

Justificaciones al diseño de las resoluciones

Enunciado uno.

Sobre la dirección

Actualmente, las agendas telefónicas se llevan en los móviles y la conectividad, a no ser buscada a tal efecto pasa ya por el simple número de teléfono móvil y una dirección de correo electrónico.

La dirección postal ha pasado a ser un dato menor. Digo esto a tenor de las soluciones bancarias online sin comisiones surgidas a lo largo de este año 2019 y sus requisitos. Así como las plataformas conectivas de compraventa que se preveen dominen este sector (sirva como ejemplo representativo salesforce y uno de sus principales clientes “Adidas”).

Así pues, si en el 2009 la dirección postal seguía siendo sumamente importante y eran guías de estructuración bien los criterios de “Correos y Telegrafos España”, bien Ministerio de Hacienda. A día de hoy, la dirección de correos postales, en una agenda telefónica de smartphone y para el uso cotidiano, no son importantes. Las trato pues como mera cadena de datos --un String de Java o un var o let de JavaScript – que no precisa más formato. Esta solución viene dada, repito por la revolución de la conectividad en la que nos hayamos inmersos. Y ante observaciones del tipo “Para hacer una factura necesitas una dirección postal fiscal” puedo responder que, el enunciado extrae los datos de una agenda de smartphone, una fuente disparatada a para tal propósito. Por que nadie le va a hacer una factura a una persona con la que se “whatsapea” normalmente. Sino que dichos datos postales serán guardados y mantenidos en un equipo para usar con la aplicación adecuada. Es por ello, que justifico mi planteamiento en base al estricto enunciado planteado y a las circunstancias de la fecha actual.

Si se quisiera un dar formato de dato para dirección habría que atender al remitido via correo electrónico por “Correos y Telegrafos” recibido en el 2009 para la aplicación VBA que realice para la empresa Cocemfe. En el cual recuerdo se admitían 83 tipos de vía distintas. Y habría que dar cabida desde direcciones postales desde “Carretera Madrid km 23, Polígono “Silobites”, Calle E, número 14, Sotano” hasta las clásicas “Calle Rue 13 del percebe, sin número”. Pero repito que dadas las características contextuales del enunciado no hace falta. Y el campo dirección se queda como he dicho.

Por qué el atributo 'tipo'

Según el INE [2018], la media de dispositivos móviles usados en España es de 2,5 por persona. Aún a pesar de este dato todavía hay gente que se resiste a prescindir de su teléfono fijo doméstico. Por otro lado, desconociendo el universo de la estadística no se puede afirmar que toda persona posea 2 móviles.

Así pues tenemos dos clases de teléfono: fijo y móvil. La segunda clase se subdivide en dos nuevas: ordinario y trabajo.

Al ser un teléfono de algún tipo necesario y suficiente para una persona (si no, no estaría en una agenda telefónica), he diseñado el archivo asignado a la etiqueta teléfono atributos {“casa”, “movil”, “trabajo”}. El nombre del atributo 'tipo' es el nombre que mejor encaja.

Sobre el indexado

Este es un punto interesante. Las antaños agendas telefónicas venían indexadas y el smartphone actual nos las ofrece indexadas también por orden alfabético, permitiendo búsquedas por otro campo.

Quizás sea esta última apreciación la más relevante. Quiero decir que se pueden definir criterios de ordenación (Alfabético por nombre, por apellidos, por valor del número de teléfono...) Y si, como es mi caso, ¿Qué pasa cuando comparto nombres en distintos alfabetos? Lo que ocurre es que la presentación de la consulta aparece, además y rpimeramente, ordenada por alfabetos.

Así pues recordando que la lección nos muestra la recomendación XML de W3C [1998] como un lenguaje de formateo de datos entendible para humanos y máquinas, y bajo ese prisma y no siendo requisito específico la indexación en este enunciado, no tiene sentido hacer indexación alguna dejando las ordenaciones para otros lenguajes

