Tecnológico de Monterrey

Lecturas y Preguntas de Lecturas

César Ignacio Saucedo Rodríguez A01712245

09 / 06 / 2025

Construcción de software y toma de decisiones (Gpo 501)

Tecnológico de Monterrey

Introducción a la Administración de Proyectos:

Este bloque introductorio tiene como objetivo darnos herramientas concretas para gestionar proyectos de forma eficiente. No se trata solo de ejecutar tareas, sino de hacerlo optimizando recursos y aplicando buenas prácticas reconocidas a nivel mundial, especialmente las definidas por el PMI (Project Management Institute), que es la autoridad global en esta área.

La guía principal que se sigue es el PMBOK (Project Management Body of Knowledge), un marco que resume el conocimiento y las mejores practicas en gestión de proyectos. Esta guía organiza todo el conocimiento en 9 áreas clave: integración, comunicación, alcance, calendario, costo, calidad, riesgos, recursos humanos y aprovisionamiento. A lo largo del curso se van a ver estas áreas tanto en teoría como en práctica, con herramientas como MS Project y análisis de casos reales.

Uno de los puntos más duros que presenta la lectura son las estadísticas de fracaso en proyectos. Por ejemplo, el *Chaos Report* del 94 muestra que solo el 16.2% de los proyectos se consideran realmente exitosos. Las causas comunes son bastante claras: falta de claridad en requisitos, poca participación de usuarios, expectativas irreales, mala comunicacion y gestión deficiente.

Al final, el mensaje es claro: para tener un proyecto exitoso no basta con buenas ideas; se necesita estructura, metodología, y una gestión que anticipe riesgos y aproveche bien los recursos humanos, técnicos y económicos.

Respuestas a las preguntas:

¿Qué es un proyecto?
Es un esfuerzo temporal, con inicio y fin definidos, que busca crear algo único: un producto, servicio o resultado. No es una rutina, es una iniciativa organizada que se va construyendo paso a paso.

¿Qué es la administración de proyectos?
Es aplicar conocimientos, habilidades, herramientas y técnicas para cumplir con los objetivos de

un proyecto. No es solo coordinar personas, sino también asegurar que el trabajo tenga dirección, sentido y resultados concretos.

¿Qué es el PMI?

Es una organización sin fines de lucro fundada en 1969 que se dedica a profesionalizar la gestión de proyectos. Ellos crearon el PMBOK y definen estándares, certifican personas y validan programas de formación.

¿Qué es la guía del PMBOK?

Es la biblia de los proyect managers. Resume todo el conocimiento necesario para administrar bien un proyecto. Desde cómo se estructura una organización hasta qué procesos debes seguir. Establece un vocabulario común y divide la gestión en áreas clave para no dejar nada al azar.

Introducción a los Sistemas de Información

La tecnología ha transformado por completo la manera en la que operan las empresas. Ya no se trata solo de tener computadoras, sino de integra los sistemas de información (SI) en la estrategia del negocio. El director de informática ya no es solo un técnico: ahora es un gestor que debe tener visión empresarial.

Un sistema de información es un conjunto de componentes interconectados que recopilan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones. Su funcionamiento básico se da en cuatro etapas: entrada, procesamiento, salida y retroalimentación, siempre interactuando con el entorno (clientes, proveedores, competidores, etc.).

Los datos, por sí solos, no significan nada; se vuelven información cuando se procesan adecuadamente y ayudan a decidir. Por eso, los SI son clave en cualquier empresa: sirven para automatizar tareas, dar seguimiento a operaciones y mejorar la toma de decisiones.

Hay varios tipos de sistemas de información, cada uno con un propósito distinto:

- TPS (Transaction Processing Systems): Procesan las transacciones del día a día (ventas, pagos, registros).

- OAS (Office Automation Systems): Apoyan actividades de oficina (Word, correo, calendarios).

- KWS (Knowledge Work Systems): Para trabajos especializados, como diseño arquitectónico.

- MIS (Management Information Systems): Dan reportes organizacionales para supervisar y planear.

- DSS (Decision Support Systems): Ayudan a resolver problemas complejos con análisis tipo "what-if".

- GDSS (Group DSS): Toman decisiones en grupo con participación colaborativa.

- ES (Expert Systems): Simulan decisiones humanas expertas (como diagnosticar fallas).

