LENGUAJE DE MARCAS

[Reconocimiento de las características de los lenguajes de marcas 2](#_Toc797514403)

[Introducción al lenguaje de marcas 3](#_Toc1081730908)

[Introducción al lenguaje de marcas XML 5](#_Toc585145425)

# Reconocimiento de las características de los lenguajes de marcas

## Introducción al lenguaje de marcas

Qué es

* Modo de codificar un documento de texto
* Se incluyen etiquetas, marcas o anotaciones (normalmente < y >)
* No es un lenguaje de programación
* Puede contener partes de código de otros lenguajes de programación

Características generales

* Texto plano
* Lenguaje compacto: las etiquetas se mezclan con el contenido
* Facilidad de procesamiento
* Flexibilidad: se puede combinar con otros lenguajes

Definición del tipo

* Está definido en un documento denomidnado DTD (Document Type Definition)
* En él se establecer las marcas, los elementos utilizados por dicho lenguaje y sus correspondientes etiquetas, atributos, sintaxis y normas de uso

Tipos

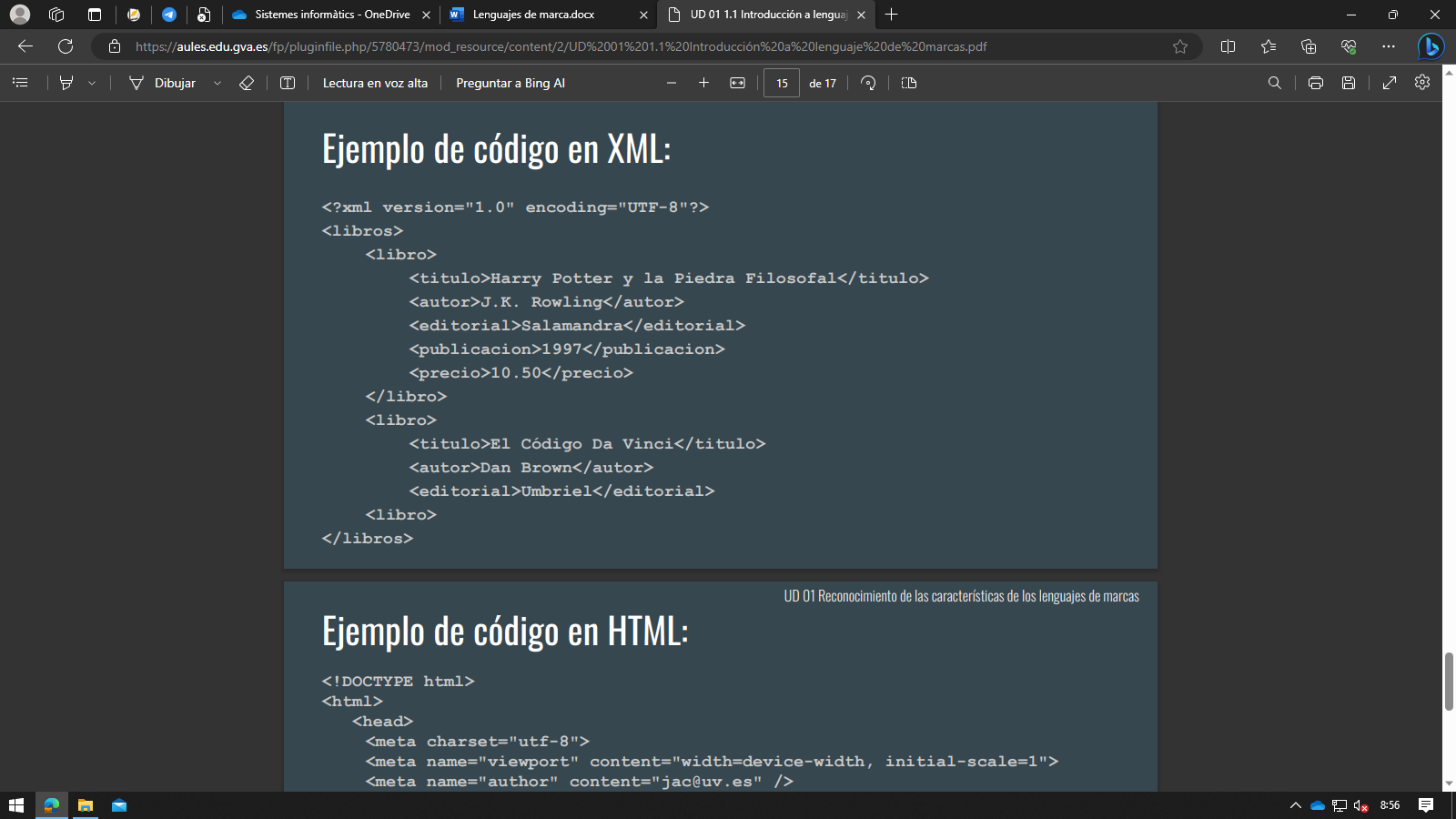
* Lenguajes de programación
  + Define el formato (apariencia) del texto
  + Suelen ocultar las etiquetas y mostrar solo el texto
  + Ejemplo: RTF (Rich Text Format) (Microsoft, 1987)
* Lenguajes de procedimientos
  + Orientado también a la presentación
  + El programa debe interpretar las etiquetas
  + Ejemplo:
    - LaTeX: representa fórmulas matemáticas complejas
    - HTML: lenguaje de maquetación y estructuración que se utiliza en el desarrollo de páginas web
* Lenguajes descriptivos o semánticos
  + Describen las diferentes partes del documento
  + Definen su contenido, pero sin identificar cómo deben representarse
  + Ejemplo: XML (metalenguaje que permite definir otros lenguajes de marca)

