

**Concurso-oposición al Cuerpo de Profesores de Secundaria
Generalitat Valenciana, 1995. Prueba práctica de Informática.**

ENUNCIADO

Se desea analizar y diseñar un Sistema de Información de control de calidad para la exclusión o admisión a comercialización, de los productos obtenidos en una industria destinada a la fabricación de diferentes modelos de tubo de acero (de diferentes diámetros y largos, roscados, sin roscar, etc.)

Para ello cada modelo de tubo dispone de un número variable de PARAMETROS DE CALIDAD, todos ellos numéricos. Un parámetro de calidad es un atributo de cada modelo de tubo capaz de demostrar un defecto de fabricación que debe estar comprendido entre un valor máximo y uno mínimo.

Por ejemplo, el valor del diámetro interior de un determinado modelo de tubo podrá ser un parámetro de calidad, así, si al modelo TR020 le corresponde idealmente un diámetro interior de 16,00 mm., evidentemente no todos los tubos que se fabriquen de este modelo tendrán un diámetro interior de 16,00 mm. ya que las condiciones de fabricación influirán en que esto se consiga o no, ahora bien por ejemplo, puede considerarse que para cada tubo de este modelo si el valor del diámetro interior está comprendido entre 15,50 mm. (Valor mínimo) y 16,50 mm. (Valor máximo) el tubo es correcto y puede comercializarse.

El sistema de producción es en serie y flexible, es decir, fácilmente puede modificarse para fabricar diferentes modelos de tubos. Cada tubo posee un número de serie único desde el comienzo de su fabricación.

Al final del proceso productivo existe una sección de toma de datos de los valores de los parámetros de calidad de cada uno de los tubos fabricados. Además, esta misma sección tendrá como función apartar del proceso aquellos tubos que no cumplan con al menos uno de los valores máximos y mínimos de cada parámetro de calidad del modelo al que pertenece ese tubo en concreto.

Este proceso actualmente es manual no pudiéndose registrar o almacenar la información. Se trata, por tanto, de mecanizar el proceso de forma que permita, al menos: introducir valores de los parámetros de calidad, almacenar la información, determinar la aceptación o no de cada tubo y obtener informes estadísticos sobre la producción.

Deberá realizarse:

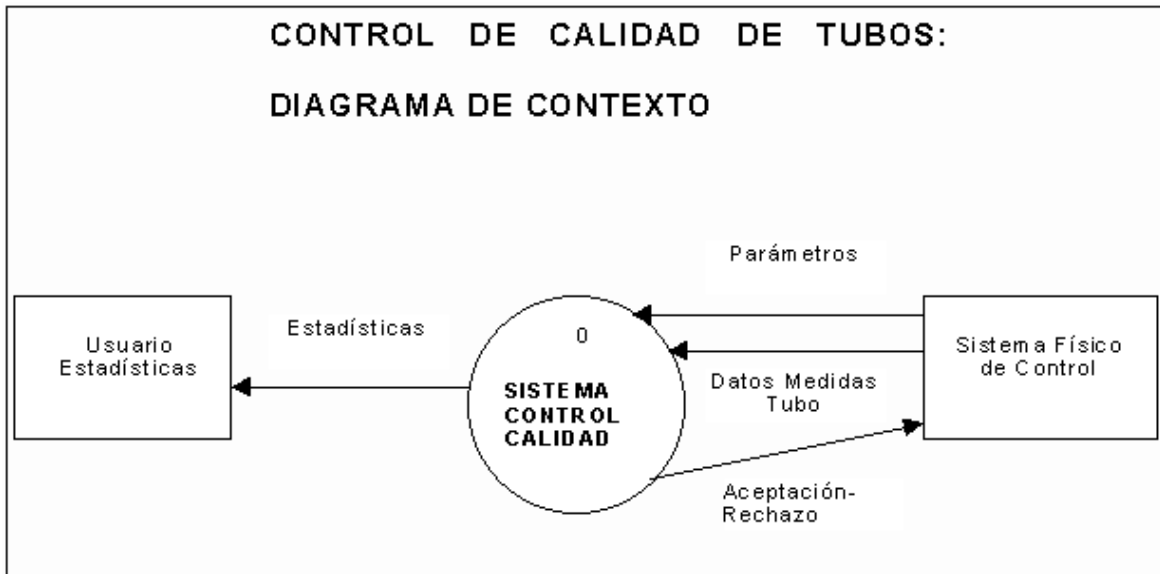
- El análisis estructurado de la aplicación informática al completo.
- El diseño, mediante pseudocódigos u ordinogramas, del proceso de toma de datos, aceptación o rechazo de cada tubo y registro de la información correspondiente.

