

Lenguaje de marcas y gestión de la información

# Tema 1

Introducción a los lenguajes de marcas

1º DAW / DAM



# Índice

## Tema 1- Introducción a los lenguajes de marcas

---

1. Lenguajes de marcas
2. Evolución de los lenguajes de marcas
  1. GML
  2. SGML
  3. HTML
  4. XML
  5. JSON
3. Herramientas
  1. Editores
  2. Procesadores
4. XML: Estructura y sintaxis
5. JSON

**Antes de comenzar...**

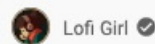
**¿Podrías identificar y clasificar los distintos fragmentos de información de los siguientes elementos?**

## Antes de comenzar...



### lofi hip hop radio - beats to relax/study to

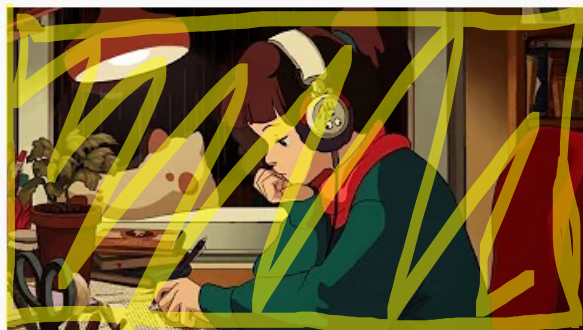
21.454 usuarios



Thank you for listening, I hope you will have a good time here | Get the latest vinyl (limited edition) ...

**EN DIRECTO**

## Antes de comenzar...



lofi hip hop radio - beats to relax/study to

21.454 usuarios



Lofi Girl



Thank you for listening, I hope you will have a good time here | Get the latest vinyl (limited edition) ..

EN DIRECTO

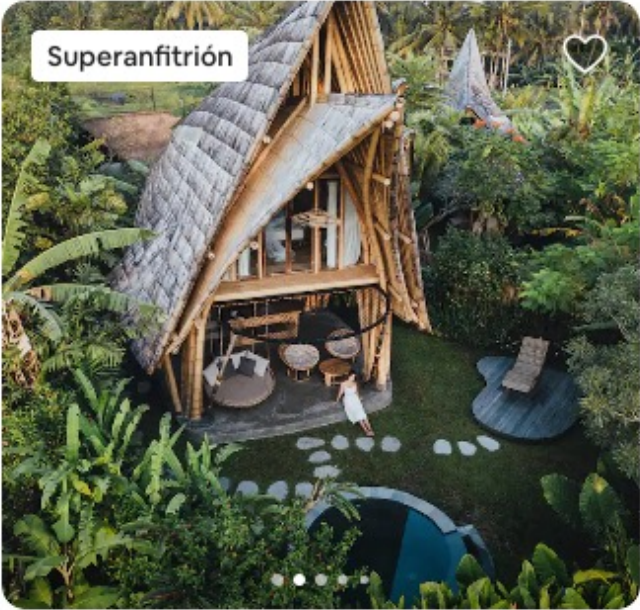
## Antes de comenzar...



## Antes de comenzar...



## Antes de comenzar...



Superanfitrión

**Choza en Ubud** ★ 4,8 (103)

Bali Bamboo House | Rescape Ubud ...

Anfitrión particular

16-21 mar

**103 € noche** · 512 € en total



## Antes de comenzar...



# 1. Lenguajes de marcas

## 1.1 Definición

Un lenguaje de marcas es **un modo de codificar un documento** donde, junto con el texto, se **incorporan etiquetas**, marcas o anotaciones **con información adicional** relativa a la estructura del texto o su formato de presentación. Es decir, es un tipo de lenguajes que nos permiten codificar o señalar qué tipo de información está contenida dentro de un documento. Ejemplo: los distintos campos que formarían una carta:

```
<carta>
<fecha>22/11/2006</fecha>
<presentacion>Estimado cliente:</presentacion>
<contenido>bla bla bla bla ...</contenido>
<firma>Don Jose Gutiérrez González</firma>
</carta>
```

De manera general, los lenguajes de marcas se pueden clasificar en: lenguajes de marcas de presentación, de procedimientos y descriptivos o semánticos.

# 1. Lenguajes de marcas

## 1.2 Elementos:

### Etiquetas

Las etiquetas definen el inicio y el fin de un fragmento con metainformación

Se escriben encerradas entre los signos `<` y `>`

Se utiliza una de inicio y otra de cierre y se distinguen por la barra inclinada del cierre:

**`<etiqueta>`** contenido **`</etiqueta>`**

# 1. Lenguajes de marcas

## 1.2 Elementos:

### DTD

Todo lenguaje de marcas está definido en un documento denominado **DTD** (Document Type Definition)

En el DTD se establecen:

- las marcas
- los elementos utilizados por dicho lenguaje
- sus correspondientes etiquetas y atributos
- su sintaxis y normas de uso

