

## 7 Tecnologia utilitzada

A continuació es comenten breument les tecnologies utilitzades al llarg d'aquest projecte.

la gestió de documents XML fora senzilla de forma que es minimitzarà el temps de desenvolupament.

En aquest aspecte C# es molt superior a C++, i donat que LeanEditor no requereix un codi totalment optimitzat es va sacrificar la gestió de la memòria en favor de la velocitat de codificació.

L'adaptació de C# de LeanGen i LeanClient encara no s'ha abordat, havent d'estudiar amb deteniment el rendiment d'un llenguatge front a un altre, perquè en aquests dos components l'optimització sí es necessària, especialment en el motor de simulació.

- **Els espais en blanc en els atributs son normalitzadors, és a dir, en cas d'aparèixer més d'un de forma consecutiva, els elimina fins a deixar un únic espai.**

Si un document XML no satisfà totes les restriccions del llenguatge es diu que el document no està ben format.

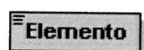
Però amés de les restriccions pròpies d el llenguatge es poden definir altres tipus de restriccions per un documents XML concret. Aquestes restriccions van destinades a la definició de l'estructura del documents es poden definir mitjançant DTD o XSD.

De la mateixa manera que es parla de documents ben formats amb les regles de construcció del document XML, al parlar del compliment de les regles de DTD o XSD es parlar de documents vàlids:

- **Documents ben format: Segueix les regles de construcció de documents XML**
- **Document vàlid: Segueix les regles del document DTD o XSD associat.**

bàsic tal com String, Integers o Booleans. Existeix una llista molt extensa d'aquest tipus simples.

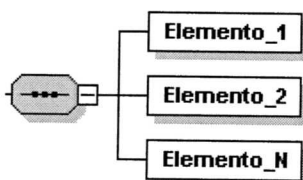
**Complexes:** (Complex Type) és l'equivalent a les classes en l'orientació a objectes. Aquesta classe de tipus potser definida pel dissenyador, igual que a les classe a l'orientació a objectes, encara que simplement és com una estructura de dades. A la definició d'un tipus complexa es poden fer servir elements de tipus simples o altres tipus complexes, a més d'altres mecanismes. En la notació gràfica es distingeix els elements simples dels complexos indicant sobre els elements simples una marca en límits superior esquerre.



Els elements que formen un document XML poden contenir, a la seva vegada altres elements, definint així una estructura complexa d'informació. Per això, els esquemes XML utilitzen entre altres dos mecanismes bàsics:

**Sequence:** Indica l'existència d'una seqüència d'element, és a dir, s'enumeren els elements que formen part d'aquesta estructura. No s'han de confondre seqüència d'elements amb una llista d'elements. El sequence es el mecanisme utilitzat per l'esquema XML, per enumerar lo que en orientació a objectes equivaldria als atributs d'una classe.

En el format gràfic es representa el component sequence de l'esquema XML de la següent forma:



El codi de l'esquema XML que representa el gràfic anterior es el següent:

```
<xs:sequence>
  <xs:element name="Elemento_1"/>
  <xs:element name="Elemento_2"/>
```

## 7.4 XPath

XPath és el resultat d'un esforç per a proporcionar una sintaxis i una semàntica comuns per a funcionalitats compartides entre XSL Transformations (XSLT) i XPathPointer.

L'objectiu principal de XPath és direccionar parts d'un document XML. Com a suport per aquest objectiu, també proporciona facilitats bàsiques per la manipulació de cadenes, números i booleans. XPath utilitza una sintaxis compacta i no XML per facilitar l'ús de XPath dintre URIs i de valors d'atributs XML. XPath opera sobre l'estructura lògica abstracta d'un document XML, més que en la seva sintaxis superficial. XPath obté la seva denominació per l'ús que d'una notació de camins, com en els URLs, per a navegar mitjançant de l'estructura jeràrquica d'un document XML.

A més a més, del seu ús per direccionar, XPath està dissenyat també de manera que té un subconjunt natural que pugui ser utilitzat per cotejar (comprovar si un node encaixa amb un patró o no); aquest ús de XPath està descrit en XSLT.

