Apunts UD1: INTRODUCCIÓ

lloc: CIFP Francesc de Borja Moll
Curs: Llenguatges de marques i sistemes de gestió d'informació
Llibre: Apunts UD1: INTRODUCCIÓ
Imprès per: Alejo Morell Bethencourt
Data: dilluns, 5 octubre 2020, 17:17

Taula de continguts

1. Introducció als llenguatges de marques

Una de les tasques bàsiques que fan els ordinadors és emmagatzemar la informació que els proporcionem per poder ser processada després. Aquesta informació pot ser de molts de tipus diferents (text, imatges, vídeos, música…) però el més important serà de quina manera l'emmagatzema l'ordinador per poder-la tractar posteriorment de manera eficient per generar més informació.

RECORDATORI:
Una definició senzilla del que és un ordinador podria ser: "Una màquina electrònica que rep i processa dades per convertir-les en informació útil".

RECORDATORI:
Les dades són representacions d'aspectes del món real i se solen recollir per fer càlculs, mostrar-les, organitzar-les, ..., amb l'objectiu que posteriorment algú en pugui fer alguna cosa amb elles, per exemple: prendre decisions, generar noves dades, …

RECORDATORI:
Podríem definir de forma molt general un sistema informàtic com un sistema on les úniques tasques que es desenvolupen consisteixen a emmagatzemar dades per processar-les per mitjà d'un programa que o bé aportarà algun tipus d'informació o bé es faran servir de nou per generar noves dades.

Característiques de les dades

Les característiques més importants de les dades en podem basar en tres aspectes:

- A qui van dirigides.
- La possibilitat de reutilitzar.
- Que es puguin compartir.

    - **A qui van dirigides:**

      Si s'intenta ser una mica més pràctic, es veurà que realment les dades tindran una forma o una altra en funció del destinatari a qui vagin dirigides:

      - Dades destinades a les persones: Aquestes dades hauran de tenir alguna estructura concreta, amb uns formats determinats, per exemple, hi apareixeran títols, caràcters en negreta, ... Generalment no serà necessari conèixer quin significat tenen les dades, ja que la interpretació es deixarà al lector.

      - Dades destinades als programes: Els programes generalment no necessiten que les dades tinguin informació sobre com s'han de representar, simplement serà necessari que siguin fàcilment identificables, que quedi clar de quin tipus són i que hi hagi alguna manera de determinar el que signifiquen per poder-les tractar automàticament.

    - **La possibilitat de reutilizar:**

      Moltes vegades, les dades es voldran reutilitzar per poder fer tasques diferents. Un error corrent és emmagatzemar-les en funció d'una tasca concreta, ja que això pot provocar que posteriorment sigui molt més complicat fer-les servir per fer altres tasques. Per tant, és bàsic disposar d'un sistema d'emmagatzematge que permeti aconseguir que les dades puguin ser reutilitzades fàcilment i si pot ser, que puguin ser reutilitzades tant per les persones com pels programes.

- **Compartició de les dades**

En els inicis de la informàtica, els ordinadors generaven i processaven la informació en el mateix lloc. Però l'aparició dels ordinadors personals, l'eclosió de les xarxes i, sobretot, l'èxit d'Internet, ha creat tota una sèrie de problemàtiques que fins al moment no existien:

> **IMPORTANT:**
>
> "Les dades generades en un lloc ara poden ser consumides en un lloc totalment diferent, per exemple, en sistemes operatius totalment diferents, en màquines que poden funcionar de maneres molt diverses".

Per tant, en un sistema informàtic modern s'ha de tenir en compte aquesta possibilitat a l'hora d'emmagatzemar dades. Hi ha la possibilitat que aquestes dades siguin compartides i, per tant, és necessari emmagatzemar-les d'alguna manera per evitar tenir problemes per emprar-les en sistemes diferents.

Emmagatzematge de dades en ordinadors

En l'actual arquitectura dels ordinadors (*Von Neumann*), la informació que s'hi pot emmagatzemar sempre es representa mitjançant uns i zeros (1, 0), és a dir, emprant el sistema binari. Això fa que per representar qualsevol classe de dades (imatges, vídeos, text…) sigui necessari fer algun tipus de procés que converteixi les dades a una representació en format binari.

Tradicionalment en els ordinadors les dades s'organitzen de dues maneres:

- **Dades binàries**

Emmagatzemar les dades de manera binària és la manera natural d'emmagatzemar dades en ordinadors. Són una tira de bits un darrere l'altre. Les dades en format binari tenen una sèrie de característiques que les fan ideals per als ordinadors:

  - Generalment estan optimitzades per ocupar només l'espai necessari.
  - Els ordinadors les llegeixen fàcilment.
  - Poden tenir estructura.
  - És relativament fàcil afegir-hi metadades (dades que defineixen i descriuen altres dades).
  - Estan disponibles immediatament per fer càlculs numèrics, ja que realment es tracta de nombres. No serà necessari fer cap transformació per poder emprar aquests números en qualsevol càlcul.

