

Proyecto KardIA

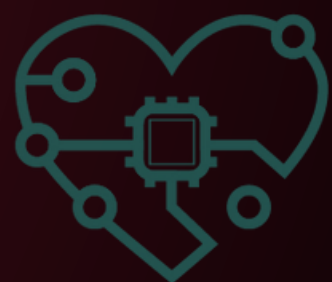
SQL | Comisión 75960 | Pre-Entrega 2

Insua, Gonzalo Nicolás

La aplicación **KardIA** es un proyecto ficticio, desarrollado únicamente con fines educativos en el marco del trabajo final obligatorio del curso. Todos los datos, funcionalidades y escenarios presentados han sido diseñados para simular un caso real, sin representar una implementación activa ni un producto disponible comercialmente.

[INGRESAR](#)

¿Qué es KardIA?



KardIA es una herramienta interna de análisis de datos diseñada para centros de salud que utiliza SQL para transformar datos clínicos en indicadores de riesgo, generando reportes en tiempo real. Ayuda a clasificar pacientes según su vulnerabilidad, mejorando la toma de decisiones y promoviendo una gestión de salud preventiva y centrada en el paciente.

**KARDIA TRANSFORMA DATOS CLÍNICOS
EN DECISIONES INTELIGENTES.**



Objetivo del Proyecto **KardIA**

El objetivo principal del proyecto es desarrollar una solución funcional que permita a los centros de salud analizar información clínica, detectar patrones de riesgo cardiovascular, y optimizar la clasificación de pacientes. Además, **KardIA** es una herramienta con múltiples aristas cross-funcionales, ya que su implementación también impacta áreas como:

- Analítica médica (detección de tendencias y patrones)
- Gestión de recursos (priorización de atención)
- Soporte a decisiones clínicas y administrativas

Problema abordado por Proyecto **KardIA**

En muchos entornos sanitarios, la información médica está dispersa, sin estructura y poco aprovechada. El Dataset original, aunque útil, no estaba diseñado para uso clínico real.

KardIA aborda esta brecha al implementar una base de datos estructurada, normalizada y optimizada, que permite realizar consultas complejas, generar alertas automáticas y centralizar la información del paciente en un solo sistema.

Base de Datos utilizada

Para el desarrollo de **KardIA**, se utilizó un Dataset de Kaggle con 10.000 registros clínicos en inglés.

Se tradujeron los datos al español y se recortó a 5.000 registros representativos para facilitar pruebas.

El Dataset fue reorganizado de tabla plana a un modelo relacional en SQL, codificando variables clave (diabetes, tabaquismo, etc.).

Esto permitió validar la arquitectura de KardIA y simular su uso con datos reales.

BD ORIGINAL

[Click aquí para acceder](#)

BD TRABAJADA

[Click aquí para acceder](#)

1

ELECCIÓN DB

2

TRADUCCIÓN EN-ES

3

DEPURACIÓN

4

NORMALIZACIÓN

5

REORGANIZACIÓN

Diagrama Entidad-Relación

InfartosDB es la base de datos relacional sobre la que se construyó **KardIA**.