# Índice

## Tema 1- Introducción a los lenguajes de marcas

---

1. Lenguajes de marcas
- 2. Evolución de los lenguajes de marcas**
  1. GML
  2. SGML
  3. HTML
  4. XML
  5. JSON
3. Herramientas
  1. Editores
  2. Procesadores
4. XML: Estructura y sintaxis
5. JSON

## 2. Evolución de los lenguajes de marcas

### Pre-GML:

Años 70

Los lenguajes de marcas surgieron, inicialmente, como lenguajes formados por el conjunto de códigos de formato que los procesadores de texto introducen en los documentos para dirigir el proceso de presentación (impresión) mediante una impresora.

```
<times 14><color verde><centrado> Este texto es un  
ejemplo para mostrar la utilización primitiva de las  
marcas</centrado></color></times 14>  
<color granate><times 10><cursiva>Para realiza este  
ejemplo se utilizan etiquetas de nuestra invención.  
</cursiva> Las partes importantes del texto pueden  
resaltarse usando la <negrita>negrita</negrita>, o el  
<subrayar>subrayado</subrayar></times 10></color>
```

Resultado:

Este texto es un ejemplo para mostrar la utilización  
primitiva de las marcas

*Para realiza este ejemplo se utilizan etiquetas de nuestra  
invención. Las partes importantes del texto pueden  
resaltarse usando la **negrita**, o el subrayado*

## 2. Evolución de los lenguajes de marcas

### GML (Generalized Markup Language)

IBM necesitaba estandarizar el formato de los documentos

### SGML (Standard Generalized Markup Language)

En 1986 ISO 8879

Muy complejo y necesitaba software costoso

```
<email>
  <remitente>
    <persona>
      <nombre> Tipo </nombre>
      <apellido> De Incógnito </apellido>
    </persona>
  </remitente>
  <destinatario>
    <direccion> mail@mail.com </direccion>
  </destinatario>
  <asunto>Sand dunes</asunto>
  <mensaje> Lorem ipsum dolor sit amet </mensaje>
</email>
```

SGML Simplificado

## 2. Evolución de los lenguajes de marcas

### HTML (HyperText Markup Language)

1989 Tim Berners-Lee creó el World Wide Web

HTML es una versión simplificada del SGML y es combinación de

- ASCII
- SGML

#### Limitaciones:

- No soporta tareas de impresión y diseño.
- El lenguaje no es flexible, ya que las etiquetas son limitadas.
- No permite mostrar contenido dinámico.
- La estructura y el diseño están mezclados en el documento.

```
<html>
  <head>
    <title>Hola mundo</title>
  </head>
  <body>
    <h1>Mi primera página web</h1>
    <p>Lorem ipsum dolor sit, amet
      consectetur adipisicing elit.
    </p>
  </body>
</html>
```

HTML



## 2. Evolución de los lenguajes de marcas

### XML (eXtensible Markup Language)

En 1998 el W3C establece el estándar XML:  
no incluye ninguna información relativa al  
diseño

#### Características:

- Permitir definir etiquetas propias.
- Permitir asignar atributos a las etiquetas.
- Utilizar un esquema para definir de forma exacta las etiquetas y los atributos.
- La estructura y el diseño son independientes

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE libro>
<libro>
  <titulo>XML practico </titulo>
  <autor>SebastienLecomte</autor>
  <autor>Thierry Boulanger</autor>
  <editorial>Ediciones Eni</editorial>
  <isbn>978-2-7460-4958-1</isbn>
  <edicion>1</edicion>
  <paginas>347</paginas>
</libro>
```

XML

## 2. Evolución de los lenguajes de marcas

### XML (eXtensible Markup Language)

Es un conjunto de estándares relacionados entre si:

- XSL (extensible Style Language)
- XML Linking Language (Xpath, Xlink y Xpointer)
- XML Namespaces (contexto de las marcas)
- XML Schemas (restricciones, actualmente DTD son los más usados)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE libro>
<libro>
  <titulo>XML practico </titulo>
  <autor>SebastienLecomte</autor>
  <autor>Thierry Boulanger</autor>
  <editorial>Ediciones Eni</editorial>
  <isbn>978-2-7460-4958-1</isbn>
  <edicion>1</edicion>
  <paginas>347</paginas>
</libro>
```

XML

## 2. Evolución de los lenguajes de marcas

### XML vs HTML

XML	HTML
<ul style="list-style-type: none"><li>• Es un perfil de SGML.</li><li>• Especifica cómo deben definirse conjuntos de etiquetas aplicables a un tipo de documento.</li><li>• Modelo de hiperenlaces complejo.</li><li>• El navegador es una plataforma para el desarrollo de aplicaciones.</li><li>• Fin de la guerra de los navegadores y etiquetas propietarias.</li></ul>	<ul style="list-style-type: none"><li>• Es una aplicación de SGML.</li><li>• Aplica un conjunto limitado de etiquetas sobre un único tipo de documento.</li><li>• Modelo de hiperenlaces simple.</li><li>• El navegador es un visor de páginas.</li><li>• El problema de la "no compatibilidad" y las diferencias entre navegadores ha alcanzado un punto en el que la solución es difícil</li></ul>

## 2. Evolución de los lenguajes de marcas

### JSON

Javascript Notation Objects

Formato ligero de intercambio de datos.

