

APUNTES

Aksel Deneken Maldonado A01711966

LECTURAS Y PREGUNTAS:

INTRODUCCIÓN A LA ADMINISTRACIÓN DE PROYECTOS

Un proyecto es algo temporal que busca lograr un resultado único. Administrar un proyecto significa aplicar conocimientos, habilidades y herramientas para cumplir con lo que se espera. El PMI es una organización que creó la guía del PMBOK, donde se explican las áreas clave para llevar bien un proyecto (como alcance, tiempo, costo, calidad, etc.). Muchos proyectos fallan por errores comunes como no tener bien definidos los requerimientos, mala planeación o comunicación, o falta de apoyo del equipo.

Preguntas:

¿Qué es un proyecto? Algo temporal que se hace para crear algo único (un producto, servicio o resultado).

¿Qué es administrar un proyecto? Usar conocimientos, técnicas y herramientas para lograr lo que el proyecto necesita.

¿Qué es el PMI? Una organización que se dedica a profesionalizar la gestión de proyectos y certificar a quienes lo hacen bien.

¿Qué es el PMBOK? Una guía que explica cómo se deben manejar los proyectos y qué se necesita saber para hacerlo bien.

¿Por qué fallan los proyectos? Porque no hay buena planeación, cambian las cosas a cada rato, hay mala comunicación o el equipo no está bien preparado.

INTRODUCCIÓN A LOS SISTEMAS DE INFORMACIÓN

Los avances tecnológicos han integrado los sistemas de información en la estrategia de las empresas, volviendo al director de informática un gestor más que un técnico. Un sistema de información es un conjunto de componentes que recopilan, procesan, almacenan y distribuyen información para apoyar decisiones y control. Se compone de entrada, procesamiento, salida y retroalimentación. Los datos por sí solos no tienen valor, pero procesados generan información útil.

Existen varios tipos de sistemas: TPS (procesan transacciones), OAS (automatizan oficina), KWS (apoyan creación de conocimiento), MIS (dan información a administradores), DSS (ayudan a tomar decisiones), GDSS (decisiones grupales), ES (imitan a expertos) y EIS (información clave para ejecutivos).

GESTIÓN DEL ALCANCE

El alcance del proyecto es todo lo que se debe entregar y hacer, ni más ni menos. Se define con objetivos SMART y se asegura que todos tengan claro qué se va a hacer y cómo. Hay dos tipos: alcance del producto (funciones/características) y alcance del proyecto (el trabajo que se hace para entregar el producto). Para gestionarlo bien hay cinco procesos: iniciación, planificación, definición, verificación y control. Todo empieza con el acta del proyecto (project charter), que formaliza el inicio. Luego se hace el enunciado del alcance, que explica el porqué del proyecto, sus objetivos y entregables. Se usa la WBS para dividir el trabajo en partes más pequeñas (paquetes de trabajo), que permiten planear y controlar. Finalmente, se revisa si lo entregado cumple con lo acordado y se manejan los cambios para evitar desviaciones (scope creep).

¿Qué nivel de detalle se debe cubrir cuando se construye una WBS?

- A. Tantos como quiera el jefe de proyecto
- B. Hasta 3 niveles y al menos 80 horas de dedicación
- C. Hasta el nivel especificado por tú gerente

D. El número necesario para controlar el proyecto de forma efectiva

DEFINICIÓN DE BASE DE DATOS Y MANEJADOR DE DBMS

Antes, los datos estaban ligados a cada programa, lo que causaba duplicidad e inconsistencia. El enfoque de bases de datos surge para evitar eso, diseñando la información de forma integral para toda la empresa. Una base de datos es un conjunto de archivos interrelacionados, diseñados para minimizar la repetición y cubrir las necesidades de distintos usuarios. Un DBMS es el sistema que gestiona esa base, permitiendo acceder, almacenar y proteger los datos de forma eficiente y segura. Los DBMS también controlan el acceso múltiple, la seguridad, las copias de

respaldo y la integridad de los datos. Se usan cuando hay muchos archivos relacionados, múltiples usuarios o grandes volúmenes de información.

NOTACIÓN MODELO ENTIDAD RELACIÓN (MER)

El diseño de bases de datos busca minimizar redundancia y facilitar la recuperación de información. Para lograrlo, usamos modelos como el Entidad-Relación (E-R), que describe la información con entidades, relaciones y atributos. Si se diseña mal una base de datos, se generan redundancias, valores nulos y dificultad para actualizar. El proceso de diseño incluye varias fases: recolección de requerimientos, diseño conceptual (modelo E-R), diseño lógico (con base en un SGBD relacional) y diseño físico. En el modelo E-R usamos rectángulos para entidades, rombos para asociaciones y elipses para atributos. Las asociaciones pueden tener distintos tipos de participación (opcional u obligatoria) y cardinalidades (1:1, 1:N, N:N). También se contemplan casos como roles, subclases (especialización), superclases (generalización), entidades débiles, y restricciones de integridad adicionales.

