países enfrentaron dificultades para:

- Registrar casos en tiempo real.
- Consolidar datos de diferentes instituciones.
- Analizar tendencias de contagio.

- Proyectar necesidades hospitalarias.

Esto demostró que la gestión de datos es un componente estratégico dentro de la salud pública. La información organizada permite transformar hechos clínicos individuales en conocimiento colectivo.

2.2 Vigilancia Epidemiológica como Proceso Sistemático

La vigilancia epidemiológica es el proceso continuo de recolección, análisis e interpretación de datos relacionados con la salud de una población. Su finalidad es detectar eventos sanitarios, controlar brotes y orientar políticas públicas.

Este proceso incluye cinco etapas fundamentales:

- a) Captura de información clínica individual.
- b) Registro estructurado en bases de datos.
- c) Análisis estadístico de los datos.
- d) Interpretación de resultados.
- e) Toma de decisiones sanitarias.

El registro individual constituye la base del sistema. Sin datos organizados desde el inicio, el análisis posterior pierde precisión. Por ello, el desarrollo de herramientas digitales que faciliten la captura de información es esencial para fortalecer la vigilancia epidemiológica.

2.3 Datos Clínicos y Estructuración de Variables

Los datos clínicos son la unidad básica de información dentro del sistema sanitario. Estos pueden clasificarse en:

- Datos demográficos: edad, sexo, residencia.
- Datos antropométricos: peso, talla.
- Datos médicos: antecedentes, síntomas, diagnósticos.
- Datos farmacológicos: medicamentos y alergias.

Desde el punto de vista estadístico, las variables pueden ser:

- Cuantitativas (edad, peso).
- Cualitativas (presencia o ausencia de síntomas).
- Temporales (fecha de diagnóstico).

La correcta estructuración implica definir el tipo de variable, su unidad de medida y su formato de almacenamiento. Una base de datos organizada permite posteriormente aplicar modelos matemáticos y generar indicadores sanitarios confiables.

2.4 Indicadores Epidemiológicos y Análisis Estadístico

Los indicadores epidemiológicos permiten medir la magnitud y evolución de una enfermedad. Entre los más importantes se encuentran:

- Tasa de incidencia: número de casos nuevos en un período determinado.
- Tasa de prevalencia: total de casos existentes en un momento específico.
- Número reproductivo básico (R_0): capacidad de transmisión de la enfermedad.

Estos indicadores requieren datos completos y actualizados. Desde la perspectiva estadística, el análisis puede incluir:

- Medidas de tendencia central (media, mediana).
- Distribución de frecuencias.
- Análisis comparativos entre grupos.
- Modelos matemáticos de propagación.

Sin una base de datos estructurada, estos cálculos no pueden realizarse de manera confiable.

2.5 Sistemas de Información y Digitalización en Salud

Un sistema de información en salud integra personas, procesos y tecnología con el fin de gestionar información clínica de manera eficiente.

La digitalización permite:

- Reducir errores manuales.
- Mejorar la accesibilidad a la información.
- Automatizar cálculos estadísticos.
- Generar reportes en tiempo real.

Durante la pandemia, la transformación digital se convirtió en un factor clave para mejorar la capacidad de respuesta de los sistemas sanitarios. La organización estructurada de datos individuales fortalece la toma de decisiones basada en evidencia.

2.6 Lenguaje R como Herramienta para la Gestión y Análisis de Datos

R es un lenguaje de programación especializado en análisis estadístico y modelado de datos. Es ampliamente utilizado en investigación científica debido a su capacidad para:

- Manipular grandes volúmenes de información.
- Aplicar modelos matemáticos.
- Generar gráficos epidemiológicos.
- Automatizar procesos analíticos.

En el presente trabajo, R se emplea para desarrollar una aplicación orientada al registro digital de datos clínicos básicos. Aunque se trata de una herramienta académica, su estructura demuestra cómo la tecnología puede apoyar la organización de información sanitaria y servir como base para análisis más avanzados.

3) APPLICACIÓN

La aplicación desarrollada en lenguaje R constituye un sistema estructurado para la captura, validación, almacenamiento y preprocesamiento de datos clínicos básicos. Desde una perspectiva técnica, puede entenderse como un módulo de gestión de datos primarios orientado a la construcción de una base estructurada que permita análisis estadísticos posteriores dentro del mismo entorno computacional.

El diseño del sistema se fundamenta en principios de modelado de datos y normalización básica de variables. Cada atributo clínico es previamente definido según su tipo y escala de medición, lo que permite garantizar coherencia semántica y compatibilidad estadística. En este sentido, las variables cuantitativas discretas (por ejemplo, edad) se diferencian de las variables cuantitativas continuas (peso y talla), así como de las variables cualitativas nominales (sexo, antecedentes médicos, alergias). Esta clasificación no es meramente descriptiva, sino que determina el conjunto de operaciones matemáticas y estadísticas que pueden aplicarse posteriormente.

El proceso de captura de información se complementa con mecanismos de validación que funcionan como filtros lógicos previos al almacenamiento. Técnicamente, estas validaciones operan como condiciones de control que restringen la entrada de valores fuera de rango o inconsistentes. Este procedimiento actúa como una forma básica de control de calidad de datos (data quality control), reduciendo la probabilidad de errores sistemáticos y aumentando la confiabilidad interna de la base generada.

El almacenamiento se realiza mediante estructuras tipo data frame, que en R representan objetos tabulares compuestos por vectores de igual longitud organizados en columnas. Esta estructura facilita la indexación, el filtrado condicional, la transformación de variables y la aplicación directa de funciones estadísticas vectorizadas. La organización tabular permite implementar operaciones como agregación por grupos, cálculo de medidas de tendencia central, dispersión y generación de nuevas variables derivadas sin necesidad de procesos externos.

Desde el punto de vista analítico, la aplicación permite la generación de indicadores secundarios a partir de variables primarias. El cálculo automático del Índice de Masa Corporal (IMC) constituye un ejemplo de transformación funcional de datos, donde se aplica una fórmula matemática que combina dos variables continuas para producir una nueva variable cuantitativa. Este procedimiento representa una etapa de ingeniería básica de características (feature engineering), concepto fundamental en análisis estadístico y modelado de datos.

Asimismo, la estructura generada es compatible con procedimientos de análisis exploratorio de datos (EDA), incluyendo la identificación de valores atípicos, análisis de distribución y evaluación preliminar de relaciones entre variables mediante correlación o visualización gráfica. La posibilidad de exportar los datos en formatos interoperables, como CSV, amplía la capacidad de integración con otros entornos analíticos o sistemas de almacenamiento relacional.

Desde una perspectiva metodológica, la aplicación puede considerarse una herramienta de preprocesamiento de datos clínicos, ya que organiza y prepara la información para etapas posteriores de modelado estadístico, inferencia o análisis multivariado.

3.1 Limitaciones del Sistema

A pesar de su funcionalidad estructural, la aplicación presenta diversas limitaciones que deben ser consideradas desde un enfoque técnico y metodológico.