Vienen a sustituir al XML

Los objetos son listas de parejas nombre / valor. El nombre y el valor están separados por dos puntos : y las parejas están separadas por comas. Los objetos se escriben entre llaves { } y los nombres de las parejas se escriben siempre entre comillas dobles.

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": 10021  
  },  
  "phoneNumbers": [  
    "212-732-1234",  
    "646-123-4567"  
  ]  
}
```

# Índice

## Tema 1- Introducción a los lenguajes de marcas

---

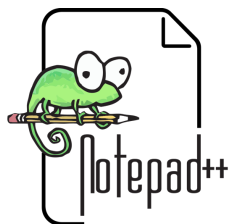
1. Lenguajes de marcas
2. Evolución de los lenguajes de marcas
  1. GML
  2. SGML
  3. HTML
  4. XML
  5. JSON
- 3. Herramientas**
  - 1. Editores**
  - 2. Procesadores**
4. XML: Estructura y sintaxis
5. JSON

### 3. Herramientas

#### Editores de Texto

Para poder crear documentos.

Ejemplos: Notepad++, Atom, SublimeText, VSCode



### 3. Herramientas

#### Procesadores:

Para poder interpretarlos. Cualquier navegador funciona como interprete, aunque Google Chrome y Firefox son más recomendados por su oferta de funcionalidades.



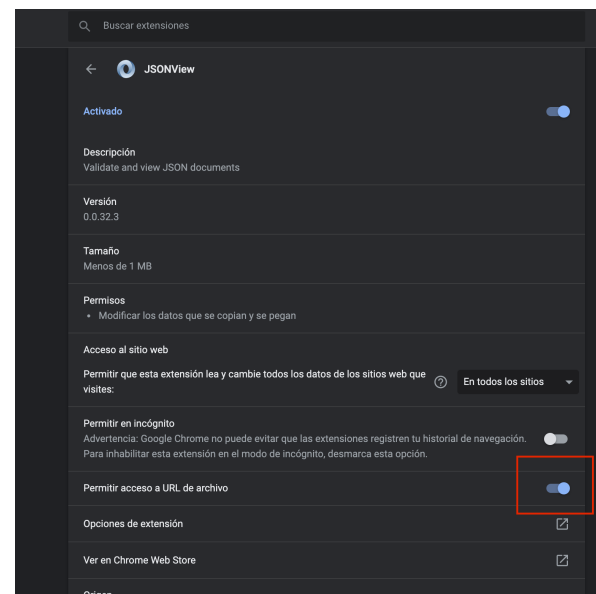
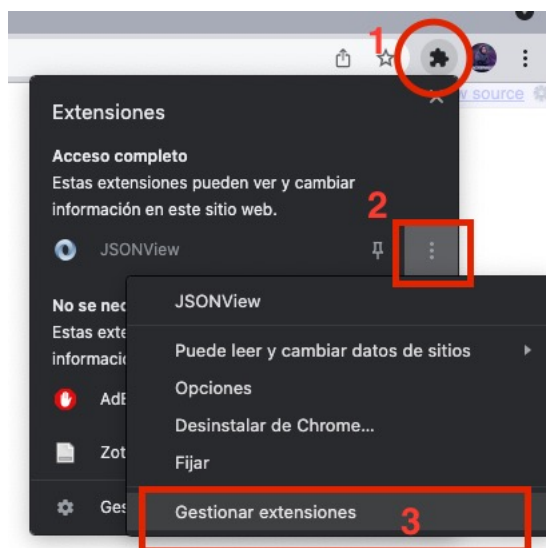
### 3. Herramientas

#### Procesadores:

JSON View es una extensión para Google Chrome que nos permite visualizar el JSON de una manera más ordenada.



[Link descarga](#)





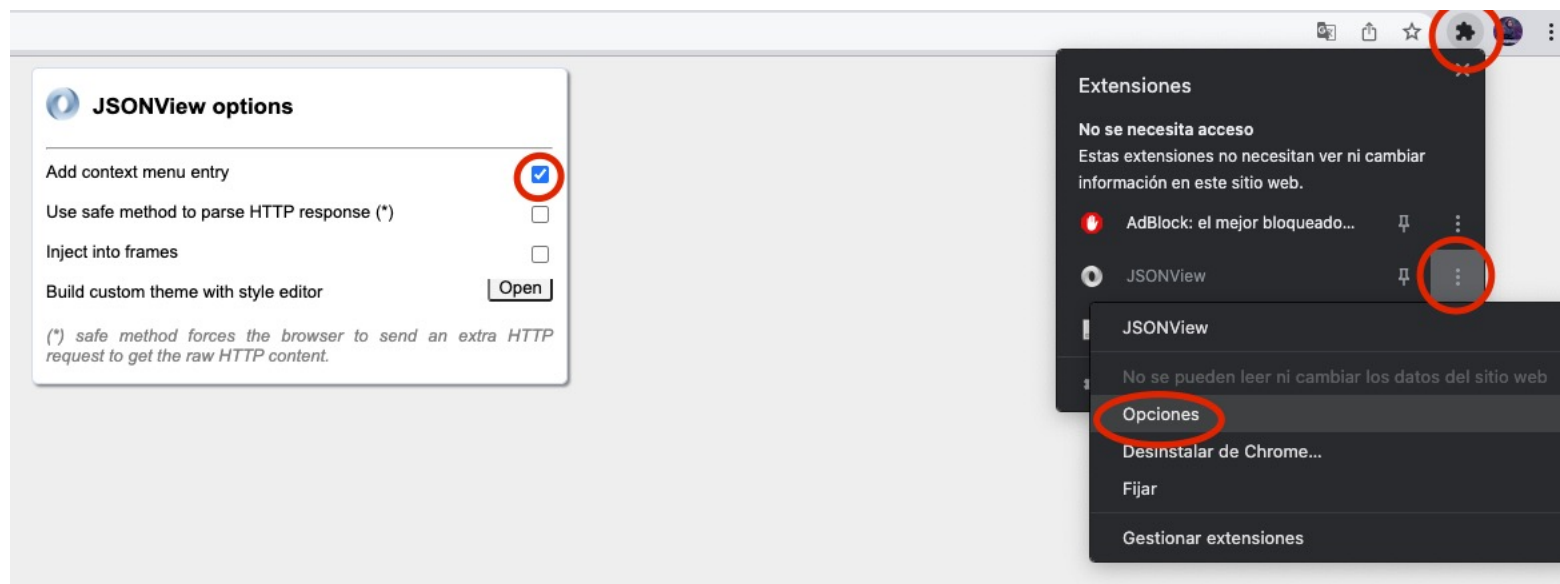
### 3. Herramientas

#### Procesadores:

JSON View es una extensión para Google Chrome que nos permite visualizar el JSON de una manera más ordenada.



[Link descarga](#)



# Índice

## Tema 1- Introducción a los lenguajes de marcas

---

1. Lenguajes de marcas
2. Evolución de los lenguajes de marcas
  1. GML
  2. SGML
  3. HTML
  4. XML
  5. JSON
3. Herramientas
  1. Editores
  2. Procesadores
- 4. XML: Estructura y sintaxis**
5. JSON

## 4. XML: Estructura y sintaxis

### Características XML:

- Dado que XML se concibió para trabajar en la Web, es directamente **compatible con** protocolos que ya funcionan, como **HTTP** y los **URL**.
- Todo documento que verifique las reglas de XML **está conforme con SGML**.
- No se requieren **conocimientos de programación** para realizar tareas sencillas en XML.
- Los documentos XML son **fáciles de crear**.
- La difusión de los documentos XML está asegurada ya que **cualquier procesador** de XML puede leer un documento de XML.
- El marcado de XML es **legible** para los humanos.
- El diseño XML es **formal y conciso**.
- XML es extensible, **adaptable** y aplicable a una gran variedad de situaciones.
- XML es **orientado a objetos**.
- Todo documento XML se compone exclusivamente de **datos de marcado y datos carácter entremezclados**.

## 4. XML: Estructura y sintaxis

### Partes principales

- Prólogo(opcional):

- Declaración del XML
  - Versión
  - Codificación
  - Autonomía
- Declaración de tipo de documento

- Ejemplar

- Elementos
  - Atributos

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE libro>
<libro>
  <titulo>XML practico </titulo>
  <autor>Sebastien Lecomte</autor>
  <autor>Thierry Boulanger</autor>
  <editorial>Ediciones Eni</editorial>
  <isbn>978-2-7460-4958-1</isbn>
  <edicion>1</edicion>
  <paginas>347</paginas>
</libro>
```