Evolución

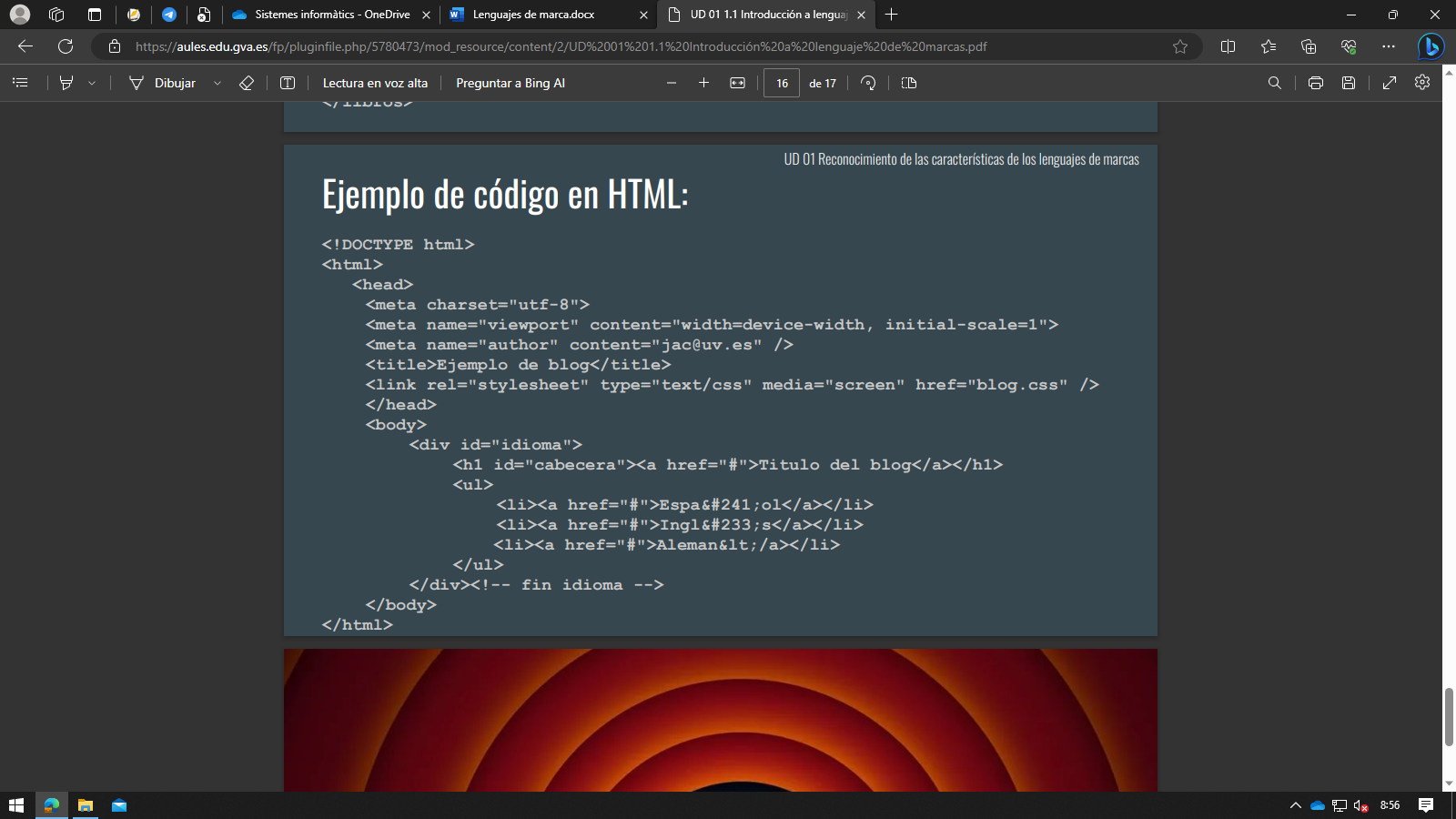
* GML
  + 1969
  + Generalized Markup Language
  + Nace ante la necesidad de gestionar documentación entre distintos departamentos y aplicaciones
* SGML
  + Standard Generalized Markup Language
  + Evolción a la estandarización del GML
  + 1986 -> ISO 8879
* HTML
  + Hyperext Markup Language
  + Aparece con el nacimiento del World Wide Web
  + Fruto de la combinación de ASCII y SGML
  + Desventajas
    - No soporta tareas de impresión y diseño
    - No flexible (etiquetas definidas)
    - No contenido dinámico
    - La estructura y el diseño están mezclados en el documento
* XML
  + EXtensible Markup Language
  + Estándar del W3C
  + Lenguaje estructural puro
  + Ventajas
    - Permiten definir etiquetas propias
    - Permiten asignar atributos a las etiquetas
    - Utilizar un esquema para definir de forma exacta las etiquetas y los atributos
    - La estructura y el diseño son intependientes
* XHTML
  + Surge para expresar el HTML como un lenguaje XML válido
  + Sin mucho éxito
* HTML5
  + 2006
  + HTML semántico
  + Las etiquetas de presentación quedan obsoletas (deprecated)