Si un programa vol emprar les dades binàries directament, necessitarà conèixer la mida en bits i, sobretot, conèixer de quina manera s'hi ha emmagatzemat la informació. Per exemple: Per emmagatzemar el nombre 150 només serà necessari convertir aquest valor decimal a la seva representació en binari i emmagatzemar-lo. És trivial comprovar que pot ser emmagatzemat en un sol byte (8 bits):

valor decimal	valor binari
150	10010110

Un problema és que les dades en format binari estan pensades per ser llegides per màquines, però no per les persones, de manera que són ideals per ser emmagatzemades en màquines, van bé per a la comunicació d'informació entre màquines, però en canvi perquè un humà les pugui fer servir serà necessari tenir un programa específic per llegir-les.

- **Dades de text**

Per solucionar el problema de recuperar les dades que hi ha en un fitxer, existeix una possibilitat que és fer el més obvi, fer el mateix que han fet les persones durant segles. Els humans en escriure ja estan fent servir una codificació i, per tant, si es fa servir la mateixa codificació, tindrem les dades en un format fàcil d'entendre i perquè es puguin emprar, per tant no hi haurà problemes perquè el codi pugui ser llegit pels programes.

AMPLIACIÓ:

- En fitxers binaris, el component d'informació més petit és el bit.
- En fitxers de text el component més petit és el caràcter.

Els fitxers de text emmagatzemen la informació lletra per lletra d'una manera similar a com ho faria una persona en escriure. Això fa que s'estigui generant una informació que es podrà llegir de la mateixa manera que es llegeix un document de paper. Per a un ordinador no hi ha gaire diferència a l'hora d'emmagatzemar els fitxers de text o els fitxers binaris, ja que els fitxers de text també són tires de bits. La diferència és que en els fitxers de text, els bits estan agrupats d'una manera estàndard i coneguda: un codi de caràcters.

AMPLIACIÓ:
Representar les dades en un ordinador en forma de text implica que per poder representar una paraula qualsevol a l'ordinador, prèviament haurà de ser codificada perquè pugui ser representada en binari (recordem que els ordinadors només poden representar dades en binari).
Aquesta codificació consisteix a determinar una quantitat de bits predefinida per marcar un caràcter, i posteriorment, s'associa un valor numèric a cada un dels caràcters.

1.1. Definició i classificació dels llenguatges de marques

Introducció

IMPORTANT:

Els llenguatges de marques o llenguatges de marcat són aquells que combinen dins un document, la informació (generalment textual) amb marques (o anotacions relatives a l'estructura del text o de la forma de representar-lo).

El llenguatge de marques és el que especifica quines seran aquestes marques possibles (etiquetes, elements, etc.) , on s'han de col·locar i el significat que tindrà cadascuna d'elles. A més, la presència de marques intercalades en el contingut fa explícita l'estructura del document o qualsevol informació addicional que es vulgui ressaltar.

Els documents que es creen amb el llenguatge de marques tenen com a avantatge la facilitat de creació i lectura. Això és gràcies al compliment d'estàndards d'emmagatzematge definits i públics, a la incorporació de metadades i a la definició de l'estructura de les dades.

Com que els fitxers de text sempre estan emmagatzemats en algun codi de caràcters conegut (per exemple: ASCII, UTF-8, etc.) s'aconsegueix que puguin ser transportats i llegits en qualsevol plataforma, sistema operatiu o programa que pugui interpretar aquests codis de caràcters. Per tant, els llenguatges de marques s'aprofitaran d'aquesta característica, en estar basats en el format de text. A més, també tindran l'avantatge que podran ser oberts i creats amb els programes d'edició de text estàndard. Des d'editors tan simples com el Bloc de notes dels sistemes Windows o el Gedit de sistemes Unix, fins a editors més complexo, passant per editors especialitzats en XML com Atom, Visual Studio Code, Sublime Text, Adobe Dreamweaver, Oxygen XML Editor, XML Copy Editor, etc, i també existeixen *frameworks* . També permeten definir les dades i la seva estructura de manera que sigui senzill per un programa poder-les interpretar.

Gràcies als avantatges que ofereixen els llenguatges de marques, aquests s'han convertit ràpidament en una de les maneres habituals de representar dades i es poden trobar contínuament en les tasques habituals amb ordinadors:

- L'exponent més popular és Internet –el Web–, que està basat totalment en els llenguatges de marques.
- Molts dels programes d'ordinador que es fan servir habitualment utilitzen en un moment o altre, algun llenguatge de marques per a emmagatzemar les seves dades de configuració o de resultats.

Les marques

Les marques són una sèrie de codis que s'incorporen als documents electrònics per determinar-ne el format, la manera com s'han d'imprimir, l'estructura de les dades, etc. Per tant, són anotacions que s'incorporen a les dades.

