

# 1.10.1 STEP

## El **formato neutro más moderno y completo** está definido en la norma ISO 10303

Se denomina **STEP**, por el acrónimo de “*Standard for the Exchange of Product model data*”

### Sus principales características son:

- √ Es una iniciativa a **largo plazo** Surge en 1984, y el núcleo básico no se materializa hasta 1994
- √ Debe permitir una representación **completa** de toda la información asociada al producto
- √ Debe describir los datos asociados al producto a lo largo de su **ciclo de vida**
- √ Debe ser **extensible**, para facilitar ampliaciones futuras
- √ Debe ser **verificable**, por lo que incluye métodos de comprobación

En realidad, STEP fue diseñado para tres tareas:

1 **Intercambio** neutro de datos de productos:

- ✓ Cada aplicación tiene su propia copia de los datos del producto en su propia forma preferida (datos descentralizados)
- ✓ Los datos en formato STEP muestran un estado fijo (una “foto fija”) del modelo que se intercambia

2 **Compartición** de datos de productos:

- ✓ Varias aplicaciones acceden y operan (incluso simultáneamente) con una sola copia “maestra” de los mismos datos de producto, la cual cambia interactivamente
- ✓ Las aplicaciones no guardan los datos en sus propios formatos preferidos
- ✓ Los datos de producto en formato STEP son los de producto integrados (globales) y no las partes que utilizan las aplicaciones particulares de datos de productos

3 **Archivo** de datos de productos (generalmente a **largo plazo**):

- ✓ Archivar requiere que los datos que cumplan con STEP para fines de intercambio se guarden de modo que puedan ser recuperados en el futuro
- ✓ Este uso posterior puede ser de intercambiar o de compartir datos de productos

Para lograr los *objetivos deseados*, STEP se apoya en dos rasgos principales:

Intercambiar, compartir y archivar, manteniendo la integridad de la interpretación

## ✓ Es complejo

- ✗ Usa una sintaxis formal, que requiere conocimientos de programación
- ✗ Incluye conceptos de generalización y restricción, y está orientado a objetos
- ✗ Requiere pruebas de conformidad

## ✓ Es legible

- ✓ Tiene un lenguaje de descripción
- ✓ El formato está en texto plano

SCHEMA ejemplo;

```
ENTITY vehiculo  
SUPERTYPE OF (ONEOF (coche, camion));  
n_de_ruedas : INTEGER;  
END_ENTITY;
```

```
ENTITY coche  
SUBTYPE OF (vehiculo);  
modelo : STRING;  
END_ENTITY;
```

```
ENTITY camion  
SUBTYPE OF (vehiculo);  
carga_maxima : REAL;  
END_ENTITY;
```

```
END_SCHEMA;
```

En consecuencia:

- ✓ Tiene una **estructura modular**
- ✓ Tiene su propio **lenguaje de desarrollo (Express)**
- ✓ Permite **ficheros “físicos” de texto plano**

## STEP tiene estructura modular, porque:

- ✓ STEP es en realidad un conjunto de estándares internacionales contruidos alrededor de una arquitectura de **partes** publicadas por separado, y desarrolladas en diferentes tiempos

Las partes que están en desarrollo se etiquetan como ISO/**PRF** 10303, ISO/**PAS** 10303 o ISO/**TS** 10303

PRF significa "Proof"

PAS significa "Publicly Available Specification"

TS significa "Technical Specification"

- ✓ La modularidad permite que cada organización desarrolle solo los **protocolos** que satisfacen sus requisitos de operación

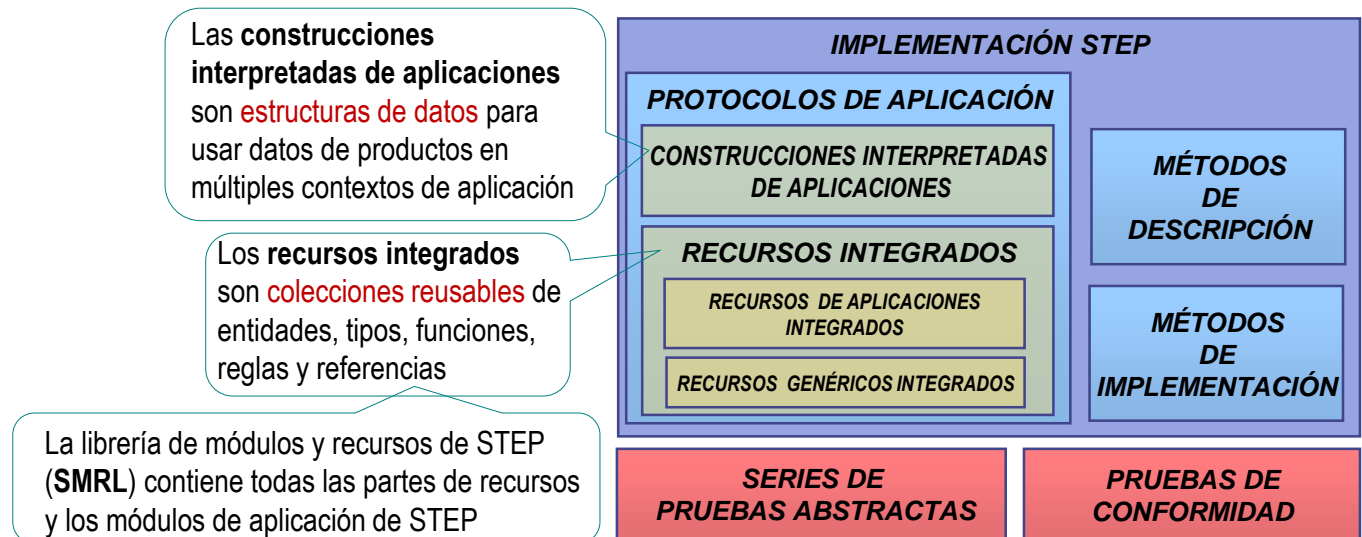
Se adoptó la estructura modular al asumir que toda la norma es demasiado extensa para que ninguna organización tenga interés en implementarla completa

# Modularidad: partes

STEP se organiza por **partes**, agrupadas en **contextos**:

- ✓ Los **métodos de descripción** proveen un **lenguaje formal** de modelado de datos
- ✓ Los **métodos de implementación** especifican los **ficheros físicos** que permiten el intercambio de información
- ✓ Las **pruebas abstractas y de conformidad** son **procedimientos para evaluar** las implementaciones
- ✓ Los **protocolos de aplicación** agrupan y organizan las **especificaciones implementables** de datos del producto en un campo específico de aplicación:

Las partes de éstos tres contextos constituyen la infraestructura



Introducción

Modularidad

Partes

Protocolos

Express

Fichero físico

Conclusiones



La numeración de las **partes** de STEP se corresponde con sus diferentes contenidos:

- ✓ La parte **1** contiene un resumen general
- ✓ Las partes **1x** contienen los métodos de descripción (lenguaje EXPRESS)
- ✓ Las partes **2x** describen los métodos de implementación (ficheros físicos)
- ✓ Las partes **3x** contienen la metodología y marco de las pruebas de conformidad (CC)
- ✓ Las partes **4x**, **5x** y **6x** contienen los recursos genéricos integrados (IGR)
- ✓ Las partes **1xx** contienen los recursos de aplicación integrados (IAR)
- ✓ Las partes **2xx** contienen los protocolos de aplicación (AP)
- ✓ Las partes **3xx** contienen las series de pruebas abstractas (ATS) para las AP
- ✓ Las partes **4xx** contienen los módulos de aplicación (AM) de alto nivel para las AP
- ✓ Las partes **5xx** contienen las construcciones interpretadas de aplicación (AIC)
- ✓ Las partes **1xxx** contienen los módulos de aplicación (AM) de menor nivel que las partes 4xx
- ✓ Las partes **3xxx** contienen los modelos de objeto de negocio (BO\_M)
- ✓ Las partes **4xxx** contienen los modelos específicos de dominio (DM)

La lista completa se puede consultar aquí:

[https://standards.iso.org/iso/10303/STEP\\_Parts\\_List.htm](https://standards.iso.org/iso/10303/STEP_Parts_List.htm)

# Modularidad: partes

En principio, las diferentes partes se ordenan jerárquicamente:

- ✓ Solo las partes de **nivel alto** cubren completamente un dominio o área

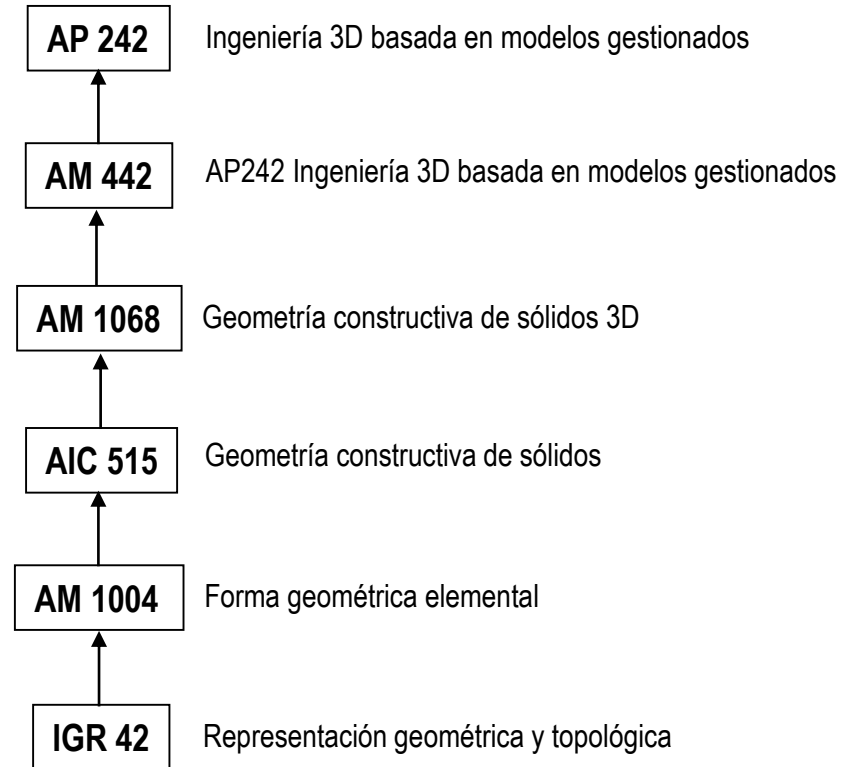
Por ejemplo, la parte AP 242 que abarca todo el diseño tridimensional de piezas y ensamblajes

- ✓ Las partes de **nivel medio** vinculan partes de nivel inferior entre sí y las especializan

Por ejemplo, la parte AP 203 se construye a partir de la parte AM 403 y se especializa en modelos CAD para la industria aeronáutica

- ✓ Las partes del **nivel bajo** son envoltorios de conceptos básicos que se usan repetidamente

Se definen en Construcciones Interpretadas de Aplicaciones (AIC), o Recursos Genéricos Integrados (IGR)



