Índex

Índex d'imatges i diagrames

Capítol 1

Introducció

1.1 Introducció

El present document és la memòria del projecte final de carrera "Sistemes gestors de Bases de dades en J2ME: MobileMoodle" per l'enginyeria tècnica en informàtica de gestió de la Facultat d'informàtica de Barcelona (FIB).

Seguint la política de l'autor d'ús de software lliure i de llicència pública, aquesta memòria ha estat integralment redactada mitjançant OpenOffice 2.2 (exceptuant els diagrames de seqüència i de model conceptual de dades de la secció 4, que han estat elaborats mitjançant Omondo UML for Eclipse).

1.2 Context del projecte

Com tot el relacionat amb les tecnologies de la informació, els dispositius mòbils han estat sotmesos a una constant (i quasi-exponencial) evolució, segons la demanda de serveis a que el sector està sotmès. Cada vegada és més freqüent trobar-se amb terminals que integren tota mena de serveis, tant orientats a les comunicacions i transmissió de dades, com a altres finalitats més lúdiques.

Tanmateix, les prestacions tecnològiques d'aquests terminals, com el suport de màquines virtuals de J2ME, fa que siguin candidats molt sòlids a la hora d'escollir-los com a "assistents personals de butxaca". Hom pot pensar, "perquè portar a sobre un mòbil, una pda, un ordinador, una videoconsola, una càmera de fotos, etc..., quan puc portar un únic terminal que integri, encara que de forma i amb prestacions lleugerament reduïdes tots els serveis que puc arribar a necessitar?".

És en el marc de J2ME que el present projecte desenvolupa una solució per a acostar a qualsevol usuari de la plataforma open source moodle a les seves dades. MobileMoodle resulta, en essència, un agregador RSS complet per al mòdul dels fòrums de la plataforma moodle.

Aprofitant els serveis de comunicacions de dades que ofereixen els actuals dispositius mòbils que implementin una màquina virtual de J2ME (anomenada KVM, acrònim de Kilobyte Virtual Machine), s'ofereix a un usuari de moodle, el descarregar els RSS dels cursos a on està subscrit aquest usuari, fer-los persistents en el terminal, mitjançant una sèrie de llibreries anomenades J2MESDLIB, una api que implementa un senzill però alhora potent sistema gestor de bases de dades relacional per J2ME, i desenvolupada per altres projectistes dirigits per Mª José Casany i Marc Alier.
A més, els usuaris de MobileMoodle, han de poder contestar als missatges d'un determinat fòrum, així com consultar i actualitzar les seves dades a voluntat.
En els capítols posteriors s'aborda amb detall quins son els procediments que s'han de dur a terme per tal de complir amb els requisits de funcionament del projecte.

Aleshores, què és moodle? En essència, moodle és una plataforma CMS (Course Management System), que permet la creació i gestió de campus virtuals, i que aporta tota una sèrie d'eines (fòrums, wikis, blogs, etc...) que faciliten, tant a usuaris alumnes com a professors, el desenvolupament, seguiment i monitorització de cursos online (és doncs, una molt potent eina groupware).

1.3 Objectius del projecte

Actualment, no existeix cap solució software en J2ME disponible per a terminals mòbils que implementi les funcionalitats descrites (més endavant veurem exemples d'aplicacions i/o plataformes virtuals diferents de moodle que implementen funcionalitats semblants).

Originalment es considerava la opció de fer servir un servidor auxiliar a mode de Proxy, que estigués sincronitzat amb la base de dades de moodle, i que rebés les peticions d'informació d'un terminal mòbil. Aquest Proxy, interpretaria la petició d'informació del client mòbil, la traduiria a una petició al servidor de moodle, i en rebre la resposta d'aquest, li remetria al client mòbil, en un format lleuger i fàcilment intel·ligible per a aquest.

Tot i semblar una solució acceptable, es va descartar, pel fet que suposa un gran impacte en l'usuari final; pensem en un usuari que vol fer servir el nostre midlet. El fet d'haver-lo implementat fent servir el recolzament d'un servidor extern, ens obligaria a disposar en tot moment d'un servei actiu, per garantitzar el correcte funcionament de MobileMoodle. Això planteja dos opcions.

1. O bé, disposar d'un software client, allotjat a l'ordinador personal de l'usuari, i destinat a rebre peticions del midlet, i sincronitzat amb la base de dades de moodle (fet que, per una altre banda, obligaria a l'administrador del sistema a facilitar informació restringida de la base de dades, a fi de garantir l'ús de cada software client de MobileMoodle)
2. Disposar d'un únic servei centralitzat, destinat a rebre totes les peticions que es facin des de un midlet MobileMoodle, i gestionat per l'administrador de sistemes d'aquest servidor de moodle concret.

Tenint en compte els inconvenients que tant una com l'altre solució aporten, es planteja una possible tercera alternativa, que consisteix en afegir un webservice al servidor de moodle, de forma que l'administrador del sistema, es lliure de decidir si vol fer compatible amb MobileMoodle o no el seu sistema, i que permetria la comunicació entre el midlet i la base de dades del servidor de moodle, a on està ubicada tota la informació que l'usuari pot arribar a necessitar durant una seqüència d'esdeveniments produïda pel midlet.

