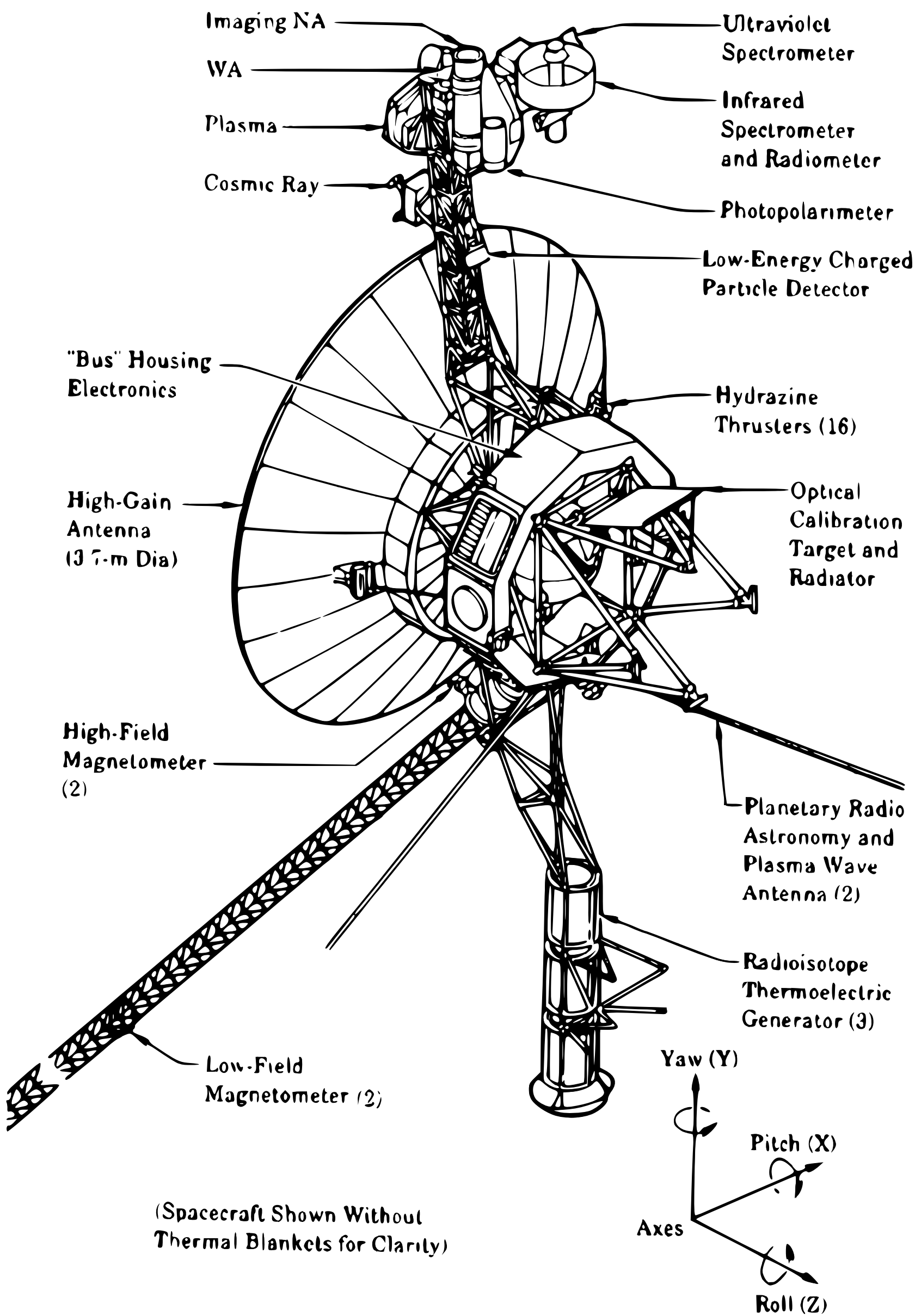


LM

Esquemas XML



# Esquemas XML

- Documento XML que define un conjunto de reglas
- Son una alternativa a los DTDs
- Más potentes que los DTDs

# Declaración del esquema

# Nodo raíz

```
<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  ...  
</xs:schema>
```

# Elementos

# Elementos

Tipo	Características
Estándar	Sin hijos y sin restricciones ni atributos
Simple	Sin hijos pero con restricciones y sin atributos
Complejo	Con hijos y/o atributos

# Estructura

```
<xs:element name="nombre_del_elemento" type="tipo" />
```

```
<xs:element name="nombre_del_elemento">
```

```
...
```

```
</xs:element>
```

# Atributos del elemento

Nombre	Descripción
name	Nombre del elemento
type	Tipo simple predefinido
maxOccurs	Nº máximo de veces que puede aparecer
minOccurs	Nº mínimo de veces que debe aparecer
ref	Para hacer referencia a un elemento declarado anteriormente