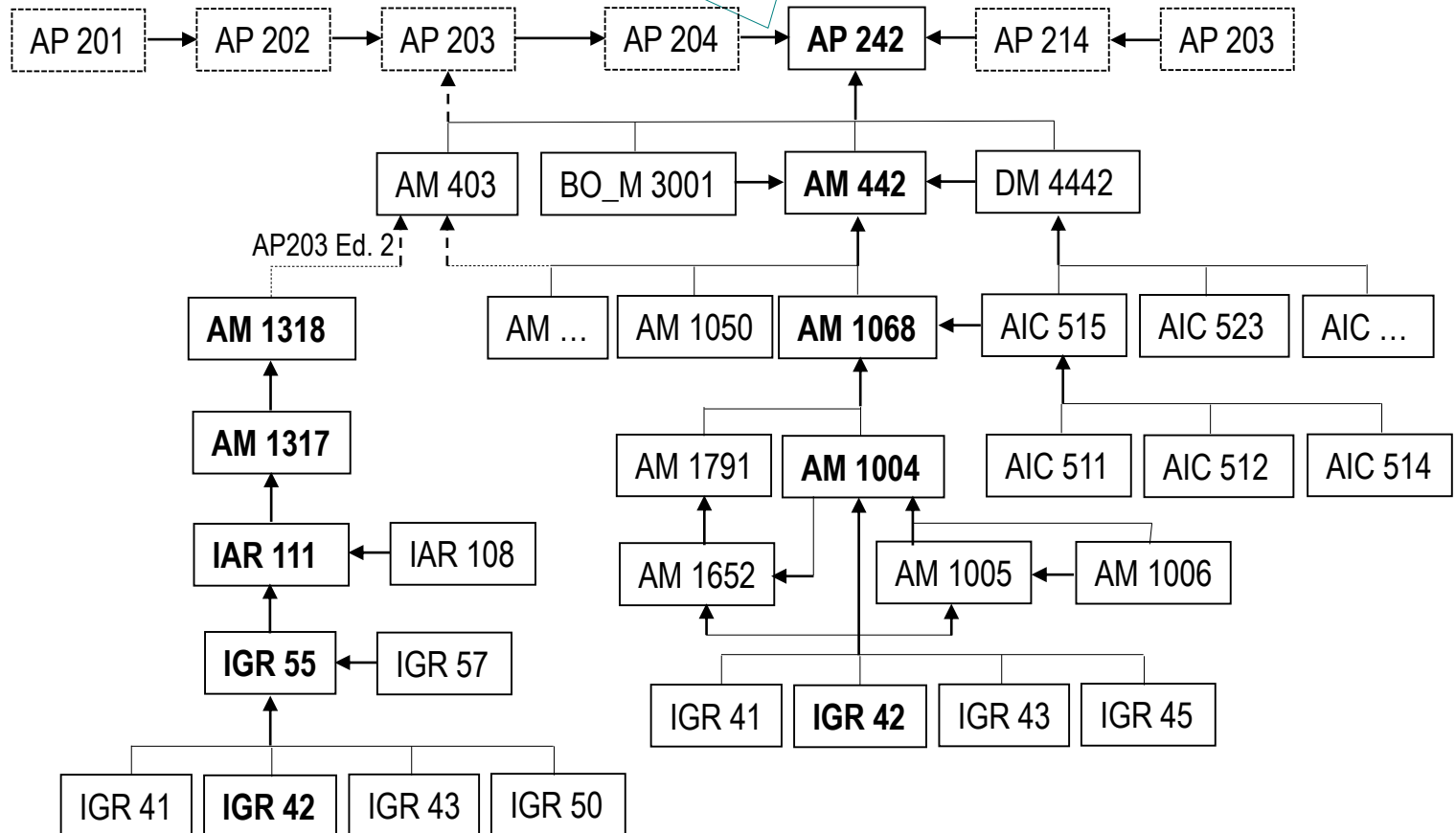


# Modularidad: partes



En realidad, las relaciones entre las diferentes partes no son estrictamente jerárquicas, dado que hay muchas dependencias mutuas, generalmente complejas:

Además, las partes de nivel superior más modernas, absorben a algunas partes de nivel superior anteriores

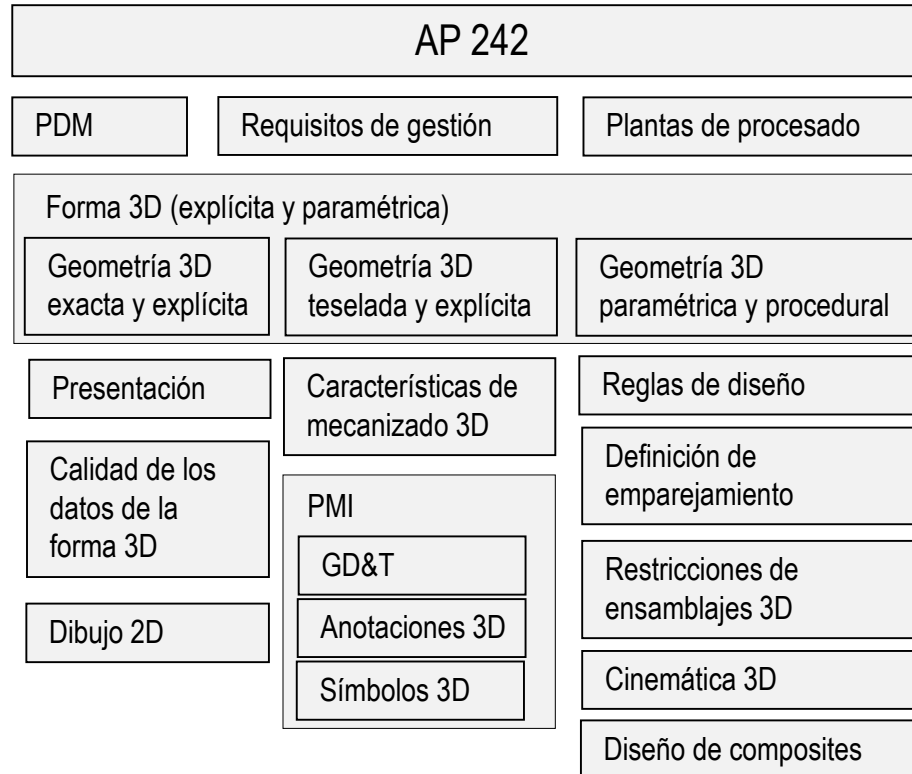


- Introducción
- Modularidad
- Partes
- Protocolos
- Express
- Fichero físico
- Conclusiones

# Modularidad: protocolos

Los protocolos de aplicación (AP) son las partes de mayor nivel

Algunos AP también son transversales, porque abarcan muchos aspectos



Los AP se subdividen, para permitir implementaciones parciales que sean útiles en ciertos campos de actividad

Introducción

**Modularidad**

Partes

**Protocolos**

Express

Fichero físico

Conclusiones

# Modularidad: protocolos

Introducción

**Modularidad**

Partes

**Protocolos**

Express

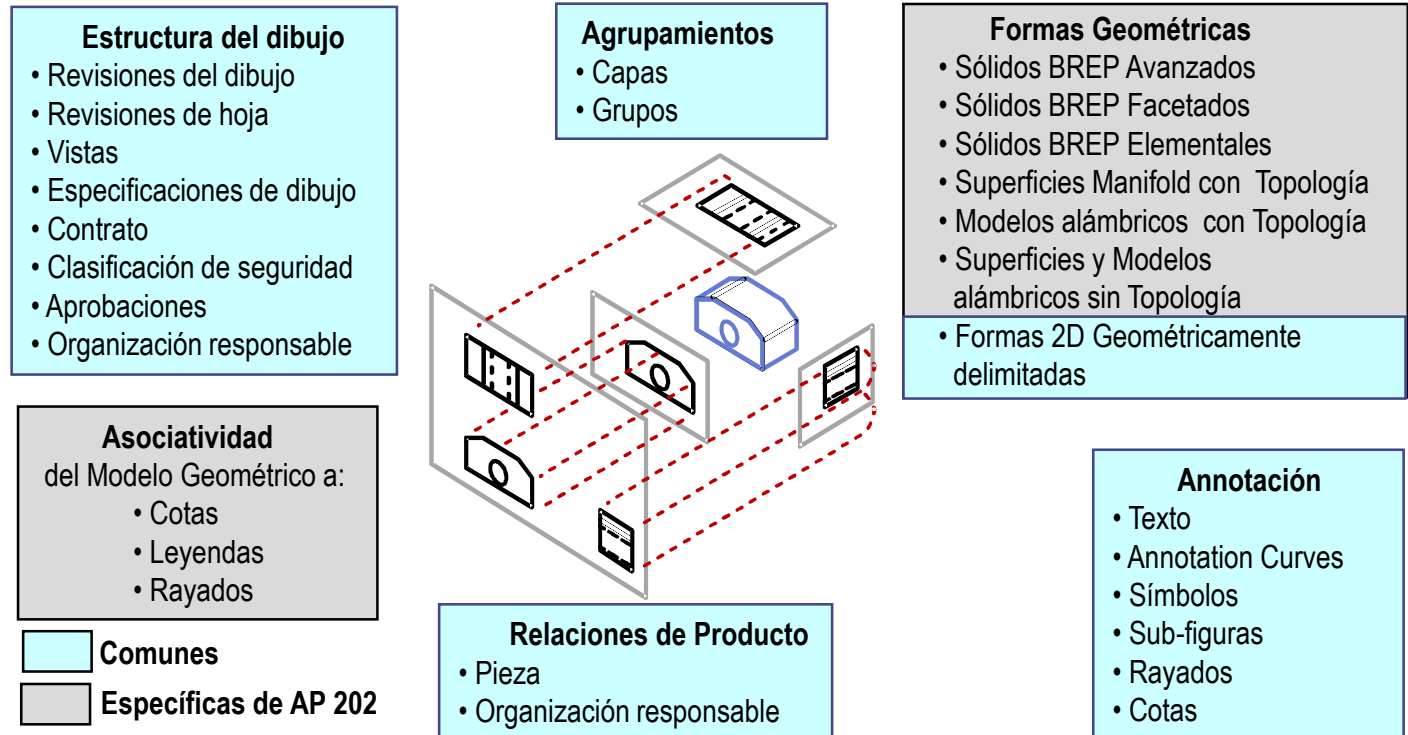
Fichero físico

Conclusiones



Un ejemplo de modularidad son los dos protocolos de aplicación para dibujos técnicos, que se diferencian porque distinguen entre dibujos explícitos y asociativos:

- ✓ AP 201 Explicit Draughting
- ✓ AP 202 Associative Draughting



# Modularidad: protocolos

Introducción

Modularidad

Partes

Protocolos

Express

Fichero físico

Conclusiones

La documentación de los AP se organiza por **capas**:

- 1 AAM (Application Activity Model) es la capa informativa y gráfica, que define las actividades y muestra los **flujos de información entre actividades** del AP mediante esquemas IDEF0