En primer lugar, el sistema opera sobre almacenamiento local en memoria mediante data frames, lo que implica restricciones en términos de escalabilidad y volumen de datos. Para conjuntos extensos de registros, esta estructura podría resultar insuficiente en comparación con sistemas de gestión de bases de datos relacionales o soluciones basadas en almacenamiento distribuido.

En segundo lugar, los mecanismos de validación implementados son de tipo básico y no incluyen procedimientos avanzados de verificación cruzada, detección automática de inconsistencias complejas o auditorías de integridad referencial. Esto significa que, aunque se reducen errores simples, no se garantiza un control exhaustivo de calidad en escenarios más complejos.

Otra limitación importante es la ausencia de un sistema robusto de gestión de usuarios y niveles de acceso. Desde el punto de vista de seguridad informática y protección de datos, la aplicación no incorpora mecanismos de autenticación, encriptación ni trazabilidad de modificaciones. En entornos clínicos reales, estas características serían indispensables para cumplir con estándares éticos y normativos.

Asimismo, el sistema no incluye herramientas de respaldo automático ni control de versiones, lo que podría representar un riesgo en caso de pérdida de información o modificaciones accidentales. Tampoco contempla integración directa con historias clínicas electrónicas ni interoperabilidad mediante estándares de intercambio de datos en salud.

Finalmente, desde el punto de vista analítico, la aplicación se centra principalmente en análisis descriptivos básicos y generación de indicadores simples. No incorpora módulos de análisis inferencial automatizado, modelado predictivo ni aprendizaje automático, lo que limita su alcance a etapas preliminares del proceso investigativo.

3.2 Mejoras Futuras del Sistema

Si bien la aplicación cumple adecuadamente su función como herramienta de registro y preprocesamiento de datos clínicos, su estructura permite proyectar diversas mejoras orientadas a optimizar su rendimiento, escalabilidad y aplicabilidad en contextos más amplios.

En primer lugar, una mejora relevante consiste en la migración del almacenamiento desde estructuras tipo *data frame* en memoria hacia un sistema de gestión de bases de datos relacional (por ejemplo, mediante integración con SQL). Esta transición permitiría manejar volúmenes mayores de información, mejorar la persistencia de los datos y optimizar las consultas estructuradas. Además, facilitaría la implementación de integridad referencial y

normalización avanzada de tablas, reduciendo redundancias y fortaleciendo la coherencia estructural del sistema.

Otra mejora importante sería la incorporación de una interfaz gráfica interactiva más robusta, por ejemplo mediante entornos como Shiny en R. Esto permitiría una experiencia de usuario más intuitiva, validaciones dinámicas en tiempo real y visualización inmediata de resultados estadísticos. Una interfaz de este tipo facilitaría la adopción del sistema por usuarios sin conocimientos avanzados de programación.

Desde el punto de vista de la seguridad y confidencialidad, el sistema podría fortalecerse mediante la implementación de mecanismos de autenticación de usuarios, control de accesos por roles y encriptación básica de datos sensibles. La incorporación de protocolos de anonimización o seudonimización contribuiría a alinear la herramienta con principios éticos y normativas de protección de datos en salud.

En el ámbito analítico, el sistema podría ampliarse mediante la inclusión de módulos de análisis inferencial automatizado. Esto implicaría la incorporación de pruebas estadísticas básicas (como comparación de medias o análisis de asociación), generación automática de reportes y visualizaciones avanzadas. Asimismo, podría integrarse con técnicas de modelado predictivo o aprendizaje automático para identificar patrones clínicos más complejos dentro de la base de datos.

Adicionalmente, se podría desarrollar un sistema de respaldo automático y control de versiones que permita registrar modificaciones realizadas sobre los datos, mejorando la trazabilidad y reduciendo el riesgo de pérdida de información. La implementación de registros de auditoría fortalecería la transparencia y el control sobre el manejo de los datos.

Finalmente, una línea de mejora estratégica sería la interoperabilidad con otros sistemas de información en salud. La posibilidad de exportar e importar datos mediante formatos estandarizados permitiría integrar la aplicación en ecosistemas digitales más amplios, ampliando su utilidad más allá del entorno académico.

En conjunto, estas mejoras proyectan la evolución del sistema desde una herramienta académica de registro hacia una plataforma más robusta de gestión y análisis de información clínica, incrementando su valor metodológico, tecnológico y formativo.

4) HERRAMIENTAS Y MÉTODOS

El desarrollo de la aplicación se fundamentó en el uso de herramientas de programación estadística y en la aplicación de métodos estructurados para la gestión y procesamiento de datos clínicos. La selección de recursos tecnológicos y metodológicos respondió a criterios de funcionalidad, accesibilidad y compatibilidad con análisis estadístico.

4.1 Herramientas Utilizadas

La herramienta principal empleada fue el lenguaje de programación R, seleccionado por su amplia capacidad para el manejo de datos, análisis estadístico y generación de visualizaciones. R proporciona un entorno flexible que permite trabajar con estructuras tabulares, aplicar funciones matemáticas vectorizadas y desarrollar procesos automatizados para la transformación de datos.

Para la edición y ejecución del código se utilizó un entorno de desarrollo integrado (IDE), que facilita la organización del script, la depuración de errores y la ejecución modular del programa.

Este entorno permite estructurar el proyecto en secciones lógicas, mejorando la trazabilidad y el mantenimiento del código.

En cuanto a la gestión de datos, se emplearon estructuras tipo *data frame*, que permiten organizar la información en formato tabular. Estas estructuras son especialmente adecuadas para bases de datos clínicas básicas, ya que garantizan correspondencia directa entre registros (filas) y variables (columnas). Además, su compatibilidad con funciones estadísticas integradas facilita la aplicación de procedimientos descriptivos y transformaciones matemáticas.

El sistema también hace uso de funciones internas del lenguaje para validación lógica, manipulación de vectores y generación de nuevas variables derivadas, como el cálculo del Índice de Masa Corporal (IMC).

4.2 Métodos de Desarrollo

El desarrollo del sistema siguió un enfoque estructurado basado en etapas secuenciales:

En primer lugar, se realizó la definición conceptual de las variables clínicas a registrar. Esta etapa implicó la identificación de los datos relevantes y su clasificación según escala de medición (cuantitativa discreta, cuantitativa continua o cualitativa). La correcta definición de variables es fundamental para garantizar coherencia estadística en fases posteriores.

En segundo lugar, se diseñó la estructura lógica del programa, estableciendo el flujo de entrada de datos, validación y almacenamiento. Este proceso incluyó la implementación de controles condicionales para asegurar la consistencia de la información ingresada. La validación previa al almacenamiento constituye un método básico de control de calidad de datos.

Posteriormente, se implementó el almacenamiento estructurado mediante la construcción progresiva del data frame, agregando registros conforme se ingresaban nuevos pacientes. Esta metodología permite mantener una base dinámica y actualizable.

Finalmente, se incorporaron métodos de análisis descriptivo, aplicando medidas de tendencia central y cálculos derivados. El cálculo del IMC representa una transformación matemática aplicada a variables continuas, constituyendo un ejemplo de ingeniería básica de variables dentro del proceso analítico.