MODELO CONCEPTUAL DE TRATAMIENTOS (MCT).

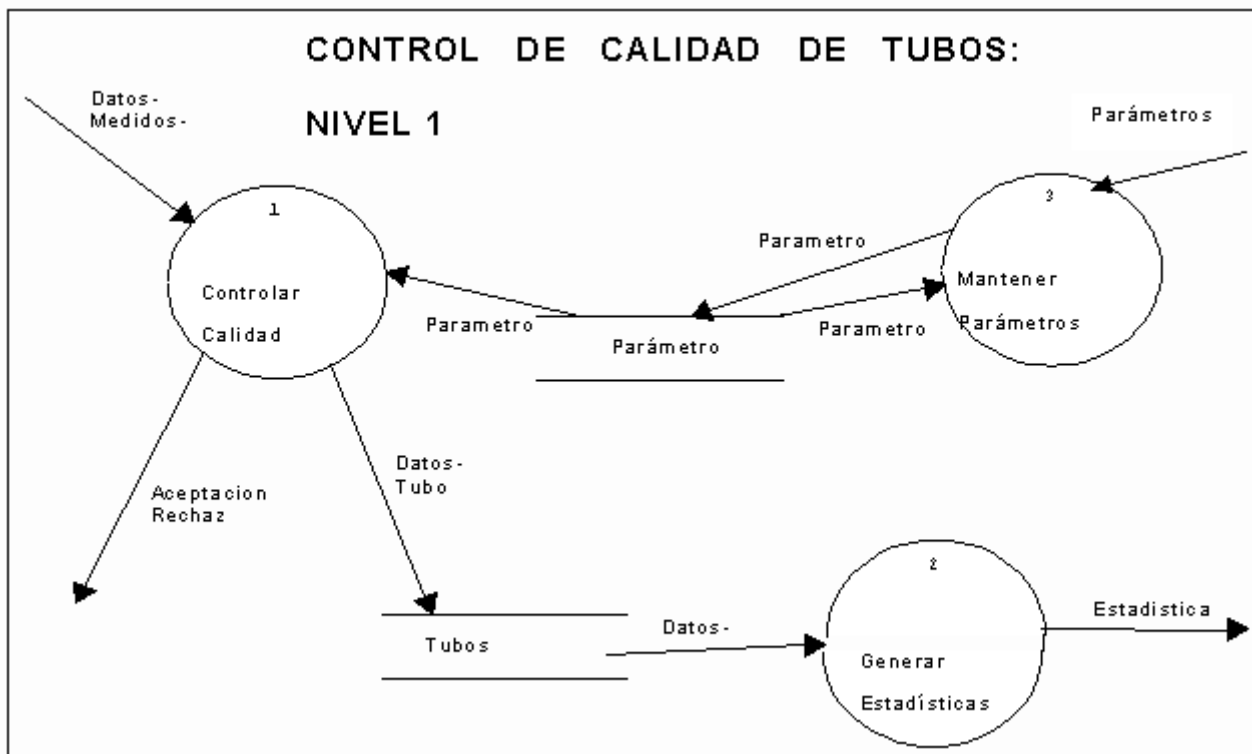
Diagrama de flujo de datos (DFD)