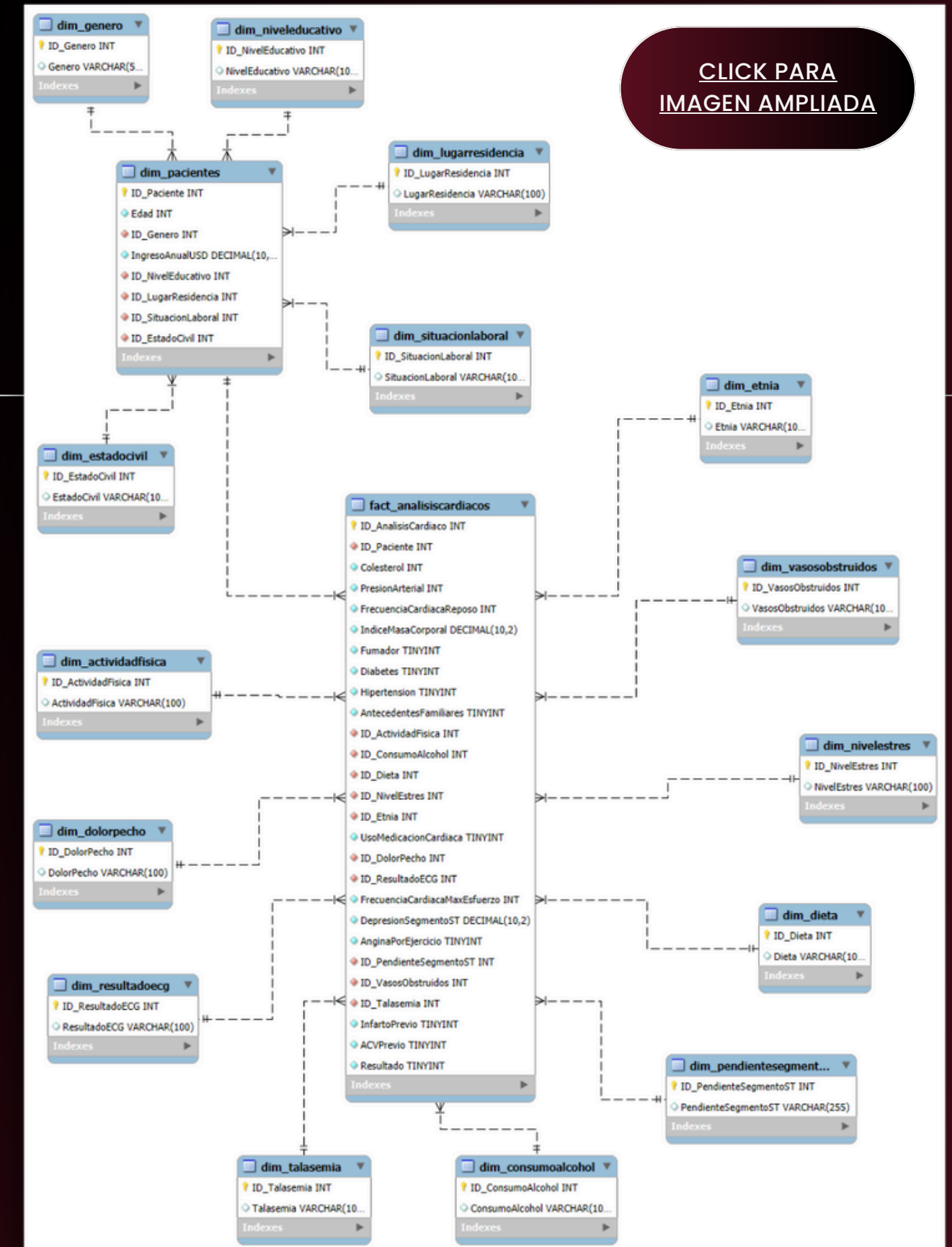
Su diseño en estrella, con la tabla de hechos Fact_AnalisisCardiacos en el centro y múltiples dimensiones normalizadas, permite analizar datos clínicos, generar reportes y segmentar pacientes de forma eficiente en MySQL.

HECHOS

Fact_AnalisisCardiacos

DIMENSIONES

16 dimensiones en total



Tablas DIM 1-3

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

Dim_ConsumoAlcohol Frecuencia de consumo de alcohol				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_ConsumoAlcohol	int	NO	PRI	auto_increment
ConsumoAlcohol	varchar(100)	YES	-	-

Dim_ActividadFisica Nivel de actividad física habitual				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_ActividadFisica	int	NO	PRI	auto_increment
ActividadFisica	varchar(100)	YES	-	-

Dim_Dieta Tipo de dieta habitual				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_Dieta	int	NO	PRI	auto_increment
Dieta	varchar(100)	YES	-	-

Tablas DIM 4-6

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

Dim_EstadoCivil Estado civil del paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_EstadoCivil	int	NO	PRI	auto_increment
EstadoCivil	varchar(100)	YES	-	-

Dim_DolorPecho Tipo de dolor de pecho reportado				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_DolorPecho	int	NO	PRI	auto_increment
DolorPecho	varchar(100)	YES	-	-

Dim_Etnia Grupo étnico al que pertenece el paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_Etnia	int	NO	PRI	auto_increment
Etnia	varchar(100)	YES	-	-

Tablas DIM 7-9

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

Dim_LugarResidencia Lugar o zona de residencia del paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_LugarResidencia	int	NO	PRI	auto_increment
LugarResidencia	varchar(100)	YES	-	-

Dim_Genero Género del paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_Genero	int	NO	PRI	auto_increment
Genero	varchar(50)	YES	-	-

Dim_NivelEducativo Nivel educativo alcanzado por el paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_NivelEducativo	int	NO	PRI	auto_increment
NivelEducativo	varchar(100)	YES	-	-