4.3 Enfoque Metodológico

Desde una perspectiva metodológica, el proyecto puede clasificarse como un desarrollo aplicado con orientación cuantitativa. El sistema no genera datos primarios de manera autónoma, sino que organiza información introducida manualmente para su posterior análisis estadístico.

El enfoque adoptado prioriza la estructuración y preparación de datos (data preprocessing), etapa fundamental dentro del ciclo de análisis estadístico. La correcta organización de la información garantiza que los resultados obtenidos en fases posteriores sean coherentes y reproducibles.

Asimismo, el método empleado permite la replicabilidad del sistema, ya que el código puede ejecutarse en distintos entornos con resultados consistentes, siempre que se mantenga la misma estructura de variables y validaciones.

5) EJEMPLO PRÁCTICO

Con el fin de ilustrar el funcionamiento de la aplicación en un contexto epidemiológico real, se plantea un escenario simulado relacionado con el registro sistemático de pacientes durante un brote de infección por SARS-CoV-2. Este ejercicio permite demostrar cómo la digitalización estructurada de datos clínicos básicos puede convertirse en un insumo fundamental para el análisis epidemiológico, la identificación de factores de riesgo y la generación de información útil para la toma de decisiones en salud pública.

La pandemia por SARS-CoV-2 evidenció que la calidad del análisis epidemiológico depende directamente de la calidad y organización de los datos primarios. En este sentido, el ejemplo práctico no solo busca mostrar el funcionamiento técnico de la aplicación, sino también contextualizar su utilidad dentro de un escenario sanitario complejo.

5.1 Escenario

Se simula el registro sistemático de pacientes que acuden a un centro de atención primaria con sospecha o confirmación diagnóstica de infección por SARS-CoV-2. En este escenario, la aplicación en lenguaje R actúa como un módulo de captura de información clínica inicial, permitiendo consolidar datos relevantes de manera organizada y estructurada.

Para cada paciente se registran variables demográficas y clínicas tales como:

- Edad
- Sexo
- Peso
- Talla
- Antecedentes médicos
- Medicamentos en uso
- Alergias

La elección de estas variables responde a criterios epidemiológicos bien establecidos. Diversos estudios han demostrado que la edad avanzada constituye uno de los principales factores asociados a mayor gravedad en COVID-19. Asimismo, la presencia de enfermedades crónicas como hipertensión arterial, diabetes mellitus, obesidad y enfermedades cardiovasculares se ha relacionado con mayor probabilidad de hospitalización y complicaciones respiratorias.

Desde el punto de vista metodológico, la aplicación no solo almacena información, sino que transforma datos individuales en variables analizables. Por ejemplo, a partir del peso y la talla se calcula automáticamente el Índice de Masa Corporal (IMC), el cual permite clasificar el estado nutricional del paciente en categorías como normopeso, sobrepeso u obesidad. Este indicador resulta relevante, ya que la obesidad ha sido identificada como un factor asociado a mayor riesgo de ventilación mecánica y complicaciones en pacientes con SARS-CoV-2.

5.2 Proceso de Registro y Validación

Durante el ingreso de información, la aplicación ejecuta controles básicos de validación que permiten mejorar la calidad del registro. Entre estos mecanismos se incluyen:

- Verificación de que la edad sea un valor numérico positivo.
- Control de rangos razonables para peso y talla.
- Prevención de campos vacíos en variables obligatorias.

Estos procedimientos reducen errores comunes en la captura de datos, lo cual es fundamental, ya que errores en el registro inicial pueden distorsionar análisis estadísticos posteriores. En el contexto de una pandemia, donde la rapidez y precisión del análisis son esenciales, la estandarización de datos se convierte en un elemento estratégico.

Una vez validados, los registros se almacenan en una estructura tipo *data frame*, organizada en filas (pacientes) y columnas (variables clínicas). Esta estructura permite aplicar directamente funciones estadísticas del entorno R sin necesidad de transformaciones adicionales complejas.

5.3 Análisis Descriptivo Inicial

Con la base de datos generada, se pueden realizar análisis descriptivos que constituyen el primer nivel de interpretación epidemiológica. Entre los análisis posibles se encuentran:

- Cálculo del promedio y mediana de edad.
- Distribución porcentual por sexo.
- Frecuencia de comorbilidades registradas.
- Clasificación nutricional según IMC.
- Identificación de proporción de pacientes con antecedentes médicos.

Estos análisis permiten observar patrones preliminares dentro del grupo registrado. Por ejemplo, podría identificarse que un alto porcentaje de pacientes con sospecha de SARS-CoV-2 pertenece a grupos etarios mayores o presenta condiciones crónicas asociadas. También podría evidenciarse una elevada prevalencia de sobrepeso u obesidad en el grupo atendido.

Aunque el ejemplo es académico y simulado, reproduce la lógica de los sistemas reales de vigilancia epidemiológica, donde la información individual se consolida para generar indicadores poblacionales.

5.4 Relación con el Enfoque Multiescalas

Desde una perspectiva multiescalar, este ejemplo representa el nivel individual del sistema epidemiológico. Cada registro corresponde a una unidad básica de información que, al agregarse con otros registros, permite construir análisis poblacionales. A su vez, estos análisis pueden alimentar modelos matemáticos de propagación o sistemas de monitoreo territorial.

En el contexto del SARS-CoV-2, la recopilación estructurada de datos clínicos permite:

- Caracterizar perfiles de riesgo.

- Identificar grupos vulnerables.
- Evaluar tendencias preliminares.
- Generar hipótesis para estudios más complejos.

De esta manera, la aplicación desarrollada en R no debe entenderse únicamente como un sistema de registro, sino como un componente inicial dentro de una cadena de procesamiento de información epidemiológica.

5.5 Interpretación Epidemiológica

La información organizada mediante la aplicación permite establecer relaciones exploratorias entre variables. Por ejemplo, podría analizarse si existe mayor proporción de comorbilidades en pacientes de mayor edad o si el sobrepeso es más frecuente en determinados grupos etarios.

Aunque estos análisis no establecen causalidad, sí permiten identificar asociaciones preliminares que podrían orientar estudios más profundos. En escenarios de crisis sanitaria, este tipo de análisis descriptivo constituye una herramienta valiosa para priorizar recursos y diseñar intervenciones preventivas.

5.6 Valor Académico y Aplicado

Este ejemplo práctico demuestra que incluso una herramienta académica sencilla puede reflejar principios fundamentales de la epidemiología aplicada. La calidad del análisis depende directamente de la calidad del registro inicial. Por lo tanto, la digitalización estructurada de datos clínicos representa el primer paso para fortalecer la gestión sanitaria basada en evidencia.

En el contexto de la pandemia por SARS-CoV-2, los sistemas de información jugaron un papel determinante en el monitoreo de casos, hospitalizaciones y factores de riesgo. La aplicación desarrollada en R reproduce a pequeña escala la lógica de estos sistemas, evidenciando cómo la informática aplicada puede contribuir al análisis y comprensión de fenómenos epidemiológicos complejos.