Dentro del nivel conceptual, en el análisis estructurado clásico, el estudio de los tratamientos se concreta en la descripción de los flujos y transformaciones de datos que se dan en el seno del sistema de información (SI). A medida que la información se mueve a través del sistema, es modificada por una serie de transformaciones. Para representar este proceso, se utiliza el denominado diagrama de flujo de datos (DFD). Consiste en una técnica gráfica sencilla, que utiliza sólo cuatro elementos denominados entidad externa, proceso, flujo de datos y almacén de datos, y que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

Seguidamente se ofrece una propuesta de solución al ejercicio planteado. El diagrama de contexto es el siguiente:



De la expansión del proceso del diagrama de contexto surge el diagrama de nivel 1 que se ofrece a continuación:



(Las definiciones y conceptos que se incluyen en estas páginas están extraídas del temario ABACUS NT. No obstante se advierte que han sido resumidas y, posiblemente, descontextualizadas.)

MODELO CONCEPTUAL DE DATOS (MCD)

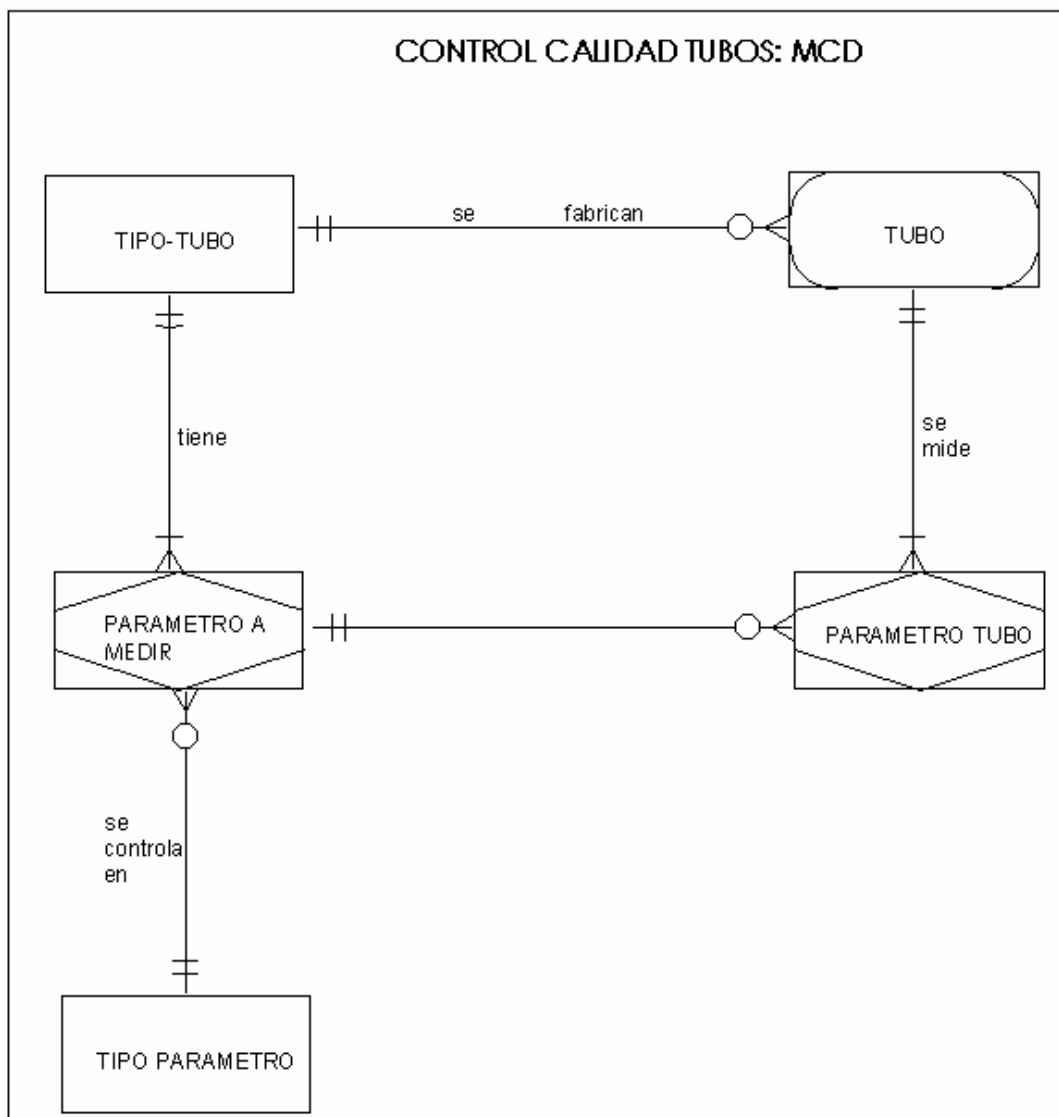
Concepto

Es posible definir el término "modelo de datos" como un conjunto de conceptos, reglas y convenciones que permiten describir los datos del universo del discurso con el fin de coadyuvar a su correcta interpretación y permitir su representación en forma de datos en el sistema de información de la organización.

Modelado de entidad-interrelación (E/R)

El modelo entidad/interrelación (E/R) es, sin género de dudas, el modelo de datos conceptual más extendido en las metodologías de diseño de bases de datos y en las herramientas CASE. Fue propuesto por Peter P. Chen en dos artículos: "The entity/relationship model: toward a unified view of data" publicado en un comunicado de la ACM en 1976 y "The entity/relationship model: a basis for the enterprise view of data" presentado en una conferencia en 1977. Según Chen, "el modelo E/R puede ser usado como una base para una vista unificada de los datos", adoptando "el enfoque más natural del mundo real que consiste en entidades e interrelaciones".