Tablas DIM 10-12

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

Dim_NivelEstres Nivel de estrés del paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_NivelEstres	int	NO	PRI	auto_increment
NivelEstres	varchar(100)	YES	-	-

Dim_PendienteSegmentoST Recuperación después de un latido				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_PendienteSegmentoST	int	NO	PRI	auto_increment
PendienteSegmentoST	varchar (255)	YES	-	-

Dim_ResultadoECG Hallazgo de alteraciones cardíacas				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_ResultadoECG	int	NO	PRI	auto_increment
ResultadoECG	varchar(100)	YES	-	-

Tablas DIM 13–15

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

Dim_SituacionLaboral Entorno ocupacional del paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_SituacionLaboral	int	NO	PRI	auto_increment
SituacionLaboral	varchar(100)	YES	-	-

Dim_Talasemia Tipo de afectación de la oxigenación sanguínea				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_Talasemia	int	NO	PRI	auto_increment
Talasemia	varchar(100)	YES	-	-

Dim_VasosObstruidos Bloqueo de los vasos sanguíneos principales				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_VasosObstruidos	int	NO	PRI	auto_increment
VasosObstruidos	varchar(100)	YES	-	-

Tablas DIM 16

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

Dim_Pacientes Datos generales de cada paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_Paciente	int	NO	PRI	auto_increment
Edad	int	YES	-	-
ID_Genero	int	NO	MUL	-
IngresoAnualUSD	decimal(10,2)	NO	-	-
ID_NivelEducativo	int	NO	MUL	-
ID_LugarResidencia	int	NO	MUL	-
ID_SituacionLaboral	int	NO	MUL	-
ID_EstadoCivil	int	NO	MUL	-

Tabla FACT 1

1 de 2 Diapositivas

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

Dim_AnalisisCardiacos Datos generales de cada paciente				
CAMPO	TIPO	NULL	KEY	EXTRA
ID_AnalisisCardiacos	int	NO	PRI	auto_increment
ID_Paciente	int	NO	MUL	-
Colesterol	int	NO	-	-
PresionArterial	int	NO	-	-
FrecuenciaCardiacaReposo	int	NO	-	-
IndiceMasaCorporal	decimal(10,2)	NO	-	-
Fumador	tinyint	NO	-	-
Diabetes	tinyint	NO	-	-

CAMPO	TIPO	NULL	KEY	EXTRA
Hipertension	tinyint	NO	-	-
AntecedentesFamiliares	tinyint	NO	-	-
ID_ActividadFisica	int	NO	MUL	-
ID_ConsumoAlcohol	int	NO	MUL	-
ID_Dieta	int	NO	MUL	-
ID_NivelEstres	int	NO	MUL	-
ID_Etnia	int	NO	MUL	-
UsoMedicacionCardiaca	tinyint	NO	-	-
ID_DolorPecho	int	NO	MUL	-

Tabla FACT 1

2 de 2 Diapositivas

Estructura de las entidades y atributos de la Base de Datos diagramada en SQL

CAMPO	TIPO	NULL	KEY	EXTRA
ID_ResultadoECG	int	NO	MUL	-
FrecuenciaCardiacaMaxEsfuerzo	int	NO	-	-
DepresionSegmentoST	decimal(10,2)	NO	-	-
AnginaPorEjercicio	tinyint	NO	-	-
ID_PendienteSegmentoST	int	NO	MUL	-

CAMPO	TIPO	NULL	KEY	EXTRA
ID_VasosObstruidos	int	NO	MUL	-
ID_Talasemia	int	NO	MUL	-
InfartoPrevio	tinyint	NO	-	-
ACVPrevio	tinyint	NO	-	-
Resultado	tinyint	NO	-	-

Vistas

1 de 2 Diapositivas

Durante el desarrollo de **KardIA**, uno de los objetivos principales fue facilitar la consulta y análisis de datos clínicos complejos, a partir de una base de datos relacional.

Pese a que dicho modelo permite una alta normalización y flexibilidad, para la gestión sanitaria resulta clave poder acceder a vistas resumidas, estructuradas y específicas, que simplifiquen la complejidad de las relaciones entre tablas.

Las vistas que se mostrarán a continuación fueron diseñadas y creadas bajo diversos criterios y distintos niveles de complejidad.