REGLAS DE TRASLADO DEL MODELO ENTIDAD RELACIÓN A TABLAS

Para pasar un modelo E-R a tablas relacionales (MR), se siguen reglas según el tipo de entidad o relación. Cada entidad se vuelve una tabla con columnas igual a sus atributos, y la llave primaria es su identificador. Las asociaciones N:N se convierten en nuevas tablas que incluyen los identificadores de las entidades relacionadas más los atributos propios de la asociación. Las relaciones 1:N se manejan agregando la llave de la entidad 1 a la tabla de la entidad N. En 1:1, basta con agregar la llave de una entidad a la otra. Las relaciones ISA heredan la llave de la superclase a las subclases. Las entidades débiles también heredan la llave de la fuerte, pero su llave incluye ambos identificadores. Los roles sirven para evitar nombres repetidos cuando una entidad se relaciona consigo misma.

COMUNICACIÓN EN LA ADMINISTRACIÓN DE PROYECTOS

La comunicación en un proyecto es clave para conectar a todos los interesados y asegurar que la información fluya bien. Se trata de recolectar, distribuir, almacenar, recuperar y eliminar información correctamente. Hay que identificar a los

interesados, planear cómo se va a comunicar la información, distribuirla, manejar expectativas y reportar desempeño. Se consideran distintos tipos de comunicación: formal/informal, interna/externa, escrita/oral, verbal/no verbal. Una buena comunicación mejora la coordinación, evita conflictos y mantiene a todos alineados.

Planear la comunicación – OBD

¿Quién necesita qué?

César coordina y necesita todo el avance. Aksel ve lo técnico. Alex las pruebas y entrega final.

¿Cuándo?

Diario al final del día y en reuniones rápidas.

¿Cómo?

Discord, en persona y Google Docs para avances. WhatsApp para avisos urgentes.

¿Quién lo entrega?

Cada quien su parte. César revisa que todo esté en orden.

¿Cómo asegurar oportunidad y confidencialidad?

Usar el canal correcto, ser puntuales y no compartir fuera del equipo.

CONCEPTOS BÁSICOS DEL MODELO RELACIONAL

El modelo relacional fue propuesto por Codd en 1970. Su idea principal es que los datos se organizan en relaciones (tablas) basadas en la teoría de conjuntos. Sus objetivos son independencia física y lógica, flexibilidad, uniformidad y sencillez. Cada relación tiene un esquema (estructura con nombre de atributos y dominios) y una extensión (conjunto de tuplas). Una tupla es una fila, y un atributo es una columna que se basa en un dominio (tipo de dato). Las llaves primarias identifican tuplas únicas, y las llaves foráneas conectan tablas entre sí. Hay restricciones que aseguran integridad de los datos (como no permitir nulos en llaves primarias o que una llave foránea coincida con una primaria). El modelo también incluye operadores algebraicos como unión, intersección, diferencia, selección, proyección, joins y producto cartesiano.

DIAGRAMAS DE SECUENCIA

Un diagrama de secuencia es un tipo de diagrama de interacción en UML que muestra cómo los objetos se comunican entre sí a lo largo del tiempo. Se usa para representar la lógica interna de un caso de uso, incluyendo los objetos involucrados, los métodos que se llaman y el orden en que sucede todo. El eje vertical muestra el tiempo y el horizontal los objetos. Las líneas de vida indican la existencia del objeto y los rectángulos finos (foco de control) indican cuándo un objeto está activo. Hay mensajes síncronos (bloquean al objeto que llama) y asíncronos (no lo bloquean). También hay fragmentos combinados como condiciones (alt), opciones (opt), bucles (loop), referencias (ref), procesos paralelos (par) y regiones críticas (critical). Los diagramas pueden ser de instancia (escenario específico) o genéricos (caso de uso general).

SQL Y ÁLGEBRA RELACIONAL

SQL, que significa "Structured Query Language", es el lenguaje que usan los sistemas de bases de datos para definir, consultar y modificar datos. Está basado en los conceptos del álgebra relacional. Toda consulta en álgebra se puede expresar con SQL. Las tres partes básicas de una consulta SQL son "SELECT", que indica qué columnas se quieren; "FROM", que indica de qué tablas se va a obtener la información; y "WHERE", que filtra los datos con una condición.