Sintaxis

* Etiquetas entre símbolos < y >
* Todas las etiquetas tienen que tener su correspondiente etiqueta de cierre
* Etiquetas y atributos en minúsculas
* Valor de los atributos entre comillas (simples o dobles)
* Ejemplo en XML:



* Ejemplo en HTML:



## Introducción al lenguaje de marcas XML

Qué es

* eXtensible Markup Language
* Lenguaje de marcas
* Metalenguaje porque define las reglas a cumplir por un lenguaje de marcas
* Creado por W3C a finales de los 90
* Define la estructura del documento
* Extensión .xml
* Documento de texto plano
* Objetivo: estandarización del HTML -> XHTML (sin éxito)
* Uso actual: intercambio de información (actualmente se usa más JSON)

Características básicas

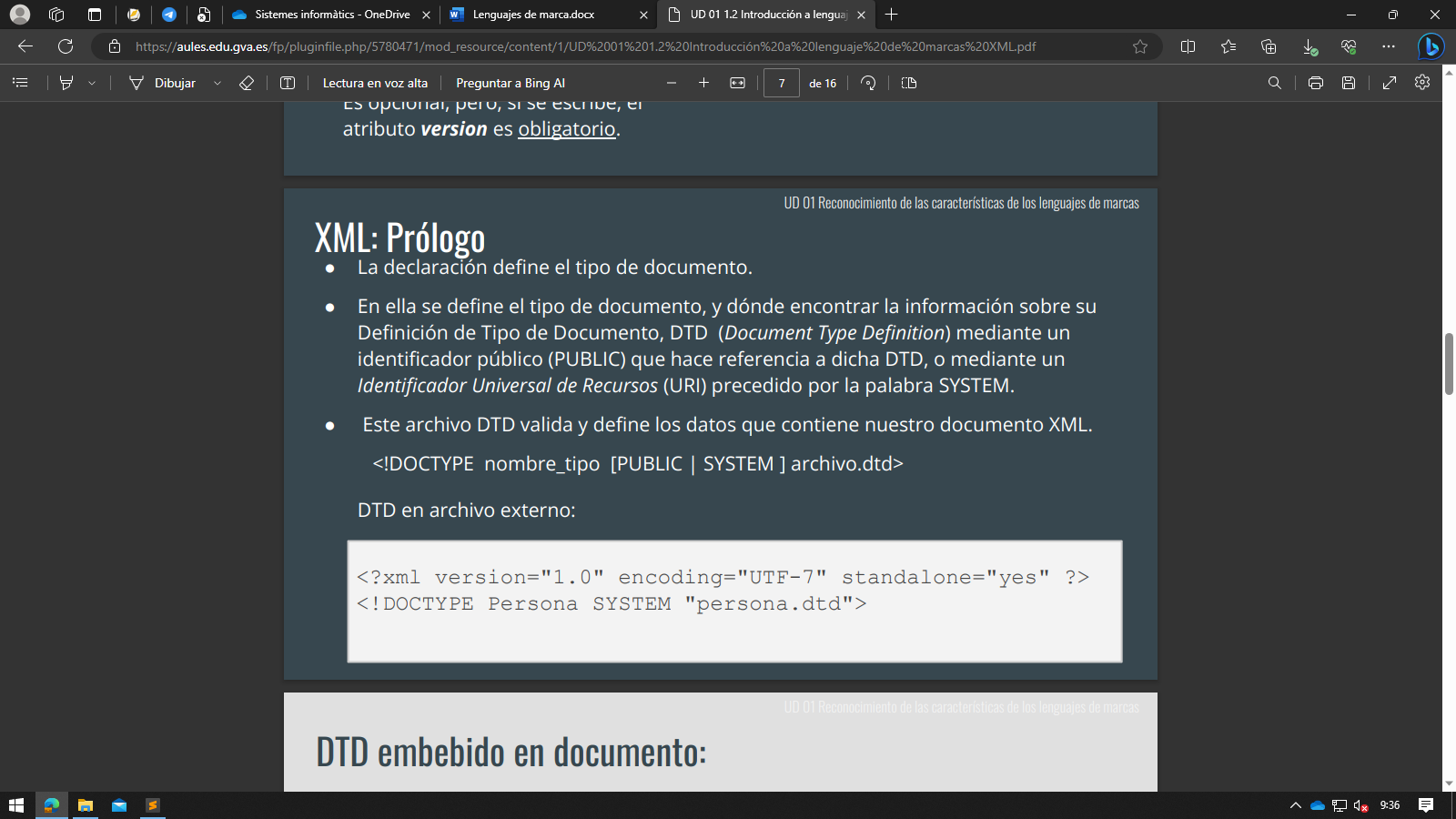
* Concebido para trabajar en la web
* No se requieren conocimientos de programación
* Documentos fáciles de crear
* Legible para los humanos
* Diseño formal y conciso
* Extensible y adaptable a variedad de situaciones
* Orientado a objetos
* Se compone exclusivamente de datos de marcado y datos carácter entremezclados