VISTA	DESCRIPCIÓN	OBJETIVO	TABLAS INVOLUCRADAS
vw_PacientesAltoRiesgo	Identifica pacientes con 5 o más factores de riesgo cardiovascular.	Facilitar la identificación de pacientes de alto riesgo.	Fact_AnalisisCardiacos
vw_InfartosPorGrupoEtario	Muestra la cantidad de infartos por grupo etario.	Proporcionar análisis de incidencia de infartos según edad.	Fact_AnalisisCardiacos, Dim_Pacientes
vw_EstiloVidaVsInfartos	Compara la frecuencia de infartos según estilo de vida.	Evaluar la relación entre estilo de vida e incidencia de infartos.	Fact_AnalisisCardiacos, Dim_Dieta, Dim_ActividadFisica, Dim_ConsumoAlcohol, Dim_NivelEstres

Funciones

1 de 2 Diapositivas

En el desarrollo de **KardIA**, resulta imprescindible contar con herramientas que permitan transformar datos clínicos complejos en información práctica y accesible. Por eso, entendemos que el uso de funciones específicas facilita la extracción, clasificación y análisis de indicadores clave de salud.

Las funciones que se describen a continuación fueron elaboradas con el propósito de simplificar el trabajo con grandes volúmenes de datos, automatizar procesos de evaluación clínica y apoyar el análisis estadístico. Gracias a estas funciones, es posible obtener insights valiosos que contribuyen a mejorar la gestión sanitaria y el estudio de factores de riesgo cardiovascular.



Funciones



2 de 2 Diapositivas

FUNCIÓN	DESCRIPCIÓN	OBJETIVO	TABLAS INVOLUCRADAS
fn_IndicePresion	Clasifica la presión arterial en tres categorías: Normal, Elevada y Alta.	Permitir la categorización de la presión arterial de los pacientes para facilitar el análisis de su estado de salud cardiovascular.	Fact_AnalisisCardiacos (para obtener el valor de presión arterial)
fn_InfartosEtniaGenero	Devuelve el número total de infartos en pacientes que pertenecen a la etnia y género especificados	Proporcionar un análisis detallado de la incidencia de infartos en diferentes grupos étnicos y de género, lo que puede ayudar en estudios epidemiológicos y en la identificación de factores de riesgo.	Fact_AnalisisCardiacos (para obtener el resultado del infarto). Dim_Etnia (para filtrar por etnia). Dim_Pacientes (para obtener el género del paciente). Dim_Genero (para filtrar por género).
fn_PorcentajeDiabeticosConInfarto	Muestra el porcentaje de pacientes diabéticos que han sufrido un infarto.	Evaluar el impacto de la diabetes en la incidencia de infartos, lo que puede ser crucial para la planificación de intervenciones de salud pública y la gestión de riesgos en pacientes diabéticos.	Fact_AnalisisCardiacos (para contar el total de pacientes diabéticos y aquellos que han sufrido un infarto).

Triggers

1 de 2 Diapositivas

En el contexto del proyecto **KardIA**, los triggers (disparadores) cumplen un rol clave para mejorar la integridad, seguridad y trazabilidad de los datos clínicos dentro de la base de datos InfartosDB.

Estas estructuras permiten ejecutar instrucciones automáticamente antes o después de eventos como inserciones, actualizaciones o eliminaciones.

Los triggers que se presentan a continuación fueron diseñados con una lógica pensada para escenarios reales de gestión sanitaria, priorizando la consistencia de los datos y la automatización de controles médicos esenciales.