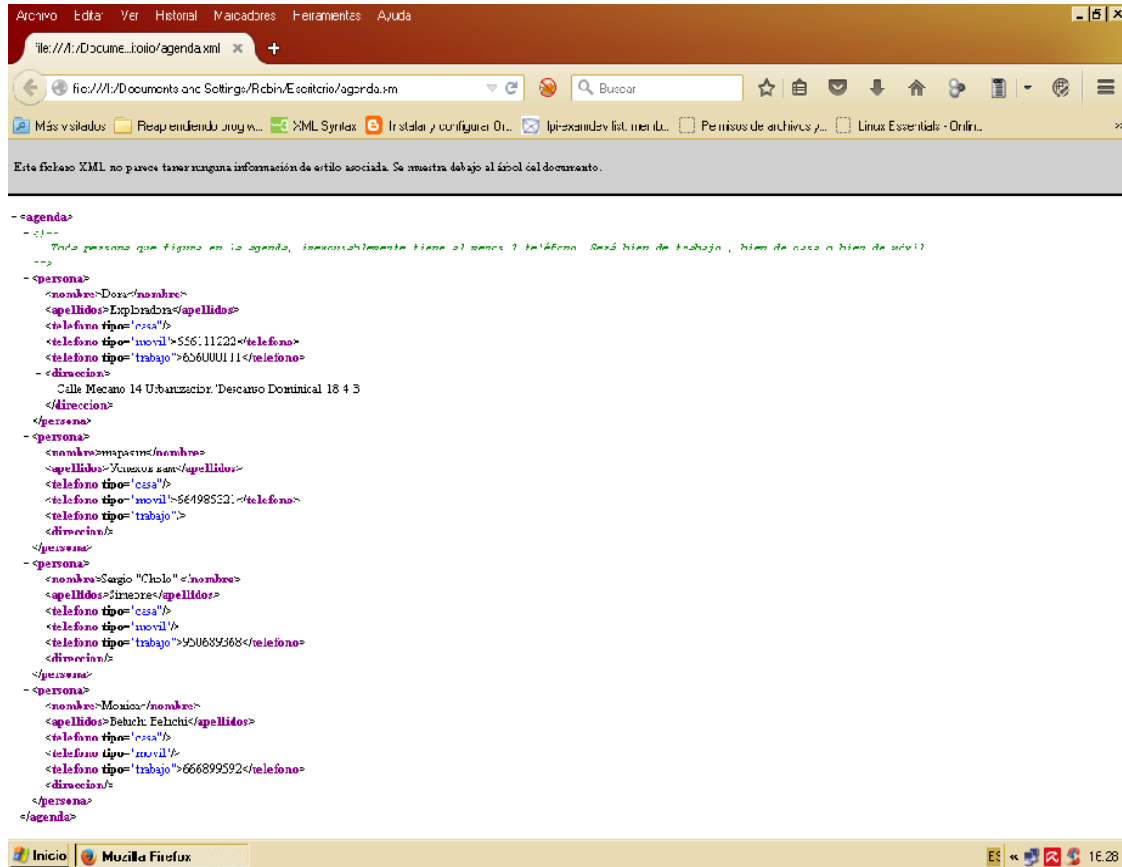
Solución adoptada

Agenda.xml creado con notepad++

Contenido del archivo:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE agenda>
<agenda><!-- Toda persona que figura en la agenda, inexcusablemente tiene al menos 1
teléfono. Será bien de trabajo , bien de casa o bien de móvil -->
  <persona>
    <nombre>Dora</nombre>
    <apellidos>Exploradora</apellidos>
    <telefono tipo="casa"></telefono>
    <telefono tipo="movil">656111222</telefono>
    <telefono tipo="trabajo">656000111</telefono>
    <direccion>Calle Mecano 14 Urbanizacion 'Descanso Dominical' 18 4
B</direccion>
  </persona>
  <persona>
    <nombre>mapasun</nombre>
    <apellidos>Учпехов вам</apellidos>
    <telefono tipo="casa"></telefono>
    <telefono tipo="movil">664985321</telefono>
    <telefono tipo="trabajo"></telefono>
    <direccion></direccion>
  </persona>
  <persona>
    <nombre>Sergio "Cholo" </nombre>
    <apellidos>Simeone</apellidos>
    <telefono tipo="casa"></telefono>
    <telefono tipo="movil"></telefono>
    <telefono tipo="trabajo">950689368</telefono>
  <persona>
    <nombre>Monica</nombre>
    <apellidos>Beluchi Beluchi</apellidos>
    <telefono tipo="casa"></telefono>
    <direccion></direccion>
  </persona>
  <
    <telefono tipo="movil"></telefono>
    <telefono tipo="trabajo">666899592</telefono>
    <direccion></direccion>
  </persona>
</agenda>
```

Verificación de la solución: visualización a través de Mozilla Firefox 39.0.3



Enunciado dos.

Lectura del enunciado

Parece , tras la lectura inicial del resultado que lo pedido parece más coherente que en el enunciado anterior.

Sin embargo una lectura en profundidad del enunciado y del dato o ejemplo sobre el que trabajar desvela incoherencias que van a determinar la justificación del diseño mucho más allá de lo explicado en la unidad didáctica.

En un principio los aspectos sobre los que voy a justificar el diseño de la solución son estos:

1. Resolver la ambigüedad producida en tanto a que un elemento receta tenga “nombre” y un ingrediente tenga “nombre”.
2. Búsquedas en el documento , en especial la de “por ingredientes”. La solución para cada tipo de búsqueda es análoga. Por ello , me centro en esta que es , en mi opinión la más difícil.
¿Por qué digo que es la más difícil?

Porque , por un lado , el ejemplo no nos habla de ingredientes puros.

Podemos ver que tanto el queso como el pan tienen sendos procesos previos -rallar y rebanar-. Por ello tengo que hacer un ejercicio de contextualización previo al formateo como hice en el enunciado uno.

Por otro lado , además del procesado de ingredientes el ejemplo presenta un **problema muy muy serio**. Un problema irrefutable por las normas del curso que impiden alterar los enunciados de las prácticas. El problema es el dato : “queso emmental suizo **O** gruyère”.

Presentar en un lenguaje comprensible para humanos y máquinas este dato como un único elemento <ingrediente> y hacer búsquedas por ingredientes con lo explicado en la unidad didáctica 1 es imposible ya que son dos quesos diferentes en sí mismos (coste, conservación , forma) y por tanto, dos ingredientes distintos. Según lo visto en la UD01- y XML, como tal en su versión mínima enseñada en esta unidad didáctica no contempla las sentencias lógicas del tipo {'and', 'or', 'not', 'xor'...}

Fuentes y tiempo

Para la realización de la solución a este enunciado se consulta en internet:

- Unidad didáctica proporcionada por JcyL
- La [wikipedia](#) (Sección 'Uso de espacio de nombres')
- [Enlace](#) de búsquedas en archivos XML
- [Tutorial de la w3school](#) respecto a sección Atributos, Namespaces, XDOM y XQuery en XML,
- Los manuales completos de la misma página para XML DOM y XML Query [Especificaciones](#) W3C para XQuery 1.0
- [Enlace](#) de youtube para ver un gestor de base de datos que permita

- consultas , cómo se instala y cómo funciona
- Ejemplos de consultas y otros tutoriales: [link1](#) , [link2](#) y este otro --bastante más claro-- [link3](#)

Cae por su propio peso que si bien la cantidad estimada de tiempo para la resolución del enunciado en el ejercicio anterior era justa al ser la primera vez que se hacía un ejercicio. Para este caso y teniendo que leer , entender y aplicar lo anteriormente listado, la hora y media indicada está fuera de lugar y sentido común e igual que entrego esta práctica quisiera una explicación de cómo han llegado al cálculo de tamaños 90 minutos sin más material que las páginas de teoría de la unidad didáctica.

Resolviendo la ambigüedad “nombre” de elementos

El enunciado nos lleva a que habrá que usar dos veces la etiqueta nombre: Una para la receta y otra para cada ingrediente.

Soluciones posibles, ordenadas por grado de elaboración:

A) Una etiqueta <nombre> y otra <Nombre> para sendos elementos

Sintácticamente posible y sencilla aunque, nada recomendable. No sería coherente con mi estilo de edición de etiquetas en el documento{todo mayúsculas, todo minúsculas, primera letra en mayúsculas, “camell”,...} que la W3C [recomienda](#) sea el mismo.

B) Una etiqueta <nombreReceta> y otra <nombreIngrediente> para los elementos <receta> e <ingrediente>

Solución más sencilla y válida sintácticamente. **Por ello, es la que adopto.**

C) Usando atributos

Al igual que soluciono el problema de nombres con estilo puedo hacerlo con atributos.

Como dicen en el [XML Tutorial](#) de la W3Cschool : *There are no rules about when to use attributes or when to use elements in XML.*

<receta nombre="nombredelareceta"></receta> y <ingrediente nombre="nombredelingrediente"></ingrediente>