# Tipos de datos

- Primitivos (string, boolean, decimal, float...)
- Derivados

Elemento complexType

# Elemento complexType

- Para definir elementos que tienen subelementos y/o atributos

# Sintaxis

```
<xs:complexType name="nombre_del_tipo_complejo">  
  <xs:sequence/all/choice>  
    ... subelementos ...  
  </xs:sequence/all/choice>  
  ... atributos ...  
</xs:complexType>
```

# Subelementos de complexType

Etiqueta	Características
sequence	Implica que deben aparecer todos los elementos y en ese orden (AND)
all	Implica que deben aparecer todos los elementos, sin importar el orden
choice	Implica que sólo debe aparecer uno de esos elementos (OR)
attribute	Se utiliza para definir atributos

# sequence

```
<xs:element name="libro" type="tipo_libro"/>
```

```
<xs:complexType name="tipo_libro">
```

```
<xs:sequence>
```

```
  <xs:element name="titulo" type="xs:string" />
```

```
  <xs:element name="autor" type="xs:string" />
```

```
  <xs:element name="editorial" type="xs:string" />
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
<libro>
```

```
  <titulo>El señor de los anillos</titulo>
```

```
  <autor>John Ronald Ruelen Tolkien</autor>
```

```
  <editorial>Tirant Lo Blanch</editorial>
```

```
</libro>
```

- Un libro tiene titulo, autor y editorial y deben aparecer exactamente en ese orden.

# all

```
<xs:element name="libro" type="tipo_libro"/>
```

```
<xs:complexType name="tipo_libro">
```

```
<xs:all>
```

```
  <xs:element name="titulo" type="xs:string" />
```

```
  <xs:element name="autor" type="xs:string" />
```

```
  <xs:element name="editorial" type="xs:string" />
```

```
</xs:all>
```

```
</xs:complexType>
```

```
<libro>
```

```
  <titulo>El señor de los anillos</titulo>
```

```
  <editorial>Tirant Lo Blanch</editorial>
```

```
  <autor>John Ronald Ruelen Tolkien</autor>
```

```
</libro>
```

- Un libro tiene titulo, autor y editorial en cualquier orden.

# choice

```
<xs:element name="libro" type="tipo_libro"/>
```

```
<xs:complexType name="tipo_libro">
```

```
<xs:choice>
```

```
  <xs:element name="titulo" type="xs:string" />
```

```
  <xs:element name="autor" type="xs:string" />
```

```
  <xs:element name="editorial" type="xs:string" />
```

```
</xs:choice>
```

```
</xs:complexType>
```

```
<libro>
```

```
  <titulo>El señor de los anillos</titulo>
```

```
</libro>
```

- Un libro puede tener o un titulo, o un autor o una editorial.



# Atributos

```
<xs:attribute name="nombre_atributo" type="tipo_atributo" use="modificador" />
```

# Propiedades de attribute

Atributo	Características
name	Nombre del tipo complejo
type	Tipo de datos con el que se identifica
use	Para definir si es obligatorio ( <b>required</b> ) u opcional

# Estructura del complexType

- Dos formas de escribirlo
- Mejor la segunda, es reutilizable

# Estructura del complexType

```
<xs:element name="contacto">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="destinatario" type="xs:string" />
      <xs:element name="remitente" type="xs:string" />
      <xs:element name="titulo" type="xs:string" />
      <xs:element name="contenido" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="fecha" type="xs:date" />
  </xs:complexType>
</xs:element>
```

# Estructura del complexType

```
<xs:element name="contacto" type="tipo_contacto"/>
```

```
<xs:complexType name="tipo_contacto">
```

```
<xs:sequence>
```

```
  <xs:element name="destinatario" type="xs:string" />
```

```
  <xs:element name="remite" type="xs:string" />
```

```
  <xs:element name="titulo" type="xs:string" />
```

```
  <xs:element name="contenido" type="xs:string" />
```

```
</xs:sequence>
```

```
<xs:attribute name="fecha" type="xs:date"/>
```

```
</xs:complexType>
```

Elemento simpleType

# Elemento simpleType

- Un tipo simple sirve para definir una serie de restricciones a un elemento o a un atributo
- Es muy útil para poner rangos, tipos enumerados...

# Elemento simpleType

```
<xs:simpleType name="nombre_del_tipo_simple">  
  <xs:restriction>  
    ... restricciones ...  
  </xs:restriction>  
</xs:simpleType>
```



# Componentes de simpleType

Tipo	Etiqueta	Características
Subelementos	restriction	Para poner rangos, patrones, enumerar posibles valores etc.
	list	Para definir un tipo de lista
	union	Para unir varios tipos definidos anteriormente en uno
Atributo	name	Para poner el nombre al tipo simple

Elemento restriction

# Elemento restriction

- Se utiliza para poner rangos, patrones, enumerar posibles valores...

# Elemento restriction

```
<xs:restriction base="xs:string">  
  <xs:nombre_restriccion value="" />  
</xs:restriction>
```

# Tipos de restricciones

Nombre	Características
enumeration	Se ponen los valores que puede tomar el elemento
minExclusive maxExclusive	Valores mínimos o máximos que puede tomar el elemento, sin incluir el último valor
minInclusive maxInclusive	Valores mínimos o máximos que puede tomar el elemento, incluyendo el último valor
pattern	Expresión regular que describe la restricción

Una expresión regular, a menudo llamada también regex, es una secuencia de caracteres que forma un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones.