Las operaciones del álgebra relacional tienen equivalentes en SQL. Por ejemplo, una selección en álgebra se expresa con "SELECT * FROM tabla WHERE condición". Una proyección corresponde a "SELECT columna1, columna2 FROM tabla". Las uniones, intersecciones y diferencias también tienen su versión en SQL, aunque algunas solo funcionan en ciertos sistemas como Oracle.

Además, SQL permite hacer consultas agregadas. Esto significa que puedes obtener datos resumidos con funciones como "SUM", "AVG", "MIN", "MAX", "COUNT". Para agrupar datos se usa "GROUP BY", y para filtrar esos grupos se utiliza "HAVING", que es parecido a "WHERE", pero aplicado después de agrupar. Por ejemplo, si se quiere saber la suma de ventas por producto, se puede agrupar por producto y aplicar la función "SUM" sobre la cantidad.

GENERACIÓN DE CASOS DE PRUEBA A PARTIR DE CASOS DE USO

Probar software es una tarea difícil y costosa, pero fundamental para asegurar calidad. Muchas veces se hace sin una metodología clara, lo cual afecta la cobertura y eficiencia de las pruebas. Una forma efectiva de organizar las pruebas desde el inicio del desarrollo es usar los casos de uso para generar los casos de prueba.

Los casos de uso describen lo que el sistema debe hacer desde la perspectiva del usuario. Incluyen actores, acciones del sistema y posibles variaciones del flujo normal. Cada caso de uso se puede describir con texto estructurado que incluye nombre, descripción, flujo de eventos, requisitos especiales, precondiciones y postcondiciones.

El flujo de eventos es la parte más importante para generar pruebas. Se divide en flujo básico (lo que normalmente pasa) y flujos alternos (variaciones o excepciones). A partir de estos flujos, se definen los diferentes escenarios de uso. Cada escenario representa un camino distinto a través del caso de uso.

Para generar casos de prueba se sigue un proceso en tres pasos:

Paso 1: Generar escenarios

Se identifican todas las combinaciones posibles entre el flujo básico y los flujos alternos. Cada combinación forma un escenario diferente. Estos escenarios son la base para definir los casos de prueba.

Paso 2: Identificar casos de prueba

Por cada escenario, se define al menos un caso de prueba. Este incluye la condición inicial, las entradas y el resultado esperado. Se recomienda usar una matriz donde se indique qué datos son válidos, inválidos o no aplican. Se debe cubrir tanto los escenarios exitosos como los casos donde algo sale mal.

Paso 3: Asignar valores concretos

Una vez definidos los casos, se sustituyen las letras "V", "I" o "n/a" por datos reales. Sin estos datos, los casos de prueba no pueden implementarse. La matriz final incluye valores específicos para entradas como usuario, contraseña, cursos seleccionados, etc., y el resultado esperado.

Usar esta metodología permite comenzar las pruebas desde etapas tempranas, detectar errores antes de que se vuelva costoso arreglarlos, asegurar una buena cobertura y mejorar la calidad del software.

CONSULTAS EN SQL USANDO ROLES Y SUBCONSULTAS

En SQL, a veces necesitamos usar una misma tabla más de una vez en una consulta. Esto sucede, por ejemplo, cuando comparamos registros o cuando una tabla representa distintos roles. Para resolverlo, se pueden usar alias que le den nombres temporales distintos a cada aparición de la tabla. Otra opción es crear sinónimos permanentes con "CREATE SYNONYM".

Un ejemplo común es cuando tenemos una tabla de "viajes" con columnas de origen y destino que hacen referencia a la misma tabla "ciudades". Para mostrar el nombre de ambas ciudades, usamos alias como "origen" y "destino" para la misma tabla. Otro ejemplo clásico es la relación entre empleados y sus jefes, donde "empleados" se relaciona consigo misma, y usamos alias como "e" y "j" para diferenciar.

También es útil cuando queremos comparar datos de distintas fechas. Por ejemplo, para calcular la diferencia de ventas de un producto entre dos días distintos, se usa dos veces la misma tabla "ventasdiarias" con alias diferentes.

Las subconsultas permiten usar el resultado de una consulta dentro de otra. Sirven para casos como: encontrar productos que no se han vendido, comparar importes o aplicar condiciones complejas. Se pueden usar con "IN", "NOT IN", "EXISTS", "NOT EXISTS" o con operadores como ">", "<", "=" combinados con "ANY", "SOME" o "ALL". Por ejemplo, para encontrar los productos con ventas mayores a cierto valor, se puede usar una subconsulta dentro del WHERE que calcule el total vendido.

NORMALIZACIÓN

Normalizar una base de datos sirve para evitar errores como la redundancia o la inconsistencia, y lograr un diseño lógico que sea fácil de mantener, escalar y adaptar a nuevos requerimientos. Implica descomponer datos complejos en registros simples y manejables. El proceso consiste en convertir los datos en registros bidimensionales, eliminar los que no dependan completamente de la llave primaria y luego las dependencias transitivas, siguiendo distintas formas normales.