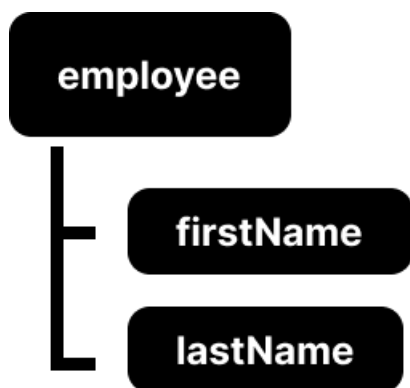
## 4. XML: Estructura y sintaxis

Estándar ISO	Código de país
<b>UTF-8 (Unicode)</b>	Conjunto de caracteres universal
<b>ISO -8859-1 (Latin-1)</b>	Europa occidental, Latinoamérica
<b>ISO -8859-2 (Latin-2)</b>	Europa central y oriental
<b>ISO -8859-3 (Latin-3)</b>	Sudoeste de Europa
<b>ISO -8859-4 (Latin-4)</b>	Países Escandinavos, Bálticos
<b>ISO -8859-5</b>	Cirílico
<b>ISO -8859-6</b>	Árabe
<b>ISO -8859-7</b>	Griego
<b>ISO -8859-8</b>	Hebreo
<b>ISO -8859-9</b>	Turco
<b>ISO-8859-10</b>	Lapón. Nórdico, esquimal
<b>EUC-JP oder Shift_JIS</b>	Japonés