Contexto y alcance, según ISO-10303-1:1994

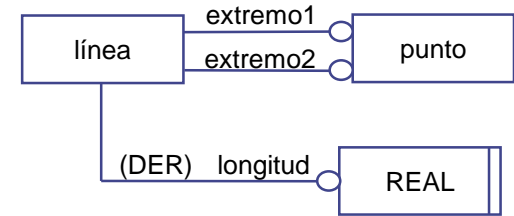
- 2 ARM (Application Reference Model) es la capa normativa y gráfica, que describe el **dominio de información** que necesita un AP mediante esquemas EXPRESS-G

Requisitos de información

- 3 AIM / MIM (Application/Module Interpreted Model) es la capa normativa y textual, que describe **la información implementable** mediante EXPRESS

Requisitos funcionales

- 4 BO Model (Business Object Model) simplifica la terminología, y facilita la implementación del AP mediante XML



```
SCHEMA ejemplo_línea;  
  ENTITY punto;  
    x : REAL;  
    y : REAL;  
  END_ENTITY;  
  ENTITY línea;  
    extremo1 : punto;  
    extremo2 : punto;  
  END_ENTITY;  
END_SCHEMA;
```

Las capas definen la semántica de la información a intercambiar, y la correspondencia de esta semántica con las estructuras de datos para representar la información

# Modularidad: protocolos

Introducción

**Modularidad**

Partes

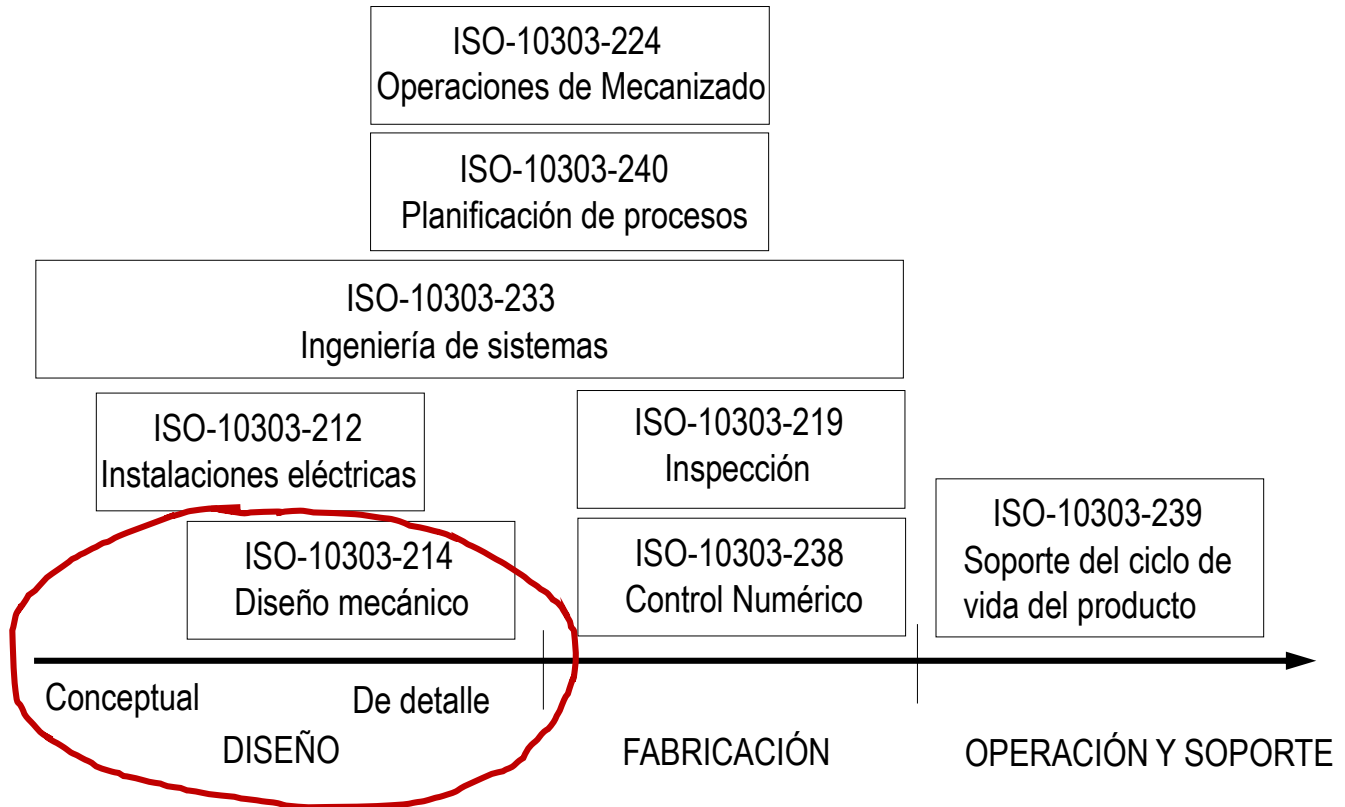
**Protocolos**

Express

Fichero físico

Conclusiones

Los protocolos de aplicación (AP) de STEP abarcan diferentes etapas del **ciclo de vida** de los productos:



En ésta lección estamos interesados en los protocolos de aplicación de modelado para la **etapa de diseño**

# Modularidad: protocolos

Introducción

**Modularidad**

Partes

**Protocolos**

Express

Fichero físico

Conclusiones

Para el campo de aplicación específico del diseño mecánico mediante modelos CAD 3D, actualmente conviven implementaciones comerciales de tres **protocolos de representación de modelos**, orientados inicialmente a contextos diferentes:

- ✓ AP 203, Configuration controlled 3D designs of mechanical parts and assemblies

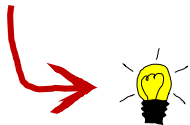
Retirado por ISO en 2014

- ✓ AP 214, Application protocol: Core data for automotive mechanical design processes

Retirado por ISO en 2014

- ✓ AP 242, Managed model based 3D engineering

Cada uno de ellos engloba y supera al anterior



Por tanto, la recomendación obvia sería **utilizar el más moderno**



Siempre que esté disponible tanto para el emisor del formato neutro, como para el receptor de dicho formato

# Modularidad: protocolos

Introducción

**Modularidad**

Partes

**Protocolos**

Express

Fichero físico

Conclusiones



Hay que tener presente que hay diferentes niveles de implementación, o **clases de conformidad** (cc) asociadas con cada AP

Las clases de conformidad son subconjuntos de un AP que se puede implementar "de manera significativa" dentro de ese dominio de aplicación sin tener que implementar todos aspectos del AP

Los usuarios deben saber qué clases de conformidad de un AP está implementada en cada aplicación CAD

No es suficiente indicar que un proveedor tiene un traductor STEP o AP 2xx ...

...porque la mayoría de los proveedores se limitan a implementar un subconjunto

## Ayuda de **SOLIDWORKS PDM**

### Opciones de conversión avanzadas (STEP)

#### Versión

##### **AP203**

El estándar STEP AP203 no dispone de implementación de colores.

##### **AP214**

Exporta colores de sólido, cara y curva. Se soportan las clases de conformidad 1, 4 y 6.

# Modularidad: protocolos

Introducción

**Modularidad**

Partes

**Protocolos**

Express

Fichero físico

Conclusiones



Las clases de conformidad (CC) especifican varios grupos seleccionados de entidades (diferentes subconjuntos del contenido total de la AP) que el software debe implementar completamente

Que una implementación sea conforme a una CC específica significa que la implementación debe admitir **todas** las entidades agrupadas dentro de esa CC

Si un vendedor afirma que su producto cumple con una CC, tiene que cumplir con **toda** esa clase de conformidad

Por ejemplo, las clases de conformidad de AP 203 son:

- ✓ CC1: información de diseño controlada por configuración, sin forma
- ✓ CC2: Clase 1 y formas representadas por modelos de estructura alámbrica delimitados geoméricamente, modelos de superficie o ambos
- ✓ CC3: Clase 1 y formas representadas por modelos de estructura alámbrica con topología
- ✓ CC4: Clase 1 y formas representadas por modelos de superficie múltiple con topología
- ✓ CC5: Clase 1 y formas representadas por representación de límite facetado
- ✓ CC6: Clase 1 y formas representadas por representación de límite avanzada (es decir, no facetada)



# Modularidad: protocolos

Introducción

**Modularidad**

Partes

**Protocolos**

Express

Fichero físico

Conclusiones

Las características principales de los protocolos para modelos 3D comúnmente implementados como formatos neutros en las aplicaciones CAD 3D comerciales son:

√ AP203 es un formato STEP de propósito general:

Aunque surgió como iniciativa de la industria Aeroespacial

- √ Define la geometría, la topología y los datos de gestión de la configuración de modelos sólidos para piezas mecánicas y conjuntos
- √ No administra colores ni capas
- × Existe una segunda edición, que soporta modelos procedurales e híbridos, pero no hay implementaciones comerciales

√ AP214 es una extensión de AP203, que agrega:

Surgió como iniciativa de la industria Automovilística

- √ Colores y capas
- √ Dimensionamiento geométrico y tolerancia
- √ Intención de diseño

# Modularidad: protocolos

Introducción

Modularidad

Partes

Protocolos

Express

Fichero físico

Conclusiones

- ✓ El protocolo AP 242 es el más moderno, engloba a los dos anteriores y contempla todo lo que faltaba:

1 Secciones paramétricas 2D

geometría explícita soportando diferentes tipos de restricciones geométricas, numéricas y algebraicas

2 Información de soporte al modelado basado en la historia (history based modeling)

Secuencia de las operaciones de modelado usadas para la construcción de un modelo, que requieren operadores para crear, interrogar y modificar entidades geométricas

3 Ensamblajes paramétricos 3D

Incluyendo restricciones paramétricas de ensamblaje entre las piezas del conjunto (Ver Tema 2)

4 Modelos anotados

Incluye anotaciones (Ver Tema 4)



Pero **no** hay implementaciones operativas de todo el protocolo AP 242

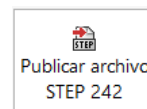
Probablemente porque solo tiene una clase de conformidad, lo que obliga a implementarlo todo



Existen implementaciones que se centran en los modelos anotados

Los modelos STEP con anotaciones solo se importan a e-Drawings, **no** a SolidWorks

Desde 2017, el módulo MBD de SolidWorks implementa un comando para escribir ficheros STEP "242" que produce modelos B-Rep con anotaciones de fabricación (PMI)



Publicar archivo STEP 242

**Publicar archivo STEP 242**  
Crea STEP 242 con datos de PMI.

La parte 11 de STEP (**ISO 10303-11**)  
define un *lenguaje de modelado de datos*  
para datos de producto llamado **EXPRESS**

Los **lenguajes de modelado de datos** son lenguajes artificiales definidos mediante un conjunto consistente de reglas, para transmitir información y conocimiento

ISO 10303 fue pionera en incluir el lenguaje de especificación como parte de la propia norma



EXPRESS provee una metodología formal de modelado de datos, apoyándose en dos **características**:

- ✓ Ayuda a definir formalmente los modelos de datos, de manera neutral respecto a la implementación y el contexto