# Expresiones regulares

```
<xs:pattern value="([a-zA-Z0-9])*"/>
```

```
<xs:pattern value="\d{2}-\d{4}"/>
```

# Tipos de restricciones

Nombre	Características
length	Longitud de un elemento de tipo texto
maxLength minLength	Longitud máxima o mínima de un elemento de tipo texto
totalDigits	Número exacto de dígitos permitidos
fractionDigits	Número máximo de decimales permitidos



Utilización del esquema

# Utilización del esquema

```
<nodo_raiz xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.miempresa.com/mi_esquema.xsd">
```

```
<nodo_raiz xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="mi_esquema.xsd">
```

すべての質問は？

Any questions?

Galderarik?

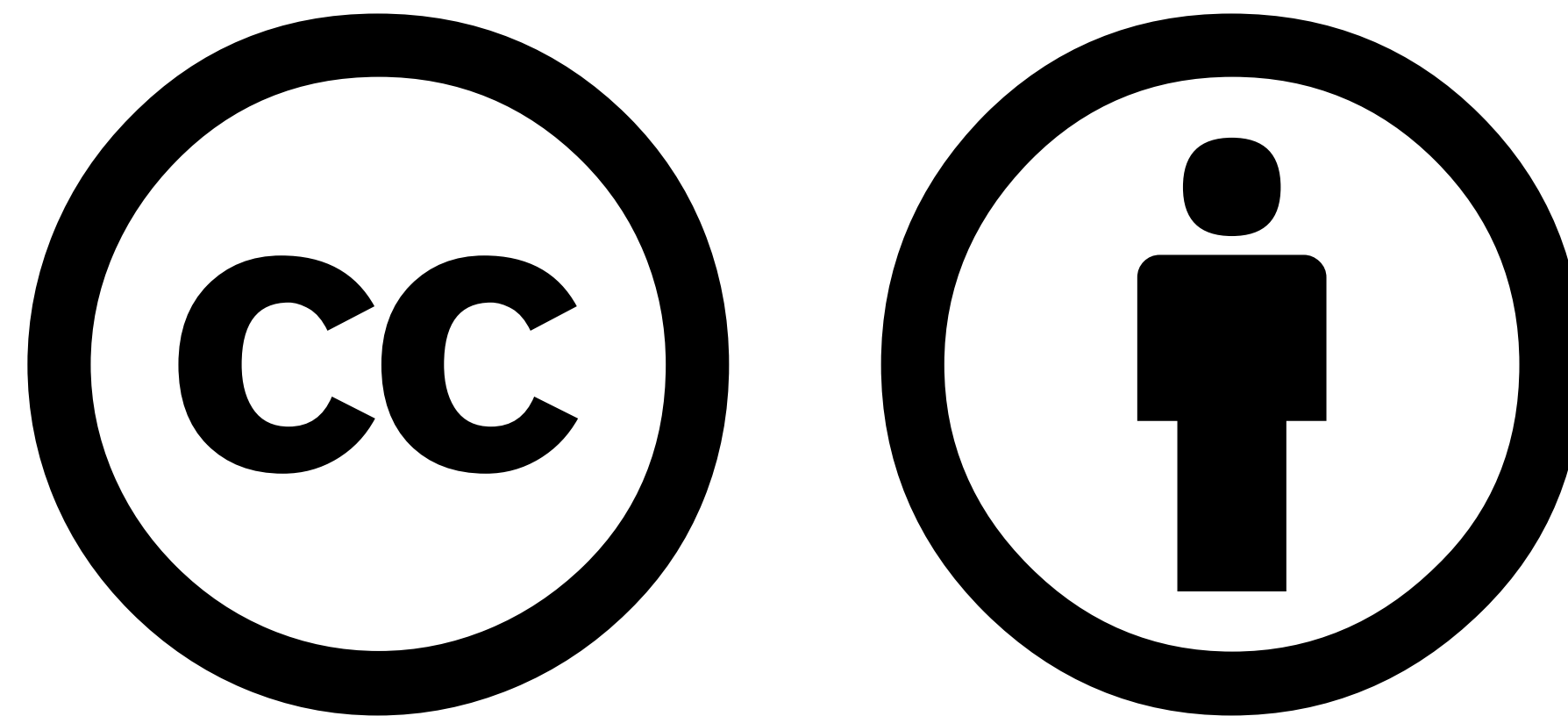
¿Alguna pregunta?

任何問題？

Alguma pergunta?

أي أسئلة؟

질문 있나요?



Excepto si se especifica lo contrario, esta presentación está bajo licencia

**<https://creativecommons.org/licenses/by/4.0/>**

© 2018 Ion Jaureguialzo Sarasola. Algunos derechos reservados.