

Base de datos 1



CONECTIVIDAD



INNOVACIÓN



EVALUACIÓN



APRENDIZAJE



"Diseñamos hoy las bases del **conocimiento** para construir el **futuro**."

Fases en el Desarrollo y Construcción de una Base de Datos



Modelaje Conceptual de Bases de Datos



Fundamentos del Modelo Relacional



Normalización



MS-SQL Lenguaje Estructurado de Consulta



```
SELECT nombre, email  
FROM Estudiante  
WHERE id_carrera = 1  
ORDER BY nombre;
```

Vista



Introducción a los nuevos sistemas de gestión de bases de datos NOSQL



Pensamiento lógico



Resolución de problemas



Toma de decisiones



Trabajo colaborativo

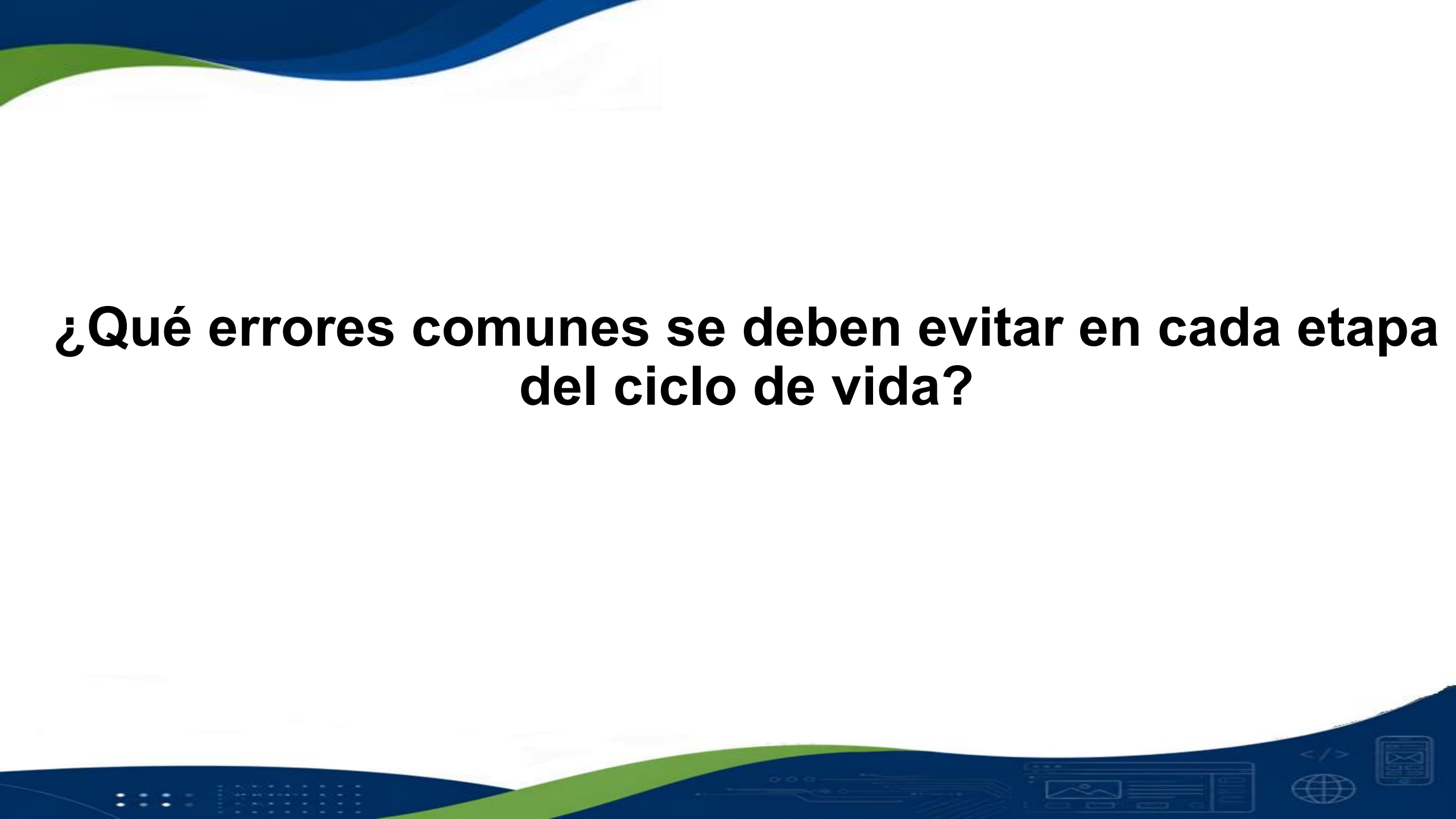
Organiza, estructura y transforma datos en **información inteligente**.