6) CONCLUSIONES

- I. El desarrollo de la aplicación permitió demostrar, desde una perspectiva técnica y metodológica, la relevancia de la estructuración sistemática de datos clínicos en contextos epidemiológicos complejos como la pandemia por SARS-CoV-2. A lo largo del proyecto se evidenció que la organización adecuada de la información no constituye únicamente un aspecto operativo, sino un elemento central dentro del proceso de análisis sanitario y generación de conocimiento.
- II. La correcta definición y clasificación de variables —como edad, sexo, comorbilidades y estado nutricional— facilita la construcción de indicadores que permiten caracterizar poblaciones y detectar patrones preliminares asociados a la evolución clínica de los pacientes. En el contexto del SARS-CoV-2, donde factores como la edad avanzada, la obesidad y la presencia de enfermedades crónicas demostraron tener impacto en la gravedad de la enfermedad, la disponibilidad de datos estructurados resulta fundamental para el análisis epidemiológico.

- III. Desde el punto de vista técnico, la utilización del lenguaje R como entorno de desarrollo proporcionó una plataforma flexible y eficiente para el registro, procesamiento y análisis descriptivo de la información. El uso de estructuras tipo *data frame* permitió organizar los datos en formato tabular, facilitando la aplicación directa de funciones estadísticas y transformaciones matemáticas. Este enfoque evidencia la importancia del preprocesamiento de datos como etapa esencial dentro del ciclo de análisis cuantitativo.
- IV. La generación de indicadores derivados, como el Índice de Masa Corporal (IMC), demostró cómo variables primarias pueden transformarse en medidas con mayor valor interpretativo. Este proceso de transformación constituye una forma básica de ingeniería de variables, permitiendo ampliar el potencial analítico del sistema y explorar asociaciones preliminares entre características clínicas. En escenarios epidemiológicos, este tipo de indicadores puede contribuir a la identificación de perfiles de riesgo y a la priorización de intervenciones.
- V. Asimismo, el proyecto pone de manifiesto que incluso herramientas académicas de alcance limitado pueden aportar al entendimiento de fenómenos epidemiológicos cuando se fundamentan en principios sólidos de estructuración de datos. La pandemia por SARS-CoV-2 evidenció la necesidad de contar con sistemas capaces de recopilar información confiable en tiempo real. En este contexto, la digitalización estructurada se convierte en un componente estratégico para la vigilancia epidemiológica, el monitoreo de tendencias y la planificación sanitaria.
- VI. Otro aspecto relevante es la relación directa entre calidad del registro y calidad del análisis. La implementación de mecanismos básicos de validación permitió reducir errores y fortalecer la consistencia interna de la base de datos. Esto reafirma que el análisis estadístico no puede considerarse independiente del proceso de captura de información; por el contrario, ambos constituyen fases interdependientes dentro de un mismo sistema.
- VII. En términos formativos, el desarrollo de la aplicación permitió integrar conocimientos de programación, estadística y gestión de información sanitaria, evidenciando el carácter interdisciplinario de la informática aplicada a la salud. El proyecto demuestra que la capacidad de transformar datos clínicos en información estructurada es una competencia esencial en contextos de emergencia sanitaria y en sistemas de salud modernos.
- VIII. Finalmente, se concluye que la digitalización y estandarización de datos clínicos no solo optimizan el análisis descriptivo, sino que constituyen la base sobre la cual pueden desarrollarse modelos predictivos, análisis inferenciales y sistemas de vigilancia más complejos. En situaciones como la pandemia por SARS-CoV-2, donde la rapidez y precisión en la interpretación de datos resultan determinantes, contar con herramientas estructuradas de registro representa un avance significativo hacia una gestión sanitaria basada en evidencia.

7) RECOMENDACIONES

- I. A partir del análisis realizado y considerando el contexto epidemiológico asociado a la pandemia por SARS-CoV-2, se plantean una serie de recomendaciones orientadas a fortalecer tanto el sistema desarrollado como su potencial aplicación en escenarios reales de gestión sanitaria.

- II. En primer lugar, se recomienda ampliar el conjunto de variables registradas, incorporando información específica relacionada con enfermedades infecciosas respiratorias. En el caso del SARS-CoV-2, variables como sintomatología presentada, fecha de inicio de síntomas, estado de vacunación, resultado de pruebas diagnósticas, saturación de oxígeno y evolución clínica permitirían enriquecer significativamente el análisis. La inclusión de estas variables facilitaría la construcción de perfiles de riesgo más precisos y permitiría realizar estudios comparativos entre grupos poblacionales.
- III. En segundo lugar, sería pertinente fortalecer los mecanismos de validación y control de calidad de los datos. La implementación de validaciones más avanzadas, como controles cruzados entre variables o detección automática de valores atípicos, contribuiría a mejorar la confiabilidad del sistema. En contextos epidemiológicos, donde las decisiones pueden depender de tendencias estadísticas, garantizar la precisión de la información resulta fundamental.
- IV. Otra recomendación relevante consiste en migrar el almacenamiento de datos hacia un sistema de gestión de bases de datos más robusto. Aunque las estructuras tipo *data frame* son adecuadas para proyectos académicos o bases de datos pequeñas, escenarios reales de vigilancia epidemiológica requieren sistemas escalables capaces de manejar grandes volúmenes de información. La integración con bases de datos relacionales permitiría mejorar la persistencia, seguridad y eficiencia en las consultas.
- V. Asimismo, se sugiere desarrollar una interfaz gráfica interactiva que facilite el ingreso y visualización de datos por parte de personal de salud. Una interfaz intuitiva reduciría la posibilidad de errores humanos y ampliaría la aplicabilidad del sistema en entornos clínicos. Además, la incorporación de paneles de visualización dinámica permitiría observar tendencias en tiempo real, lo cual es especialmente útil en situaciones de brotes o emergencias sanitarias.
- VI. Desde el punto de vista analítico, se recomienda integrar módulos de análisis inferencial y modelado estadístico. La aplicación podría evolucionar hacia la implementación de pruebas de asociación, análisis multivariado o modelos predictivos que permitan estimar probabilidades de complicaciones en pacientes con SARS-CoV-2 u otras enfermedades infecciosas. Esto transformaría la herramienta de un sistema descriptivo a una plataforma con mayor capacidad interpretativa.
- VII. En términos de seguridad y ética, resulta indispensable incorporar mecanismos de protección de datos, tales como autenticación de usuarios, control de accesos por niveles y protocolos de anonimización. En contextos de salud pública, el manejo de información sensible debe cumplir estándares de confidencialidad y protección de datos personales.

- VIII. Finalmente, se recomienda considerar la interoperabilidad con otros sistemas de información en salud. La posibilidad de exportar e importar datos mediante formatos estandarizados permitiría integrar la aplicación en redes de vigilancia epidemiológica más amplias. En escenarios como la pandemia por SARS-CoV-2, la capacidad de compartir información estructurada entre instituciones es un factor determinante para la coordinación y planificación sanitaria.
- IX. En conjunto, estas recomendaciones proyectan la evolución del sistema hacia una herramienta más robusta, segura y analíticamente avanzada, capaz de contribuir de manera significativa a la gestión de información clínica y epidemiológica en contextos actuales y futuros.