- EIS (Executive Information Systems): Sistemas a la medida para directivos, con dashboards visuales y acceso rápido a información crítica.

Los SI bien implementados no solo automatizan: dan ventaja estratégica, permiten ver el panorama completo y actuar con inteligencia de negocio.

Respuestas a las preguntas

¿Qué es un sistema de información?

Es un conjunto de elementos que trabajan juntos para recolectar, procesar, guardar y entregar

información que ayuda en la toma de decisiones dentro de una organización. Básicamente, es el cerebro operativo del negocio.

¿Cuál es la diferencia entre dato e información?
El dato es solo una representación cruda (como un número o palabra suelta). La información es cuando esos datos están organizados y tienem un significado útil. Ejemplo: "8.5" es un dato, pero "Juan sacó 8.5 en la materia" ya es información.

¿Qué tipos de sistemas de información existen?

- TPS para procesar operaciones rutinarias.
- OAS para automatizar la oficina.
- KWS para conocimiento especializado.
- MIS para generar reportes gerenciales.
- DSS para apoyar decisiones difíciles.
- GDSS para decisiones grupales.
- ES para simular expertos humanos.
- EIS para dar visión estratégica a los directivos.

¿Qué función cumple la retroalimentación en un SI?
Permite evaluar los resultados y mejorar el sistema constantemente. Es lo que cierra el ciclo y convierte a los SI en herramientas inteligentes que aprenden y se ajustan.

Gestión del Alcance

Tecnológico de Monterrey

Este módulo trata de un aspecto clave de cualquier proyecto: el alcance. Básicamente, se refiere a todo lo que se debe hacer —y nada más que eso— para que el proyecto se considere completo y exitoso. Incluir de más o de menos es igual de peligroso.

El alcance se puede dividir en dos:

- Alcance del producto: lo que debe incluir el producto o servicio (funciones, características).

- Alcance del proyecto: las tareas necesarias para entregar ese producto.

La gestión del alcance implica cinco procesos:

1. Iniciación: Se autoriza formalmente el proyecto, se definen objetivos generales, se asigna un jefe y se asegura que haya recursos. Aquí nace el acta del proyecto, un documento clave que da respaldo oficial y claridad sobre lo que se quiere lograr.

2. Planificación del alcance: Se redacta el enunciado del alcance, que describe los objetivos, entregables y justificación. Es el acuerdo base entre cliente y equipo.

3. Definición del alcance: Se descompone el trabajo en tareas manejables con una WBS (Work Breakdown Structure). Esto permite controlar mejor el proyecto, estimar tiempos y costos, y asignar responsabilidades claras.

4. Verificación del alcance: Se revisan entregables con inspecciones, auditorías o pruebas, y se obtiene la aceptación formal del cliente.

5. Control de cambios del alcance: Se asegura que cualquier cambio al alcance pase por un proceso oficial. Así se evita el temido scope creep (cuando se mete trabajo extra sin control).

La WBS es el corazón del control del alcance. Divide todo el trabajo en niveles jerárquicos hasta llegar a unidades pequeñas llamadas paquetes de trabajo, que ya se pueden calendarizar, estimar y monitorear.

También se mencionan herramientas adicionales como:

- OBS (Organizational Breakdown Structure): Para definir qué equipo se encarga de qué parte.

- RAM (Responsibility Assignment Matrix): Para dejar claro quién es responsable de cada tarea.

Respuestas a las preguntas

¿Que es el alcance del proyecto?
Es todo el trabajo necesario —y solo el necesario— para entregar los productos o servicios del proyecto según lo acordado. Define claramente qué está incluido y qué no lo está.

¿Cuál es la diferencia entre el alcance del producto y el alcance del proyecto?
El alcance del producto define qué características y funciones debe tener el producto. El alcance del proyecto detalla qué tareas se deben ejecutar para entregar ese producto.

¿Qué es una WBS?
Es una estructura jerárquica que divide el trabajo del proyecto en componentes más pequeños y manejables. Ayuda a planificar, controlar y asignar responsabilidades.

¿Para qué sirve el acta del proyecto?
Para formalizar el inicio del proyecto. Define objetivos, restricciones, criterios de éxito y asigna roles. También le da autoridad al jefe del proyecto para usar recursos.

¿Qué se busca con la verificación del alcance?
Obtener la aceptación formal de los entregables por parte del cliente. Se hace mediante inspecciones y revisiones para asegurar que todo cumple con lo requerido.