XPath modela un document XML com un arbre de nodes. Hi ha diferents tipus de nodes, incloent nodes element, nodes atribut i nodes text. XPath defineix un mode de calcular un valor de cadena per a cada tipus de node. Alguns tipus de node també tenen noms. XPath és totalment compatible amb XML Namespaces. Així, el nombre d'un node es modela com un parell consistent en un part local i un (potser nul) URI d'espai: d'això se'n diu un nombre expandit.

únic resultat. Aquest document defineix una biblioteca de funcions que totes les implementacions de XPath han de suportar. Per les funcions de la biblioteca bàsica de funcions, els arguments i el resultat són dels quatre tipus bàsics. Tant XSLT com XPointer exten a XPath mitjançant la definició de funcions addicionals; algunes d'aquestes funcions operen sobre els quatre tipus bàsics, altres operen sobre tipus de dades addicionals definits per XSLT i XPointer.

Les declaracions d'espais de noms consisteixen en una correspondència de prefixes a URIs d'espais de noms.

Les assignacions de variables, biblioteca de funcions i declaracions d'espais de noms utilitzades per avaluar una sub-expressió són sempre les mateixes que les que s'emplenen per avaluar l'expressió que la conté. El node contextual, la posició contextual i la mida contextual utilitzats per avaluar una sub-expressió són a vegades diferents dels que s'emplenen per avaluar l'expressió que la conté. Varis tipus d'expressions canvien el node contextual; sols els predicats canvien la posició contextual i la mida contextual. Al descriure l'avaluació d'un tipus d'expressió, sempre es fa constar explícitament si el node contextual, la posició contextual i la mida contextual canvien per l'avaluació de sub-expressions, si res es diu sobre el node contextual, la posició contextual i la mida contextual, romandran inalterats en l'avaluació de sub-expressions d'aquest tipus d'expressió.

Les expressions XPath sovint apareixen en atributs XML. La gramàtica especificada en aquesta secció s'aplica al valor de l'atribut després de la normalització XML1.0. Així per exemple, si la gramàtica utilitza el caràcter <, aquest no ha d'aparèixer en el codi font XML com < sinó que ha de ser tractat conforme a les regles de XML 1.0 introduint-lo, per exemple com &lt;. Dintre de les expressions, les cadenes literals es delimiten mitjançant cometes simples o dobles, les quals s'emplenen també per delimitar atributs XML. Per evitar que una marca d'entrecomillat en una expressió sigui interpretada per el processador XML com a terminador del valor de l'atribut, la marca d'entrecomillat pot introduir-se com una referència de caràcter (&quot; O

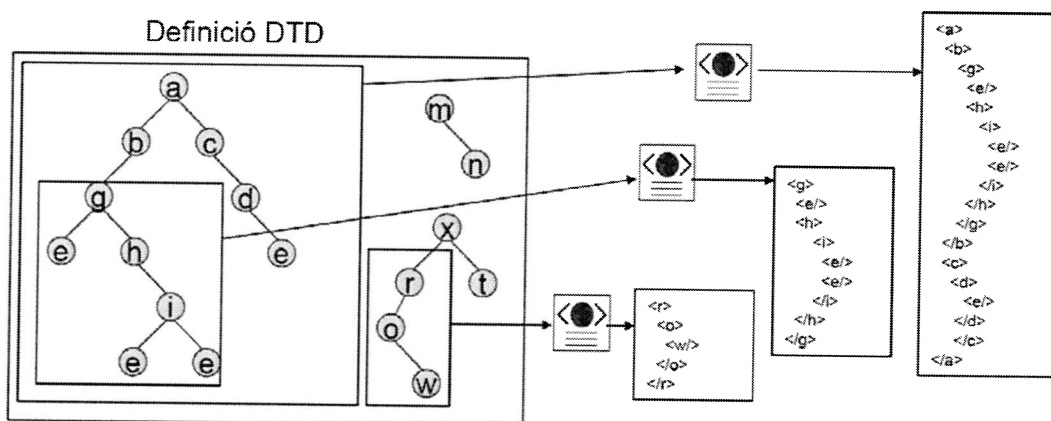
## 7.5 DTD

Un document XML pot :

- **Estar ben format:** si segueix les regles per a la construcció de documents XML.
- **Ser vàlid:** si compleixen les regles de la DTD que té associat.