“Aplicativo Web para el Registro y Análisis de Comorbilidades en Estudiantes”

Paso 1

```
library(shiny)
library(shinydashboard)
library(DT)
library(ggplot2)
```

Esta parte carga las librerías necesarias:

- **shiny** → crea aplicaciones web interactivas en R
- **shinydashboard** → permite diseño tipo panel profesional
- **DT** → muestra tablas interactivas
- **ggplot2** → genera gráficos estadísticos

paso 2

```
# -----
# UI - Interfaz de Usuario
# -----
ui <- dashboardPage(
  dashboardHeader(title = "Comorbilidad UNICA"),
  dashboardSidebar(
    sidebarMenu(
      menuItem("Dashboard & Hipótesis", tabName = "dashboard", icon = icon("chart-line")),
      menuItem("Aregar Estudiante", tabName = "agregar", icon = icon("user-plus")),
      menuItem("Consultar Estudiante", tabName = "consultar", icon = icon("search")),
      menuItem("Actualizar/Eliminar", tabName = "actualizar", icon = icon("edit")),
      menuItem("Mostrar Tabla", tabName = "mostrar", icon = icon("table"))
    )
  ),
  dashboardBody(
    tabItems(
```

Aquí se construye la interfaz visual:

- **dashboardHeader** → título del sistema
- **dashboardSidebar** → menú lateral
- **menuItem** → secciones de la app
- **dashboardBody** → contenido principal

paso 3

```
# Dashboard con Contraste de Hipótesis
tabItem(tabName = "dashboard",
  fluidRow(
    box(width = 12, title = "Contraste de Hipótesis Estadístico", status = "info", solidHeader = TRUE,
        column(width = 6,
               tags$h4(tags$b("Hipótesis Nula (H0):")),
               tags$p("No existe correlación entre el IMC y la Morbilidad. ( $\rho = 0$ )"),
               tags$hr(),
               tags$h4(tags$b("Hipótesis Alternativa (H1):")),
               tags$p("Existe una correlación positiva entre el IMC y la Morbilidad. ( $\rho > 0$ )")
        ),
        column(width = 6,
               tags$h4(tags$b("Interpretación Visual:")),
               tags$p("Si la linea de tendencia roja es plana, mantenemos H0."),
               tags$p("Si la linea tiene una pendiente ascendente clara, aportamos evidencia a favor de H1.")
        )
    ),
    fluidRow(
      valueBoxOutput("totalEstudiante"),
      valueBoxOutput("promEdad"),
      valueBoxOutput("morbilidadAlta")
    ),
    fluidRow(
      box(width = 8, title = "Análisis de Regresión: Contraste H0 vs H1", status = "success", solidHeader = TRUE,
          plotOutput("grafHipotesis")),
      box(width = 4, title = "Estado de Morbilidad", status = "warning", solidHeader = TRUE,
          plotOutput("grafMorbilidad"))
    ),
    fluidRow(
      box(width = 6, title = "Frecuencia por Enfermedad", status = "primary", solidHeader = TRUE,
          plotOutput("grafEnfermedad")),
      box(width = 6, status = "danger",
          actionButton("cerrar_app", "Cerrar Aplicación",
                      style = "color: white; background-color: red; width: 100%;"))
    )
  )
),
```

Dashboard con Contraste de Hipótesis

Este módulo corresponde al panel principal del aplicativo, donde se realiza el análisis estadístico y la visualización de los datos relacionados con el índice de masa corporal (IMC) y la morbilidad de los estudiantes.

Paso 4

```
# Agregar Estudiante
tabItem(tabName = "agregar",
  box(width = 6, status = "primary", solidHeader = TRUE, title = "Nuevo Registro",
      numericInput("id", "ID", value = 1, min = 1),
      textInput("nombre", "Nombre"),
      selectInput("sexo", "Sexo", choices = c("Masculino", "Femenino", "Otro")),
      numericInput("edad", "Edad", value = 18, min = 0),
      dateInput("fecha", "Fecha de nacimiento", value = Sys.Date()),
      numericInput("peso", "Peso (kg)", value = 0, min = 0),
      numericInput("talla", "Talla (m)", value = 0, step = 0.01, min = 0),
      selectInput("sangre", "Tipo de sangre", choices = c("A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-")),
      numericInput("morbilidad", "Morbilidad (%)", value = 0, min = 0, max = 100),
      textInput("enfermedad", "Enfermedad"),
      textInput("medicamentos", "Medicamentos"),
      textInput("alergias", "Alergias"),
      textInput("contacto", "Contacto de emergencia"),
      actionButton("add", "Agregar Paciente", style = "background-color: green; color: white;"))
  )
),
```

El módulo Agregar Estudiante permite registrar información personal y médica mediante un formulario interactivo. Los datos ingresados se almacenan en la base del sistema y se utilizan para el análisis estadístico y la visualización del estado de salud de los estudiantes.

Paso 5

```
# Consultar
tabItem(tabName = "consultar",
        box(width = 6, status = "info", solidHeader = TRUE, title = "Búsqueda por ID",
            numericInput("id_consulta", "ID del Estudiante", value = 1),
            actionButton("buscar", "Consultar"),
            br(), br(),
            verbatimTextOutput("resultado_consulta")
        )
    ),

# Actualizar/Eliminar
tabItem(tabName = "actualizar",
        box(width = 6, status = "warning", solidHeader = TRUE, title = "Modificar Registros",
            numericInput("id_upd", "ID del Estudiante a Modificar", value = 1),
            actionButton("buscar_upd", "Cargar Datos"),
            hr(),
           textInput("nombre_upd", "Nombre"),
            selectInput("sexo_upd", "Sexo", choices = c("Masculino", "Femenino", "Otro")),
            numericInput("edad_upd", "Edad", value = 0),
            dateInput("fecha_upd", "Fecha nacimiento"),
            numericInput("peso_upd", "Peso (kg)", value = 0),
            numericInput("talla_upd", "Talla (m)", value = 0, step = 0.01),
            selectInput("sangre_upd", "Tipo de sangre", choices = c("A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-")),
            numericInput("morbilidad_upd", "Morbilidad (%)", value = 0),
            textInput("enfermedad_upd", "Enfermedad"),
            textInput("medicamentos_upd", "Medicamentos"),
            textInput("alergias_upd", "Alergias"),
            textInput("contacto_upd", "Contacto de emergencia"),
            actionButton("actualizar", "Actualizar Registro", style = "background-color: orange; color: white;"),
            actionButton("eliminar", "Eliminar Estudiante", style = "background-color: red; color: white;")
        )
    ),

# Mostrar Tabla
tabItem(tabName = "mostrar",
        box(width = 12, status = "success", solidHeader = TRUE, title = "Base de Datos",
            DTOutput("tabla_Estudiante"),
            downloadButton("export_csv", "Exportar a CSV")
        )
    )
)
```

El aplicativo cuenta con módulos que permiten consultar, actualizar, eliminar y visualizar los registros de estudiantes, garantizando una gestión organizada y confiable de la información de salud.