Seguidamente se continúa con la propuesta de solución al ejercicio planteado. El diagrama de E/R, resuelto según el simbolismo gráfico de Martin (que en realidad es una mezcla de los niveles conceptual y lógico de datos), es el siguiente:



(Las definiciones y conceptos que se incluyen en estas páginas están extraídas del temario ABACUS NT. No obstante se advierte que han sido resumidas y, posiblemente, descontextualizadas.)

MODELO LÓGICO DE DATOS (MLD)

Concepto

El diseño lógico estándar parte del esquema conceptual resultante de la etapa de diseño conceptual de datos y, teniendo en cuenta los requisitos de usuario y los imperativos económicos y tecnológicos, lleva a la construcción de un esquema lógico estándar.

Esquema lógico estándar relacional

El esquema lógico estándar ha de describirse empleando preferentemente un lenguaje de definición de datos (DDL, Data Definition Language) que se corresponderá con la metodología seleccionada, metodología que hoy por hoy será, casi con seguridad, el modelado relacional de datos.

Sin embargo, a modo de resumen, es posible presentar el diseño de tablas y sus restricciones de clave principal y ajena utilizando un sencillo sistema que ofrecemos seguidamente:

Tipo-Tubo (SerieRaiz [p], OtrosDatosTipoTubo)

Tipo Parametro (IdParametro [p], DescripcionParametro)

Tubo ((SerieRaiz[e], NumTubo)[p], AceptadoSiNo) @ Es débil de Tipo-Tubo

Parametro A Medir ((SerieRaiz[e], IdParametro[e])[p], MinimoAceptable, MaximoAceptable)

Parametro Tubo → Presenta una estructura compleja por lo que la vamos a dividir en atributos y claves:

- Atributos: (SerieRaiz, IdParametro, NumTubo, Valor, DentroDeRangoSiNo)
- Clave Principal: (SerieRaiz, IdParametro, NumTubo)
- Clave Ajena 1: (SerieRaiz, IdParametro) → Apunta a la entidad Parametro A Medir
- Clave Ajena 2: (SerieRaiz, NumTubo) → Apunta a la entidad Tubo