Separa la estructura física de datos de la semántica de los datos

- ✓ No es un lenguaje de programación, aunque tiene características comunes con ellos:
  - ✓ Tiene influencias de varios lenguajes de programación (Ada, Algol, C ++,...)
  - ✓ Permite comparaciones que incorporan muchas de las operaciones matemáticas y lógicas incluidas en los lenguajes de programación
  - ✓ Admite procedimientos y funciones programadas como parte de la expresión de restricciones, y por tanto en el proceso de evaluación

EXPRESS está diseñado específicamente para representar datos de productos, a través de entidades que tienen propiedades y están sujetas a restricciones

Se basa en la programación orientada a objetos (OOP), en lugar de la programación estructurada

Las unidades fundamentales de EXPRESS son:

- ✓ Esquemas Son contenedores de información relacionada
- ✓ Entidades Definen conceptos del mundo real mediante objetos abstractos
- ✓ Atributos Definen las propiedades de los objetos
- ✓ Tipos Delimitan los rangos válidos para los objetos
- ✓ Reglas Describen las relaciones válidas entre objetos

# Express

Introducción

Modularidad

**Express**

Fichero físico

Conclusiones

El lenguaje es analizable por humanos y por computadoras con el propósito de validación formal

EXPRESS tiene dos representaciones formales posibles:

EXPRESS-G  
es la notación **gráfica**



EXPRESS  
es la notación **textual**



Solo soporta una parte del lenguaje

Pero es apropiado para facilitar una visión global

```
TYPE Día_de_la_semana = ENUMERATION OF
(Lunes
Martes,
Miércoles,
Jueves,
Viernes,
Sábado,
Domingo);
END_TYPE;

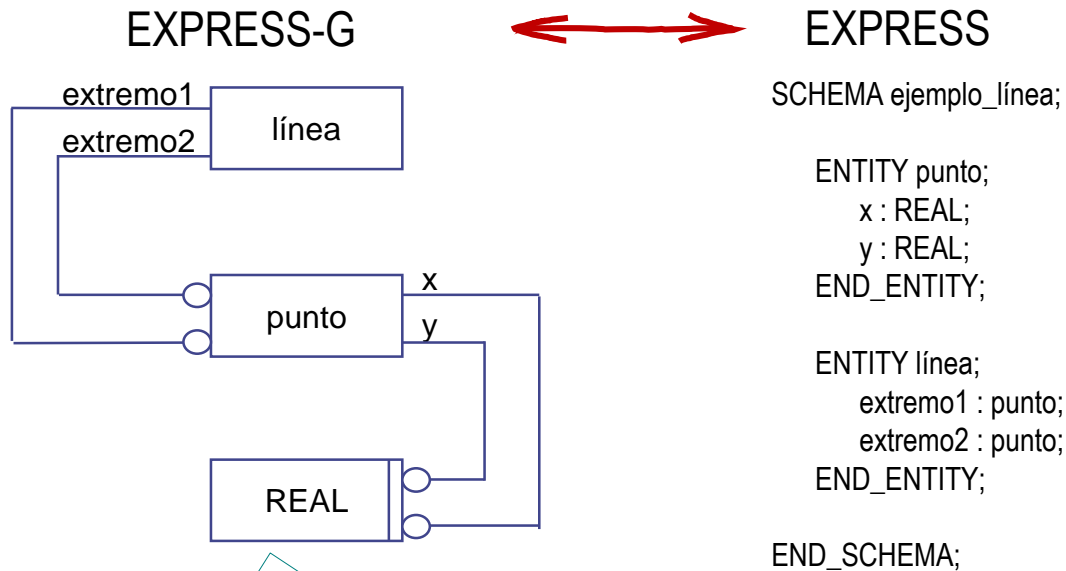
ENTITY Fecha;
    día : Día_de_la_semana;
END_ENTITY;
```

Un ejemplo de descripción *informal* sería:

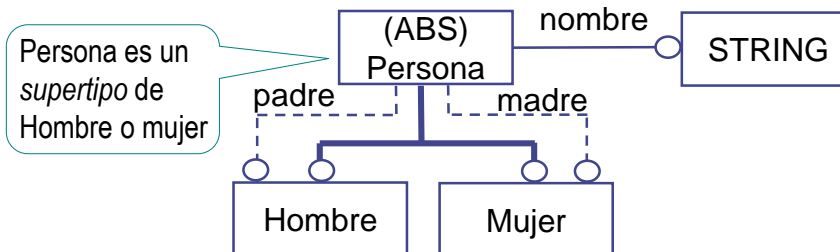
Los días se agrupan en semanas, denominándose respectivamente lunes, martes, miércoles, jueves, viernes, sábado y domingo los días consecutivos que forman cada semana

# Express

Los esquemas sencillos se pueden representar de ambas formas:



Los esquemas se complican cuando se usan tipos relacionados jerárquicamente entre sí:



Introducción

Modularidad

**Express**

Fichero físico

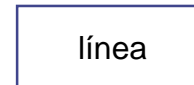
Conclusiones

EXPRESS-G es una representación mediante **representaciones de datos en red**, que muestran las relaciones entre las entidades geométricas que se relacionan para constituir un modelo CAD:

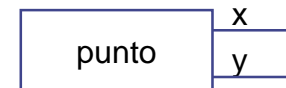


Más información sobre visualización gráfica de datos en lección 3.5

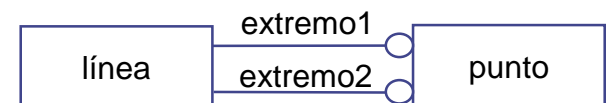
- ✓ Una **entidad** es una clase de objetos
- ✓ Las entidades se representan con una etiqueta encerrada en un rectángulo



- ✓ Los **atributos** son propiedades que tienen los objetos
- ✓ Se usan líneas para conectar las entidades con sus atributos
- ✓ Los nombres de los atributos se escriben con etiquetas encima de las líneas



- ✓ Las **relaciones** expresan dependencia o interacción entre entidades
- ✓ Las relaciones se dibujan mediante líneas orientadas que conectan rectángulos de clases
- ✓ El sentido de la orientación se indica mediante un círculo en el extremo de la línea



# Express

Introducción

Modularidad

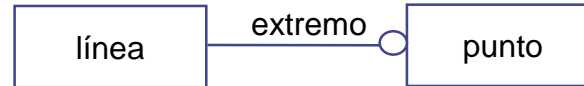
**Express**

Fichero físico

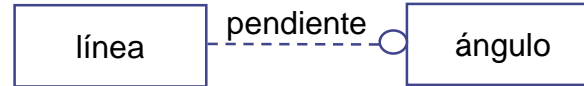
Conclusiones

✓ Los diferentes **tipos de relaciones** se representan mediante diferentes tipos de líneas y etiquetas:

✓ Relación **obligatoria**, con línea fina continua



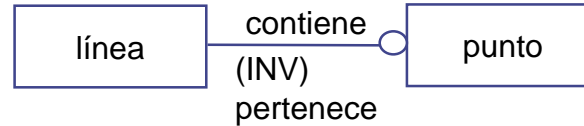
✓ Relación **optativa**, con línea fina a trazos



✓ Relación **jerárquica**, con línea gruesa



✓ Relación **inversa**, con etiqueta doble



✓ Relación **múltiple**, con etiquetas de:

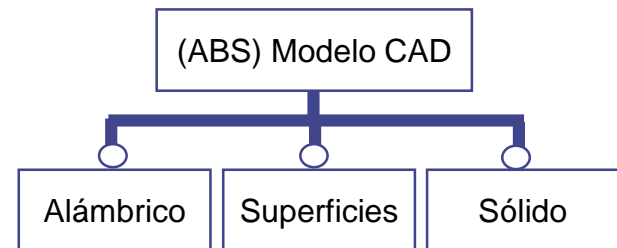


ENTITY polilínea;  
líneas : LIST [1:?] OF línea;  
END\_ENTITY;

- ✓ Conjuntos sin orden ni repeticiones (SET)
- ✓ Listas con orden y sin repeticiones (LIST)
- ✓ Conjuntos sin orden y con repeticiones (BAG)
- ✓ Matrices con orden y tamaño fijo (ARRAY)

Se indica el rango [mínimo:máximo]

✓ Las entidades son **ABSTRACTAS (ABS)** cuando no pueden existir por ellas mismas, sino que se limitan a ser conceptos generales (**supertipos**) que agrupan otras entidades que sí que existen por si mismas (**subtipos**)

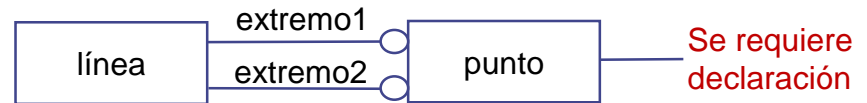




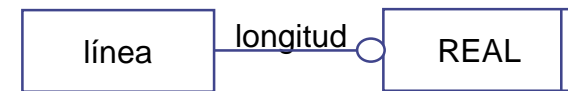
# Express

✓ Hay dos tipos principales de atributos:

✓ Los **compuestos** se tienen que declarar explícitamente



✓ Los tipos de datos **básicos** (o predefinidos) son los componentes elementales del lenguaje, porque no se pueden descomponer en nada más pequeño que ellos:



- ✓ El tipo de datos *booleano* solo admite los valores cierto y falso (equivalentes a 0 o 1)
- ✓ El tipo de datos *binario* es una secuencia de unos y ceros (por ejemplo 10011)
- ✓ El tipo de datos *entero* incluye cualquier número natural (sin decimales)
- ✓ El tipo de datos *lógico* incluye los datos cierto, falso o desconocido
- ✓ El tipo de datos *número* incluye indistintamente números enteros o reales
- ✓ El tipo de datos *real* incluye números con parte decimal
- ✓ El tipo de datos *tira de caracteres* acepta cualquier secuencia de caracteres

BOOLEAN

BYNARY

INTEGER

LOGICAL

NUMBER

REAL

STRING



Al utilizar los tipos de datos numéricos es cuando el lenguaje da lugar a modelos CAD que pueden causar errores de precisión