Pero, si bien en este tutorial indica que la información puede formatearse en elementos y atributos, aconseja no usar estos últimos cuando sean expandibles jerárquicamente pues conjunto de valores de los atributos serán una lista sin jerarquía ninguna.

Por lo tanto aunque válida sigo eligiendo la opción B.

D) Usando Namespace

En el enunciado no indican que el xml que vamos a generar tengamos que mezclarlo con ningún otro documento. Por tanto no se va a producir ninguna ambigüedad elemental más allá de nombrar una receta o nombrar un ingrediente. Aún así estamos en nuestro derecho y no hay prohibición de usar prefijos para hacer únicos los elementos y prevenir ambigüedades. El caso planteado en la unidad didáctica de nombre de alumnos y profesores que asisten a una reunión es un “matar bombas a cañonazos” para este ejercicio. Solución posible descartada por poco eficiente.

Búsqueda por nombre de ingrediente , cantidad de comensales o nombre de la receta

La exigencia, siguiendo [este enlace](#) ,quedará resuelta con XQuery -o Xpath , como se ve en tutorial. Aunque por simplificar , me basaré exclusivamente en XQuery-, siempre y cuando el xml esté bien construido sintácticamente. Es decir, los nombres de los elementos sean únicos. Así pues, nada contradice el uso de la solución B como criterio de desambiguación.

Solución al dato : “queso emmental suizo o gruyere”

En mi opinión es el tema más importante para darPor último queda un tema muy importante para dar solución a este enunciado como un fichero xml que tiene que permitir la búsqueda por ingredientes (entendiendo que dicha búsqueda será en base al nombre de ingrediente) entre otras búsquedas.

Y es que , tanto emmental suizo como gruyere son dos tipos de quesos dentro del gran abanico de quesos que podemos usar para cocinar . Son dos ingredientes distintos y a la vez excluyentes.

Así pues, dado que el xml standar estudiado en la UD 01 y los manuales consultados no añaden lógica en ningún momento a un documento xml y debido a la indexación por ingredientes, estimo como solucionable el problema de la 'o' generando las recetas “sopa de cebolla al emmental” y sopa de cebolla al gruyere” con sendos ingredientes.

Una solución alternativa sería dejar el dato como está y modificando la consulta del nombre de ingrediente a substrings generados por la disyuntiva 'o'. Creo que si la unidad 01 ,base de esta práctica, se queda corta para este enunciado , pasa a ser nimia si se adoptará esta segunda solución.

Por ello como he remarcado anteriormente me quedo con la solucion de xml que contiene dos recetas cada una con ingredientes únicos.

Una observación

No indica nada el enunciado de la práctica pero hay que ser conscientes de que los datos de procesado son secuenciales. La alteración en dicho proceso en el caso menor hace imposible la receta.

Sin embargo y del mismo se ha de entender que la lectura de un archivo xml por un navegador es secuencial y tanto LET como FOR en consultas Xquery toman los contenidos de forma secuencial.

Por ello y además por no ser explícitamente pedido en enunciado no añadiremos atributo alguno a la etiqueta <fase>.

De todas formas y para que quede visualmente demostrado repetiré la tercera consulta , modificandola para ver el proceso de la receta.

Solución

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE libroRecetas>
<libroRecetas>
  <receta>
    <comensales>4</comensales>
    <nombreReceta>Sopa de cebolla al emmental</nombreReceta>
    <ingrediente>
      <nombreIngrediente>cebolla</nombreIngrediente>
      <unidad>kilogramo</unidad>
      <cantidad>1</cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>caldo de carne</nombreIngrediente>
      <unidad>litro</unidad>
      <cantidad>2</cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>mantequilla</nombreIngrediente>
      <unidad>gramos</unidad>
      <cantidad>100</cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>harina</nombreIngrediente>
      <unidad>cucharada</unidad>
      <cantidad>1</cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>queso emmental suizo rallado</nombreIngrediente>
      <unidad>gramos</unidad>
      <cantidad>100</cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>pan tostado </nombreIngrediente>
      <unidad>rebanada</unidad>
      <cantidad></cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>Tomillo </nombreIngrediente>
      <unidad></unidad>
      <cantidad></cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>Laurel</nombreIngrediente>
      <unidad>hoja</unidad>
      <cantidad>1</cantidad>
    </ingrediente>
    <ingrediente>
      <nombreIngrediente>pimienta</nombreIngrediente>
      <unidad></unidad>
      <cantidad></cantidad>
    </ingrediente>
    <proceso>
      <fase>Pelar y partir las cebollas en rodajas finas.</fase>
      <fase>Rehogarlas con la mantequilla, sal y pimienta a fuego lento hasta que estén
```

transparentes sin dorarse.</fase>
 <fase>Añadir la harina sin dejar de remover.</fase>
 <fase>Ponerlo en una cazuela con el caldo, el tomillo y el laurel.</fase>
 <fase>Dejar cocer a fuego lento durante unos 15 minutos.</fase>
 <fase>Poner las rebanadas de pan encima, espolvorear el queso y gratinar al
 horno.</fase>
 </proceso>

</receta>

<receta>

<comensales>4</comensales>

<nombreReceta>Sopa de cebolla al gruyère</nombreReceta>

<ingrediente>

<nombreIngrediente>cebolla</nombreIngrediente>

<unidad>kilogramo</unidad>

<cantidad>1</cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>caldo de carne</nombreIngrediente>