Les marques més emprades són les que estan formades per textos descriptius i estan envoltades dels símbols de "més petit" (<) i "més gran" (>) i normalment n'hi sol haver una al principi i una al final:

<marca>....</marca> de forma general

Aparició i evolució dels llenguatges de marques

- La idea del marcat procedeix de l'anglès "*marking up*", terme amb el qual es referien a la tècnica de marcar manuscrits amb llapis de color per fer anotacions com ara la tipografia a emprar en les impremtes. Aquest mateix terme s'ha utilitzat per als documents de text que contenen ordres o anotacions.
- Les possibles anotacions o indicacions incloses en els documents de text han donat lloc a llenguatges (entenent que en realitat són formats de document i no llenguatges en el sentit dels llenguatges de programació d'aplicacions) anomenats llenguatges de marques, llenguatges de marcat o llenguatges d'etiquetes.
- Es considera a Charles Goldfarb com el pare dels llenguatges de marques. Es tracta d'un investigador d'IBM que va proposar idees perquè els documents de text tenguessin la possibilitat d'indicar el seu format. Va contribuir a definir el llenguatge GML d'IBM, el qual va posar les bases del llenguatge SGML (pare de HTML i XML) ideat per Goldfarb.
- A finals dels anys 80 dins el CERN (*Conseil Européen pour la Recherche Nucléaire*) es va crear un llenguatge de marcat pensat per compartir informació usant les xarxes d'ordinadors i, de forma més general, a través d'Internet. Aquest llenguatge es basava en alguns principis de SGML i ho van denominar HTML (*Hyper-text de marques*). L'aparició d'aquest llenguatge va suposar d'alguna manera una revolució en la forma de compartir informació, gràcies principalment a la senzillesa de la seva sintaxi i del programari necessari per a interpretar-lo. En poc temps el llenguatge HTML es va estendre i va començar a créixer de forma a vegades descontrolada i gairebé sempre influenciat per raons merament comercials.
- A mitjans dels anys 90 el consorci W3C (*World Wide Web Consortium*) va començar una iniciativa per intentar dotar la web d'un llenguatge més potent i que pogués donar una estructurar semàntica a aquesta. Per a això es van marcar l'objectiu de crear un nou llenguatge de marques basat en SGML i que fos senzill com HTML. Finalment, l'any 1998, W3C va fer públic un nou estàndard que van denominar XML (*eXtended Markup Language*), més senzill que SGML i més potent que HTML.

Característiques dels llenguatges de marques

Els llenguatges de marques han destacat per una sèrie de característiques que els han convertit en els tipus de llenguatges més emprats en la informàtica actual per emmagatzemar i representar les dades. Entre les característiques més interessants que ofereixen els llenguatges de marques es troben:

- Que es basen en el text pla.
- Que permeten fer servir metadades.
- Que són fàcils d'interpretar i processar.
- Que són fàcils de crear i bastant flexibles per representar dades molt diverses.
- Les aplicacions d'Internet i molts dels programes d'ordinador que es fan servir habitualment, fan servir d'alguna manera o altra algun llenguatge de marques.

Avantatges dels llenguatges de marques

- Es poden interpretar directament perquè que fan servir el format de text.
- Són independents de la plataforma, del sistema operatiu o del programa.
- El fet que estiguin basats en format de text fa que siguin fàcils de crear i de modificar perquè només requereixen un simple editor de textos.
- Facilitat de procés: Permeten que el processament de les dades que contenen pugui ser automatitzat d'alguna manera, ja que el fitxer conté l'estructura de les dades i això fa que programa pugui interpretar cada una de les dades d'un fitxer de marques per representar-lo o tractar-lo convenientment, ja que mostren l'estructura de les dades que contenen. Posteriorment un programa podrà interpretar gràcies a les marques què és el que significa cada una de les dades del document.

Classificació dels llenguatges de marques

Podem classificar els llenguatges de marques en dos grans grups basats en el seu objectiu:

1. Llenguatges descriptius o semàntics: Orientats a descriure l'estructura de les dades que conté. En aquests llenguatges es descriu quina estructura lògica té el document ignorant de quina manera serà representada en els programes. Només es posen les marques amb l'objectiu de definir les parts que donen estructura al document. L'exemple més important és l'XML.

Exemple fragment document XML:

```
<carta>
   <data>01/11/2020</data>
   <salutacio>Estimat company:</salutacio>
   <contingut> contingut de la carta ...</contingut>
   <firma>Adela Bujosa</firma>
</carta>
```

2. Llenguatges procedimentals i de presentació: Orientats a especificar com s'ha de representar la informació. En aquests llenguatges el que es fa és indicar de quina manera s'ha de fer la presentació de les dades. Ja sigui per mitjà d'informació per al disseny (marcar negretes, títols, ...) o de procediments que ha de fer el programari de representació. L'exemple més popular d'aquests llenguatges és l'HTML però n'hi ha molts més: TeX, Wikitext… En aquests casos els documents ens poden servir per determinar de quina manera es mostrarà el document a qui el llegeixi.