En una DTD es defineixen les característiques estructurals d'un conjunt de tipus de documents XML. Aquesta definició es fa a través d'un conjunt de regles que defineixen la gramàtica del/s document/s.

Es poden determinar cert tipus de propietats sintàctiques. És a dir, una DTD pot definir diferents estructures XML, aquestes poden compartir definicions d'elements. Per tant diferents documents XML:



La DTD conté la informació relativa a:

- **Estructura del document XML (descripció de l'arbre).**
- **Els elements que es poden usar. Per cada un d'ells podem especificar:**
  - **Els elements fills que aquest pot tenir. Per cada un d'ells es pot especificar:**
    - Si són obligatoris o opcionals.
    - El nombre de cops que pot aparèixer.

La declaració de contingut pot incloure:

- La paraula clau #PCDATA, que indica que l'element pot tenir dades de tipus caràcter, es dir, no de tipus de marca. Podem fer referència tant a:

1. Text, com números com dates Per exemple la següent de DTD:

```
<!ELEMENT autor (#PCDATA)>
Reconeix XMLs que siguin del tipus:
<autor> J.R.R. Tolkien </autor>
```

2. El nom d'altres elements: estarem especificant que l'element està format per altres elements/nodes. Per exemple les següents declaracions de DTD:

```
<!ELEMENT Persona (Nom, Cognom, Cognom)>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Cognom (#PCDATA)>
```

Reconeix XMLs que siguin del tipus:

```
<Persona>
<Nom>Miquel</Nom>
<Cognom>Trilla</Cognom>
<Cognom>Romero</Cognom>
```

3. La paraula reservada #PCDATA i el nom d'altres elements: en aquest cas estarem dient que el node pot estar format tant per elements com per com per cadenes. Per exemple la següent DTD:

```
<!ELEMENT Persona (#PCDATA|Nom|Cognom|Cognom) *>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Cognom (#PCDATA)>
```

Reconeix XMLs que siguin del tipus:

```
<Persona>
Fitxa d'una persona.
<Nom>Miquel</Nom>
<Cognom>Trilla</Cognom>
```



## 7.6 XQuery

D'un temps a l'actualitat la comunitat de desenvolupadors ha vist l'aparició de moltes noves tecnologies. Tecnologies que, mentre solucione problemes i obren possibilitats de desenvolupament (com XML i els Serveis Web), també provoquen nous requeriments. En el present punt es pretén introduir un altre nova tecnologia que sorgeix com la necessitat de consultar documents i bases de dades XML: el XQuery.

XQuery és un llenguatge de consultes estàndard, publicat pel W3C (World Wide Web Consortium) que utilitza la notació XML per definir consultes i dominar els resultats XQuery és lo suficientment flexible com per consultar un ampli espectre d'origens de dades, incloent bases de dades relacionals, documents XML, Serveis Web, aplicacions i sistemes heretats.