<unidad>litro</unidad>

<cantidad>2</cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>mantequilla</nombreIngrediente>

<unidad>gramos</unidad>

<cantidad>100</cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>harina</nombreIngrediente>

<unidad>cucharada</unidad>

<cantidad>1</cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>queso gruyère rallado</nombreIngrediente>

<unidad>gramos</unidad>

<cantidad>100</cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>Pan tostado </nombreIngrediente>

<unidad>rebanada</unidad>

<cantidad></cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>Tomillo </nombreIngrediente>

<unidad></unidad>

<cantidad></cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>Laurel</nombreIngrediente>

<unidad>hoja</unidad>

<cantidad>1</cantidad>

</ingrediente>

<ingrediente>

<nombreIngrediente>pimienta</nombreIngrediente>

<unidad></unidad>

<cantidad></cantidad>

</ingrediente>

```

    <proceso>
      <fase>Pelar y partir las cebollas en rodajas finas.</fase>
      <fase>Rehogarlas con la mantequilla, sal y pimienta a fuego lento hasta que estén
transparentes sin dorarse.</fase>
      <fase>Añadir la harina sin dejar de remover.</fase>
      <fase>Ponerlo en una cazuela con el caldo, el tomillo y el laurel.</fase>
      <fase>Dejar cocer a fuego lento durante unos 15 minutos.</fase>
      <fase>Poner las rebanadas de pan encima, espolvorear el queso y gratinar al
horno.</fase>
    </proceso>

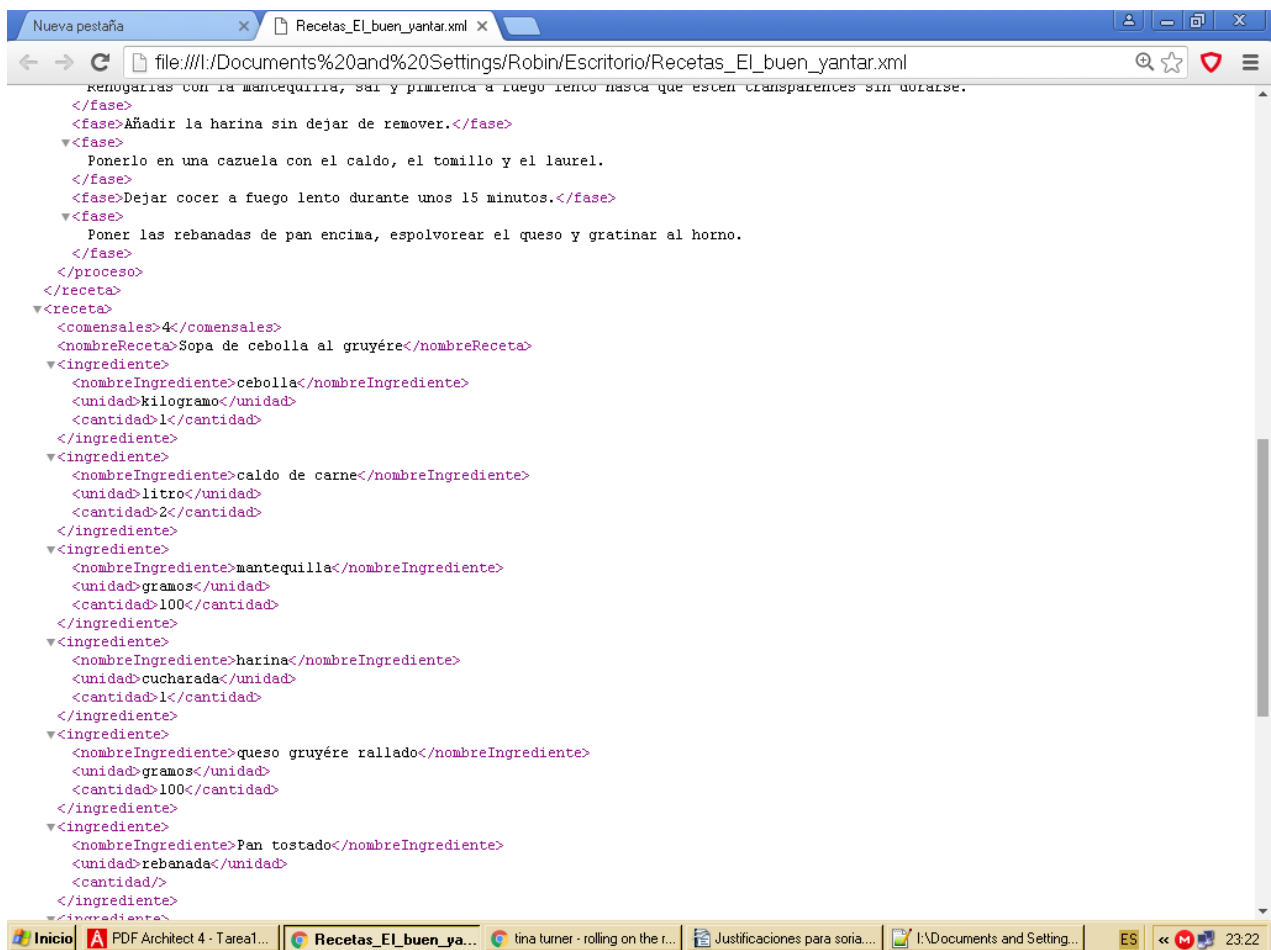
  </receta>

</libroRecetas>

```

Comprobación sobre la solución.

Por un lado, con Chrome 49.0 , da correcto.



Segunda comprobación:

XQuery, revisando este [enlace](#), instalo **baseX** y realizo una consulta de todos los ingredientes sobre el documento tras la lectura de las [especificaciones](#) W3C para XQuery 1.0 y varios manuales de ejemplos de consulta.