[p] → clave principal

[e] → clave ajena (extranjera)

DISEÑO MEDIANTE PSEUDOCÓDIGO ESTRUCTURADO CLÁSICO

Seguidamente se incluye un pseudocódigo estructurado clásico, que emplea archivos planos, y cuya sintaxis está fuertemente inspirada en Pascal. No obstante, se podía perfectamente haber empleado un pseudocódigo más parecido a C/Java.

```
PROGRAM Tubo; {Diseño control de calidad en una empresa}
```

```
TYPE
```

```
TCodigoS = string[6];  
TCodigoR = real;
```

```
{En todos los diseños de registros se ha obviado la inclusión del campo de  
otros datos -OtrosDatosTipoTubo ...-}
```

```
RTipoTubo= RECORD
```

```
    SerieRaiz :TCodigoS;  
    DescripcionTipoTubo :string[40];
```

```
END;
```

(Las definiciones y conceptos que se incluyen en estas páginas están extraídas del temario ABACUS NT. No obstante se advierte que han sido resumidas y, posiblemente, descontextualizadas.)

RTipoParametro = RECORD

IdParametro :TCodigoS;
DescripcionParametro :string[40];

END;

RTubo = RECORD

SerieRaiz :TCodigoS;
NumTubo :TCodigoR;
AceptadoSiNo :Boolean;

END;

RParametroAMedir = RECORD

SerieRaiz :TCodigoS;
IdParametro :TCodigoS;
MinimoAceptable :real;
MaximoAceptable :real;

END;

RParametroTubo = RECORD

SerieRaiz :TCodigoS;
NumTubo :TCodigoR;
IdParametro :TCodigoS;
Valor :real;
DentroDeRangoSiNo :boolean;

END;

FTipoTubo = FILE OF RTipoTubo;

FTipoParametro = FILE OF RTipoParametro;

FParametroAMedir = FILE OF RParametroAMedir;

FParametroTubo = FILE OF RParametroTubo;

FTubo = FILE OF RTubo;

{El siguiente procedimiento consiste en una toma de los datos de cada tubo producido, suponemos que los tipos de tubos, parámetros etc. ya se han introducido por lo que el único dato relevante es el valor de cada parámetro para un tubo determinado.}

PROCEDURE TomaDatos;

VAR

RTiTu :RTipoTubo;
RTiPa :RTipoParametro;
RPaAM :RParametroAMedir;
RPaTu :RParametroTubo;
RTubo :RTubo;
Serie :TCodigoS;
Numero :TCodigoR;
AceptadoSiNo :boolean;
Valor :real;

(Las definiciones y conceptos que se incluyen en estas páginas están extraídas del temario ABACUS NT. No obstante se advierte que han sido resumidas y, posiblemente, descontextualizadas.)

{Para simplificar se ha obviado el uso de rutinas de confirmación de datos introducidos por el usuario}

BEGIN

REPEAT

{Se supone que la fabricación es por series de tubos}

REPEAT

```
WriteLn('Introduzca la serie de producción');
ReadLn(Serie);
IF NOT CondEsc THEN
```

```
IF NOT BuscarEnIndice(PaAM,1,Serie,RPaAM) THEN {1er parámetro de la serie}
```

```
WriteLn('Serie no válida, escoja otra')
```

ELSE BEGIN

```
BuscarEnIndice(TiTu,1,Serie,RTiTu);
ClrScr;
WriteLn('Proceso de toma de datos para el tubo tipo ',
RTiTu.DescripcionTipoTubo);
```

END;

UNTIL CondEsc OR NOT(EOF(PaAM));

IF NOT (CondEsc) THEN

REPEAT {Tubos de una serie}

REPEAT

```
WriteLn('Introduzca el nº del tubo');
ReadLn(Numero);
BuscarEnIndice (Tubo,1,(Serie+Numero),RTubo);
IF NOT(EOF(Tubo)) THEN WriteLn('Error tubo existente, use otro código.');
```