# Express

Introducción

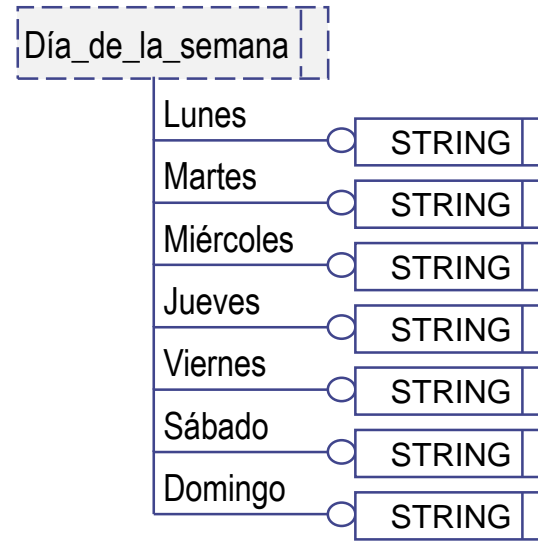
Modularidad

**Express**

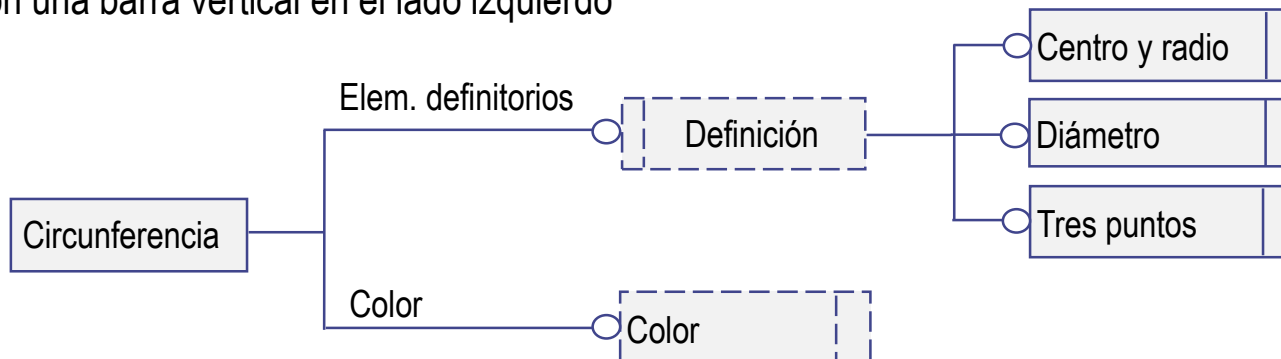
Fichero físico

Conclusiones

- ✓ Las **enumeraciones** detallan el rango de valores (de tira de caracteres) que puede tomar un atributo
- ✓ Las entidades enumeradas se representan con una etiqueta encerrada en un rectángulo de líneas de trazos con una barra vertical en el lado derecho



- ✓ Las **selecciones** permiten elegir entre opciones alternativas
- ✓ El nombre de la selección se representan con una etiqueta encerrada en un rectángulo de líneas de trazos con una barra vertical en el lado izquierdo



# Express

Introducción

Modularidad

**Express**

Fichero físico

Conclusiones

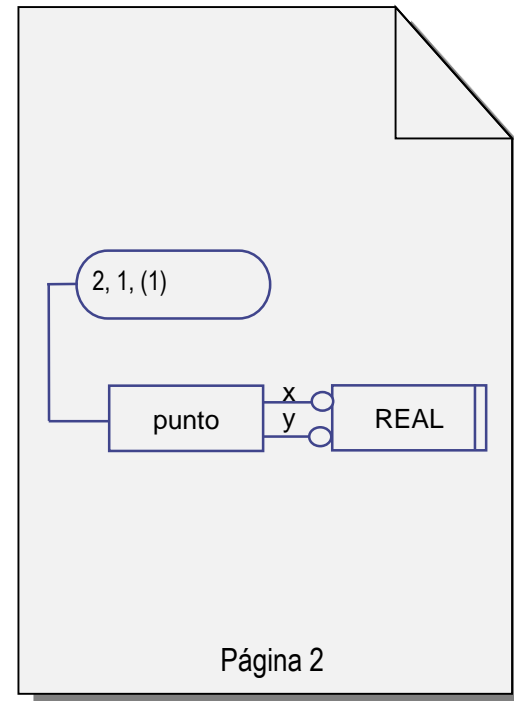
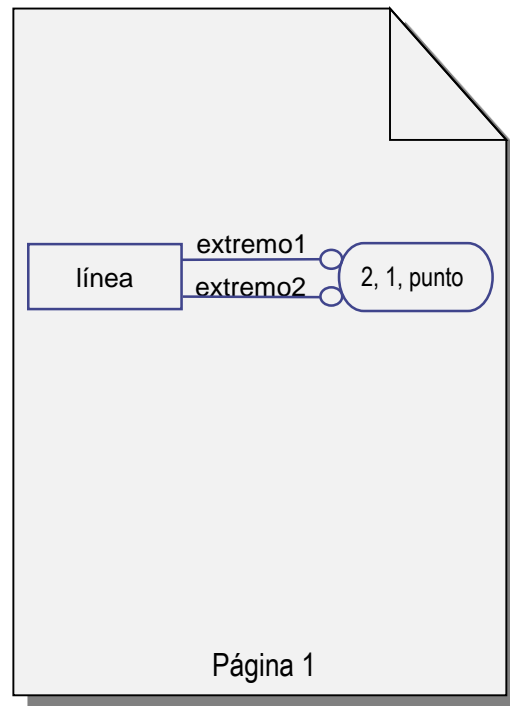
√ Dado que los diagramas pueden abarcar más de una página, las interconexiones (**referencias cruzadas**) se identifican mediante etiquetas encerradas en cajas redondeadas:

√ En la página donde se hace la referencia, la etiqueta de la caja se denomina ONTO ANOTHER PAGE, y contiene el número de página, el número de referencia y el nombre de la entidad a que hace referencia

Página, ref, nombre

√ En la página donde se define la referencia, la etiqueta se denomina ONTO THIS PAGE, y contiene el número de página, el número de referencia y el número de la página donde se usa (entre paréntesis)

Página, ref, (página)



# Express

Introducción

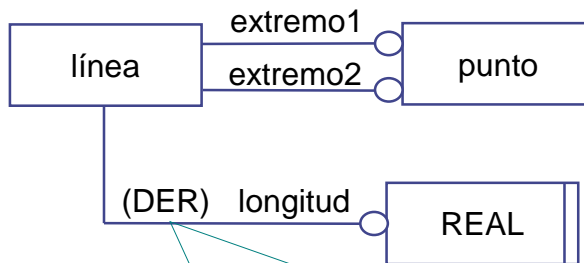
Modularidad

**Express**

Fichero físico

Conclusiones

- ✓ Los diagramas pueden contener prefijos en las etiquetas de los atributos, para indicar que se guían por **reglas**



```
ENTITY línea;  
  extremo1 : punto;  
  extremo2 : punto;  
DERIVE  
  longitud : distance:= SQRT ((extremo1.xcoord – extremo2.xcoord)**2 +  
                               (extremo1.ycoord – extremo2.ycoord)**2);  
END_ENTITY;
```

- ✗ Pero los recursos más sofisticados del lenguaje EXPRESS no se pueden representar mediante diagramas EXPRESS-G

Por ejemplo, el uso de WHERE para definir el rango válido de un atributo

```
TYPE día_del_mes = INTEGER;  
WHERE  
  wr1: ((1<=SELF) AND (SELF<=31));  
END_TYPE;
```

# Fichero físico

Introducción

Modularidad

Express

**Fichero físico**

Conclusiones

La parte 21 de STEP (ISO 10303-21) define los **ficheros de intercambio** mediante un formato de texto

También se llaman **ficheros físicos**

Hay tres hechos que lo caracterizan:

- ✓ Evoluciona a lo largo del tiempo

ISO 10303-21:1994	Primera edición
ISO 10303-21:2002	Segunda edición (soluciona fallos de la primera edición)
ISO 10303-21:2016	Tercera edición (añade funcionalidades)

- ✓ El fichero físico es el **método de implementación** más usado en STEP, porque produce un formato legible

Es legible debido a:

- ✓ Su estructura ASCII
- ✓ Su organización en una instancia por línea

- ✓ Hay otros **métodos de implementación**, pero se usan menos

La parte 28 proporciona una representación de los datos de acuerdo con la sintaxis del lenguaje de marcado extensible (**XML**)

# Fichero físico

El fichero físico tiene una estructura rígida:

- ✓ El fichero debe comenzar con la **palabra clave**:

*ISO-10303-21;*

Todas las instrucciones acaban con punto y coma

- ✓ A continuación, el fichero contiene dos **secciones**:

1 Cabecera

2 Datos

En algunas versiones, esta sección puede ser múltiple

- ✓ Existen palabras clave para indicar el final de las secciones y el final del fichero:

```
ISO-10303-21  
HEADER;  
...  
ENDSEC;  
DATA;  
...  
ENDSEC;  
END-ISO-10303-21;
```

# Fichero físico: cabecera

Introducción

Modularidad

Express

Fichero físico

Cabecera

Datos

Conclusiones

La sección de cabecera también tiene una **estructura fija**

↳ La estructura tiene entre tres y seis **grupos ordenados**:

- 1 Descripción del fichero
- 2 Nombre del fichero
- 3 Esquema del fichero
- 4 Población del fichero
- 5 Idioma de la sección
- 6 Contexto de la sección

Es una instrucción con dos campos:

```
FILE_DESCRIPTION(('Descripción'), 'Nivel');
```

✓ La *descripción* es un texto libre

Suele indicar el protocolo al que se ajusta el fichero (Por ejemplo "STEP AP203")

✓ El *nivel de implementación* incluye dos números:

✓ La versión (1 para la versión inicial, 2 para la versión con correcciones y 3 para la más reciente)

El valor 3 se usa para ficheros con varias secciones de datos, esquemas múltiples y/o soporte de FILE\_POPULATION

✓ La opción de conformidad puede ser 1 para conformidad interna, o 2 para conformidad externa

Mejora la interoperabilidad entre API's, porque minimiza el uso de subtipos complejos

¡No se usa!

# Fichero físico: cabecera

Introducción

Modularidad

Express

Fichero físico

Cabecera

Datos

Conclusiones

La sección de cabecera también tiene una **estructura fija**

↳ La estructura tiene entre tres y seis **grupos ordenados**:

- 1 Descripción del fichero
- 2 Nombre del fichero
- 3 Esquema del fichero
- 4 Población del fichero
- 5 Idioma de la sección
- 6 Contexto de la sección

Es una instrucción con siete campos:

```
FILE_NAME(  
  'Pieza1.STEP',           // Nombre  
  '2017-05-16T14:14:18', // Fecha y hora  
  ('),                    // Autor  
  ('),                    // Empresa  
  'SwSTEP 2.0',           // Software STEP  
  'SolidWorks 2016',     // Software CAD  
  ');                    // Autorización
```

En formato ISO 8601

Identifica el software que ha generado el fichero STEP

Identifica el software que ha generado el modelo CAD



La sección de cabecera también tiene una **estructura fija**

↳ La estructura tiene entre tres y seis **grupos ordenados**:

- 1 Descripción del fichero
- 2 Nombre del fichero
- 3 Esquema del fichero
- 4 Población del fichero
- 5 Idioma de la sección
- 6 Contexto de la sección

Es una instrucción con un campo:

```
FILE_SCHEMA(  
  'AUTOMOTIVE_DESIGN { 1 0 10303 214 2 1 1}');
```

El campo indica el esquema EXPRESS que se ha utilizado para obtener el modelo STEP

SolidWorks utiliza dos esquemas:

```
STEP203, CONFIG_CONTROL_DESIGN  
STEP214, AUTOMOTIVE_DESIGN
```

# Fichero físico: cabecera

Introducción

Modularidad

Express

Fichero físico

Cabecera

Datos

Conclusiones

La sección de cabecera también tiene una **estructura fija**

↳ La estructura tiene entre tres y seis **grupos ordenados**:

1 Descripción del fichero

2 Nombre del fichero

3 Esquema del fichero

4 Población del fichero

5 Idioma de la sección

6 Contexto de la sección

Estos grupos solo se usan, a partir de la versión 3, para diferenciar múltiples grupos de datos

# Fichero físico: cabecera

Introducción

Modularidad

Express

Fichero físico

**Cabecera**

Datos

Conclusiones



Un ejemplo de cabecera creado por SolidWorks al exportar un modelo a STEP es el siguiente:

```
HEADER;  
FILE_DESCRIPTION (( 'STEP AP203' ),  
  '1');  
FILE_NAME ('Pieza1.STEP',  
  '2017-05-16T14:14:18',  
  ( " ),  
  ( " ),  
  'SwSTEP 2.0',  
  'SolidWorks 2016',  
  " );  
FILE_SCHEMA (( 'CONFIG_CONTROL_DESIGN' ));  
ENDSEC;
```

En las instrucciones con varios campos es habitual emplear una línea para cada campo para hacerlos más legibles para los humanos

Los indentados y los huecos en blanco son ignorados en el procesamiento informático

# Fichero físico: datos

La sección de datos contiene la información del modelo CAD, para lo que sigue unos principios básicos:

- ✓ La información se descompone en instrucciones o “instancias”

Cada instancia ocupa una o más líneas, y termina con punto y coma

- ✓ Cada instancia se identifica con una **etiqueta** o “nombre”

La etiqueta precede a la instancia y es el símbolo # seguido de un número

#000 Instancia;

- ✓ Las instancias deben estar ordenadas con números crecientes, aunque no necesariamente correlativos
- ✓ Las instancias se relacionan entre sí mediante sus números

El orden en el que están colocadas las instancias NO afecta a su interpretación

```
A= 1;  
B= 2;  
C= A + B;
```

≡

```
C= A + B;  
A= 1;  
B= 2;
```

# Fichero físico: datos

√ Cada instancia define una **entidad**

√ Las entidades **independientes** se definen en una instancia que contiene:

√ La *etiqueta* de la instancia

√ El *signo* igual

√ El *nombre* de la entidad

√ Los *atributos* de la entidad Entre paréntesis, y separados por comas

```
#23 = CARTESIAN_POINT ( 'NONE', ( 0.000, 0.000, 0.000 ) );
```

Los atributos múltiples se ponen entre paréntesis anidados

√ Las entidades **dependientes** se definen mediante un grupo de instancias:

√ Las entidades dependientes referencian a otras en sus atributos

```
#32 = DIRECTION ( 'NONE', ( -0.000, -0.000, -1.000 ) );
```

```
#112 = VECTOR ( 'NONE', #32, 1000.000 );
```

La entidad referenciada como atributo debe estar presente en la sección de datos

¡Pero el orden es indiferente!

√ Las dependencias se pueden encadenar

```
#13 = LINE ( 'NONE', #23, #112 );
```

```
#23 = CARTESIAN_POINT ( 'NONE', ( 0.000, 0.000, 0.000 ) );
```

```
#112 = VECTOR ( 'NONE', #32, 1000.000 );
```

```
#32 = DIRECTION ( 'NONE', ( -0.000, -0.000, -1.000 ) );
```

Introducción

Modularidad

Express

**Fichero físico**

Cabecera

**Datos**

Conclusiones

# Fichero físico: datos

Introducción

Modularidad

Express

**Fichero físico**

Cabecera

**Datos**

Conclusiones

Las entidades del fichero físico corresponden a diversos tipos de información del modelo CAD:

- 1 **Datums y sistemas de datums**, son “PLACEMENTS”, que localizan los elementos geométricos respecto a un sistema de coordenadas absoluto

Se asume que existe un **sistema cartesiano de coordenadas global**, al cual se referencian todas las demás entidades

- 2 **Geometría del modelo**, son “SOLID\_MODELS”, que representan la forma nominal de un producto

Incluye los sólidos de barrido, los sólidos primitivos, los B-Rep y los volúmenes recortados

- 3 **Anotaciones**, que incluyen las anotaciones administrativas de gestión del modelo, y las anotaciones de geometría, fabricación o diseño que enriquecen a los modelos CAD



Más detalles sobre Anotaciones en Tema 4

# Fichero físico: datos

Introducción

Modularidad

Express

**Fichero físico**

Cabecera

**Datos**

Conclusiones

Los datums y sistemas de datums se construyen a partir de los elementos geométricos básicos:

√ Un punto es una entidad independiente con cuatro atributos:

- 1 El nombre del punto
- 2 La coordenada X respecto al sistema global
- 3 La coordenada Y respecto al sistema global
- 4 La coordenada Z respecto al sistema global

Se define mediante una instancia independiente:

```
#23 = CARTESIAN_POINT ('', ( 0.000, 0.000, 0.000 ));
```

√ Una dirección es una entidad independiente con cuatro atributos:

- 1 El nombre de la dirección
- 2 La coordenada X del extremo del vector dirección
- 3 La coordenada Y del extremo del vector dirección
- 4 La coordenada Z del extremo del vector dirección

Se define mediante una instancia independiente:

```
#31 = DIRECTION ('', ( 0.000, 0.000, 1.000 ));
```

# Fichero físico: datos

Introducción

Modularidad

Express

**Fichero físico**

Cabecera

**Datos**

Conclusiones

✓ Un vector es una entidad dependiente con cuatro atributos:

- 1 El nombre del vector
- 2 La referencia a la dirección
- 3 El módulo

Se define mediante dos instancias dependientes:

```
#11 = VECTOR ( ' , #31 , 100.000 ) ;
```

```
#31 = DIRECTION ( ' , ( 0.000 , 0.000 , 1.000 ) ) ;
```

✓ Un sistema de coordenadas se define con una entidad dependiente que requiere cuatro atributos:

- 1 El nombre del sistema
- 2 Un punto de localización del origen de coordenadas
- 3 La dirección del primer eje de un sistema ortogonal dextrógiro
- 4 La dirección del segundo eje de un sistema ortogonal dextrógiro

Se define mediante cuatro instancias dependientes:

```
#49 = AXIS2_PLACEMENT_3D ( ' , #113 , #87 , #93 ) ;
```

```
#113 = CARTESIAN_POINT ( ' , ( 0.000 , 0.000 , 0.000 ) ) ;
```

```
#087 = DIRECTION ( ' , ( 0.000 , 0.000 , 1.000 ) ) ;
```

```
#093 = DIRECTION ( ' , ( 1.000 , 0.000 , 0.000 ) ) ;
```



# Fichero físico: datos

Las operaciones de modelado se especifican mediante instancias dependientes que definen alguno de los diferentes tipos de sólidos:

✓ Para definir un modelo B-Rep se describe:

✓ Su topología (todas las caras que lo delimitan):

```
#50 = ADVANCED_BREP_SHAPE_REPRESENTATION ( ' , ( #78, #49), #1 );
```

```
#078 = MANIFOLD_SOLID_BREP ( ' , #76 );
```

```
#076 = CLOSED_SHELL ( ' , ( #140, #xxx, ... ) );
```

El atributo múltiple incluye una etiqueta por cada cara

```
#140 = ADVANCED_FACE ( ' , ( #148), #172, .F. );
```

```
#148 = FACE_OUTER_BOUND ( ' , #227, .T. );
```

```
#227 = EDGE_LOOP ( ' , ( #121, #yyy, ... ) );
```

✓ Su geometría (los elementos geométricos que definen las caras, así como su posición y tamaño):

```
#121 = ORIENTED_EDGE ( ' , * , * , #222, .T. );
```

Los asteriscos indican atributos derivado (por herencia)

```
#222 = EDGE_CURVE ( ' , #15, #176, #178, .T. );
```

```
#015 = VERTEX_POINT ( ' , #4 );
```

```
#004 = CARTESIAN_POINT ( 'NONE', ( -25.000, 25.000, 0.0000 ) );
```

```
#176 = VERTEX_POINT ( ' , #6 );
```

```
#006 = CARTESIAN_POINT ( 'NONE', (( 25.000, -25.000, 0.0000 ) );
```

```
#178 = LINE ( ' , #227, #111);
```

```
#011 = CARTESIAN_POINT ( 'NONE', (( 25.000, -25.000, 0.0000 ) );
```

```
#11 = VECTOR ( ' , #31 , 100.000);
```

```
#31 = DIRECTION ( ' , ( 0.000, 0.000, 1.000 ) );
```

```
#172 = PLANE ( ' , #49)
```

```
#49 = AXIS2_PLACEMENT_3D ( ' , #113, #87, #93 );
```

Introducción

Modularidad

Express

Fichero físico

Cabecera

Datos

Conclusiones

# Fichero físico: datos

Introducción

Modularidad

Express

**Fichero físico**

Cabecera

**Datos**

Conclusiones

- ✓ Para definir modelos de barrido, se invoca la instancia del tipo particular de barrido, y se añade la información del perfil al que se aplica el barrido:

- ✓ Las operaciones de barrido:

```
#131 = SHAPE_DEFINITION_REPRESENTATION ( $, #171 );
```

```
#171 = PROCEDURAL_SHAPE_REPRESENTATION ( ' ', (#101), $ );
```

```
#171 = PROCEDURAL_SHAPE_REPRESENTATION_SEQUENCE ( ' ', (#78, #xx, ...), $, ' ' );
```

```
#078 = EXTRUDED_FACE_SOLID ( ' ', #76, #122, 55.000 );
```

```
#076 = FACE_SURFACE ( ' ', (#023, #xxx, ...), #045, .T. );
```

```
#023 = FACE_BOUND ( ' ', #025, .T. );
```

```
#025 = VERTEX_LOOP ( ' ', (#26, #xxx, ...));
```

```
#026 = VERTEX_POINT ( ' ', #27);
```

```
#027 = CARTESIAN_POINT ( ' ', ( 0.000, 0.000, 0.000 ));
```

```
...
```

```
#31 = DIRECTION ( ' ', ( 0.000, 0.000, 1.000 ));
```

```
#045 = SWEPT_SURFACE ( ' ', #023, #045, .T. );
```

```
#122 = DIRECTION ( ' ', ( 1.000, 0.000, 0.000 ));
```

- ✓ Tanto los barridos como los perfiles a los que se aplican los barridos pueden definirse paramétricamente:

```
#078 = EXTRUDED_FACE_SOLID ( ' ', #76, #122, #250 );
```

```
#250 = BOUND_PARAMETER_ENVIRONMENT ( #251, #252 );
```

```
#251 = INSTANCE_ATTRIBUTE_REFERENCE ( ' ', #078);
```

```
#252 = BOUND_MODEL_PARAMETER ( ' DEPTH', #253, ' ', $ );
```

```
#253 = FINITE_REAL_INTERVAL ( 50.000, .CLOSED., 60.000, .CLOSED.);
```

Ver NISTIR 7433  
Kim J. Pratt M.J., Iyer R, Sriram R.  
Data Exchange of Parametric CAD  
Models Using ISO 10303-108