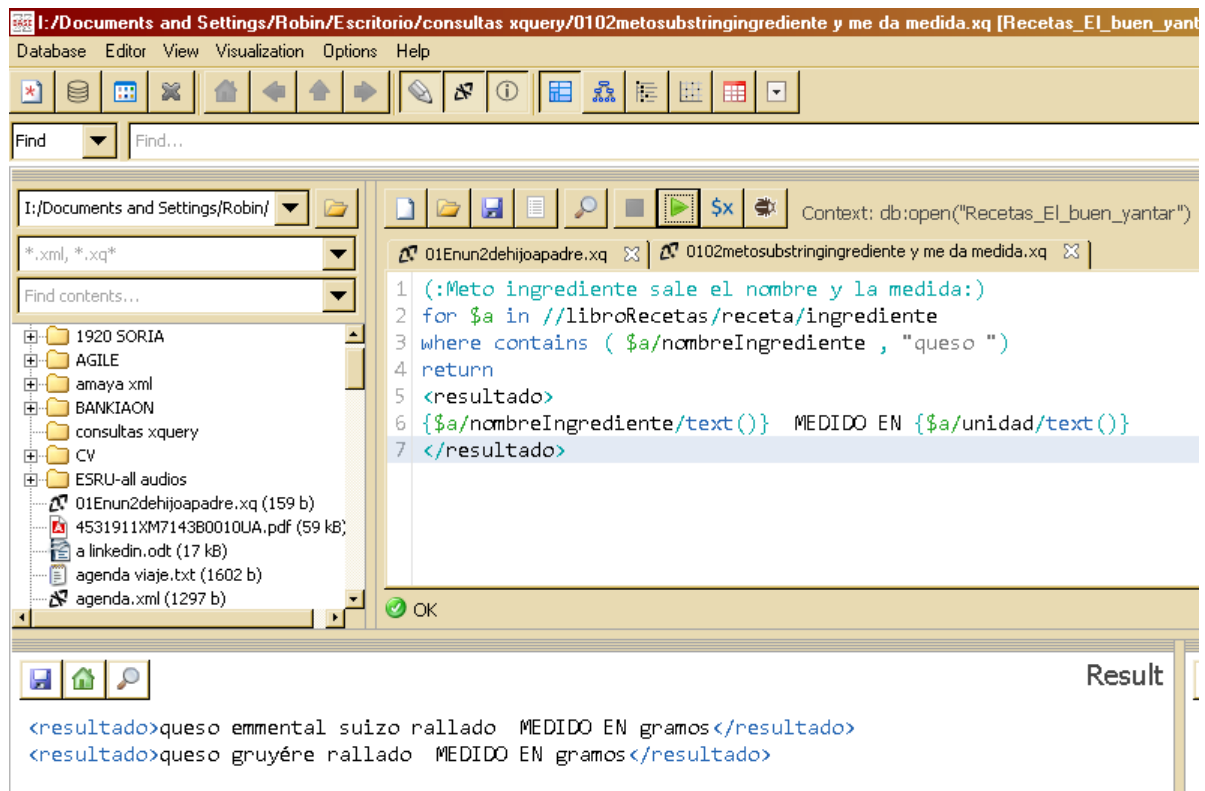
The screenshot shows the BaseX 9.2.4 application window. The title bar is "file* [Recetas_El_buen_yantar] - BaseX 9.2.4". The menu bar includes Database, Editor, View, Visualization, Options, and Help. The toolbar contains various icons for file operations and database management. The main window is divided into several panes:

- Left Pane:** A file explorer showing the directory "I:/Documents and Settings/Robin/". It lists several files and folders, including "1920 SORIA", "AGILE", "amaya.xml", "BANKIAON", "CV", "ESRU-all audios", "4531911XM7143B0010UA.pdf (59 kB)", "a linkedin.odt (17 kB)", "agenda viaje.txt (1602 b)", "agenda.xml (1297 b)", "amaya-WinXP-11.4.7.exe (14 MB)", and "ATX vs ITX.pdf (3346 kB)".
- Editor Pane:** Contains an XQuery query. The context is "db:open('Recetas_El_buen_yantar')". The query is:

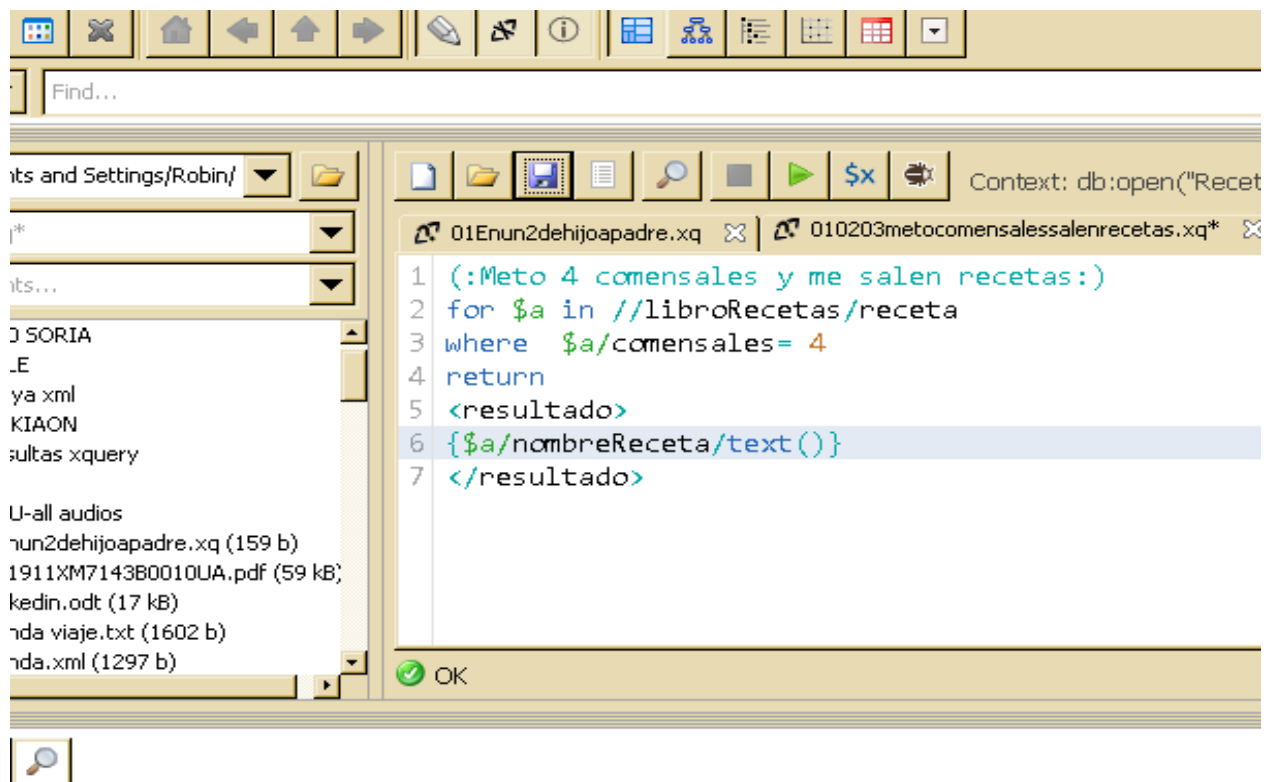
```
1 for $a in //libroRecetas/receta/ingrediente
2
3 return
4 <inventario>
5 {$a/nombreIngrediente/text()} → PARA {$a/ancestor::receta/nombreReceta/
6   text()}
7 </inventario>
```
- Right Pane:** A tree view showing the document structure. It includes "libroRecetas" and "receta", with sub-elements like "ing.." and "proc..".
- Result Pane:** Displays the results of the query. It shows a list of ingredients and their corresponding recipe names, each wrapped in an XML element:

```
<inventario>harina → PARA Sopa de cebolla al emmental</inventario>
<inventario>queso emmental suizo rallado → PARA Sopa de cebolla al emmental</inventario>
<inventario>pan tostado → PARA Sopa de cebolla al emmental</inventario>
<inventario>Tomillo → PARA Sopa de cebolla al emmental</inventario>
<inventario>Laurel → PARA Sopa de cebolla al emmental</inventario>
<inventario>pimienta → PARA Sopa de cebolla al emmental</inventario>
<inventario>cebolla → PARA Sopa de cebolla al gruyère</inventario>
<inventario>caldo de carne → PARA Sopa de cebolla al gruyère</inventario>
<inventario>mantequilla → PARA Sopa de cebolla al gruyère</inventario>
<inventario>harina → PARA Sopa de cebolla al gruyère</inventario>
<inventario>queso gruyère rallado → PARA Sopa de cebolla al gruyère</inventario>
<inventario>Pan tostado → PARA Sopa de cebolla al gruyère</inventario>
<inventario>Tomillo → PARA Sopa de cebolla al gruyère</inventario>
```
- Bottom Pane:** A status bar showing the current database path: "db:open('Recetas_El_buen_yantar', 'Recetas_El_buen_yantar.xml')/libroRecetas". It also shows the file size "22 MB" and the current time "1:05".

Ya, visto funciona. Procedo a la realización de consultas pedidas en el enunciado:



La consulta de los comensales



The screenshot shows the BaseX 8.5.1 interface. The top menu bar includes Database, Editor, View, Visualization, Options, and Help. The toolbar contains icons for file operations and navigation. The left sidebar shows a file explorer with a tree view of the 'pakico' directory, including subdirectories like .cache, .config, .dbus, and .gnupg. The main editor window displays an XQuery script:

```
1 (:Meto parte de receta , ej. gruyère , y salen ingredientes:)
2 for $a in //libroRecetas/receta
3 where contains ( $a/nombreReceta , 'gruyère')
4 return
5 <resultado>
6 { $a/nombreReceta/text() :
7   { $a/ingrediente }
8 </resultado>
```

The right sidebar shows a tree view of the 'libroRecetas' database structure. The bottom panel displays the query result as an XML document:

```
<resultado>Sopa de cebolla al gruyère :
<ingrediente>
  <nombreIngrediente>cebolla</nombreIngrediente>
  <unidad>kilogramo</unidad>
  <cantidad>1</cantidad>
</ingrediente>
<ingrediente>
  <nombreIngrediente>caldo de carne</nombreIngrediente>
  <unidad>litro</unidad>
  <cantidad>2</cantidad>
</ingrediente>
<ingrediente>
  <nombreIngrediente>mantequilla</nombreIngrediente>
```

The bottom status bar indicates 'Time needed: 52.65 ms' and '20 MB'.

(He cambiado al Ubuntu 18 de la biblioteca)

Comprobada pues la solución conforme pedido en enunciado.
Añado última imagen sobre lo comentado para fases de proceso.

The screenshot shows the BaseX 8.5.1 interface. The top menu bar includes Database, Editor, View, Visualization, Options, and Help. The toolbar contains icons for file operations and navigation. The left sidebar shows a file explorer with a tree view of the 'pakico' directory, including subdirectories like .cache, .config, .dbus, and .gnupg. The main editor window displays an XQuery script:

```
1 (:Meto parte de receta , ej. gruyère , y salen ingredientes:)
2 for $a in //libroRecetas/receta
3 where contains ( $a/nombreReceta , 'gruyère')
4 return
5 <resultado>
6 { $a/nombreReceta/text() :
7   { $a/proceso }
8 </resultado>
```

The right sidebar shows a tree view of the 'libroRecetas' database structure. The bottom panel displays the query result as an XML document:

```
<resultado>Sopa de cebolla al gruyère :
<proceso>
  <fase>Pelar y partir las cebollas en rodajas finas.</fase>
  <fase>Rehogarlas con la mantequilla, sal y pimienta a fuego lento hasta que estén
transparentes sin dorarse.</fase>
  <fase>Añadir la harina sin dejar de remover.</fase>
  <fase>Ponerlo en una cazuela con el caldo, el tomillo y el laurel.</fase>
  <fase>Dejar cocer a fuego lento durante unos 15 minutos.</fase>
  <fase>Poner las rebanadas de pan encima, espolvorear el queso y gratinar al horno.</fase>
</proceso>
</resultado>
```

The bottom status bar indicates 'Time needed: 18.33 ms' and '28 MB'.