El módulo **Consultar Estudiante** permite buscar datos mediante el ID, mostrando rápidamente la información personal y médica registrada. El módulo **Actualizar/Eliminar** facilita la modificación o eliminación de registros, asegurando que la base de datos se mantenga actualizada y libre de errores. Por su parte, el módulo **Mostrar Tabla** presenta todos los registros en una tabla interactiva con opciones de búsqueda y ordenamiento, además de permitir la exportación a formato CSV para su uso en Excel u otras herramientas.

En conjunto, estos módulos permiten administrar la información de manera eficiente, apoyar el análisis del estado de salud estudiantil y facilitar la toma de decisiones basadas en datos.

Paso 6

```
# -----
# SERVER - Lógica
# -----
server <- function(input, output, session) {

  file_path <- "datos_estudiantes.csv"

  # Carga inicial
  initial_data <- if (file.exists(file_path)) {
    read.csv(file_path, stringsAsFactors = FALSE)
  } else {
    data.frame(ID=integer(), Nombre=character(), Sexo=numeric(),
               FechaNacimiento=as.character(), Peso=numeric(), Talla=numeric(), IMC=numeric(),
               Sangre=character(), Morbilidad=numeric(), Enfermedad=character(),
               Medicamentos=character(), Alergias=character(), Contacto=character(),
               stringsAsFactors=FALSE)
  }

  Estudiante <- reactiveVal(initial_data)

  guardar_datos <- function(df) {
    write.csv(df, file_path, row.names = FALSE)
    Estudiante(df)
  }
}
```

Esta sección corresponde a la lógica del servidor en la aplicación Shiny, encargada de gestionar el almacenamiento, carga y actualización de los datos de los estudiantes, conectando la interfaz con la base de datos en formato CSV.

Primero, se define la ruta del archivo **datos_estudiantes.csv**. Si el archivo existe, los datos se cargan automáticamente; si no, se crea un data frame vacío con la estructura necesaria, permitiendo que la aplicación funcione desde su primera ejecución.

Luego, se emplea **reactiveVal** para almacenar los datos en memoria de forma reactiva, de modo que cualquier cambio se refleje automáticamente en toda la aplicación sin recargarla.

Finalmente, la función **guardar_datos** guarda los cambios en el archivo CSV y actualiza el objeto reactivo, asegurando la persistencia de la información incluso después de cerrar la aplicación

Paso 7

```
# AGREGAR
observeEvent(input$add, {
  df <- Estudiante()
  peso_val <- if(is.na(input$peso)) 0 else input$peso
  talla_val <- if(is.na(input$talla)) 0 else input$talla
  imc <- if(talla_val > 0) round(peso_val / (talla_val^2), 1) else 0

  nuevo <- data.frame(
    ID = input$id, Nombre = input$nombre, Sexo = input$sexo, Edad = input$edad,
    FechaNacimiento = as.character(input$fecha), Peso = peso_val, Talla = talla_val,
    IMC = imc, Sangre = input$sangre, Morbilidad = input$morbilidad,
    Enfermedad = input$enfermedad, Medicamentos = input$medicamentos,
    Alergias = input$alergias, Contacto = input$contacto, stringsAsFactors = FALSE
  )
  guardar_datos(rbind(df, nuevo))
  showNotification("Estudiante agregado", type = "message")
})
```

Cuando se presiona el botón **Agregar**, se capturan los datos ingresados en el formulario, se validan valores como peso y talla, y se calcula automáticamente el **IMC**. Luego, toda la información se organiza en un nuevo registro (data frame) y se guarda junto a los datos existentes mediante **guardar_datos**. Finalmente, se muestra una notificación confirmando que el estudiante fue agregado correctamente.

Paso 8

```
# INDICADORES
output$totalEstudiante <- renderValueBox({
  valueBox(nrow(Estudiante()), "Total Estudiantes", icon = icon("users"), color = "blue")
})

output$promEdad <- renderValueBox({
  df <- Estudiante()
  prom <- if(nrow(df) > 0) round(mean(df$Edad, na.rm = TRUE), 1) else 0
  valueBox(prom, "Edad Promedio", icon = icon("hourglass-half"), color = "purple")
})

output$morbilidadAlta <- renderValueBox({
  df <- Estudiante()
  conteo <- if(nrow(df) > 0) sum(df$Morbilidad > 70, na.rm = TRUE) else 0
  valueBox(conteo, "Morbilidad Critica (>70%)", icon = icon("exclamation-triangle"), color = "red")
})

# CONSULTAR
observeEvent(input$buscar,{
  df <- Estudiante()
  res <- df[df$ID == input$id_consulta, ]
  output$resultado_consulta <- renderPrint({ if(nrow(res) == 0) "No encontrado" else res })
})
```

Este bloque de código muestra **indicadores y la función de consulta** dentro de la app Shiny.

Primero, se crean indicadores visuales (*valueBox*) que muestran: el total de estudiantes registrados, la edad promedio y la cantidad de casos con morbilidad crítica (>70%). Estos valores se calculan automáticamente a partir de los datos almacenados.

Luego, en la sección **Consultar**, al presionar el botón de búsqueda se filtra la base de datos por el ID ingresado. Si el estudiante existe, se muestran sus datos; si no, aparece el mensaje “*No encontrado*”.

Paso 8

```
# CARGAR PARA ACTUALIZAR
observeEvent(input$buscar_upd,{
  df <- Estudiante()
  idx <- which(df$ID == input$id_upd)
  if(length(idx) == 0) {
    showNotification("ID no encontrado", type = "warning")
    return()
  }
  updateTextInput(session, "nombre_upd", value = df$Nombre[idx])
  updateSelectInput(session, "sexo_upd", selected = df$Sexo[idx])
  updateNumericInput(session, "edad_upd", value = df$Edad[idx])
  updateDateInput(session, "fecha_upd", value = as.Date(df$FechaNacimiento[idx]))
  updateNumericInput(session, "peso_upd", value = df$Peso[idx])
  updateNumericInput(session, "talla_upd", value = df$Talla[idx])
  updateSelectInput(session, "sangre_upd", selected = df$Sangre[idx])
  updateNumericInput(session, "morbilidad_upd", value = df$Morbilidad[idx])
  updateTextInput(session, "enfermedad_upd", value = df$Enfermedad[idx])
  updateTextInput(session, "medicamentos_upd", value = df$Medicamentos[idx])
  updateTextInput(session, "alergias_upd", value = df$Alergias[idx])
  updateTextInput(session, "contacto_upd", value = df$Contacto[idx])
})
```

Este bloque se encarga de **cargar la información de un estudiante para poder actualizarla** dentro de la aplicación.

Cuando el usuario ingresa un ID y presiona buscar, el sistema revisa la base de datos reactiva para encontrar ese registro.

- Si el ID no existe, se muestra una notificación de advertencia indicando que no fue encontrado.
- Si el ID existe, la aplicación rellena automáticamente todos los campos del formulario (nombre, sexo, edad, peso, talla, enfermedades, etc.) con los datos actuales del estudiante.