Tecnológico de Monterrey

¿Qué es el scope creep y cómo se controla?

Es cuando se agregan tareas o requisitos no aprobados al proyecto. Se controla con un sistema formal de control de cambios, para que todo pae por revisión y aprobación.

Definición de base de datos y DBMS

Durante mucho tiempo, las empresas manejaban la información a través de archivos ligados directamente a los programas que los usaban. Esto provocaba duplicación de datos, errores, y pérdida de confianza en los sistemas. A partir de los años 60 surgió el enfoque de bases de datos, que propone diseñar una estructura de datos única para toda la organización, compartida entre distintas áreas, evitando redundancias y errores.

Una base de datos es una colección de archivos interrelacionados, pero organizados de forma lógica y centralizada. Por ejemplo, en una universidad podría haber un archivo de alumnos (matrícula, nombre), otro de cursos (clave, semestre, profesor) y otro de inscripciones (matrícula, curso). Lo importante es que estén bien diseñados, sin repeticiones innecesarias y cumpliendo con las necesidades globales de la organización.

Para que esto funcione bien, se necesita un sistema de gestión de base de datos o DBMS (Database Management System). El DBMS es un conjunto de programas que permiten acceder, manipular y proteger los datos. Sin el, una base de datos sería solo un conjunto de archivos sin control real.

Usar bases de datos es especialmente útil cuando:

- Hay varios archivos interrelacionados.
- Distintos usuarios o departamentos necesitan acceder a la misma información.
- El volumen de registros es muy alto (como en bancos, gobiernos, o servicios de telefonía).

Los DBMS resuelven los grandes problemas de los sistemas antiguos, como:

- Redundancia e inconsistencia: se evita tener datos duplicados o diferentes en distintos archivos.
- Dificultad de acceso: ya no hay que hacer un programa nuevo cada vez que alguien necesita un dato.
- Aislamiento de datos: facilita obtener información desde diferentes puntos sin reescribir código.
- Errores de acceso concurrente: protege los datos cuando varias personas trabajan al mismo tiempo.
- Seguridad: permite dar permisos específicos a cada usuario.
- Integridad: asegura que los datos sigan reglas lógicas (como que no haya saldos negativos).
- Respaldo y recuperación: permite restaurar la base si algo falla.

En resumen, el DBMS actúa como puente entre los datos en disco y los usuarios, haciendo que todo funcione bien: rápido, seguro, y con sentido.

Notación del Modelo Entidad-Relación (MER)

El propósito de diseñar una base de datos relacional no es solo almacenar datos, sino estructurarlos de forma que se minimice la redundancia y se facilite el acceso y actualización de la información. La clave está en usar esquemas bien pensados, y el modelo entidad-relación (E-R) es la herramienta base para lograrlo.

Tecnológico de Monterrey

Problemas comunes en diseños mal hechos

Cuando se manejan archivos por separado, cada uno vinculado a una aplicación, se vuelve inevitable la duplicación y la inconsistencia de los datos. Esto, además de complicar las actualizaciones, hace que los sistemas pierdan credibilidad. Aunque parezca cosa del pasado, muchas empresas siguen funcionando así, ahora con hojas de cálculo o "bases de datos de escritorio". El enfoque de base de datos nace para resolver justamente esto: crear una estructura unificada que represente toda la información de la empresa.

Fases del diseño de bases de datos

1. Observación del mundo real
2. Recolección de requerimientos
3. Diseño conceptual (con modelos como el MER)
4. Diseño lógico (ajustado al SGBD específico)
5. Diseño físico (estructura de almacenamiento)

Entidades

Una entidad representa algo del mundo real que queremos modelar: clientes, empleados, productos. Se representan con rectángulos y deben tener existencia propia. Su identificación única se logra mediante uno o más atributos clave.

Asociaciones

Las asociaciones describen relaciones entre entidades. Se representan con romvos y deben tener nombres claros, en forma de verbo. También pueden tener atributos propios, como una fecha o un precio asociado a la relación.

Tecnológico de Monterrey

Cardinalidad

La cardinalidad indica cuántas veces se relaciona una entidad con otra. Existen tres tipos principales:

- 1:1 (uno a uno)
- 1:N (uno a muchos)
- N:M (muchos a muchos)

Esta notación también permite representar la participación de las entidades:

- Total (línea doble): todas las ocurrencias deben participar.
- Parcial (línea simple): solo algunas lo hacen.