# Fichero físico: datos

- √ Para definir primitivas CSG se utiliza alguna de las instancias específicas, y se indica su colocación:

```
#056 = BASE_SOLID_SELECT ( #78 );
```

```
#078 = BLOCK ( ' ', #79, 25, 25, 50 );
```

```
#079 = AXIS2_PLACEMENT_3D ( ' ', #aaa, #bbb, #ccc );
```

- √ Las unidades se especifican, al menos, para cada sólido:

```
#221 = MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION ( " ", ( #040 ), #001 );
```

```
#040 = STYLED_ITEM ( 'NONE ' , (#xxx) , #yyy);
```

```
...
```

```
#001= ( GEOMETRIC_REPRESENTATION_CONTEXT ( 3 )  
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT ( ( #2 ) )  
GLOBAL_UNIT_ASSIGNED_CONTEXT ( ( #3, #4, #5 ) )  
REPRESENTATION_CONTEXT ( 'NONE', 'WORKSPACE' ) );
```

```
#002 = UNCERTAINTY_MEASURE_WITH_UNIT ( LENGTH_MEASURE(  
1.000000000000000100E-005), #3, 'distance_accuracy_value', 'NONE');
```

```
#003 = ( LENGTH_UNIT ( ) NAMED_UNIT ( * ) SI_UNIT ( .MILLI., .METRE. ) )
```

```
#004 = ( NAMED_UNIT ( * ) PLANE_ANGLE_UNIT ( ) SI_UNIT ( $, .RADIAN. ) );
```

```
#005 = ( NAMED_UNIT ( * ) SI_UNIT ( $, .STERADIAN. ) SOLID_ANGLE_UNIT ( ) );
```

Introducción

Modularidad

Express

**Fichero físico**

Cabecera

**Datos**

Conclusiones

# Fichero físico: datos

Los ficheros físicos gestionan dos tipos de anotaciones:

1 Anotaciones administrativas, que identifican el contenido

Las **anotaciones administrativas** permiten la trazabilidad de proceso de creación y edición del modelo CAD:

✓ Hay anotaciones para indicar **autoría y propiedad**:

```
#077 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT (#43, #11, (#7));
```

```
#043 = PERSON_AND_ORGANIZATION (#114, #49);
```

```
#114 = PERSON ('UNSPECIFIED', 'UNSPECIFIED', 'UNSPECIFIED',  
('UNSPECIFIED'), ('UNSPECIFIED'), ('UNSPECIFIED'));
```

```
#49 = ORGANIZATION ('UNSPECIFIED', 'UNSPECIFIED', );
```

```
#011 = PERSON_AND_ORGANIZATION_ROLE ('creator');
```

```
#007 = PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE (  
'ANY', ", #101, .NOT_KNOWN. );
```

```
#101 = PRODUCT ('Pieza2 203', 'Pieza2 203', ", (#143));
```

```
#143 = MECHANICAL_CONTEXT ('NONE', #134, 'mechanical');
```

```
#134 = APPLICATION_CONTEXT ('configuration controlled  
3d designs of mechanical parts and assemblies');
```

✓ Hay anotaciones para controlar las **fechas y accesos**:

```
#138 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT (#155, #131, (#1));
```

```
#155 = DATE_AND_TIME (#23, #152);
```

```
#023 = CALENDAR_DATE (2018, 16, 5);
```

```
#152 = LOCAL_TIME (0, 36, 46.000000000000000000, #126);
```

```
#126 = COORDINATED_UNIVERSAL_TIME_OFFSET (1, 0, .AHEAD.);
```

```
#131 = DATE_TIME_ROLE ('classification_date');
```

```
#001 = SECURITY_CLASSIFICATION ("", #52);
```

```
#52 = SECURITY_CLASSIFICATION_LEVEL ('unclassified');
```

Los traductores comerciales siguen usando algunas entidades obsoletas

2 Anotaciones del modelo CAD

Introducción

Modularidad

Express

Fichero físico

Cabecera

Datos

Conclusiones

## Los ficheros físicos gestionan dos tipos de anotaciones:

1 Anotaciones administrativas, que identifican el contenido

2 Anotaciones del modelo CAD

Las anotaciones CAD semánticas de los protocolos AP 203 y AP214 están principalmente vinculadas a los dibujos (ISO 10303-1206:2014 Draughting annotations:

```
#723= DRAUGHTING_MODEL_ITEM_ASSOCIATION ('', '#723, #724, #725);
```

```
#723= FLATNESS_TOLERANCE ('', #217, #130);
```

```
#217= LENGTH_MEASURE_WITH_UNIT (LENGTH_MEASURE(0.1), $);
```

```
#130= SHAPE_ASPECT ('', 'GDT', #049, .F.);
```

```
#049= PRODUCT_DEFINITION_SHAPE ('', #228);
```

```
#0228= PRODUCT_DEFINITION ('', #4368);
```

```
#724= DRAUGHTING_MODEL ('', (#711, #xxx, ...));
```

```
#711= ANNOTATION_PLANE ('', (#712, #xxx, ...), $, $);
```

...

A veces se apoyan en símbolos predefinidos, pero, en otras ocasiones se usan símbolos “delineados” sin ninguna carga semántica:

```
#725= DRAUGHTING_CALLOUT('Flatness', (#790));
```

```
#790= TESSELLATED_ANNOTATION_OCCURRENCE('', (#aaa), #bbb);
```

# Fichero físico: datos



Las anotaciones no predefinidas se guardan en el formato STEP como trazos geométricos (sin ningún tipo de semántica):

Introducción

Modularidad

Express

Fichero físico

Cabecera

Datos

Conclusiones



```
#33=COORDINATES_LIST(",2,((-2.128,25,-2.600),(-7.479,34.070,-2.600));  
#32=TESSELLATED_CURVE_SET("#33,((1,2)));
```

```
#35=COORDINATES_LIST(",2,((-2.128,25,-2.600),(8.573,43.141,-2.600));  
#34=TESSELLATED_CURVE_SET("#35,((1,2)));
```

```
#37=COORDINATES_LIST(",2,  
((8.573,43.141,-2.600),(15.889,43.141,-2.600));  
#36=TESSELLATED_CURVE_SET("#37,((1,2)));
```

```
#38=COORDINATES_LIST(",31,  
((12.301,53.369,-2.600),(13.466,53.369,-2.600),(15.809,44.903,-2.600),(14.604,44.903,-2.600),(14.128,46.651,-  
2.600),(11.652,46.651,-2.600),(11.149,44.903,-2.600),(9.944,44.903,-2.600),(12.896,51.124,-2.600),(12.589,50.0336,-  
2.600),(12.281,48.943,-2.600),(12.127,48.397,-2.600),(12.088,48.261,-2.600),(12.0793,48.227,-2.600),(12.076,48.218,-  
2.600),(12.076,48.216,-2.600),(12.0793,48.216,-2.600),(12.088,48.216,-2.600),(12.126,48.216,-2.600),(12.278,48.216,-  
2.600),(12.883,48.216,-2.600),(13.489,48.216,-2.600),(13.640,48.216,-2.600),(13.678,48.216,-2.600),(13.688,48.216,-  
2.600),(13.690,48.216,-2.600),(13.690,48.218,-2.600),(13.688,48.227,-2.600),(13.678,48.261,-2.600),(13.641,48.397,-  
2.600),(13.492,48.943,-2.600));
```

```
#39=COMPLEX_TRIANGULATED_SURFACE_SET("#38,31,  
((0.,0.,1.)),(),((8,7,6),(19,6,20),(15,8,6),(17,16,6),(19,18,6),(17,6,18),(20,6,21),(15,6,16),(14,13,8),(12,1,8),(13,12,8),(11,  
1,12),(1,10,9),(10,1,11),(14,8,15),(22,21,5),(24,23,5),(22,5,23),(4,3,5),(26,25,5),(25,24,5),(5,3,27),(21,6,5),(31,30,2),(9,  
2,1),(2,9,31),(29,3,30),(27,26,5),(28,3,29),(3,28,27),(2,30,3)),());
```

```
#27=TESSELLATED_GEOMETRIC_SET('note',(#32,#34,#36,#39));  
#26=TESSELLATED_ANNOTATION_OCCURRENCE('Elemento de detalle1',(#31),#27);  
#40=DRAUGHTING_CALLOUT('Elemento de detalle1',(#26));
```



```
#38=COORDINATES_LIST(",31,  
((12.301,53.369,-2.600),(13.466,53.369,-2.600),(15.809,44.903,-2.600),(14.604,44.903,-2.600),(14.128,46.651,-  
2.600),(11.652,46.651,-2.600),(11.149,44.903,-2.600),(9.944,44.903,-2.600),(12.896,51.124,-2.600),(12.589,50.0336,-  
2.600),(12.281,48.943,-2.600),(12.127,48.397,-2.600),(12.088,48.261,-2.600),(12.0793,48.227,-2.600),(12.076,48.218,-  
2.600),(12.076,48.216,-2.600),(12.0793,48.216,-2.600),(12.088,48.216,-2.600),(12.126,48.216,-2.600),(12.278,48.216,-  
2.600),(12.883,48.216,-2.600),(13.489,48.216,-2.600),(13.640,48.216,-2.600),(13.678,48.216,-2.600),(13.688,48.216,-  
2.600),(13.690,48.216,-2.600),(13.690,48.218,-2.600),(13.688,48.227,-2.600),(13.678,48.261,-2.600),(13.641,48.397,-  
2.600),(13.492,48.943,-2.600));
```

```
#39=COMPLEX_TRIANGULATED_SURFACE_SET("#38,31,  
((0.,0.,1.)),(),((8,7,6),(19,6,20),(15,8,6),(17,16,6),(19,18,6),(17,6,18),(20,6,21),(15,6,16),(14,13,8),(12,1,8),(13,12,8),(11,  
1,12),(1,10,9),(10,1,11),(14,8,15),(22,21,5),(24,23,5),(22,5,23),(4,3,5),(26,25,5),(25,24,5),(5,3,27),(21,6,5),(31,30,2),(9,  
2,1),(2,9,31),(29,3,30),(27,26,5),(28,3,29),(3,28,27),(2,30,3)),());
```

# Fichero físico: STEP

Introducción

Modularidad

Express

**Fichero físico**

Conclusiones



Pese a su popularidad como forma de intercambio de modelos CAD, el fichero físico de STEP tiene varios inconvenientes destacables:

- ✗ Es difícil de interpretar, porque no se puede leer secuencialmente
- ✗ La estructura modular de STEP lo hace ineficiente, porque las instancias dependientes descomponen las operaciones en muchas entidades, y no reutiliza instancias comunes
- ✗ Permite “aromas”, dado que una misma operación se puede expresar mediante instancias, o conjuntos de instancias, diferentes

Por ejemplo, para colorear un sólido se utilizan muchas instancias:

```
#095 = PRESENTATION_LAYER_ASSIGNMENT ( " , " , (#40) )
#040 = STYLED_ITEM ( ' NONE ' , (#250) , #112);
#250 = PRESENTATION_STYLE_ASIGNEMENT ( (#004) );
#004 = SURFACE_STYLE_USAGE ( . BOTH . , #244);
#244 = SURFACE_SIDE_STYLE ( ' ' , #033);
#033 = SURFACE_STYLE_FILL_AREA ( #002);
#002 = FILL_AREA_STYLE ( ' ' , #106);
#106 = FILL_AREA_STYLE_COLOUR ( ' ' , #149);
#149 = COLOUR_RGB ( ' ' , 0.05 , 0.00 , 0.00);
```

Por claridad, las instancias se han mostrado ordenadas, para lo que se ha alterado su secuencia

El segundo problema es que, típicamente, puede haber definiciones de color duplicadas

# Fichero físico: STEP

Introducción

Modularidad

Express

**Fichero físico**

Conclusiones

x Aunque están en desuso, existen **versiones cortas** para los nombres de las entidades...