Posibles valores propiedad encoding

## 4. XML: Estructura y sintaxis

### Conversión de información a un .xml



```
<employee>
  <firstName>Carl</firstName>
  <lastName>Johnson</lastName>
</employee>
```



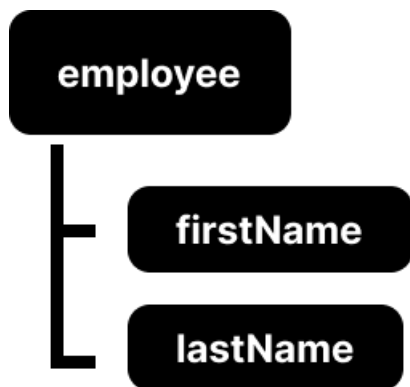
```
<employee>
  <first-name>Carl</first-name>
  <last-name>Johnson</last-name>
</employee>
```



```
<employee>
  <first_name>Carl</first_name>
  <last_name>Johnson</last_name>
</employee>
```

## 4. XML: Estructura y sintaxis

### Conversión de información a un .xml



✗

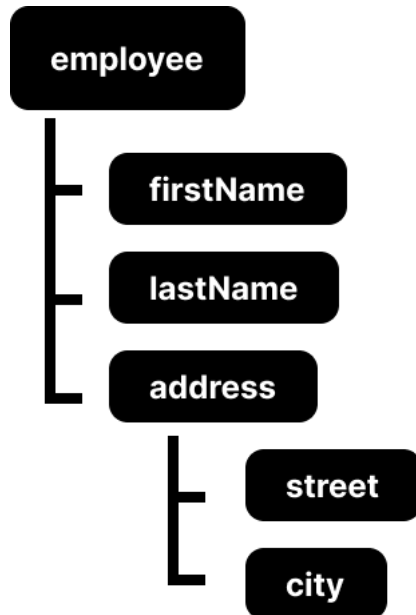
```
<employee >
  <firstName>Carl</first-Name>
  <lastName>Johnson<lastName>
</employee>
```

✗

```
<employee >
  <first Name>Carl</first Name>
  <last Name>Johnson</last name>
</employee>
```

## 4. XML: Estructura y sintaxis

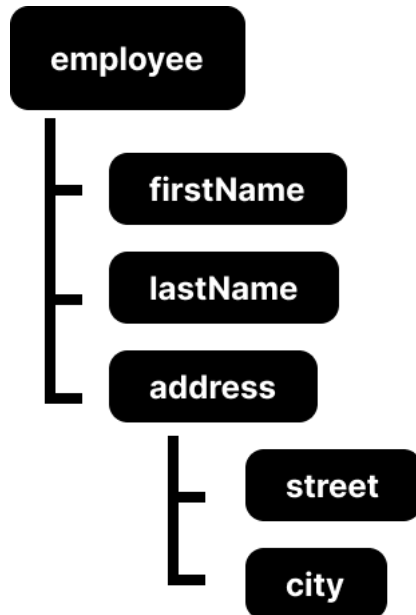
### Conversión de información a un .xml





## 4. XML: Estructura y sintaxis

### Conversión de información a un .xml



```
<employee>
  <first_name>Carl</first_name>
  <last_name>Johnson</last_name>
  <address>
    <street>
      Grove St
    </street>
    <city>Los Santos</city>
  </address>
</employee>
```

## 4. XML: Estructura y sintaxis

### Atributos

Es otra forma de añadir contenido a los elementos

#### Precauciones:

- No permiten varios valores
- No permiten estructuras anidadas
- No son fácilmente adaptables a cambios futuros

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?
<!DOCTYPE biblioteca >
<biblioteca>
  <ejemplar tipo Ejem="libro" titulo="XML práctico" editorial="Ediciones Eni">
    <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"></libro> </tipo>
    <autor nombre="Sebastien Lecomte"></autor>
    <autor nombre="Thierry Boulanger"></autor>
    <autor nombre="Angel Belinchon Calleja" funcion="traductor"></autor>
    <prestado lector="Pepito Grillo">
      <fecha_pres dia="13" mes="mar" año="2009"></fecha_pres>
      <fecha_devol dia="21" mes="jun" año="2009"></fecha_devol>
    </prestado>
  </ejemplar>
</biblioteca>
```

Ejemplo de uso de atributos

## 4. XML: Estructura y sintaxis

### Namespaces

Los namespaces o dominios, permiten importar múltiples ficheros sin que colisionen los nombres de las propiedades que se llamen igual.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"
<!DOCTYPE alumnos>
<alumnos>
  <nombre>Fernando Fernández González</nombre>
  <nombre>Isabel González Fernández</nombre>
  <nombre>Ricardo Martínez López</nombre>
</alumnos>
```