### 7.6.1 Antecedents

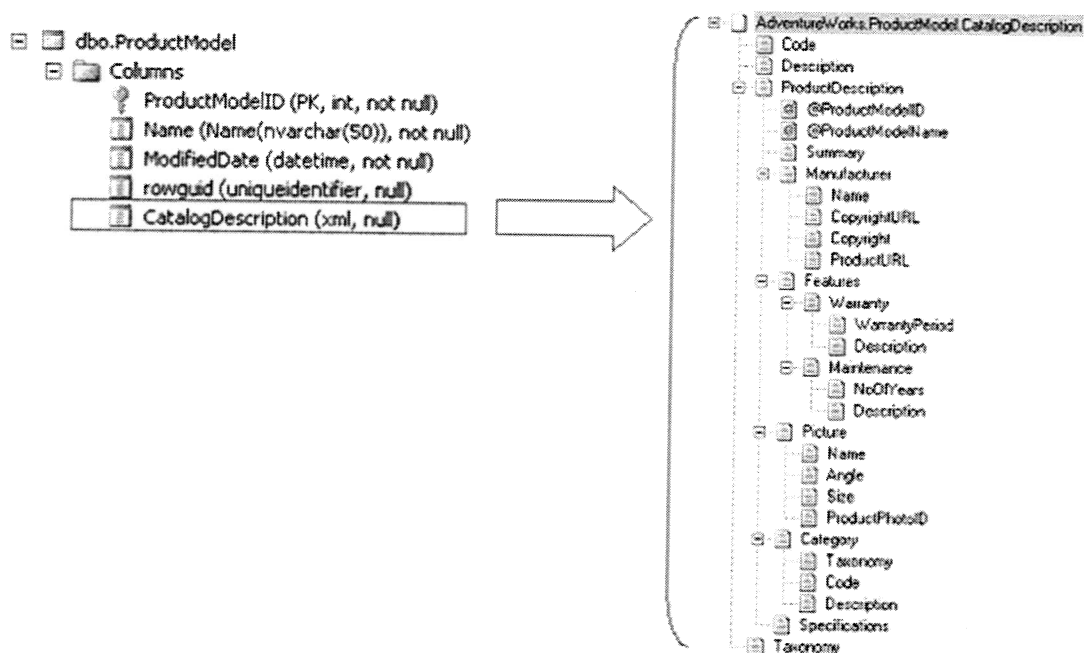
XML ha significat molt per el desenvolupament de sistemes; qüestions tals com la possibilitat de comunicar de manera transparent sistemes que pertanyen a diferents plataformes haurien estat impensades en altres temps. Obstant XML és un pas important, per si sol no és de gran utilitat. El que realment fa potent aquesta tecnologia és el conjunt d'estàndards que s'han desenvolupat (i els que encara estan en desenvolupament) entorn a la mateixa.

XQuery és un llenguatge. Proveeix mecanismes per extraure informació de bases de dades XML natives, així com d'un altre tipus d'origens de dades (como ser base de dades relacionals). Entre d'altres cosses, permet la possibilitat d'obtenir dades d'arxiu XML i una taula de la base de dades relacional amb una sola consulta. XQuery es presenta com un llenguatge funcional, enlloc executat comands com ho faria un llenguatge procedural, cada

Progress”, encara que s’està treballant en elles. Com es pot veure, Xquery és una d’elles.

XPath es descriu millor amb exemples que amb especificacions formals de sintaxi, veiem alguns.

Pels exemples utilitzarem la base de dades AdventureWorks que està inclosa en el SQL Server 2005 Beta 1 (Yukon). En particular utilitzarem la columna CatalogDescription de la taula ProductModel que es de tipus XML (la nova versió SQL Server permet emmagatzemar XML de manera nativa) Segons veiem a continuació:



A continuació es veuen algunes de les dades que conté la columna CatalogDescription. És a dir, aquestes son les dades XML que es troben en un registre de la taula ProductModel:

```
<ProductDescription ProductModelID="19"
ProductModelName="Mountain100">
  <Summary>
    Our top-of-the-line competition mountain bike.
    Performance-enhancing options include the
    innovative HL Frame,
```

Xquery es basa en el llenguatge per consultes XPath existent, amb un increment de la compatibilitat per aconseguir una millor iteració, millors resultats de l'ordenació i la possibilitat de general XML necessari. Xquery opera segons el model de dades Xquery. Es tracta d'una abstracció de documents XML, i els resultats d'Xquery poden tenir tipus o no. La informació del tipus es basa en el tipus proposat per el llenguatge d'esquemes XML de W3C. Sinó es disposa de la informació de tipus, Xquery tracta les dades com sense tipus. Això és similar al mode en XPath versió 1.0 tracta el XML.

# 8 Proves d'avaluació de les tecnologies

L'etapa de proves és molt important el desenvolupament de software que funcioni correctament. Les proves han de permetre determinar si una implementació del sistema satisfà la funcionalitat, les restriccions i el rendiment establert a l'anàlisi de requeriments i especificacions. Aquesta tasca no és trivial , perquè és necessari tenir un bon coneixement dels requisits del programari i no es pot deixar per quan el sistema està completament implementat.