...que hacen menos legible el fichero

Entity name	Short name
cartesian_point	cpt
vertex	vx
Edge	ed
edge_logical_structure	ed_strc
edge_loop	ed_loop

```
ISO-10303-21;
HEADER; FILE_DESCRIPTION((THIS FILE CONTAINS A SMALL SAMPLE STEP
MODEL), 4;2);
FILE_NAME('EXAMPLE STEP FILE #1',
'2013-02-11T15:30:00',
('JOHN DOE',
'ACME INC.',
'METROPOLIS USA'),
('ACME INC. A SUBSIDIARY OF GIANT INDUSTRIES','METROPOLIS USA'),
'CIM/STEP VERSION2',
'SUPER CIM SYSTEM RELEASE 4.0',
'APPROVED BY JOE BLOGGS');
FILE_SCHEMA(('EXAMPLE_GEOMETRY'));
ENDSEC;
DATA;
/*
THE FOLLOWING 13 ENTITIES REPRESENT A TRIANGULAR EDGE LOOP
*/
#1=CPT(0.0,0.0,0.0); /* THIS IS A CARTESIAN POINT ENTITY */
#2=CPT(0.0,1.0,0.0);
#3=CPT(1.0,0.0,0.0);
#11=VX(#1); /* THIS IS A VERTEX ENTITY */
#12=VX(#2);
#13=VX(#3);
#16=ED(#11,#12); /* THIS IS AN EDGE ENTITY */
#17=ED(#11,#13);
#18=ED(#13,#12);
#21=ED_STRC(#17,.F.); /* THIS IS AN EDGE LOGICAL STRUCTURE ENTITY */
#22=ED_STRC(#18,.F.);
#23=ED_STRC(#16,.T.);
#24=ED_LOOP((#21,#22,#23)); /* THIS IS AN EDGE LOOP ENTITY */
/*
OTHER SYNTACTICAL REPRESENTATIONS WERE POSSIBLE. THE PREVIOUS
EXAMPLE IS REPRESENTATIVE OF ONE POSSIBLE APPROACH.
*/
ENDSEC;
END-ISO-10303-21;
```

Fuente: [www.steptools.com](http://www.steptools.com)



# Conclusiones

Introducción

Modularidad

Express

Fichero físico

**Conclusiones**

1 El formato neutro STEP es la mejor alternativa de **intercambio** de modelos CAD mediante ficheros neutros

- ✓ Es bastante completo
- ✓ Está suficientemente implantado
- ✓ Tiene perspectivas de futuro, porque se sigue desarrollando

2 Escribir un fichero STEP es potencialmente complejo, porque hay que ajustarse a un **método de implementación**:

- ✓ muy extenso
- ✓ muy estricto
- ✓ tiene muchas variantes

3 El lenguaje de modelado **EXPRESS** es suficientemente claro para:

- ✓ Permitir descripciones de modelos sencillos a usuarios con poca formación
- ✓ Permitir revisiones y comparaciones de los modelos usados por diferentes traductores de modelos CAD

# Conclusiones

Introducción

Modularidad

Express

Fichero físico

**Conclusiones**

- 4 El formato STEP también tiene potencial para llegar a emplearse como fichero de almacenamiento propio de diferentes aplicaciones CAD

El objetivo final es una base de datos de información de productos integrada que sea accesible y útil para todos los recursos necesarios para respaldar un producto durante su ciclo de vida

- 5 El poder de las herramientas de modelado STEP permite desarrollar mejores modelos

Permite aumentar la calidad de los modelos

- 6 Pero las herramientas y la metodología de STEP son extraordinariamente difíciles de dominar

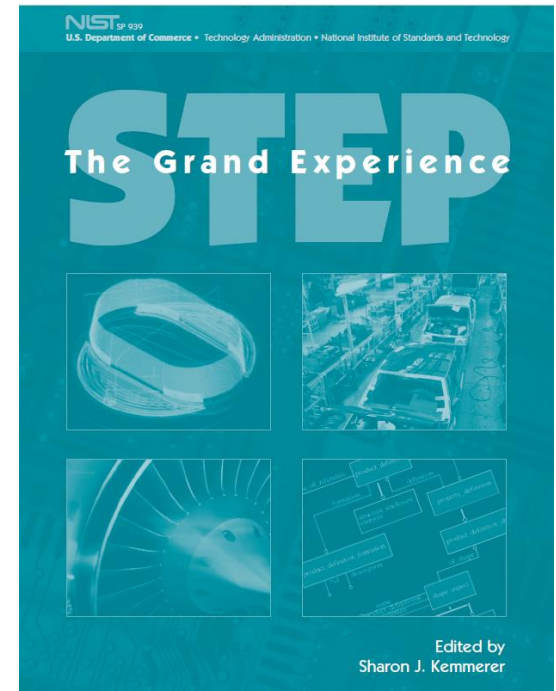
Debido a su extensión, y a su desarrollo mediante trabajo en paralelo durante mucho tiempo:

- ✓ Falta armonización
- ✓ Hay solapes

# Para saber más

## Para saber más de STEP:

- √ Subcomite 4 del Comité 184 de ISO:  
<https://www.iso.org/committee/54158.html>
- √ The CAX Implementor Forum:  
<http://www.cax-if.org>
- √ PDES Inc.:  
<http://www.pdesinc.org>
- √ STEP AP242:  
<http://www.ap242.org/>
- √ WIKISTEP:  
<http://www.wikistep.org>
  
- √ NIST special publication 939  
<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication939.pdf>



# Para saber más

---

## Para saber más del lenguaje EXPRESS:

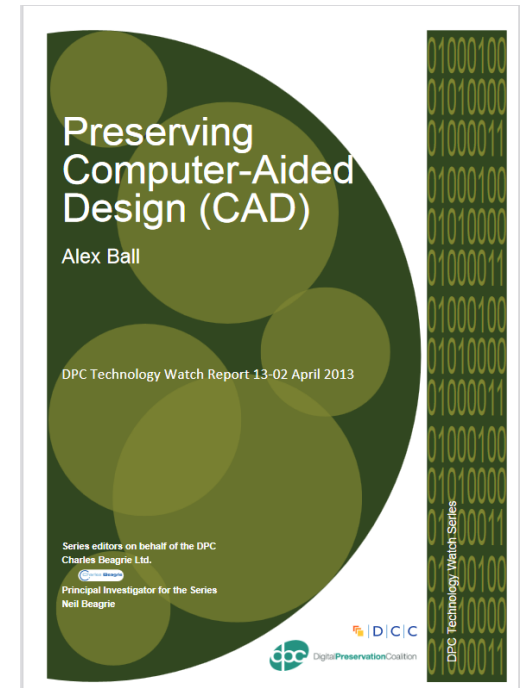
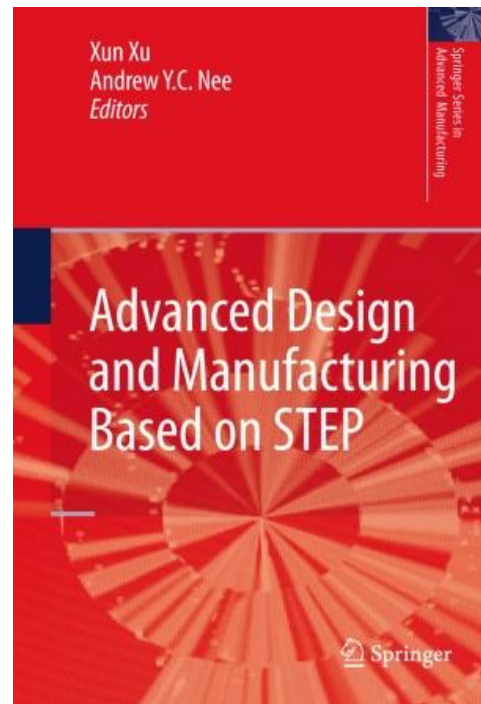
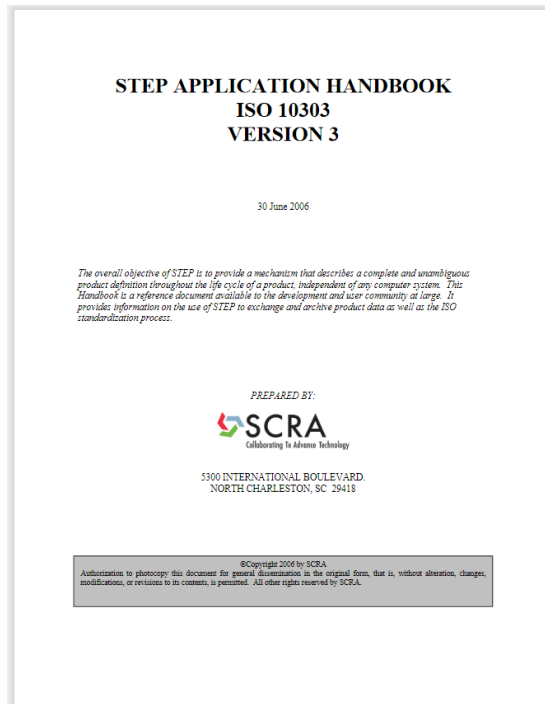
- √ [https://en.wikipedia.org/wiki/EXPRESS\\_\(data\\_modeling\\_language\)](https://en.wikipedia.org/wiki/EXPRESS_(data_modeling_language))
- √ [http://www.steptools.com/library/standard/p21e3\\_dis\\_paris.html#clause-11](http://www.steptools.com/library/standard/p21e3_dis_paris.html#clause-11)

## Para saber más del fichero físico:

- √ <http://www.jsdai.net/support/about-step/step-file>
- √ [http://www.steptools.com/library/standard/p21e3\\_dis\\_paris.html#clause-11](http://www.steptools.com/library/standard/p21e3_dis_paris.html#clause-11)

# Para saber más

## Otros documentos para contextualizar STEP:



Disponible en:

<http://www.asd-ssg.org/>

[c/document\\_library/get\\_file?](http://www.asd-ssg.org/c/document_library/get_file?)

[uuid=1a27ecc6-6570-40cd-b611-f02bac2c2687&groupId=11317](http://www.asd-ssg.org/c/document_library/get_file?uuid=1a27ecc6-6570-40cd-b611-f02bac2c2687&groupId=11317)

ISBN 978-1-84882-739-4

ISSN: 2048-7916

DOI: <http://dx.doi.org/10.7207/twr13-02>