Enunciado tres.

Asignando características a la información

Tenemos un módulo con: nombre, ciclo al que pertenece, un nombre homologado sobre el que se proyecta, horas semanales, carácter, fecha comienzo, fecha fin y alumnos matriculados.

Unos alumnos caracterizados por: cadena de nombre, cadena de apellidos, tipo documento identificativo, cadena codificada del documento, teléfono (puede ser fijo o móvil pero inicialmente no se le da importancia al tipo, un email, una dirección que desglosaré en el próximo párrafo, faltas de asistencia y notas (aunque no deja claro si "apto" es una posible nota definitiva o es una nota parcial de algún control, algún parcial o qué exactamente).

Las direcciones de los alumnos se caracterizan por: un tipo de vía abreviado, un nombre de tipo de vía, un nombre de vía y una altura; una población y un código postal (denominado CP) con un valor numérico.

¿Notas o notas?, valores

Leído en el enunciado que un alumno puede tener más de una nota, tomo como referencia el usado en mi centro: Cada alumno tiene una nota final.

Pero como he dicho anteriormente puede, además tener dos notas parciales además de poder acumular unas notas de prácticas (6).

El "por ahora" del enunciado evidencia que no corresponde a nota final y la falta de ausencia en el dato para el otro alumno que la nota del parcial está recurrida y por tanto no es oficial. (no pueden ser notas de prácticas porque el resultado sería numérico).

Las prácticas tienen un periodo límite de entrega y una UD asociada; los parciales unas UD Asociadas y unas fechas y la nota final no tiene fecha asociada al ser la nota obtenida al la fecha fin del módulo.

Sin embargo, el como se realizan las UD corresponde a cada centro o en su defecto Comunidad Autónoma. Es decir, el temario es el que es, seguro corresponde a algún perfil tecnológico pero la creación de parciales o de prácticas puntuales pertenece al veni adocenci del profesor o en su defecto al sistema educativo al que se suscribe el centro. Por ello, la nota sí o sí es la nota final y el resto de notas (parciales y prácticas) son "apaños" para justificarla. Apaños que según el enunciado no se comunican entre máquinas y que se podría dejar todo en un único campo denominado, nota final que correspondiendo con las fechas en que se transmite la información.

Es decir el elemento "notaFinal" se puede emitir todos los días del curso pero el que realmente será significativo será el emitido el último día de curso.

¿El módulo pertenece a un único ciclo o a varios?

No parece corresponde a esta asignatura abordar las cardinalidades de las relaciones. Sin embargo el propósito de esta pregunta es meditar si es el módulo el que anida al elemento ciclo (sólo pertenece a este ciclo) o es el elemento ciclo el que anida al módulo (El módulo pertenece a más ciclos además de este).

Es el enunciado el que nos saca a cualquier duda: "de un módulo del ciclo formativo

DAW”.

Definir los atributos adecuados

Ya en las justificaciones del enunciado dos hablé sobre la conveniencia o no de usar atributos o elementos. En mi caso no los usaré, prefiero trabajar con elementos. (otra cosa es que si el documento de trabajo lleva , los use. Pero yo estoy definiendo el documento xml y prefiero que todo este perfectamente jerarquizado. Son más líneas , y el documento queda mucho más extenso, pero voy a usar Geany, notepad++ o Amaya - dependiendo el equipo en que termine la práctica- y estos editores llevan ayudas para que se controle cierre de etiquetado , palabras reservadas o cualquier pequeño detalle que al editar o extender un xml hacen no quede bien formado.

¿Es conveniente desglosar el elemento email?

La verdad es que esta es una gran pregunta. Pero la respuesta es que no. XML no procesa la información sino que sirve de transporte entre dos máquinas.

Por ejemplo, una persona llamada Xavier Lomeras Martínez puede tener este correo electrónico:

xml@miweb.aquenopuedesimaginartemidominio.eh.eh.masalladegondor.muchomas.mas.alla.com

Y al decirle que los 32 caracteres reservados en la BD para meter su correo no caben , que nos mande el que usa normalmente, nos manda este:

xml@grr.la --un correo temporal-- o para apuntar maneras
drop@table'Recetas_El_Buen_Yantar'.com

La verdad es que hay que tener sumo cuidado con cadenas de texto que vamos a procesar pero, XML no procesa. Simplemente,transporta.

Debido a ello , adopto aquí el criterio usado en el enunciado uno de esta práctica para direcciones postales en una agenda de teléfono móvil. Recuerdo fue : Si existen , se trata como una única cadena.

Claro que para el enunciado 1 no tenían el porque de existir sin embargo aquí el email viene impuesto por enunciado, tratado en los datos a aplicar y tiene todos el sentido común para esta formación online.

Bueno pues con estas mínimas justificaciones hago esqueleto:

```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <!DOCTYPE cicloFormativo>
3  <cicloFormativo>
4    <nombreCiclo>DAW</nombreCiclo>
5    <modulo>
6      <identificador>LMSG</identificador>
7      <nombreModulo>Lenguajes de Marcas y Sistemas de Gestion de Información</nombreModulo>
8      <horasSemanales>4</horasSemanales>
9      <caracter>obligatorio</caracter>
10     <!-- Adoptado formato fechas dd/mm/yyyy -->
11     <fechaComienzo>15/09/2010</fechaComienzo>
12     <fechaFin>30/06/2011</fechaFin>
13     <matriculados>
14       <nombreAlumno></nombreAlumno>
15       <tipoIdentificacion></tipoIdentificacion>
16       <codigoIdentificacion></codigoIdentificacion>
17       <email></email>
18       <telefono></telefono>
19       <direccion>
20         <tipoVia></tipoVia>
21         <descripcionVia></descripcionVia>
22         <nombreVia></nombreVia>
23         <hito></hito>
24         <poblacion></poblacion>
25         <codificacionPostal></codificacionPostal><!-- Si vive en USA es ZIP -->
26         <codigoPostal></codigoPostal>
27       </direccion>
28       <faltasAsistencia></faltasAsistencia>
29       <notaFinal></notaFinal>
30     </matriculados>
31   </modulo>
32 </cicloFormativo>
33