De esta manera, el usuario puede modificar solo los campos necesarios y guardar los cambios, evitando errores y facilitando la edición de registros.

Parte 9

```
# ACTUALIZAR
observeEvent(input$actualizar,{
  df <- Estudiante()
  idx <- which(df$ID == input$id_upd)
  if(length(idx) == 0) return()
  imc <- if(input$talla_upd > 0) round(input$peso_upd/(input$talla_upd^2), 1) else 0
  df[idx,] <- list(input$id_upd, input$nombre_upd, input$sexo_upd, input$edad_upd,
    as.character(input$fecha_upd), input$peso_upd, input$talla_upd,
    imc, input$sangre_upd, input$morbilidad_upd, input$enfermedad_upd,
    input$medicamentos_upd, input$alergias_upd, input$contacto_upd)
  guardar_datos(df)
  showNotification("Registro actualizado", type = "message")
})

# ELIMINAR
observeEvent(input$eliminar, {
  df <- Estudiante()
  idx <- which(df$ID == input$id_upd)
  if(length(idx) == 0) return()
  guardar_datos(df[-idx,])
  showNotification("Registro eliminado", type = "warning")
})
```

Este bloque permite **actualizar y eliminar registros de estudiantes** dentro de la aplicación.

En la sección **Actualizar**, el sistema busca el estudiante por su ID. Si lo encuentra, recálcula el IMC con los nuevos datos y reemplaza toda la información del registro con los valores actualizados. Luego, guarda los cambios y muestra un mensaje de confirmación.

En la sección **Eliminar**, la aplicación también busca el ID del estudiante. Si existe, elimina ese registro de la base de datos, guarda los cambios y muestra una notificación indicando que el registro fue eliminado.

Esto garantiza una gestión completa de los datos, permitiendo mantener la información correcta y actualizada.

Parte 10

```
# GRÁFICOS
output$tabla_Estudiante <- renderDT({ datatable(Estudiante(), options = list(scrollX = TRUE)) })

output$export_csv <- downloadHandler(
  filename = function() { paste0("Estudiante_", Sys.Date(), ".csv") },
  content = function(file) { write.csv(Estudiante(), file, row.names = FALSE) }
)

output$grafHipotesis <- renderPlot({
  df <- Estudiante()
  if(nrow(df) < 2) return(NULL)
  ggplot(df, aes(x = IMC, y = Morbilidad)) +
    geom_point(aes(color = Sexo), size = 4) +
    geom_smooth(method = "lm", color = "red", fill = "pink") +
    theme_minimal() +
    labs(title = "Evidencia Empírica de la Relación Nutricional",
      x = "IMC (Índice de Masa Corporal)",
      y = "Porcentaje de Morbilidad")
})

output$grafEnfermedad <- renderPlot({
  df <- Estudiante()
  if(nrow(df) == 0) return(NULL)
  ggplot(df, aes(x = Enfermedad)) + geom_bar(fill = "#3c8dbc") + theme_minimal()
})

output$grafMorbilidad <- renderPlot({
  df <- Estudiante()
  if(nrow(df) == 0) return(NULL)
  df$Estado <- ifelse(df$Morbilidad > 70, "Alta", "Normal")
  ggplot(df, aes(x = factor(1), fill = Estado)) + geom_bar(width = 1) + coord_polar("y") + theme_void()
})

observeEvent(input$cerrar_app, { stopApp() })
```

Este bloque gestiona la **visualización y exportación de datos** en la aplicación.

Primero, muestra la tabla de estudiantes en formato interactivo y permite **exportar los datos a un archivo CSV** con fecha automática en el nombre.

Luego, genera tres gráficos:

- **Relación IMC vs. Morbilidad:** muestra la relación entre el índice de masa corporal y el porcentaje de morbilidad, con una línea de tendencia.
- **Frecuencia de Enfermedades:** gráfico de barras que indica cuántos estudiantes presentan cada enfermedad.
- **Estado de Morbilidad:** gráfico circular que clasifica a los estudiantes en morbilidad alta o normal.

Finalmente, se incluye un botón para **cerrar la aplicación**, finalizando su ejecución de forma segura.

Parte 11

```
# -----
# Lanzar la APP
# -----
rshinyApp(ui, server)
```

Este último bloque es el encargado de **iniciar la aplicación Shiny**.

La función shinyApp(ui, server) conecta la interfaz de usuario (**UI**) con la lógica del servidor (**server**), permitiendo que ambos componentes trabajen juntos. Al ejecutarse esta línea, la aplicación se lanza y queda lista para que el usuario interactúe con ella.

En resumen, es el punto de arranque que pone en funcionamiento todo el sistema.

#Códigos utilizados

```
dashboardHeader()
Crea la cabecera del panel.
Fundamento: Muestra el título y elementos superiores del dashboard.

dashboardSidebar()
Crea la barra lateral de navegación.
Fundamento: Permite acceder a las diferentes secciones de la aplicación.

dashboardBody()
Contiene el contenido principal.
Fundamento: Aquí se visualizan formularios, gráficos y tablas.

sidebarMenu()
Agrupa las opciones del menú lateral.
Fundamento: Organiza las secciones para facilitar la navegación.

menuItem()
Crea una opción dentro del menú.
Fundamento: Permite abrir cada módulo o pestaña del sistema.

tabItems()
Contenedor de todas las pestañas.
Fundamento: Permite cambiar entre diferentes vistas del dashboard.

tabItem()
Define una pestaña individual.
Fundamento: Cada pestaña muestra una función específica (agregar, consultar, etc.).

fluidRow()
Crea filas adaptables.
Fundamento: Organiza los elementos de forma responsiva según el tamaño de pantalla.

box()
Crea contenedores visuales con título.
Fundamento: Agrupa información relacionada en paneles ordenados.

column()
Divide el espacio en columnas.
Fundamento: Permite distribuir el contenido horizontalmente.
--
```

`valueBoxOutput()`
Muestra indicadores numéricos.
Fundamento: Visualiza estadísticas clave como totales o promedios.

`plotOutput()`
Muestra gráficos.
Fundamento: Permite representar datos de forma visual.

`numericInput()`
Campo para ingresar números.
Fundamento: Se usa para datos numéricos como edad, peso o morbilidad.

`textInput()`
Campo para ingresar texto.
Fundamento: Permite registrar nombres, enfermedades u observaciones.

`selectInput()`
Lista desplegable.
Fundamento: Permite seleccionar opciones predefinidas (sexo, tipo de sangre).

`dateInput()`
Selector de fecha.
Fundamento: Facilita el ingreso correcto de fechas.