Exemple fragment document HTML:

```
<html>
 <head>
  <title>Exemple senzill</title>
 </head>
 <body>
  <p>Aquest text és un paràgraf.</p>
 </body>
</html>
```

AMPLIACIÓ:
Sistema d'etiquetatge: Tant si el sistema és descriptiu com de presentació, les marques no han estat col·locades de qualsevol manera sinó que s'ha anat seguint un sistema determinat. Sovint les marques envolten el contingut que volem que tingui un significat o que sigui representat d'una manera determinada. No es poden col·locar les marques de qualsevol manera, ja que una de les coses que cal evitar són possibles mal interpretacions. Per això, a més de definir les marques que s'hi posaran, els llenguatges de marques defineixen unes regles d'ús que especifiquen com han de ser les marques, en quines condicions es permet fer-les sevir i a vegades fins i tot què signifiquen.

Utilització de llenguatges de marques en entorns web

Una pàgina web és un document electrònic adaptat per a la *World Wide Web* que, normalment, forma part d'un lloc web.

La pàgina web, està composta principalment per informació (només text o mòduls multimèdia) així com per hiperenllaços; a més, pot contenir o associar dades d'estil per especificar com ha de visualitzar-se, i també aplicacions embegudes per fer-la interactiva (Una aplicació embeguda es tracta d'un programa categoritzat dins de la familia del software de sistema que està directament integrat en

un sistema de hardware i la seva finalitat és controlar màquines o dispositius. Generalment està dissenyat pel hardware particular en el que s'executa i a més compleix una única funció per el que no pot ser utilitzat en altres situacions).

Les pàgines web estan escrites en un llenguatge de marques que proporciona la capacitat de gestionar i inserir hiperenllaços, generalment, HTML.

El contingut de la pàgina pot ser predeterminat (pàgina web estàtica) o generat en el moment de la seva visualització o en sol·licitar-la a un servidor web (pàgina web dinàmica). Pel que fa a l'estructura de les pàgines web, alguns organismes, especialment el W3C, solen establir directives amb la intenció de normalitzar el disseny, per tal de facilitar i simplificar la visualització i interpretació del contingut.

1.2. Organitzacions desenvolupadores

Dins de les organitzacions que s'han encarregat de desenvolupar els llenguatges de marques es troben:

- Organització Internacional per a l'Estandardització (*ISO, International Organization for Standardization*)

Es va formar després de la Segona Guerra Mundial (23 de febrer de 1947) i és l'organisme encarregat de promoure el desenvolupament de normes internacionals de fabricació, comerç i comunicació per a totes les branques industrials a excepció de l'elèctrica i l'electrònica.

La seva funció principal és la de cercar i definir l'estandardització de normes de productes i seguretat per a les empreses o organitzacions en l'àmbit internacional. És una xarxa dels instituts de normes nacionals de 163 països, sobre la base d'un membre per país, amb una Secretaria Central a Ginebra (Suïssa) que coordina el sistema.

Les normes desenvolupades per ISO són voluntàries, ja que és un organisme no governamental i no depèn de cap altre organisme internacional, per tant, no té autoritat per imposar les seves normes a cap país. El contingut dels estàndards està protegit per drets d'autor i per accedir-hi al públic en general ha de comprar cada document. Aquesta organització després de l'èxit que va tenir GML i, després d'un llarg procés, va publicar el 1986 *l'Standard Generalized Markup Languaje* (SGML) amb rang d'estàndard internacional amb el codi ISO 8879.

- W3C (*World Wide Web Consortium*)

El W3C es va crear el 1994 per Tim Berners-Lee al MIT, actual seu central del consorci. Posteriorment es va unir l'abril de 1995, l'INRIA a França, reemplaçat pel ERCIM el 2003 com l'hoste europeu del consorci i la Universitat de Keiō (*Shonan Fujisawa Campus*) al Japó el setembre de 1996 com a hoste asiàtic.

La seva funció principal és tutelar el creixement i organització de la web. El seu primer treball va ser normalitzar el llenguatge HTML, el llenguatge de marques amb què s'escriuen les pàgines web. En créixer l'ús del web, van créixer les pressions per ampliar l'HTML. El W3C va decidir que la solució no era ampliar l'HTML, sinó crear unes regles perquè qualsevol pogués crear llenguatges de marques adequats a les seves necessitats, però mantenint unes estructures i sintaxi comunes que permetessin compatibilitzar i tractar-los amb les mateixes eines. Aquest conjunt de regles és l'XML, la primera versió es va publicar en 1998.

1.3. Etiquetes, elements i atributs dels llenguatges de marques

A l'apartat anterior hem explicat que són els llenguatges de marques, i s'han introduït alguns conceptes que ara es fa necessari explicar en detall:

Hi ha tres termes emprats per tots els llenguatges de marques, que s'utilitzen per descriure les parts d'un document de llenguatges de marques:

- **Elements**: Representen estructures mitjançant les quals s'organitzarà el contingut del document o accions que es desencadenen quan el programa navegador interpreta el document. Consten de l'etiqueta d'inici, l'etiqueta de cap i de tot allò que es troba entre les dues. Alguns elements no tenen contingut. Se'ls denomina elements buits i no han de dur cap etiqueta.
- **Etiqueta o tag**: És un text que va entre el símbol menor que (<) i el símbol més gran que (>). Existeixen etiquetes d'inici (ex. <nom>) i etiquetes de fi (ex. </ nom>).
- **Atribut**: És un parell nom-valor que es troba dins de l'etiqueta d'inici d'un element i indiquen les propietats que poden portar associades els elements.