Atributos

Son las propiedades que describen entidades o asociaciones. Se representan con elipses y pueden ser:

- Simples o compuestos
- Monovaluados o multivaluados
- Derivados (como la edad, calculada a partir de la fecha de nacimiento)

El conjunto de posibles valores de un atributo se llama dominio. Además, existen atributos clave, que son los que permiten identificar de forma única una instancia.

Tecnológico de Monterrey

Entidades fuertes y débiles

Una entidad fuerte existe por sí sola. Una entidad débil necesita de otra para identificarse, como el familiar que depende del empleado. Estas se representan con doble trazo y su relación con la entidad fuerte lleva una línea doble con cardinalidad 1:N.

Modelo ER extendido

Cuando los requerimientos son más complejos, se usa una versión extendida del modelo E-R que incorpora:

- Roles: una misma entidad puede tener diferentes papeles en una asociación.
- Generalización y especialización: una entidad padre da origen a subclases que heredan sus atributos y agregan otros propios.
- Superclases y subclases: permiten representar estructuras jerárquicas.

Reglas de integridad

No todo se representa gráficamente. Existen restricciones lógicas que deben cumplirse, como:

- Una fecha de fin no puede ser anterior a la fecha de inicio.
- El sueldo de un empleado debe ser menor al del jefe.
- La suma de cantidades embarcadas no puede superar la cantidad pedida.

También pueden existir cotas de cardinalidad, que establecen mínimos y máximos, por ejemplo: un alumno debe inscribirse a por lo menos uno y como máximo cinco cursos.

Tecnológico de Monterrey

Estas reglas adicionales son fundamentales para asegurar la calidad del modelo y evitar errores en la implementación final.

Resumen Reglas de traslado del modelo E-R al modelo relacional

Cuando ya tenemos listo nuestro modelo entidad-relación (MER), el siguiente paso es traducirlo a tablas relacionales. Este proceso se conoce como traslado del modelo E-R a MR (modelo relacional). El objetivo es generar una estructura en forma de tablas donde los datos se almacenen de manera eficiente y sin perder las relaciones del modelo original.

Procedimiento general

- Cada entidad se transforma en una tabla. Sus columnas son los atributos, y la clave primaria se define usando los identificadores. Puede elegirse una clave artificial si no existe un atributo natural que funcione como identificador único.

- Cada asociación con cardinalidad N:N también se transforma en una tabla. Esta tabla incluye los identificadores de las entidades que se están relacionando, y estos identificadores juntos forman la clave primaria. Si la asociación tiene atributos propios, también se añaden.

- Cada asociación con cardinalidad 1:N se maneja agregando el identificador de la entidad del lado “1” como clave foránea en la tabla del lado “N”. No se crea una tabla nueva.

- Cada asociación con cardinalidad 1:1 puede resolverse igual que la 1:N, agregando el identificador de una entidad en la otra. No importa el orden.

Ejemplo aplicado

Dado un modelo con entidades A, B, C y asociaciones X (N:N) y Y (1:N), se obtiene el siguiente resultado en el modelo relacional:

- A(a1, a2, a3)
- B(b1, b2, a1)
- C(c1, c2, c3, c4)
- X(a1, c1, x1, x2)

Se observa cómo se integran las claves y cómo se mantienen las asociaciones correctamente sin perder consistencia.

Reglas adicionales del traslado

Relaciones ISA

Cuando hay relaciones de generalización/especialización (ISA), se hereda el identificador de la entidad generalizada. Las tablas que representan las subclases incluyen ese mismo identificador como clave primaria.

Ejemplo:

- G(g1, g2, g3)
- Ea(g1, a1, a2)
- Eb(g1, b1)

Entidades fuertes y débiles

Las entidades débiles no tienen clave propia y dependen de una entidad fuerte. En el modelo relacional, la tabla de la entidad débil hereda la clave primaria de la fuerte y la combina con su propia clave parcial para formar su clave primaria.

Ejemplo:

- F(f1, f2)
- D(f1, d1, d2)

Roles

Cuando una entidad se relaciona consigo misma o tiene múltiples relaciones con otra, se utilizan roles para distinguir las columnas que contienen claves foráneas. Esto evita ambigüedad en los nombres de las columnas y mantiene la estructura legible.