```

Ahora toca llenarlo con los datos mostrar la solución.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE cicloFormativo>
<cicloFormativo>
  <nombreCiclo>DAW</nombreCiclo>
  <modulo>
    <identificador>LMSGI</identificador>
    <nombreModulo>Lenguajes de Marcas y Sistemas de Gestion de
Información</nombreModulo>
    <horasSemanales>4</horasSemanales>
    <caracter>obligatorio</caracter>
    <fechaComienzo>15/09/2010</fechaComienzo>
    <fechaFin>30/06/2011</fechaFin>
    <matriculados>
      <nombreAlumno>Ana Fernández Gutiérrez</nombreAlumno>
      <tipoIdentificacion>nif</tipoIdentificacion>
      <codigoIdentificacion>16965696L</codigoIdentificacion>
      <email>ana.fdezgtrrez@hotmail.com</email>
      <telefono>789654321</telefono>
      <direccion>
        <tipoVia>C</tipoVia>
        <descripcionVia>Calle</descripcionVia>
        <nombreVia>El Percebe</nombreVia>
        <hito>13</hito>
        <poblacion>Santander</poblacion>
        <codificacionPostal>CP</codificacionPostal>
        <codigoPostal>39401</codigoPostal>
      </direccion>
    </matriculados>
  </modulo>
</cicloFormativo>

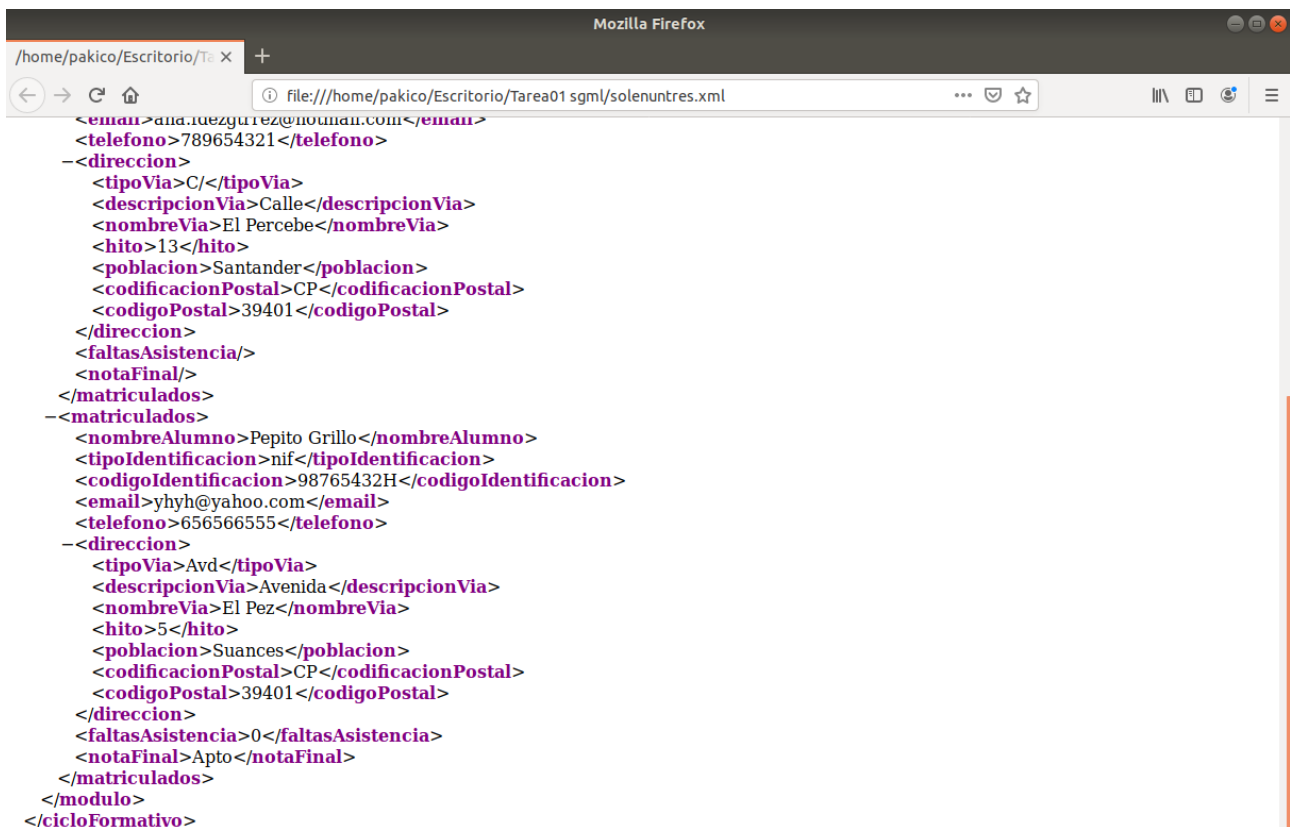
```

```

        <faltasAsistencia></faltasAsistencia>
        <notaFinal></notaFinal>
    </matriculados>
    <matriculados>
        <nombreAlumno>Pepito Grillo</nombreAlumno>
        <tipoIdentificacion>nif</tipoIdentificacion>
        <codigoIdentificacion>98765432H</codigoIdentificacion>
        <email>yhyh@yahoo.com</email>
        <telefono>656566555</telefono>
        <direccion>
            <tipoVia>Avd</tipoVia>
            <descripcionVia>Avenida</descripcionVia>
            <nombreVia>El Pez</nombreVia>
            <hito>5</hito>
            <poblacion>Suances</poblacion>
            <codificacionPostal>CP</codificacionPostal>
            <codigoPostal>39401</codigoPostal>
        </direccion>
        <faltasAsistencia>0</faltasAsistencia>
        <notaFinal>Apto</notaFinal>
    </matriculados>
</modulo>
</cicloFormativo>

```

Comprobamos y vemos que los campos vacios reaccionan bien



Con esto queda terminada la práctica que, repito, ha superado con creces el tiempo de realización estimado de menos de 6 horas.