    Anem a veure un exemple:

```
<adreça>
   <client>
      <nom> Maria </nom>
      <llinatges> Más López </llinatges>
   </client>
   <carrer> Dels tarongers, 12 </carrer>
   <provincia ciutat="Palma de Mallorca"> Illes Balears</provincia>
   <codi_postal> 07002 </codi_postal>
</adreça>
- Tenim:
 • Elements:
    ◦ Un element pare <adreça>, que a la vegada conté 4 elements fills <client
      >, <carrer>, <provincia> i <codi_postal>. I un element fill <client> que
       té dos fills directes <nom> i <llinatges>
 • Etiquetes:
    ◦ Les etiquetes són totes les paraules que estan entre els símbols < >
 • Atributs:
    ◦ Un atribut, que està dins l'element <provincia> i que és ciutat.
```

1.4. Introducció al llenguatge XML

XML (*eXtensible Markup Language*, Llenguatge de Marcat eXtensible) és un llenguatge desenvolupat per W3C (*World Wide Web Consortium*) què està basat SGML (S*tandard Generalized Markup Language*, Llenguatge de Marcat Generalizat Estàndard). Aquest llenguatge s'utilitza per a l'emmagatzemament i intercanvi de dades estructurades en distintes plataformes.

L'XML és un llenguatge simple de descripció d'informació:

- És un estàndard que permet dissenyar i desenvolupar llenguatges de marques.
- És un format de text estandarditzat que serveix per representar i transportar informació estructurada.

Es diu que XML és un metallenguatge, això vol dir que pot ser emprat per definir altres llenguatges anomenats dialectes XML. Alguns dels llenguatges que es poden definir a partir d'XML són:

- GML (*Geography Markup Language*, Llenguatge de Marcat Geogràfic),
- MathML (*Mathematical Markup Language*, Llenguatge de Marcat Matemàtic),
- RSS (*Really Simple Syndication*, Sindicació Realment Simple),
- SVG (*Scalable Vector Graphics*, Gràfics Vectorials Escalables),
- XHTML (*eXtensible HyperText Markup Language*, Llenguatge de Marcat d'Hipertexte eXtensible), ...

Com en tots els llenguatges de marques, els documents XML es componen de dades caràcter (que seria la informació pròpiament dita) i marcat (marques XML). El marcatge afegeix informació addicional que possibilita una nova manera de tractar la informació, ja que permet realitzar sobre els documents tasques informàtiques com són recerques més precises, filtrats, generació automàtica d'informes, etc.

En un document XML tota la informació es representa com a text. No hi ha tipus de dades numèriques, binaries, lògiques, ... Les marques en un document XML van entre els símbols "<" i ">", o bé, en el cas de les referències importants, comencen per "&" i acaben amb ";".

XML és extensible, això és, que en XML les marques no estan predefinides, sinó que podem definir les nostres pròpies marques, per tal que compleixin els requisits establerts en el llenguatge, i que veurem més endavant. Gràcies a aquesta informació XML s'adapta a qualsevol classe de situació, necessitats de l'autor i programari de processament, ja que en funció dels requeriments es pot fer servir un programari més senzill o més complex. Al llarg d'aquest apartat del llibre veurem exemples clars de marques XML.

Estructura d'un document XML

L'estructura general d'un document XML està formada per dues parts:

```
- Pròleg (opcional):
  Conté una seqüència d'instruccions de processament i/o declaració del tipus de
document. Es pot dividir en dues parts:
     Declaració XML: Estableix la versió d'XML, el tipus de codificació i si és
 un document autònom.
     Declaració de tipus de document: Estableix el tipus de document que és.
- Cos:
  És el contingut informatiu de document, organitzat com un arbre únic d'element
s marcats.
```

- **Pròleg**

El pròleg afegeix informació sobre el document. En concret, declara que el document és un document XML i inclou informació sobre la versió d'XML utilitzada per a escriure-ho. A més, pot incloure informació sobre el tipus de codificació de caràcters utilitzat en el document, si és autònom (conté en si mateix tota la informació necessària per processar-) o no i el tipus al qual s'ajusta el document.

Tot i que el pròleg és opcional, la seva inclusió és molt recomanable, ja que facilita un processament fiable i robust de la informació continguda en el document. Com hem vist abans, el pròleg pot al seu torn dividir-se en dues parts:

La declaració XML és una instrucció de processament especial i compleix diverses funcions:

- Marca el document com a text XML,
- Inclou la declaració de la versió d'XML utilitzada en el document,
- Aporta informació sobre la codificació emprada per representar els caràcters,
- Indica si el document és autònom o no,

Si està present, la declaració XML ha de ser la primera línia del document. Un exemple de declaració XML completa podria ser:

```
<? xml versió = "1.0" encoding = "ISO-8859-1" standalone = "yes"?>
```

Els camps dins de la declaració XML han de seguir l'ordre estricte que veiem a l'exemple anterior.