Ejemplo:

- E(e1, e2)
- R(e1, RolDeE1, r1, r2)

El uso de roles es obligatorio para evitar columnas duplicadas con el mismo nombre y definir claramente los distintos papeles que una entidad puede jugar en una relación.

Comunicación

La comunicación en los proyectos no se trata solo de informar, sino de hacerlo bien: a tiempo, con claridad, y enfocada en quien realmente necesita la información. En proyectos con muchos actores involucrados, una buena gestión de la comunicación puede ser la diferencia entre el éxito y el fracaso.

¿Qué es la gestión de la comunicación?

Es el conjunto de procesos para recopilar, distribuir, almacenar, recuperar e incluso destruir información del proyecto de manera adecuada y oportuna. La comunicación eficaz conecta a personas con distintos niveles, roles, intereses y culturas.

Se reconocen múltiples dimensiones:

- Interna vs. externa
- Formal vs. informal
- Vertical vs. horizontal
- Verbal vs. no verbal
- Escrita vs. oral

Procesos clave

1. Identificar a los interesados

Consiste en reconocer a todas las personas u organizaciones que se verán afectadas por el proyecto y documentar su nivel de interés, su influencia, y su impacto en el éxito del proyecto. Esto permite priorizar a quién dirigir los esfuerzos y diseñar estrategias para maximizar su apoyo y minimizar su resistencia.

2. Planear la comunicación

Aquí se define:

- Quién necesita qué información.

- Cuándo y cómo se le entregará.

- Quién será responsable de transmitirla.
  Se debe asegurar que la información sea oportuna, pertinente y, si es necesario, confidencial.

3. Distribuir la información

Este proceso se ejecuta durante todo el proyecto y consiste en entregar la información según lo planeado. Se utilizan técnicas como selección de medios, presentaciones, conducción de reuniones y retroalimentación de los interesados.

4. Administrar las expectativas

No basta con informar, también hay que trabajar con los interesados, anticiparse a sus inquietudes, negociar y resolver problemas. Aquí entra la parte más sensible de la comunicación: escuchar, persuadir, explicar y ajustar el enfoque cuando sea necesario.

5. Reportar el desempeño

Es el proceso para entregar reportes sobre el estado del proyecto: qué se ha logrado, qué riesgos están activos, qué problemas han surgido, y qué se espera para el siguiente periodo. También se comparan estimaciones con datos reales, lo que permite ajustar el plan si hace falta.

Conceptos básicos del modelo relacional

El modelo relacional surge como una evolución de cómo estructurar bases de datos. Introducido por Codd en los años 70, propone representar los datos en forma de relaciones (tablas), independientes del almacenamiento físico y con reglas claras para mantener su integridad. Se

fundamenta en la teoría de conjuntos, y su enfoque facilita la manipulación y consulta de información de forma lógica, flexible y estandarizada.

Entre sus características más importantes están:

- Independencia física y lógica de los datos.
- Uniformidad y simplicidad en la estructura.
- Lenguaje estandarizado: SQL.
- Acceso a datos mediante vistas lógicas (subconjuntos).

Relación

El concepto base es la relación, que representa un conjunto de tuplas. Cada relación puede tener uno o más dominios (atributos) y su representación más común es en forma de tabla. Las columnas representan los atributos y las filas las tuplas.

Ejemplo: una relación Comida con los dominios Entrada, Sopa y Plato se representa como Comida(Entrada, Sopa, Plato), y cada fila combina elementos de estos dominios

Una relación es formalmente un subconjunto del producto cartesiano de sus dominios. Es decir, no todas las combinaciones posibles tienen que estar presentes.

Noción de relación

Hay dos formas de describir una relación:

- Intensión: estructura fija de la tabla (atributos).

- Extensión: conjunto de datos (tuplas) que contiene en un momento dado.

La representación tabular es la más práctica. Por ejemplo, la relación AUTOR(NOMBRE, NACIONALIDAD, INSTITUCIÓN) define la estructura, mientras que una tabla con Pepe, John y Pierre es su extensión.

Dominio y atributo

Un dominio es el conjunto de posibles valores para un atributo. Debe ser homogéneo (todos del mismo tipo) y atómico (no divisible).

Un atributo es el papel que juega el dominio dentro de la relación. Ejemplo: un dominio de colores puede usarse como atributo para definir el color de autos, camisas o banderas.