UNTIL CondEsc OR EOF(Tubo);

IF NOT CondEsc THEN BEGIN

```
AceptadoSiNo:=true;
IniciarTransaccion();
REPEAT {Parámetros de un tubo}
```

```
BuscarEnIndice (TiPa,1,RPaAM.IdParametro,RTiPa);
```

```
WriteLn('Introduzca el valor de ',RTiPa.DescripcionParametro);
```

```
ReadLn(Valor);
```

```
RPaTu.SerieRaiz:=Serie;
```

```
RPaTu.NumTubo:=Numero;
```

```
RPaTu.IdParametro:=RPaAM.IdParametro;
```

```
RPaTu.DentroDeRangoSiNo:=((Valor>=RPaAM.MinimoAceptable) AND
(Valor<=RPaAM.MaximoAceptable));
```

```
RPaTu.Valor:=Valor;
```

```
NuevoReg(PaTu);
```

```
Write(PaTu,RPaTu);
```

```
AceptadoSiNo:=AceptadoSiNo AND RPaTu.DentroDeRangoSiNo;
```

```
AvanzarReg(PaAM,RPaAM,1);{siguiente parámetro del tipo de tubo}
```

(Las definiciones y conceptos que se incluyen en estas páginas están extraídas del temario ABACUS NT. No obstante se advierte que han sido resumidas y, posiblemente, descontextualizadas.)

```
UNTIL EOF(PaAM) OR RPaAM.SerieRaiz<>Serie;
NuevoReg(Tubo);
RTubo.SerieRaiz:=Serie;
RTubo.NumTubo:=Numero;
RTubo.AceptadoSiNo:=AceptadoSiNo;
Write(Tubo,RTubo);
Commit();
IF NOT(AceptadoSiNo) THEN BEGIN

    WriteLn('Atención el tubo ',SerieRaiz,Numero,' no ha sido aceptado,');
    WriteLn('proceda pues a retirarlo. ');
    WriteLn('Para continuar pulse una tecla. ');
    ReadLn;

END;
{Preparar los parámetros a medir para el siguiente tubo}
BuscarEnIndice(PaAM,1,Serie,RPaAM); {de nuevo 1er parámetro de la serie}
```

```
END; {IF NOT CondEsc}
```

```
UNTIL CondEsc;
```

```
UNTIL CondEsc;
END; {PROCEDURE TomaDatos}
```

```
VAR
```

```
TiTu :FTipoTubo;
TiPa :FTipoParametro;
PaAM :FParametroAMedir;
PaTu :FParametroTubo;
Tubo :FTubo;
Op :integer;
```

```
Assign(TiTu, 'TIPO TU') (*<Indizado Por SerieRaiz>*);
Assign(TiPa, 'TIPO PA') (*<Indizado Por IdParametro>*);
Assign(Tubo, 'TUBOS') (*<Indizado Por SerieRaiz+NumTubo>*);
Assign(PaAM, 'PARAM') (*<Indizado Por SerieRaiz+IdParametro>*);
Assign(PaTu, 'PARATU') (*<Indizado Por SerieRaiz+NumTubo+IdParametro>*);
Reset(TiTu); Reset(TiPa); Reset(Tubo); Reset(PaAM); Reset(PaTu);
ClrScr;
Visualizar(Menu,Principal);
REPEAT
```

```
    ReadLn(Op);
    IF NOT CondEsc THEN
```

```
        CASE Op OF
```

```
            1:TomaDatos;
            2:
            ELSE Error('Opción incorrecta');
```

```
        END;{CASE}
```

```
UNTIL (Op=2) OR CondEsc;
Close(TiTu); Close(TiPa); Close(Tubo); Close(PaAM); Close(PaTu);
END.
```

(Las definiciones y conceptos que se incluyen en estas páginas están extraídas del temario ABACUS NT. No obstante se advierte que han sido resumidas y, posiblemente, descontextualizadas.)