Triggers



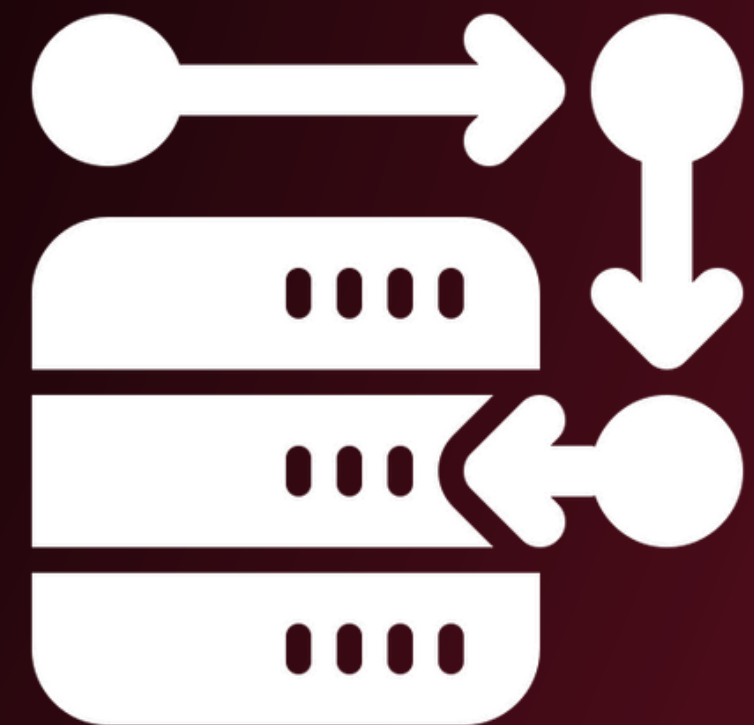
2 de 2 Diapositivas

TRIGGER	DESCRIPCIÓN	OBJETIVO
trg_PrevenirCambioDatosPaciente	Se ejecuta antes de una actualización en Dim_Pacientes y bloquea cualquier intento de modificar los campos ID_Paciente o ID_Genero, lanzando un error personalizado.	Protege la integridad de los datos evitando cambios en claves críticas, garantizando la consistencia estructural y evitando errores en relaciones con otras tablas.
trg_PrevenirBorradoPaciente	Se activa antes de eliminar un paciente y verifica si tiene análisis clínicos registrados; si los tiene, impide la eliminación y lanza un error.	Evita la pérdida de historial clínico, protege la integridad referencial y asegura la trazabilidad de los datos cumpliendo buenas prácticas médicas.
trg_ValidarPresionAlta	Se ejecuta antes de insertar un nuevo registro en la tabla Fact_AnalisisCardiacos y verifica si el valor de PresionArterial es mayor a 180. Si es así, se lanza un error con un mensaje de advertencia.	Garantizar que los registros de presión arterial extremadamente alta sean validados antes de su inserción en la base de datos, promoviendo una revisión médica adecuada y previniendo datos erróneos o peligrosos.

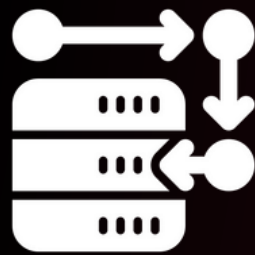
Stored Procedures

1 de 2 Diapositivas

En el proyecto **KardIA**, los Stored Procedures (procedimientos almacenados) son esenciales para automatizar y optimizar la gestión de datos clínicos dentro de la base de datos InfartosDB. Estas rutinas predefinidas permiten ejecutar operaciones complejas de manera eficiente y consistente, mejorando la precisión en el manejo de análisis, la clasificación de pacientes y la detección de riesgos, garantizando la integridad y trazabilidad de la información médica.



Stored Procedures



2 de 2 Diapositivas

STORED PROCEDURE	DESCRIPCIÓN	OBJETIVO
<code>sp_InsertarAnalisisCardiaco</code>	Permite insertar un nuevo registro en la tabla Fact_AnalisisCardiacos. A través de este procedimiento, se pueden ingresar datos detallados de un análisis cardíaco de un paciente.	El objetivo de este procedimiento es simplificar y automatizar la inserción de nuevos análisis cardíacos en la BD, asegurando que todos los valores necesarios se ingresen de forma consistente y estructurada en la tabla principal.
<code>sp_BuscarPacientesAltoRiesgoClinico</code>	Recupera registros de pacientes cuya condición clínica supera ciertos umbrales de riesgo definidos por el usuario. Evalúa específicamente colesterol, presión arterial, IMC y frecuencia cardíaca en reposo.	Este procedimiento identifica pacientes que presentan un riesgo clínico elevado en función de sus mediciones básicas, lo que puede ser útil para priorizar atención médica, monitoreo o intervenciones preventivas.

Scripts de SQL

ESTRUCTURA

[Click aquí
para acceder](#)

VALORES

[Click aquí
para acceder](#)

CONSULTAS

[Click aquí
para acceder](#)

VISTAS

[Click aquí
para acceder](#)

FUNCIONES

[Click aquí
para acceder](#)

TRIGGERS

[Click aquí
para acceder](#)

STORED PROCEDURES

[Click aquí
para acceder](#)

[CLICK PARA
REPOSITORIO GITHUB](#)

Gracias por ver

Detrás de cada línea de código **hay una decisión mejor informada!**