Llave primaria

Es el atributo (o conjunto de atributos) que permite identificar de forma única cada tupla. Puede ser:

- Simple: una sola columna.

- Compuesta: dos o más columnas.

Ejemplo: en una tabla Empleados, la combinación de Departamento y Puesto puede ser la llave si individualmente no garantizan unicidad.

Llave foránea

Es un atributo que referencia la llave primaria de otra tabla. Sirve para enlazar relaciones distintas. Los valores deben coincidir con los de la tabla referida o ser nulos. En la práctica, es clave para mantener integridad referencial.

Ejemplo: la tabla Empleados puede tener como claves foráneas los atributos Departamento y Puesto, que deben coincidir con los registros en las tablas respectivas.

Restricciones

Se dividen en:

- Inherentes: no hay duplicados, el orden de filas o columnas no importa.
- De usuario: reglas adicionales impuestas por lógica de negocio (por ejemplo, no permitir edades negativas).

Incluye la integridad referencial: si una tupla referencia otra (por medio de una llave foránea), esa otra debe existir o el valor debe ser nulo.

Valores nulos

El modelo relacional acepta valores nulos como indicador de información desconocida o no aplicable. Estos deben manejarse con cuidado para no afectar la lógica de las consultas o la integridad.

Álgebra relacional

El álgebra relacional define operaciones que pueden realizarse sobre las relaciones, permitiendo combinar, filtrar y transformar datos

Principales operadores:

- Unión: combina todas las tuplas de dos relaciones compatibles.
- Intersección: obtiene solo las tuplas comunes.

- Diferencia: obtiene las tuplas que están en una pero no en la otra.
- Proyección: selecciona columnas específicas.
- Selección: filtra filas según una condición.
- Join natural: une tablas basándose en columnas comunes.
- Teta-join: unión con condición explícita.
- Producto cartesiano: todas las combinaciones posibles entre dos relaciones.
- División: encuentra tuplas que cumplen una condición para todas las tuplas de otra relación.

Estas operaciones forman el núcleo de la manipulación de datos en SQL.

Diagramas de secuencia con UML

Los diagramas de secuencia son una herramienta para modelar cómo interactúan los objetos en un sistema a lo largo del tiempo. Se utilizan en el diseño de software para detallar la implementación de escenarios definidos previamente en los casos de uso. Estos diagramas permiten visualizar la dinámica de un método o una funcionalidad concreta, mostrando los mensajes que se intercambian los objetos en el orden en que ocurren.

Fundamentos

Un diagrama de secuencia es un tipo de diagrama de interacción. En el eje horizontal se representan los objetos y en el eje vertical el paso del tiempo. Su principal característica es que hace énfasis en el orden temporal de los mensajes.

También existe otro tipo llamado diagrama de colaboración que se enfoca más en la estructura que en el tiempo.

Elementos principales

- Línea de vida: línea vertical discontinua que representa la existencia del objeto.
- Foco de control: rectángulo delgado sobre la línea de vida que indica cuándo un objeto está ejecutando una acción.
- Mensajes:
    - Síncronos: el objeto que envía el mensaje se bloquea hasta recibir respuesta. Se representan con flecha de cabeza sólida.
    - Asíncronos: no bloquean al emisor, y pueden generar nuevos hilos. Se representan con flecha de cabeza abierta.
    - Retornos: respuestas a un mensaje. Se pueden omitir si el final de la activación lo deja claro.

También pueden mostrarse la creación (<<create>>) y destrucción (<<destroy>>) de objetos como estereotipos opcionales.

Tipos de diagramas de secuencia

- De instancia: describe un caso particular de ejecución (una instancia de un caso de uso).
- Genérico: describe la interacción completa del caso de uso, usando condiciones, bifurcaciones y bucles.

Fragmentos combinados

Tecnológico de Monterrey

Son estructuras que permiten incluir decisiones, repeticiones y condiciones dentro del diagrama:

- alt: representa alternativas (if...else).
- opt: opción ejecutada solo si se cumple la condición (equivale a un if sin else).
- loop: repite el fragmento según una condición.
- ref: hace referencia a otro diagrama ya definido, útil para reutilización.
- par: indica que los fragmentos se ejecutan en paralelo.
- critical: solo permite un proceso activo en el fragmento, útil para evitar condiciones de carrera.