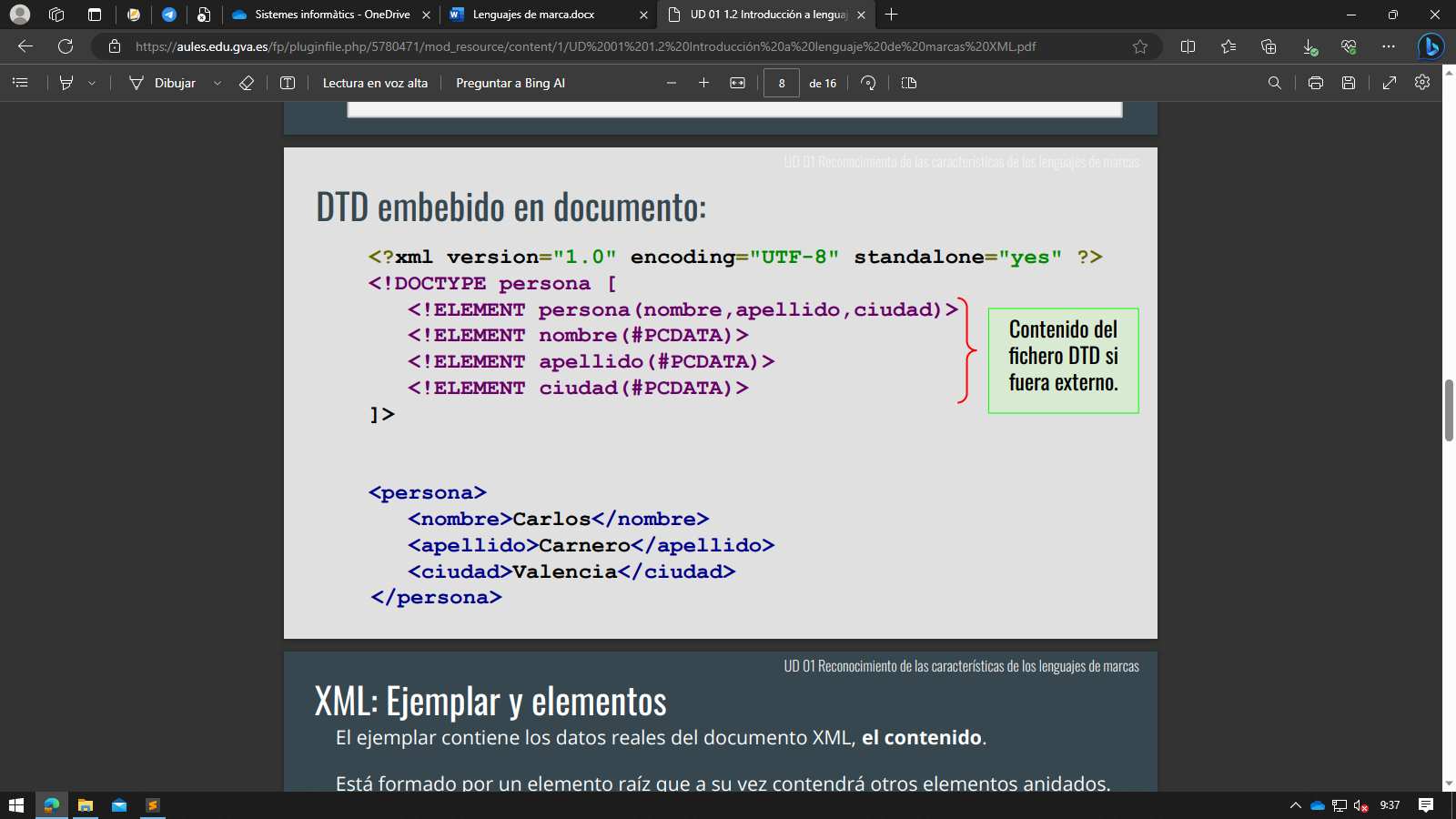
Algunos conceptos

* Caracteris y codificación (ISO-889-59-1, UTF-8, etc.)
* Procesador XML (XML parser)
  + La aplicación recurre a un procesador XML para leer un documento XML
  + Este procesador es el que lee, analiza y pasa la información en un formato estructurado a la aplicación
* Etiquetas (tags) o marcas (mark-up)
  + Las etiquetas son marcas que sirven para identificar un contenido concreto respecto al resto
  + Tipos de etiquetas:
    - Etiquetas de apertura (start-tag): <apartado>
    - Etiquetas de cierre (end-tag): </apartado>
    - Etiquetas vacías (empty-tag): <saldo-de-línea />