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE profesores>
<profesores>
  <nombre>Pilar Ruiz Pérez</nombre>
  <nombre>Tomás Rodríguez Hernández</nombre>
</profesores>
```

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE miembros>
<alumnos xmlns:alumnos="http://ASIR/alumnos">
<profesores xmlns:profesores="http://ASIR/profesores">
<asistentes>
  <alumnos:nombre>Fernando Fernández González</alumnos:nombre>
  <alumnos:nombre>Isabel González Fernández</alumnos:nombre>
  <alumnos:nombre>Ricardo Martínez López</alumnos:nombre>
  <profesores:nombre>Pilar Ruiz Pérez</profesores:nombre>
  <profesores:nombre>Tomás Rodríguez Hernández</profesores:nombre>
</asistentes>
```

# Índice

## Tema 1- Introducción a los lenguajes de marcas

---

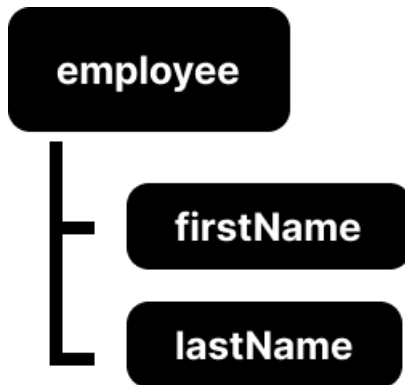
1. Lenguajes de marcas
2. Evolución de los lenguajes de marcas
  1. GML
  2. SGML
  3. HTML
  4. XML
  5. JSON
3. Herramientas
  1. Editores
  2. Procesadores
4. XML: Estructura y sintaxis
- 5. JSON**

## 5. JSON

### JSON

Las etiquetas en los objetos JSON, se sustituyen por pares de valores entre comillas dobles. Cada pareja de valores representa una clave y un valor (nombre de la etiqueta y contenido en xml). Entre si están separados por dos puntos, y de la siguiente pareja de clave-valor por una coma. Todo el objeto tiene que ir entre llaves { }.

employee.json

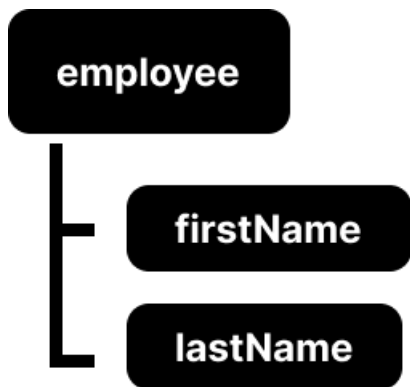


```
{  
  "firstName": "Carl",  
  "lastName": "Johnson"  
}
```

## 5. JSON

### JSON

Las etiquetas en los objetos JSON, se sustituyen por pares de valores entre comillas dobles. Cada pareja de valores representa una clave y un valor (nombre de la etiqueta y contenido en xml). Entre si están separados por dos puntos, y de la siguiente pareja de clave-valor por una coma. Todo el objeto tiene que ir entre llaves { }.



employee.json

```
{  
  "employee": {  
    "firstName": "Carl",  
    "lastName": "Johnson"  
  }  
}
```

alternativamente se podría poner así 🙅  
(aunque para acceder a los datos de employee, tendríamos que acceder a la propiedad employee dentro del archivo employee, lo cual resulta un poco redundante )

## 5. JSON

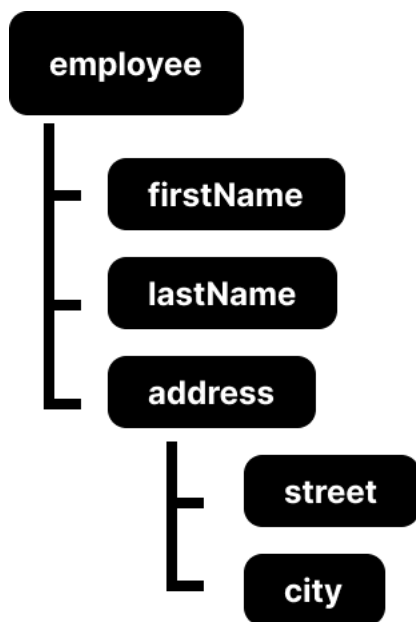
### JSON: estructuras avanzadas

Los valores de las propiedades de los objetos JSON también pueden consistir en:

- **Objetos:** conjunto de propiedades anidadas (véase la propiedad “address” de la imagen)
- **Arrays:** conjunto de valores entre corchetes y separados por comas (propiedad “phoneNumbers”)
- **Arrays de objetos:** combinación de las dos anteriores, se verán en detalle un poco más adelante.