S'ha anat realitzant proves a mesura que s'implementava, procurant veure que funcionessin tal i com s'havia previst. Si el resultat no era l'esperat, s'ha consultat l'especificació i el disseny del sistema, s'ha modificat la implementació corregint l'error i s'ha tornat a enviar la prova. D'aquesta forma, la tasca de provar el sistema s'ha anat repartint durant l'etapa de implementació.

Encara que no s'hagin assolit els objectius secundaris durant tot el projecte s'han anat fent implementacions de mòduls per tal de poder veure realment quina era la funcionalitat que aportaven cada un dels elements.

Es va implementar un petit mòdul que atacava contra la base de dades i feia servir el tipus de dades XML que aporta SQL Server 2005, els resultats varen ser espectaculars ja que SQL Server 2005 aporta moltes noves funcionalitats per treballar amb elements XML a més de la implementació de XPath i XQuery amb algunes millores sobre versions anteriors.

També es va implementar un mòdul de comunicacions per veure les facilitats que donava .NET en aquest sentit i realment va facilitar molt la tasca.

## 8.1 Caixa Negra

És un mètode on no es té en compte l'estructura interna, és a dir, el sistema veu com una caixa negra desconeixent completament la seva estructura interna. Permet obtenir entrades que proven tots els requisits funcionals del programa. Amb aquest mètode s'intenta trobar:

- **Funcions incorrectes o absentes**
- **Errors d'interfaç**
- **Errors en estructures de dades o en accessos a la base de dades externes.**
- **Errors de rendiment.**
- **Errors d'inicialització i terminació.**

## 8.3 Proves realitzades

La realització de les proves s'ha realitzat paral·lelament a la implementació del sistema, per així poder evitar que els errors inicials foren detectades en una fase molt avançada del desenvolupament.

Aquestes proves s'han basat en els conceptes de la metodologia de caixa negra i caixa blanca vistes anteriorment.

Les tècniques de caixa blanca s'han utilitzat en les proves de les modificacions realitzades sobre el sistema, al tenir un profund coneixement de l'estructura interna d'aquestes modificacions, i amb la finalitat de detectar errors a la fase de la implementació. Les de la caixa negra han estat destinades a verificar el compliment dels requisits funcionals.

Les tècniques de caixa negra també han estat utilitzades per provar els diferents components de LeanSim® abans de la seva utilització i modificació. Això ha estat necessari ja que és una eina encara en fase de desenvolupament i algunes modificacions recents encara no han estat completament provades.

L'elecció de caixa negra es deu a la magnitud d'alguns components, com per exemple el motor de simulació, sent impossible realitzar unes proves molt exhaustives dintre de l'àmbit d'un projecte final de carrera.

En qualsevol cas per molt exhaustiva que sigui al fase de proves s'ha de tenir en compte que mai es pot garantir l'absència d'errors. Una prova pot ajudar a detectar un error, però en cap cas podrà assegurar que no existeixin més. De la bondat i de la exhaustivitat d'aquesta fase a gran mesura la qualitat del programari desenvolupat.

## 9 Planificació del projecte

Un cop finalitzades totes les etapes del projecte és moment de realitzar una valoració del temps invertit en la realització d'aquestes etapes i del cost real del projecte.

En aquest capítol es fa un repàs a totes les fases – o etapes- del projecte, es comenten alguns detalls del calendari de treball seguit per posteriorment mostra la planificació final, amb la distribució d'hores entre els diferents perfils i un recompte total d'hores dedicades.

Quant a l'estudi econòmic es presenten els costos de cadascun dels recursos i perfils utilitzats per, partint de la planificació anterior, fer el càlcul de costos total.

- Validació: Verificació del codi implementat.
- Redacció de la memòria del projecte: Redacció del document final , a partir de la recopilació dels documents de treball generats en cada una de les etapes anteriors.
- Preparació de la presentació: Preparació del material per la defensa del projecte.

## 9.1.2 Calendari de treball

Tot i que el projecte s'ha prolongat durant un total de dos anys el temps real dedicació ha estat menor, perquè han estat diferents motius els que m'han impossibilitat dedicar-me al projecte de forma continuada al llarg d'aquest període.

Per motius laborals i personals m'ha estat impossible portar a terme una continuació en el projecte, una jornada laboral de 8 hores fa molt difícil portar una continuïtat en el desenvolupament del projecte.