Estos fragmentos se enmarcan y etiquetan según el operador correspondiente. Se usan para modelar estructuras de control de flujo similares a las de la programación estructurada.

Álgebra relacional, SQL básico y funciones agregadas

La lectura propone un objetivo claro: vincular el álgebra relacional con los comandos prácticos del lenguaje SQL. Esto tiene mucho sentido porque si bien el álgebra relacional es una base teórica que estructura cómo se manipulan los datos en las bases de datos, SQL es el lenguaje que realmente usamos para ejecutar esas operaciones. En otras palabras, el álgebra relacional describe qué queremos hacer, y SQL nos da las herramientas para hacerlo.

El trabajo es individual, y lo importante aqui es entender cómo las operaciones que ya conocemos del álgebra (como unión, intersección, proyección, selección, producto cartesiano, joins, etc.) se traducen en sentencias SQL concretas. Por ejemplo, la proyección se relaciona con SELECT, la selección con WHERE, y las uniones o intersecciones con los distintos tipos de JOIN.

Además, se aborda el uso de funciones agregadas en SQL (como SUM, AVG, COUNT, MAX, MIN), que nos permiten obtener estadísticas o resúmenes de grandes volúmenes de datos, algo que no está tan presente en el álgebra relacional pura, pero es fundamental para análisis reales en bases de datos.

En resumen, esta lectura tiene un enfoque práctico: bajar los conceptos abstractos del álgebra relacional a sentencias SQL que sí usamos en la práctica profesional. Es un paso clave para integrar la teoría con la implementación.

Metodología para diseñar casos de pruebas a partir de casos de uso

En la mayoría de los proyectos de software, las pruebas consumen entre el 30 % y el 50 % del costo total, pero aun así muchas veces no se prueba bien antes de liberar. La razón es clara: se suele probar sin una metodología sólida y se deja todo al final. Este enfoque es de alto riesgo, porque si se detectan errores graves al terminar, corregirlos puede costar mucho tiempo y dinero.

El artículo propone una estrategia concreta: utilizar los casos de uso como base para generar los casos de prueba, y así empezar las pruebas desde las primeras etapas del desarrollo. Los casos de uso, que describen cómo interactúan los usuarios o sistemas externos con la aplicación, permiten definir qué se espera que haga el sistema, por lo tanto, también qué se debe probar.

Cómo se estructuran los casos de uso

Los casos de uso se redactan con una estructura textual que incluye:

- Nombre
- Descripción breve
- Flujo de eventos (principal y alternos)
- Requisitos especiales

- Precondiciones y postcondiciones

El flujo de eventos es la parte clave para pruebas. El flujo básico describe lo que ocurre cuando todo va bien. Los flujos alternativos representan excepciones o caminos opcionales.

Cada combinación entre flujo básico y alternos se llama escenario. Por ejemplo: un escenario puede ser el flujo básico completo, otro puede incluir el básico más una desviación por error. Cada uno de estos escenarios se transforma luego en uno o más casos de prueba.

Proceso para generar casos de prueba

El artículo propone un proceso de tres pasos:

1. Generar los escenarios del caso de uso
   Se identifican todas las combinaciones posibles entre flujo básico y alternos. Por ejemplo: "registro exitoso", "usuario no identificado", "catálogo no disponible", etc.

2. Identificar los casos de prueba
   Para cada escenario, se definen condiciones y situaciones que se deben probar. Esto incluye tanto los caminos exitosos como los casos con errores. Aquí se pueden usar matrices con columnas como: ID del estudiante, cursos seleccionados, prerequisitos, disponibilidad, etc.

3. Asignar valores de prueba
   Finalmente, se sustituyen las condiciones generales (como "válido" o "inválido") por datos reales que se usarán en la ejecución de las pruebas. Así se completa cada caso de prueba con entradas concretas y resultados esperados.

Conclusión

Vincular los casos de prueba directamente con los casos de uso permite:

- Iniciar las pruebas desde el inicio del proyecto
- Asegurar que se cubran todos los caminos del sistema
- Detectar errores antes, cuando aún es barato corregirlos
- Aumentar la calidad del software entregado

Consultas en SQL usando roles y subconsultas

En SQL, hay situaciones donde una misma tabla se usa más de una vez dentro de una misma consulta. Esto ocurre cuando comparamos registros entre sí o cuando una tabla cumple distintos roles en una relación. Para resolverlo, se utilizan alias o roles, que nos permiten diferenciar las instancias de la tabla sin ambigüedad.