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": 10021  
  },  
  "phoneNumbers": [  
    "212-732-1234",  
    "646-123-4567"  
  ]  
}
```

## 5. JSON

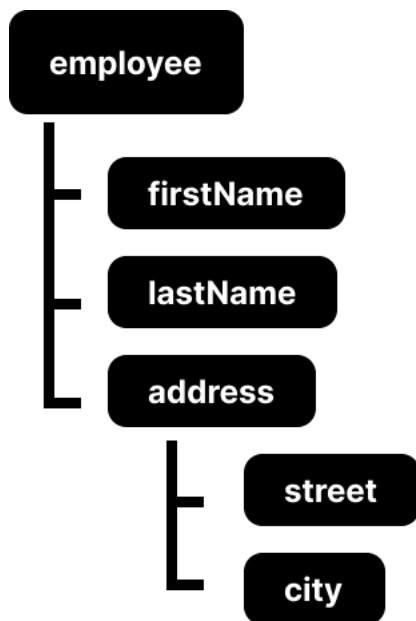


```
employee.json
{
  "firstName": "Carl",
  "lastName": "Johnson",
  "address": {
    "street": "Grove St",
    "city": "Los Santos"
  }
}
```

La propiedad `address` **es a su vez un objeto**, es decir, **está dividida a su vez en dos propiedades**: `street` y `city`. Para indicar propiedades con "sub-propiedades" se utilizan llaves `{ }`



## 5. JSON

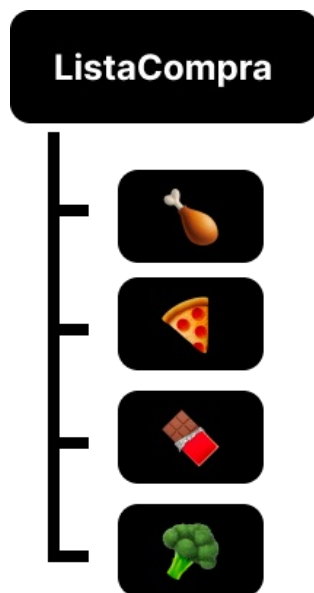


employee.json

```
{  
  "employee": {  
    "firstName": "Carl",  
    "lastName": "Johnson",  
    "address": {  
      "street": "Grove St",  
      "city": "Los Santos"  
    }  
  }  
}
```

alternativamente se podría poner así 🙅  
(aunque nuevamente ocurre lo de la redundancia)

## 5. JSON

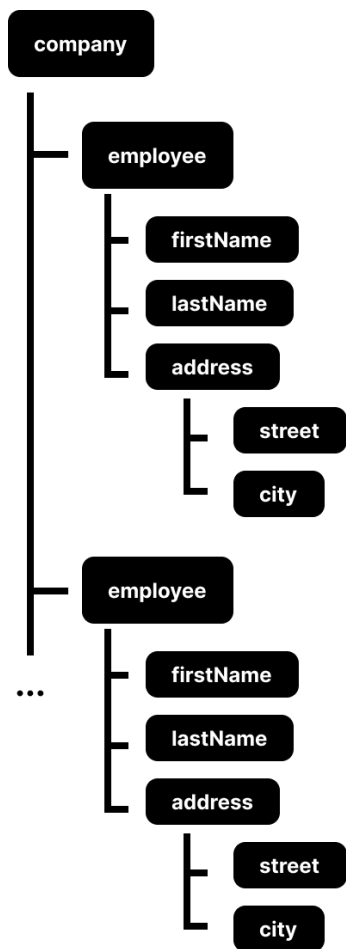


listaCompra.json

```
{  
  "listaCompra": [  
    "🍗",  
    "🍕",  
    "🍫",  
    "🥦"  
  ]  
}
```

**Las listas (arrays)** de elementos de igual importancia jerárquica van entre corchetes [ ], con el contenido entre comillas dobles y separados por comas de los siguientes elementos.

## 5. JSON

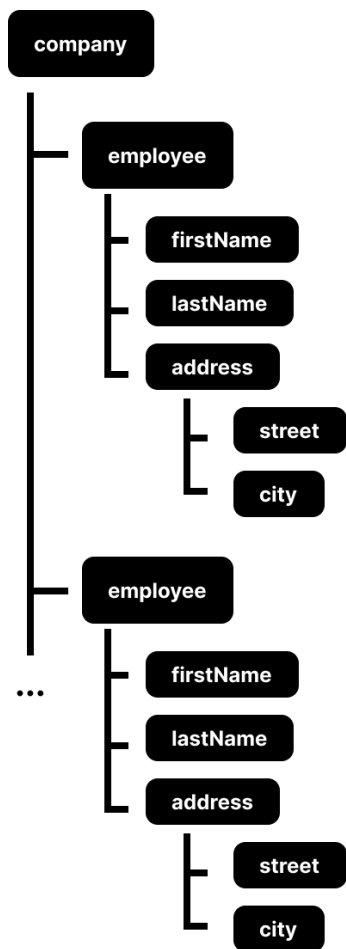


company.json