- Atribut "version": Permet indicar la versió per a la qual es va elaborar el document (per exemple a l'exemple versió 1.0) i permetre que els documents s'adaptin a l'evolució de l'estàndard.
- Atribut "encoding": Permet indicar el joc de caràcters utilitzat en el document. El valor per defecte és UTF-8.

Incís sobre la codificació i representació de caràcters

ISO i l'ús del bit vuitè

Davant el problema de no poder representar certs caràcters amb el sistema AS CII, es va centrar l'atenció en el vuitè bit, que fins ara era utilitzat com bit de paritat. Amb aquest bit es va veure una forma de poder representar 12 8 nous codis.

Es va desenvolupar l'estàndard ISO, i va comportar l'aparició de diversos ti pus, cadascun dels quals era capaç de representar els caràcters de certs alf abets. Entre els diferents tipus de codificació que ens podem trobar estan e ls següents:

- ISO-8859-1:També conegut amb el sobrenom de Latin-1: Es tracta d'una cod ificació molt utilitzada en gran part d'Europa i el continent americà. A questa codificació inclou els caràcters des de la a-z (tant minúscules c om majúscules), els números i símbols d'ús habitual, a excepció del símb ol del €.
- ISO-8859-15:Es tracta d'un sistema de codificació similar al vist anteri orment, encara que inclou algunes diferències com és la inclusió del sím bol del €.
- ISO-8859-5:Representa a l'alfabet ciríl·lic amb les que es poden escriur e en rus, ucraïnès o serbi.
- ISO-8859-6:És l'estàndard de l'alfabet àrab. Comprèn les lletres bàsique s de la llengua àrab, tot i que no inclou les extensions necessàries per al persa ni el pakistanès. Encara que conté les bases de l'àrab, cal ten ir en compte que les lletres d'aquesta llengua poden tenir fins a quatre formes de representació diferent, de manera que per a la correcta presen tació en una pàgina cal sovint un programa independent que analitzi el c ontext en què es troben les lletres i li de la interpretació adequada.

UNICODE

És un sistema per aglutinar totes les codificacions que han existit al llarg del temps i, a causa de l'augment d'ús d'Internet que ha suposat la conversi ó de documents a un major nombre de llengües. L'estàndard ISO s'ha mostrat a mb el temps insuficient per atendre tots els caràcters utilitzats en qualsev ol part del món. Així pues, era necessària una forma global de representar t ots els possibles caràcters existents i UNICODE ser l'encarregat de dur a te rme aquesta tasca.

Aquest estàndard va ser desenvolupat per la UTC (*Unicode Technical Committee*) i té com a particularitat que no és un joc de caràcters, sinó que és un es tàndard que s'encarrega d'assignar un codi numèric únic a cada element que v olem representar.

A l'hora de fer referència a un caràcter Unicode, es fa utilitzant el següen t format: O+XXXX, on les X fan referència al Codi de caràcter en base hexade cimal. Per posar un exemple, la "À" té l'assignat el valor U+00C1. Podeu veu re un llistat d'aquests caràcters en el següent enllaç: https://home.unicod e.org/.

Com ja s'ha dit, UNICODE només és l'encarregat d'indicar el valor que se li assigna a cada element que volem representar, però no indica com s'ha de re presentar aquests elements de manera binària, l'idioma utilitzat en els disp ositius informàtics. És aquí on apareix la codificació més utilitzada en l'a ctualitat:

Codificació UTF-8

UTF-8 és el format de codificació de caràcters més utilitzat en l'actualitat i es caracteritza per utilitzar un número de bytes variable depenent del caràcter que es vulgui representar. Però a més d'aquesta característica, hi ha altres que ho fan molt atractiu a l'hora de ser utilitzat en la codificació de pàgines web, entre les quals podem destacar:

- Capaç de representar qualsevol caràcter Unicode.
- Utilització d'1 a 4 bytes per representar els diferents caràcters, depenent del seu valor Unicode.
- Permet la representació de qualsevol missatge ASCII sense necessitat d'haver de fer cap canvi en la seva codificació.

UTF-8 divideix els caràcters Unicode en diversos grups, depenent del nombre de bytes necessaris per a la seva codificació. Aquests grups són els següents:

- Caràcters codificats amb 1 byte. Aquí hi ha els que són representats en el format ASCII.
- Caràcters codificats amb 2 bytes. Aquest grup inclou els caràcters romanços més els signes diacrítics i els alfabets grec, ciríl·lic, armeni, hebreu i àrab entre d'altres.
- Caràcters codificats amb 3 bytes. Format pels caràcters del pla bàsic multilingüe d'Unicode, que junt al grup anterior, inclou la majoria de caràcters d'ús comú a tot el món, també el xinès, japonès i coreà.
- Caràcters codificats 4 bytes. Aquí es troben els símbols matemàtics i alfabets clàssics per a ús acadèmic com l'alfabet persa, fenici, etc.