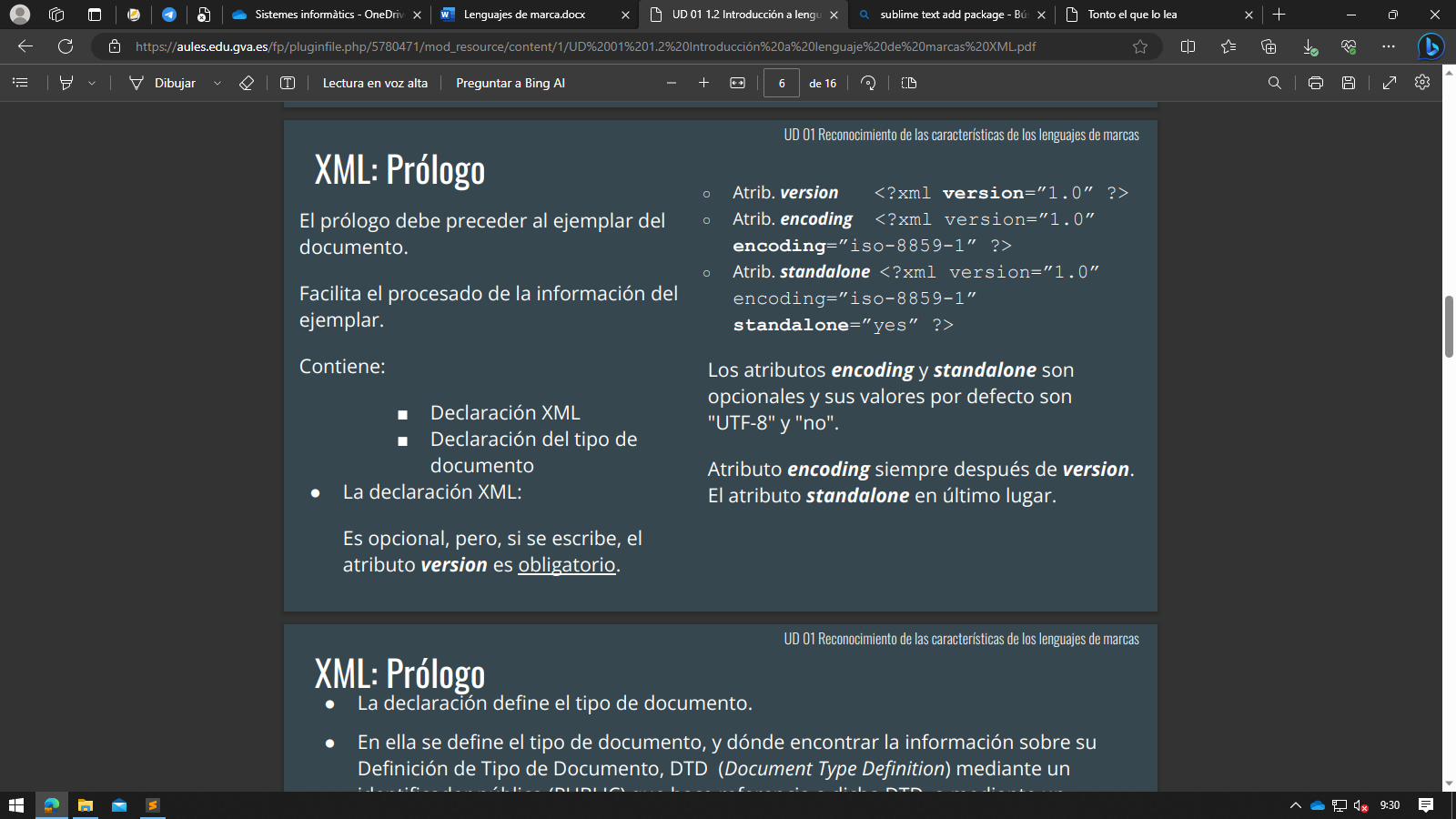
Partes

* Prólogo
  + Facilita el procesado de la información del ejemplar
    - Define el tipo de documento
    - Define dónde encontrar la información sobre su Definición de Tipo de Documento, DTD (Document Type Definition)
    - Mediante un identificador público (PUBLIC) que hace referencia a dicha DTD
    - O mediante un Identificador Universal de Recursos (URI) precedido por la palabra SYSTEM
    - Este archivo DTD valida y define los datos que contiene nuestro documento XML
    - <! DOCTYPE nombre\_tipo [PUBLIC | SYSTEM] archivo.dtd>
    - DTD en archivo externo:

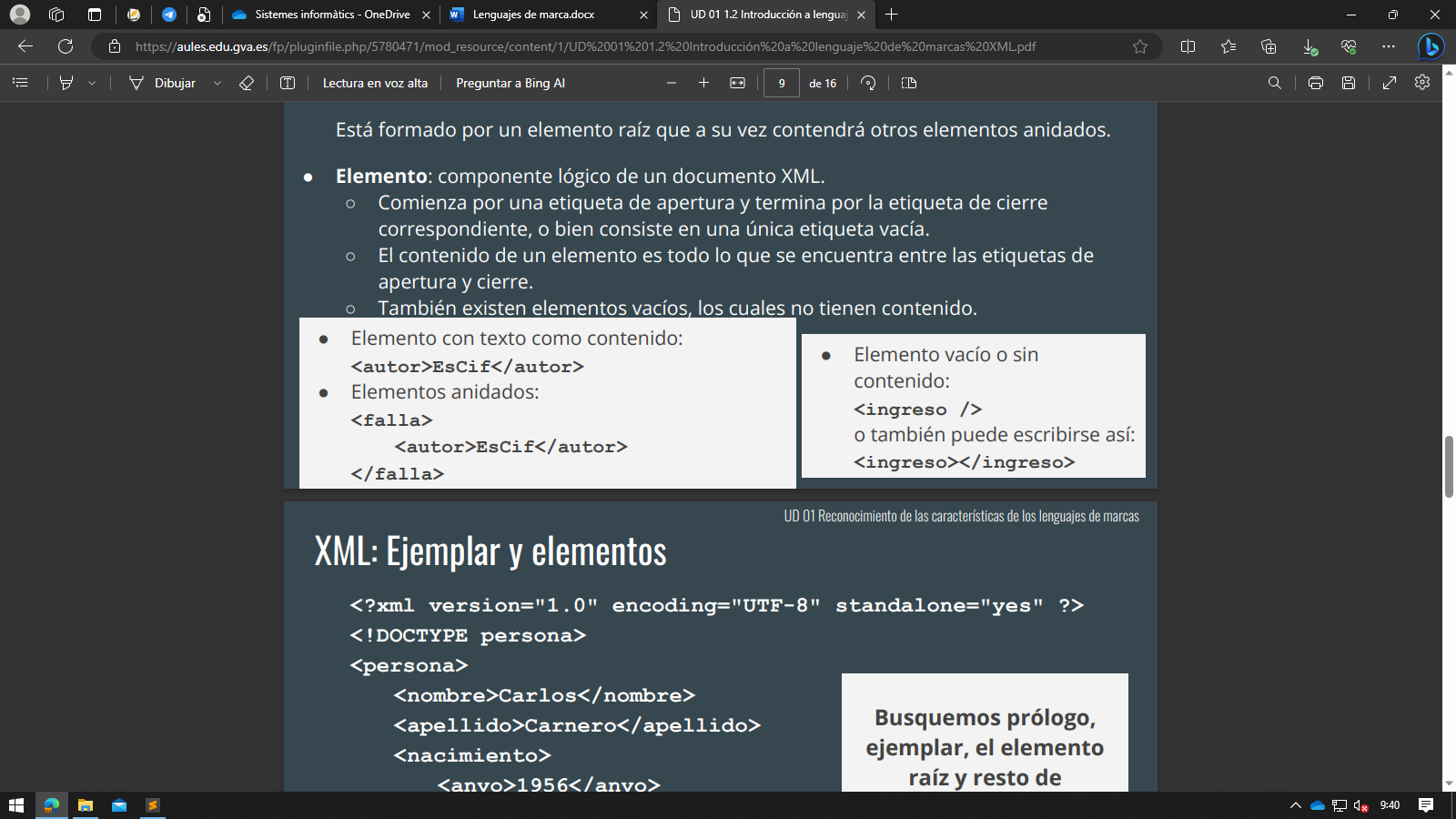




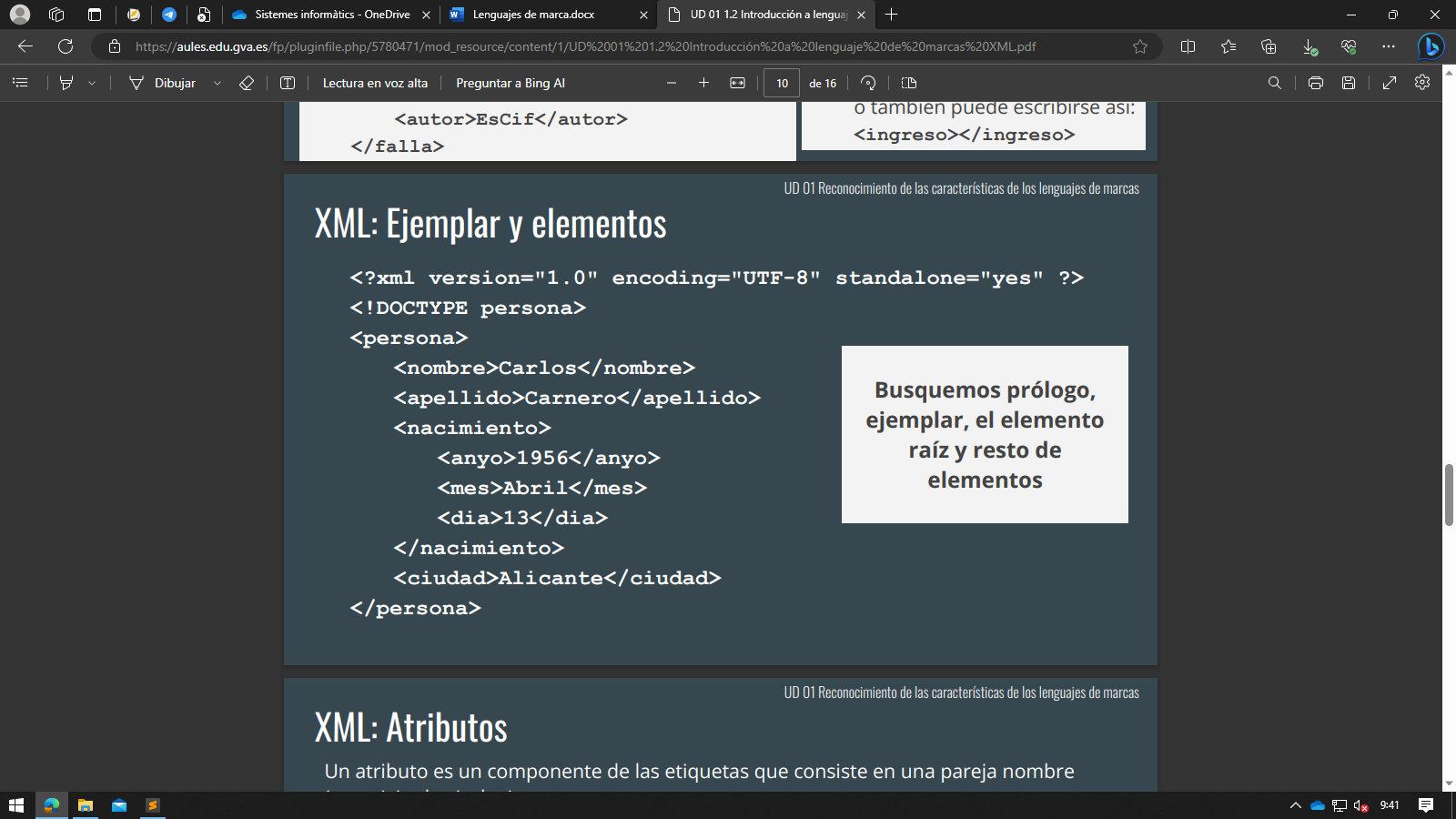
* + Opcional
    - Si se escribe el atributo “version” es obligatorio
  + Contiene:
    - Declaración XML
    - Declaración del tipo de documento
  + Los atributos “encoding” y “standalone” son opcionales y sus valores por defecto son “UTF-8” y “no”
  + El atributo “encoding” siempre después de versión
  + El atributo “standalone” en último lugar



* Ejemplar: contenido
  + Contiene los datos reales del documento XML
  + Está formado por un elemento raíz que a su vez contendrá otros elementos anidados
  + Elemento:
    - Componente lógico de un documento XML
    - Comienza por una etiqueta de apertura y termina por la etiqueta de cierre correspondiente, o bien consiste en una única etiqueta vacía
    - El contenido de un elemento es todo lo que se encuentra entre las etiquetas de apertura y cierre
    - También existen elementos vacíos, los cuales no tienen contenido



* Ejemplo:



* + 2 primeras líneas: prólogo
  + Resto: ejemplar
* Atributos
  + Componente de las etiquetas que consiste en una pareja name/value
  + Se puede encontrar en las etiquetas de apertura o en las etiquetas vacías
  + No puede haber dos atributos con el mismo nombre
  + Sintaxis: nombreAtributo=”valorAtributo”