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
LIC. EN DESARROLLO Y GESTION DE SOFTWARE
MATERIA: BASE DE DATOS I (COD. 8398)
GRUPO: 2GS211 - II SEMESTRE DE 2026 - COD. HORARIO: 3971
PROFESORA: MARLENIS A. PIMENTEL M.

SEMANA 2 – VIERNES 28 DE AGOSTO DE 2026

Unidad II: Fases en el desarrollo y construcción de una base de datos





¿Qué errores comunes se deben evitar en cada etapa del ciclo de vida?



1. Planificación

Errores comunes

1. No definir claramente los objetivos

- Se inicia el proyecto sin tener una visión clara de lo que se desea lograr.
- **Ejemplo:** Una universidad solicita un sistema académico, pero nunca especifica si debe manejar matrículas, pagos o gestión docente.

2. Alcance mal definido

- Intentar incluir demasiadas funciones desde el inicio.
- **Ejemplo:** Desarrollar simultáneamente módulos de matrícula, biblioteca, finanzas y recursos humanos sin una planificación adecuada.

3. Ignorar limitaciones de recursos

- No considerar presupuesto, tiempo o personal disponible.
- **Consecuencia:** El proyecto queda incompleto o se retrasa.

2. Recolección y Análisis de Requisitos

Errores comunes

1. No involucrar a los usuarios finales

- Los analistas obtienen información únicamente de los directivos.
- **Ejemplo:** Diseñar un sistema de matrícula sin consultar a las secretarías que realizan el proceso diariamente.

2. Asumir información

- Suponer procesos sin verificarlos.
- **Ejemplo:** Asumir que un estudiante puede matricular una sola carrera cuando la universidad permite doble titulación.

2. Recolección y Análisis de Requisitos

Errores comunes

3. Requisitos incompletos

- No documentar reglas de negocio importantes.
- **Ejemplo:** Olvidar registrar que ciertos cursos requieren prerrequisitos.

4. Mala comunicación

- Interpretar incorrectamente las necesidades de los usuarios.
- **Consecuencia:** Se desarrolla un sistema diferente al esperado.

3. Diseño Conceptual

Errores comunes

1. Identificar incorrectamente las entidades

- Crear entidades innecesarias o ignorar entidades importantes.

2. Relaciones mal definidas

- No identificar adecuadamente las cardinalidades.
- **Ejemplo:**
- Decir que un estudiante solo puede matricular un curso cuando realmente puede matricular varios.

3. No validar el modelo con los usuarios

- Diseñar el modelo sin revisarlo con quienes utilizarán el sistema.

4. Diseño Lógico

Errores comunes

1. Falta de normalización

- Provoca redundancia de datos.

Tabla incorrecta

Estudiante	Curso	Profesor
Ana	Base de Datos	Pérez
Ana	Programación	Gómez

Aquí se repite información innecesariamente.

4. Diseño Lógico

Errores comunes

2. Exceso de normalización

- Dividir demasiado la información generando consultas complejas.

3. No definir claves primarias

- **Ejemplo:**
- Una tabla de estudiantes sin un identificador único.

4. No definir claves foráneas

- Se pierde la integridad referencial.