```
{
  "company": [
    "employee",
    "employee"
  ]
}
```

Las listas de objetos (o **arrays de objetos**) son parecidos a los arrays normales, pero en vez de tener elementos simples (como Strings, número, etc) tienen objetos compuestos por múltiples propiedades y sub-propiedades. Cada objeto está contenido por las llaves, y separado del siguiente por una coma

## 5. JSON



company.json

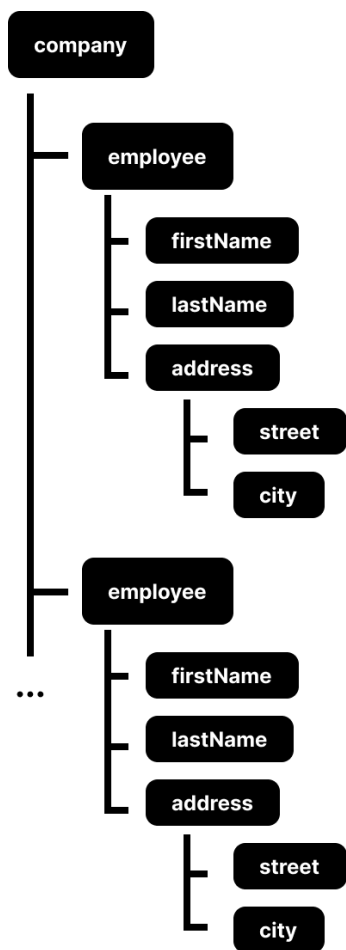
```
{
  "company": [
    employee,
    employee
  ]
}
```

```
{
  "firstName": "Carl",
  "lastName": "Johnson",
  "address": {
    "street": "Grove St",
    "city": "Los Santos"
  }
}
```

```
{
  "firstName": "Big",
  "lastName": "Smoke",
  "address": {
    "street": "Grove St",
    "city": "Los Santos"
  }
}
```

Las listas de objetos (o **arrays de objetos**) son parecidos a los arrays normales, pero en vez de tener elementos simples (como Strings, número, etc) tienen objetos compuestos por múltiples propiedades y sub-propiedades. Cada objeto está contenido por las llaves, y separado del siguiente por una coma

## 5. JSON



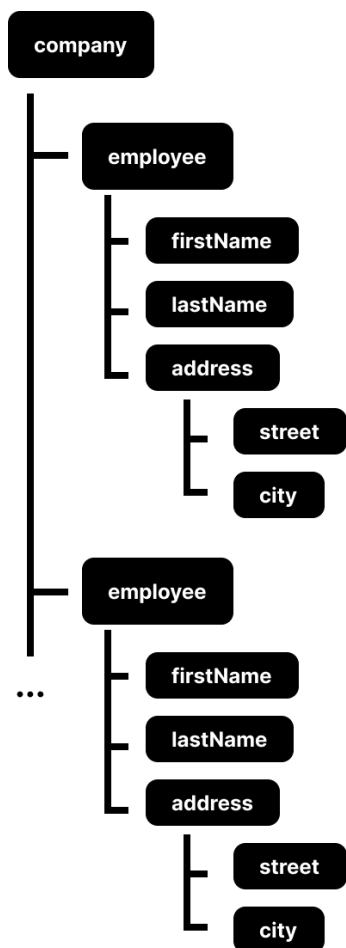
company.json

```
{
  "company": [
    {
      "firstName": "Carl",
      "lastName": "Johnson",
      "address": {
        "street": "Grove St",
        "city": "Los Santos"
      }
    },
    {
      "firstName": "Big",
      "lastName": "Smoke",
      "address": {
        "street": "Grove St",
        "city": "Los Santos"
      }
    }
  ]
}
```

👉 Resultado final ✓

Las listas de objetos (o **arrays de objetos**) son parecidos a los arrays normales, pero en vez de tener elementos simples (como Strings, número, etc) tienen objetos compuestos por múltiples propiedades y sub-propiedades. Cada objeto está contenido por las llaves, y separado del siguiente por una coma

## 5. JSON



company.json

```
{  
  "company": [  
    {  
      "firstName": "Carl",  
      "lastName": "Johnson",  
      "address": {  
        "street": "Grove St",  
        "city": "Los Santos"  
      }  
    },  
    {  
      "firstName": "Big",  
      "lastName": "Smoke",  
      "address": {  
        "street": "Grove St",  
        "city": "Los Santos"  
      }  
    }  
  ]  
}
```

Esto indica que empieza una lista

Primer objeto

Coma para separar

Segundo objeto

Esto indica que acaba la lista

Las listas de objetos (o **arrays de objetos**) son parecidos a los arrays normales, pero en vez de tener elementos simples (como Strings, número, etc) tienen objetos compuestos por múltiples propiedades y sub-propiedades. Cada objeto está contenido por las llaves, y separado del siguiente por una coma

# Resumen

## Tema 1- Introducción a los lenguajes de marcas

---

1. Lenguajes de marcas
2. Evolución de los lenguajes de marcas
  1. GML
  2. SGML
  3. HTML
  4. XML
  5. JSON
3. Herramientas
  1. Editores
  2. Procesadores
4. XML: Estructura y sintaxis
5. JSON