Com a resultat d'aquest calendari de dedicació tan dispar la planificació no es mostrarà tota la evolució d'aquests dos anys, sinó que s'ha optat per presentar una planificació amb el treball constant, amb l'objectiu d'obtenir una major claredat.

A continuació es mostra el diagrama de Gantt:



Anàlisi dels requisits	40	40	0
Especificació del sistema	40	40	0
Disseny de l'arquitectura del sistema	150	150	0
Implementació	80	10	70
Validació	20	0	20
Redacció de la memòria	100	80	20
Preparació de la presentació	16	16	0
<b>TOTAL</b>	<b>776</b>	<b>666</b>	<b>110</b>

La duració total del projecte final de carrera es de 776 hores, el què amb una dedicació a mitja jornada (3 hores) equival a 259 dies.

Segons això el cost total del projecte resulta:

Perfil	Hores	Cost/Hora	Cost (Euros)
Analista	666	24	15984
Programador	110	15	1650
Material	776	0.67	519.92
<b>Total</b>			<b>18153.92</b>

És a dir que el cost del projecte és de 18152.92

## 10 Conclusions

A continuació es presenten les conclusions finals del present projecte, primer un repàs sobre la consecució dels objectius i seguidament les millores o línies de futur que es proposen per complementar la feina.

Just quan terminava la meua etapa al Laboratori de Càlcul desenvolupant LeanSim van sorgir dos projectes nous de dos companys del laboratori, el primer aportava un llenguatge d'script al motor de simulació i el segon donava un gir de 180 graus a tot el model de dades passant dels fitxers SCN que es com s'emmagatzemava la informació del model a un nou model basat en la tecnologia XML.

Més endavant mentre desenvolupava el projecte es van crear dos clients nous, LeanStatistics i LeanTraining, cada un aportant noves funcionalitats i aportant canvis en la missatgeria del sistema.

L'anàlisi i comprensió de l'aportació de tots aquests projectes, i l'estudi de quins canvis i repercussions que té la implantació d'una eina de debug sobre tots aquests components esdevé per si sol un projecte final de carrera.

# 11 Agraïments

A Marina per donar-me el suport moral que tant he necessitat.

Als meus amics que m'ajudaran amb els tràmits finals...

A la meva família per tenir tantes ganes com jo d'acabar el projecte.

A Pau Fonseca, per saber-me escoltar i orientar-me en els temes clau del projecte.

A Josep Casanovas per haver-me introduït en el món de la simulació i haver-me escollit per el repte de la tercera pista d'aterratge.

A Esmail Abbas per estar sempre disposats a donar un cop de mà.

A Carlos Carmona per haver-me ajudat en els moments més crítics.

A Monty per aconsellar-me amb la seva saviesa i experiència.

A tots els companys de la facultat que he conegut durant aquests anys.

## 12 Bibliografía

### 12.1 Simulación

-Modelado y simulación. Aplicación a proyectos logísticos de fabricación y servicios. Antoni Guasch, Miquel Angel Piera, Josep. Casanovas, Jaume Figueras. Edicions UPC, febrero 2002.

-Distributed Discrete-Event Simulation. Jayadev Misra. Department of Computer Sciences, The University of Texas at Austin, Texas, marzo 1986.

### 12.2 LeanSim

-Manual de LeanSim®. Jordi Montero. 2005

-Creació de models en LeanSim®. La màquina genèrica. Pau Fonseca, Jordi Montero.

-LeanSim® virtual reality distributed simulation suite. P. Fonseca, Josep Casanovas, Jordi Montero. Proceedings MSO 2004.

-Implantació de tecnologies XML i tecnologies criptogràfiques al simulador LeanSim®. Francesc Guim. PFC-FIB 2003

-LeanStatistics. Narcís Verdaguer. PFC-FIB 2005

-Estudi de determinats assajos clínics mitjançant el simulador genèric LeanSim®. Albert Riera. PFC-FIB 2004

### 12.3 Ingeniería del software

-Ingeniería del software: Especificació. Dolors Costal, María Ribera Sancho, Ernest Teniente. Edicions UPC, 1999