(Aquesta informació està treta de la pàgina acensTechnologies, una companyia de telelònica)

- Atribut "standalone": És la declaració de document autònom i pot valer "yes" o "no". El valor "yes", indica que el document conté en el seu interior tota la informació rellevant per a la seva interpretació. Només cal dir per ara que poden existir certs continguts, fora del document actual, que modifiquin la forma en què es processarà el document, aquesta característica implica que el document no és autònom.

La declaració del Tipus de Document és opcional, ja que està inclosa en el pròleg i té un format especial, diferent de les marques i de les instruccions de processament. Proveeix una sèrie de mecanismes que aporten funcionalitat a XML. Gràcies a ella és possible definir una sèrie de restriccions addicionals que han de complir els documents. També incorpora la possibilitat d'utilitzar certes eines que facilitaran a l'usuari XML algunes tasques. Totes aquestes propietats addicionals s'engloben sota el que s'anomena un tipus. Els documents que tenen un tipus associat, i que compleixen amb ell, podran distingir de la resta i formaran el que s'anomena un tipus de documents o una classe de documents.

La declaració de tipus del document no sempre és necessària. És perfectament possible treballar amb XML sense emprar-les, sobretot en entorns en els quals els documents XML es generen automàticament per programes i no cal comprovar certes condicions. Un exemple d'una declaració de tipus de document és el següent:

```
<! DOCTYPE CIFP FRANCESC DE BORJA MOLL SYSTEM "http://www.cifpfbmoll.eu">
```

- **cos**

És la part més important i conté la informació del document, és a dir, les dades a les quals s'ha afegit el marcat.

Elements XML**:** Els documents XML estan formats per text pla (és a dir, sense format) i contenen marques (etiquetes) definides pel desenvolupador. Les marques han de ser el més descriptives possibles.

- La sintaxi general d'una marca és:

  ```
  <etiqueta>valor</etiqueta>
  ```

  Per exemple, si en un document XML es vol guardar el nom Aina, es pot escriure:

  ```
  <nom>Aina</nom>
  ```

  Per tant, seguint la sintaxi bàsica, l'etiqueta d'inici seria <nom> i l'etiqueta de fi </nom>. El valor en aquest cas seria Aina.

- Elements buits:

  En un document XML, un element pot no tenir cap, en aquest cas hauríem d'escriure: <etiqueta></etiqueta> o bé simplement: <etiqueta/>.

- Relacions pare-fill entre elements XML:

  A XML, un element (pare) pot contenir un altre(s) element(s) (els fills). En el següent exemple podem veure que tenim un element "pare", que conté 4 elements "fills" (nom, dona, data_de_naixement i ciutat), i a la vegada un element fill (data_de_naixement) té tres elements "fills" (dia, mes, any).

  ```
  <persona>
    <nom>Aina</nom>
    <dona/>
    <data_de_naixement>
        <dia>10</dia>
        <mes>3</mes>
        <any>2000<any>
     </data_de_naixement>
     <ciutat>Palma de Mallorca</ciutat>
  </persona>
  ```

- Element arrel d'un document XML

  Qualsevol document XML ha de tenir un únic element arrel (pare), d'aquesta manera, qualsevol document XML es pot representar com un arbre invertit d'elements. Es diu que els elements són els que donen estructura semàntica al document.

- Elements amb contingut mixt

  Un element d'XML pot contenir contingut mixt, és a dir, text i altres elements. Per exemple:

  ```
  <persona>
     <nom>Aina</nom> fa <alçada>1,65</alçada>
  </persona>
  ```

Normes de sintaxi d'XML

En un document XML, tots els noms dels elements es diu que són "case sensitive", és a dir, sensibles a lletres minúscules i majúscules. Hauran de complir les següents normes:

- Poden contenir lletres minúscules, lletres majúscules, números, punts ("."), guions ("-") i guions baixos ("_"), també poden contenir el caràcter dos punts (":") però només per definir espais de noms.
- El primer caràcter ha de ser una lletra o un guió baix.
- Darrere el nom d'una etiqueta es permet escriure un espai en blanc o un salt de línia, però mai abans del nom de l'etiqueta.
- Les lletres "á", "Á", "ñ", "Ñ" estan permeses però és recomanable no utilitzar-les per reduir possibles incompatibilitats amb programes que no les suportin.
- El guió ("-") i el punt ("."), encara que estan permesos per nomenar etiquetes, s'aconsella evitar el seu ús., ja que el guió es pot confondre amb el signe menys de la resta i el punt es podria interpretar com una propietat d'un objecte (veurem les propietats dels objectes més endavant).