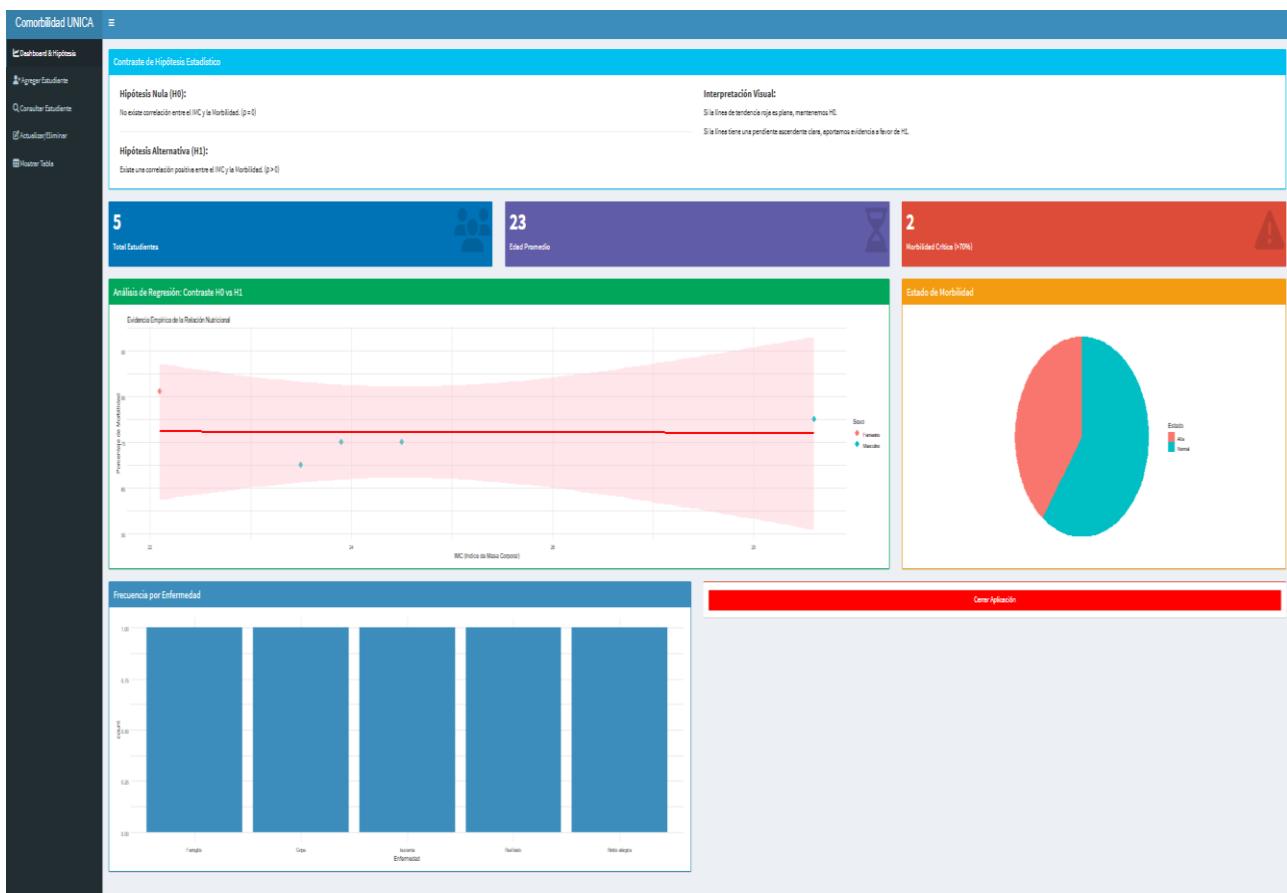
`actionButton()`
Botón de acción.
Fundamento: Ejecuta eventos como agregar, buscar, actualizar o eliminar.

`verbatimTextOutput()`
Muestra texto sin formato.
Fundamento: Permite visualizar resultados de consultas de forma clara.

`DTOutput()`
Muestra tablas interactivas.
Fundamento: Permite visualizar datos con búsqueda, ordenamiento y desplazamiento.

`downloadButton()`
Botón de descarga.
Fundamento: Permite exportar datos a archivos como CSV

Resultados



Nuevo Registro

ID: 5

Nombre: Flores Estalla Angel Alejandro

Sexo: Masculino

Edad: 25

Fecha de nacimiento: 2001-04-06

Peso (kg): 75

Talla (m): 1.62

Tipo de sangre: A-

Morbilidad (%): 75

Enfermedad: Rinitis alérgica

Medicamentos: Loratadina

Alergias: Picazon

Contacto de emergencia: Jorge Castillo – Vecino – 943218765

Agregar Paciente

Comorbilidad UNICA

[Dashboard & Hipótesis](#)

[Agregar Estudiante](#)

[Consultar Estudiante](#)

[Actualizar/Eliminar](#)

[MostrarTabla](#)

Búsqueda por ID

ID del Estudiante

Consultar

ID	Nombre	Sexo	Edad	FechaNacimiento	Peso	Talla
3	Aquije Mendoza Wilfredo	Masculino	30	1996-05-21	70	1.71
	IMC Sangre Morbilidad Enfermedad Medicamentos Alergias					
3	23.9	A+	70	leucemia	amoxicilina	hinchazon
				Contacto		
3	Miguel Ramos	- Hermano -	965432187			

Comorbilidad UNICA

[Agregar Estudiante](#)

[Consultar Estudiante](#)

[Actualizar/Eliminar](#)

[MostrarTabla](#)

ID del Estudiante a Modificar

Cargar Datos

Nombre

Sexo

Edad

Fecha nacimiento

Peso (kg)

Talla (m)

Tipo de sangre

Morbilidad (%)

Enfermedad

Medicamentos

Alergias

Contacto de emergencia

[Actualizar Registro](#) [Eliminar Estudiante](#)

Comorbilidad UNICA

[Dashboard & Hipótesis](#)

[Agregar Estudiante](#)

[Consultar Estudiante](#)

[Actualizar/Eliminar](#)

[MostrarTabla](#)

Base de Datos

Show 10 entries

Search:

ID	Nombre	Sexo	Edad	FechaNacimiento	Peso	Talla	IMC	Sangre	Morbilidad	Enfermedad	Medicamentos	Alergias	Contacto
1	Apaza Quispe Abraham	Masculino	20	2005-10-30	61	1.61	23.5	O+	65	Resfrido	paracetamol	ninguna	Carlos Mendoza - Padre - 934567812
2	Lopez Astoyauri Ronaldo	Masculino	19	2023-01-18	65	1.63	24.5	AB-	70	Faringitis	Ibuprofeno	Sarpullido	Ana López - Madre - 976543218
3	Aquije Mendoza Wilfredo	Masculino	30	1996-05-21	70	1.71	23.9	A+	70	leucemia	amoxicilina	hinchazon	Miguel Ramos - Hermano - 965432187
4	Torres Magallanes Maria	Femenino	21	2005-08-03	51	1.52	22.1	B+	81	Gripe	Loperamida	Dificultad para respirar	Patricia Gómez - Tía - 954321876
5	Flores Estalla Angel Alejandro	Masculino	25	2001-04-06	75	1.62	28.6	A-	75	Rinitis alérgica	Loratadina	Picazón	Jorge Castillo - Vecino - 943218765

Showing 1 to 5 of 5 entries

[Exportar a CSV](#)

Previous 1 Next