La 1FN exige que cada celda tenga un solo valor, sin filas repetidas ni columnas duplicadas.

La 2FN añade que los atributos no clave deben depender totalmente de la clave primaria.

La 3FN elimina las dependencias transitivas, donde un dato depende de otro que a su vez depende de la clave.

La BCNF es una versión más estricta de la 3FN: todos los determinantes deben ser llaves candidatas.

La 4FN resuelve dependencias multivaluadas, separando en tablas distintas cada conjunto de valores independientes.

La 5FN aborda dependencias más complejas (de producto), evitando pérdida de información al dividir tablas con relaciones entre especializaciones o roles múltiples.

NOTAS DE CLASE ABAJO ↓

MR
 Entidad (a_1, a_2, a_3)
 clave PK
 Relación ($q_1, otra$)
 Cardinalidad de la relación
 A(oo, a1)
 P(1, a1)
 Arquitectura de aplicación de los sistemas. Muestra arbolitos del sistema como nodos, los cuales son conectados mediante enlaces de comunicación.
 Nodos: definición de firma, entidad de representación tanto dispositivos hardware como entornos de aplicación software.
 Nodos: representa un tipo de recurso computacional, sobre el que se puede desplegar arbolitos, para la aplicación.
 dispositivos hardware y entornos de ejecución.
 Componentes
 hardware
 hardware-soft
 Software
 Creador de...
 Interfaz...
 Software
 Servidor Linux
 Apache
 Multiplex
 E. re...
 E. null
 Any N...
 be con...
 into
 fact...
 NFA:
 Multiplex
 interf...
 direct...

Diagrama de clases: Relaciones

Nombre Tipo - Frase Verbal - Nombre Tipo

Compartir

Empleado 1 una
Persona 1 asignar n
Vuelo n
Agrupar n
Agrupar n

Superclase

* solo a nés 1. * a uno a nés 1. * 40 de uno a 40
5 solo 5 3, 5, 8 solo 3, 5, 8

Navegabilidad

estudiante otro alumno Agrupar la clase asignar a conoce a la clase estudiante, no al revers.

Composición

compuesto tiene parte de

Cochete 1
Rueda 4
Motor 1
Cilindro 4

Agresión | el compuesto puede ser más de uno

◊ ← hueso ◀ herencia

Generalización

Identificar cosas comunes y hacer Superclases y Sub

MEP

Se crea D. relacional y software que permite crear, administrar y consultar base de datos organizadas en tablas

Entidades → Círculos

Círculos → Rectángulos

relación

Rede existir más de una asociación

Asociación descriptiva

As. cliente

vende

As. cliente

Tipos Relación Representación

1:1 Cardinalidad 1:1 1

1:N Una a muchas N

N:M Muchas a Muchas N N

Parcial, no siempre tiene que haber relación

Atributos, son características, propiedades

%o valores de las entidades

Provedores

R.F.C.
Patria Social
Dirección
Teléfono
Email

Materiales

Clave
Descripción
Especificaciones

Atributos Compuestos \rightarrow conjunto de atributos
Simple \rightarrow no se pueden dividir

Mora binomial \rightarrow solo un valor por atributo
Multi valorable \rightarrow más de un valor

Derivados \rightarrow el atributo se puede conseguir en
función de los otros atributos

Relación ISA "es un"

Entidades Reales "Independientes"
Entidades derivadas "dependientes"

Grupos de normalidad. O/S

Gestión de Proyecto

- Justificación del proyecto
- Producto del proyecto
- Antecedentes
- SMART

Modulo
Requisitos
Cada elemento
modulable

Linea
binaria

Work Breakdown Structure

Software

Subestructura

Modulo 1 Modulo 2

Requisitos análisis Diseño Codificación Pruebas

Cada elemento de representar un entregable medible

Diagrama de secuencia, describe interacción "temporales"

Diagrama de colaboración, organización estructural

Secuencia, Modelar interacción entre objetos.

Línea de vida:

Línea discontinua vertical, existencia de un objeto

Foco de control, rectangular, tiempo, ejecución una acción

Objeto Objeto

Diagramas de estado

Comportamiento de un sistema, todos los estados posibles en los que puede estar un objeto particular

Estados y transiciones

Estado \rightarrow condición o situación en la vida de un objeto

Acción \rightarrow operación

```
graph TD; A((nombre Acción)) --> B[Nombre Estado / actividad]; B -- "nombre Acción" --> C[ ]; B -- "nombre Estado" --> D[ ]
```