Atributs a XML

Un atribut serveix per proporcionar informació extra sobre l'element que el conté. Els elements d'un document XML poden tenir atributs definits a l'etiqueta d'inici. Per exemple:

```
Definim els següents atributs:
codi: F001
Nom: Falda
Color: blau
Preu: 24,90
La seva representació en un document XML podria ser:
<article codi="F001">
   <nom color="blau" preu="24,90">Falda</nom>
</article>
```

- **Normes de sintaxi dels atributs**

  1. Els noms dels atributs han de complir les mateixes normes de sintaxi que els noms dels elements.
  2. A més els atributs d'un element han de ser únics.
  3. Els atributs continguts dins d'un element s'han de separar amb espais en blanc.

Com crear un document XML

Podem escriure un document XML amb qualsevol editor de text pla, com per exemple el bloc de notes.

Un exemple podria ser:

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <llibre>
    <títol>Cien años de soledad</títol>
    <autor>Gabriel García Márquez</autor>
    <any_de_publicació="1967/>
  </llibre>
  <llibre>
    <títol>Ulisses</títol>
    <autor Datanaixement="02/02/1882">James Joyce</autor>
    <any_de_publicació="1923"/>
  </llibre>
</biblioteca>
```

1.5. Eines d'edició per XML

Creació de documents XML

Crear un document XML manualment és molt més senzill que crear un document binari, ja que no difereix gaire de crear un document de text. Simplement necessitem un editor de text que no enriqueixi el text.

- **Editors**

  Els documents XML són simples documents de text en què hem afegit algun tipus de metadades. Això permet que la creació de documents XML sigui realment senzilla, ja que es pot fer servir l'editor més senzill que trobem en qualsevol sistema operatiu per poder crear els nostres documents. A pesar d'això també han aparegut tota una sèrie d'editors pensats per fer l'edició de documents XML més senzilla. Aquests editors són molt diversos i normalment ofereixen diferents tipus d'assistència per evitar que es cometin errors en crear el document, comproven interactivament que el document sigui correcte, aconsellen etiquetes, ... Molts dels editors especialitzats en XML normalment a més ofereixen moltes altres funcions com generació d'expressions XPath, creació de fulls d'estil, depuració de transformacions, peticions XQuery...

  L'única cosa que s'ha de fer, és crear un document XML és un editor de text normal i corrent que no enriqueixi el text (els programes editors de text com són Microsoft Word, l'OpenOffice.org Writer, el LibreOffice Writer, ..., no són adequats). Els editors que si es poden emprar, són per exemple: Gedit de Linux, Bloc de notes,...., i en alguns casos aquests editors més senzills fins i tot detecten que s'està editant un document XML i marquen amb colors diferents les etiquetes i les dades (per exemple Gedit).

  - **Editors amb suport d'XML**

  Són editors que donen suport a XML. Aquests editors normalment ofereixen una assistència mínima en editar XML, com acoloriment de les diferents seccions, comprovació automàtica del tancament de les etiquetes, o fins i tot se'n poden trobar amb auto completament d'etiquetes.

  - **Editors especialitzats en XML**

  Aquests editors estan dissenyats específicament per crear i editar documents XML de manera eficient i senzilla minimitzant les possibilitats que es cometin errors en l'edició. Generalment tots ofereixen un entorn amb un grup de finestres amb diferents vistes de l'edició per intentar que no es perdi la visió de conjunt, del que s'està creant.

  A part de la simple edició de documents XML, aquests editors permeten tot un ampli ventall de tasques amb les tecnologies relacionades amb l'XML com:

  - Editar documents XML restringint les etiquetes que s'hi fan servir;
  - Definir un esquema;
  - Crear, convertir i depurar esquemes XML, XSLT, XPath, XQuery, WSDL, SOAP... ;
  - Ajudes per crear documents en vocabularis basats en XML,...;

  Una característica interessant que ofereixen és la possibilitat d'editar documents XML des de diferents punts de vista. El més corrent sol ser fer-ho per mitjà de vistes de text o diferents vistes gràfiques destinades a amagar la complexitat dels documents XML als usuaris que fan servir l'editor.

  A part de l'edició de text molts editors també ofereixen vistes que permeten que un usuari pugui crear dades estructurades de manera gràfica sense que l'usuari ni tan sols sàpiga que està creant un document XML. Una d'aquestes vistes alternatives és la vista d'arbre. La vista d'arbre permet editar el document visualment a partir de l'estructura jeràrquica, de manera que no cal que el que

està creant l'arbre conegui la sintaxi XML. Mentre l'usuari va creant l'arbre, l'editor en segon pla va creant el document XML corresponent. Amb la mateixa idea de fer que l'edició dels documents XML sigui més fàcil per als usuaris no especialitzats, també hi ha la vista de graella.

La popularitat del format XML està fent que el nombre d'editors especialitzats no pari de créixer i que per tant es faci difícil triar l'editor que s'adapta més bé a les necessitats que un usuari pugui tenir.

Alguns exemples d'editors peer XML són: File Viewer Plus 3, XML Explorer, XML Notepad 2007, EditiX XML Editor, Essential XML Editor, XML Tree Editor, ... i n'hi ha d'altres en línia, com per exemple: xmlGrid.net, XML Viewer,...