Un ejemplo típico es el de los empleados que reportan a otros empleados. Como la relacion es consigo misma, usamos alias como empleado y jefe para distinguir entre quien reporta y a quién se reporta. El uso de estos alias evita confusión y permite que la consulta sea clara y funcional.

Otra estrategia útil es la creación de sinónimos permanentes con CREATE SYNONYM, que permite definir nombres alternos para las tablas y facilitar la lectura en consultas que se repiten constantemente.

Este mismo enfoque se aplica cuando se quiere comparar registros de una misma tabla en distintos momentos, como cuando se analiza la diferencia en ventas de un producto entre dos fechas. Se usa la misma tabla dos veces con distintos alias (primero, segundo) y se comparan los valores para obtener la diferencia.

Por otro lado, las subconsultas permiten usar el resultado de una consulta como condición dentro de otra. Se pueden usar para filtrar, buscar registros que no existen en otra tabla o comparar valores agregados.

Ejemplos comunes:

Para encontrar productos que no se han vendido:

```
SELECT idproducto FROM productos

WHERE idproducto NOT IN (SELECT idproducto FROM ventasdiarias);
```

O con NOT EXISTS:

```
SELECT idproducto FROM productos p

WHERE NOT EXISTS (

  SELECT * FROM ventasdiarias v WHERE v.idproducto = p.idproducto

);
```

También se pueden construir subconsultas que comparen valores calculados, como el total vendido por producto, y luego seleccionar solo aquellos que superen cierto monto. En este caso, se puede usar una subconsulta en el WHERE o en la lista de selección.

En resumen, tanto los roles como las subconsultas son herramientas esenciales cuando se necesita estructurar consultas más complejas, ya sea porque una tabla cumple múltiples funciones, o porque la lógica requiere comparar datos que dependen de otras búsquedas. Estas técnicas permiten mantener claridad en las sentencias y evitan ambigüedades al trabajar con relaciones internas o condiciones más elaboradas.

Normalización

La normalización es una técnica clave en el diseño de bases de datos. Su objetivo es estructurar los datos de manera lógica y estable para evitar redundancia, facilitar el mantenimiento y permitir modificaciones sin afectar la integridad del sistema. Al normalizar, se descompone la información en tablas más simples, eliminando dependencias innecesarias entre campos.

¿Por qué normalizar?

- Para identificar relaciones pertinentes entre datos.
- Para facilitar consultas y reportes sin inconsistencias.
- Para hacer más fácil insertar, modificar o eliminar registros.
- Para preparar el modelo ante futuras reorganizaciones o ampliaciones.

Proceso de normalización

1. Se descomponen los datos en tablas bidimensionales.
2. Se eliminan relaciones basadas solo en la llave primaria.
3. Se eliminan dependencias transitivas.

Este proceso se guía por un conjunto de reglas conocidas como formas normales. Cada forma mejora la estructura de la anterior.

Formas normales

Primera forma normal (1FN)

Una tabla está en 1FN si cada celda contiene un valor atómico (no arreglos, listas o repeticiones). Las columnas deben tener nombres únicos y las filas no deben ser duplicadas.

Segunda forma normal (2FN)

Se cumple si ya está en 1FN y todos los atributos no clave dependen completamente de la llave primaria. Es decir, no se permiten dependencias parciales.

Tecnológico de Monterrey

Tercera forma normal (3FN)

Está en 3FN si ya cumple 2FN y no tiene dependencias transitivas. Esto significa que ningún atributo no clave debe depender de otro atributo no clave.

Forma normal de Boyce-Codd (FNBC)

Una tabla está en FNBC si cada determinante de una dependencia funcional es una llave candidata. Es una versión más estricta que 3FN, útil cuando hay múltiples llaves candidatas.

Cuarta forma normal (4FN)

Se cumple cuando no existen dependencias de valores múltiples. Estas ocurren cuando un solo valor de clave determina múltiples valores en dos columnas que no se relacionan entre sí.

Quinta forma normal (5FN)

Se aplica cuando una relación puede dividirse en varias subrelaciones sin pérdida de información, pero no puede reconstruirse correctamente. Se enfoca en dependencias de producto.