5. Diseño Físico

Errores comunes

1. No considerar el rendimiento

- Diseñar bases de datos sin índices.
- **Consecuencia:** Consultas lentas cuando aumenta el volumen de datos.

2. Ignorar la seguridad

- No definir permisos de acceso.
- **Ejemplo:** Todos los usuarios pueden modificar notas académicas.

5. Diseño Físico

Errores comunes

3. No planificar respaldos

- La pérdida de datos puede convertirse en una situación crítica.

4. No estimar crecimiento futuro

- Diseñar para pocos usuarios sin considerar la expansión de la organización.

6. Implementación

Errores comunes

1. No seguir los estándares de programación

- Nombres inconsistentes para tablas y campos.

- **Incorrecto**

```
CREATE TABLE dato1
```

- **Mejor**

```
CREATE TABLE Estudiantes
```

6. Implementación

Errores comunes

2. No documentar la base de datos

- Falta de diccionario de datos y comentarios.

3. Omisión de restricciones

- No utilizar:
 - PRIMARY KEY
 - FOREIGN KEY
 - UNIQUE
 - CHECK

4. Migración incorrecta de datos

- Importar datos sin verificar su calidad.

7. Pruebas

Errores comunes

1. Probar únicamente casos normales

- No considerar situaciones excepcionales.
- **Ejemplo:**
- Probar una matrícula válida, pero no una matrícula con datos faltantes.

2. No validar la integridad de los datos

- Permitir registros huérfanos.

7. Pruebas

Errores comunes

3. Ignorar pruebas de rendimiento

- La base de datos funciona bien con 100 registros, pero falla con 100,000.

4. No involucrar usuarios

- Los usuarios detectan problemas después de la puesta en producción.

8. Operación y Mantenimiento

Errores comunes

1. No realizar respaldos periódicos

- Puede ocasionar pérdida total de información.

2. No monitorear el rendimiento

- Las consultas se vuelven lentas con el crecimiento de los datos.

3. No actualizar la seguridad

- Mantener versiones antiguas del SGBD aumenta vulnerabilidades.

8. Operación y Mantenimiento

Errores comunes

4. No documentar cambios

- Genera dificultades para futuros administradores.

5. Falta de capacitación

- Los usuarios desconocen nuevas funcionalidades o procedimientos.

Caso Práctico

- Supongamos que una universidad desarrolla un sistema de matrícula:
- **Error en análisis**
 - No se entrevistó a los estudiantes.
 - No se identificó que un estudiante puede retirar materias.
- **Error en diseño**
 - La tabla Matrícula no contempla retiros.

Caso Práctico

- **Error en implementación**
 - No existen restricciones de integridad.
- **Error en pruebas**
 - Nunca se probó el retiro de materias.
- **Resultado**
 - El sistema entra en producción y no permite registrar retiros académicos, obligando a realizar cambios costosos.

Recomendación general

- La mayoría de los problemas en una base de datos se originan en las primeras etapas, especialmente en la **recolección de requisitos y el diseño conceptual**.
- Un buen análisis inicial reduce significativamente los errores, los costos de corrección y el tiempo de desarrollo.
- Por ello, siempre se recomienda validar cada fase con los usuarios antes de avanzar a la siguiente.

¿Cómo evitar errores en la fase de análisis de requisitos?



¿Cómo evitar errores en la fase de análisis de requisitos?

- La **fase de análisis de requisitos** es una de las más críticas en el desarrollo de una base de datos, ya que los errores cometidos aquí suelen propagarse a las etapas posteriores. De hecho, corregir un requisito mal definido durante la implementación puede ser mucho más costoso que detectarlo durante el análisis.

Estrategias para evitar errores en el análisis de requisitos

1. Involucrar a todos los usuarios clave

- Un error frecuente es entrevistar únicamente a directivos o administradores.

¿Qué hacer?

- Identificar todos los actores que interactuarán con el sistema:
 - Administradores
 - Operadores
 - Gerentes
 - Clientes
 - Técnicos

Estrategias para evitar errores en el análisis de requisitos

1. Involucrar a todos los usuarios clave

- **Ejemplo**
- En un sistema universitario:
- Entrevistar a:
 - Estudiantes
 - Docentes
 - Secretarias
 - Coordinadores académicos
 - Administradores del sistema
- Error:
 - Consultar únicamente al director académico.

Estrategias para evitar errores en el análisis de requisitos

2. Utilizar varias técnicas de determinación de hechos

- No depender de una sola fuente de información.
- **Recomendación**
- Combinar:
 - Entrevistas
 - Cuestionarios
 - Observación
 - Revisión de documentos
 - Muestreo

Estrategias para evitar errores en el análisis de requisitos

2. Utilizar varias técnicas de determinación de hechos

Ejemplo

- Una secretaria afirma que el proceso de matrícula tarda 5 minutos.
- Sin embargo, mediante observación se descubre que realmente tarda 15 minutos por verificaciones manuales adicionales.

Estrategias para evitar errores en el análisis de requisitos

3. Documentar claramente los requisitos

- Todo requisito debe quedar escrito y aprobado.
- **Requisito mal definido**
 - El sistema deberá generar reportes.
- **Requisito bien definido**
 - El sistema deberá generar reportes de matrícula por semestre, carrera y estudiante en formato PDF y Excel.
 - Mientras más específico sea el requisito, menor será la posibilidad de interpretaciones erróneas.

Estrategias para evitar errores en el análisis de requisitos

4. Validar la información con los usuarios

- Después de recopilar los requisitos, se debe confirmar que fueron comprendidos correctamente.
- **Técnica recomendada**
 - Presentar un resumen de requisitos a los usuarios.
- **Ejemplo**
 - El analista documenta:
 - Un estudiante puede matricular hasta 6 materias por semestre.
 - Durante la revisión, los coordinadores indican que los estudiantes sobresalientes pueden matricular hasta 8.
 - Si no se valida el requisito, el sistema se desarrollaría incorrectamente.

Estrategias para evitar errores en el análisis de requisitos

5. Identificar reglas de negocio

- Las reglas de negocio son restricciones que deben cumplirse.
- **Ejemplos**
 - Un estudiante debe aprobar Matemáticas I para cursar Matemáticas II.
 - Un profesor no puede impartir más de cinco cursos por semestre.
 - Una beca solo puede asignarse a estudiantes con promedio superior a 85.
 - Muchas veces estas reglas no aparecen en los formularios y solo se descubren mediante entrevistas detalladas.

Estrategias para evitar errores en el análisis de requisitos

6. Evitar asumir información

- Nunca se debe dar algo por hecho.
- **Error**
 - El analista supone que cada estudiante tiene un único número telefónico.
- **Realidad**
 - Algunos estudiantes registran:
 - Teléfono personal
 - Teléfono residencial
 - Teléfono de emergencia
 - La mejor práctica es siempre preguntar y verificar.

Estrategias para evitar errores en el análisis de requisitos

7. Elaborar modelos preliminares

- Crear diagramas iniciales ayuda a detectar errores tempranamente.

Estrategias para evitar errores en el análisis de requisitos

8. Registrar excepciones y casos especiales

- Los usuarios suelen describir situaciones normales y olvidar las excepciones.
- **Caso normal**
 - Un estudiante se matricula en un curso.
- **Casos especiales**
 - Retiro de materias.
 - Cambio de grupo.
 - Convalidaciones.
 - Cursos repetidos.
 - Estudiantes suspendidos.
 - Estas situaciones deben documentarse desde el inicio.

Estrategias para evitar errores en el análisis de requisitos

9. Mantener una comunicación constante

- La comunicación debe existir durante todo el proyecto.
- **Buenas prácticas**
 - Reuniones periódicas.
 - Actas de reunión.
 - Correos de confirmación.
 - Aprobación formal de requisitos.
 - Esto evita malentendidos entre analistas y usuarios.