Així doncs, l'objectiu del projecte consisteix en desenvolupar una solució software, que permeti a un usuari de moodle (ja sigui alumne, o professor), consultar i participar de forma senzilla, i des de qualsevol lloc, els fòrums dels cursos a on està subscrit.

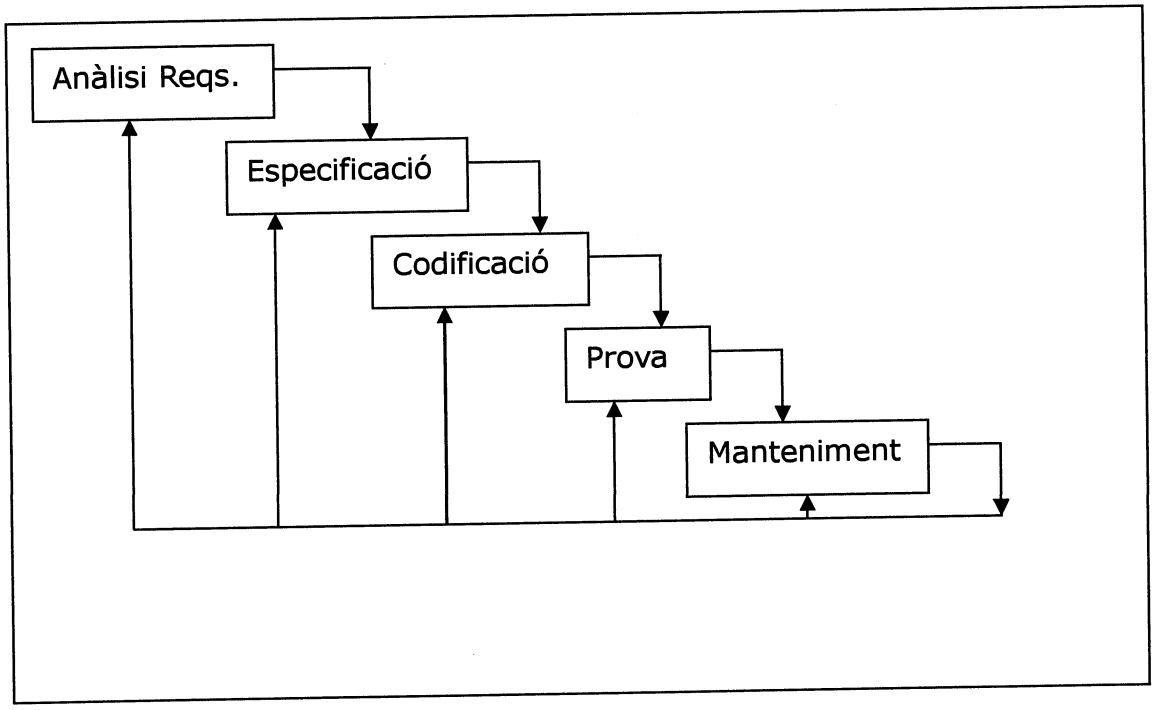
El projecte també contempla la integració d'un webservice implementat especialment , que mitjançant RMI (Remote Method Invocation), accedeix a la informació adequada, així com l'ús de les llibreries J2MESDLIB, essent el present projecte un banc de proves ideal per aquestes llibreries, encara en desenvolupament i perfeccionament.

1.4 Procés de desenvolupament

El paradigma de desenvolupament del sistema software escollit ha estat el conegut com a "Cicle de vida clàssic", o model de "salt d'aigua". El procés de cicle de vida clàssic estableix una sèrie de fases, ordenades de forma jeràrquica, de la que s'obtenen una sèrie de documents, que permeten l'avaluació i en cas de ser necessari, refinament del treball realitzat fins al moment, de forma que, fins que una tasca concreta no es completa de forma satisfactòria, no se'n comença una d'un nivell inferior.

Cal comentar que les diferents fases de que consta el procés escollit, permeten establir un grau d'iteració, de forma que si en una fase es detecta un error concret, es pot tornar a la fase anterior per refinar-ho.

El cicle de vida clàssic respon al següent diagrama:



· Fig. 1. Cicle de vida clàssic.

- Anàlisi de requisits i especificació. A partir de l'anàlisi dels requeriments especificats per l'usuari, es realitza una especificació del sistema software, que consta d'un diagrama del model de dades conceptual, així com el model de casos d'ús.

- Disseny. A la fase de disseny s'adapta la solució obtinguda després de la fase d'especificació, a una tecnologia concreta, tot seguint una arquitectura de software definida. Aquesta fase també contempla el disseny de la capa de dades, així com el de la capa de domini i la de presentació (interfície d'usuari).

- Codificació. En aquesta fase es duu a terme la implementació del sistema ja dissenyat, fent servir el llenguatge de programació i tecnologies escollides. També és en aquesta fase que s'apliquen les possibles correccions tant al model de dades com de casos d'ús, segons els errors que es puguin detectar durant aquesta fase.

- Prova. Aquesta fase estableix la creació d'un banc de proves, per testejar la implementació ja operativa del nostre sistema; és necessari establir un banc de proves exhaustives, que exposi els errors d'implementació del sistema software, a fi de corregir-les.

- Documentació. Aquesta fase consta de l'elaboració de la documentació relativa al sistema software. A fi d'evitar incoherències, degudes a possibles canvis en el disseny i/o implementació produïts durant les successives iteracions en les fases del procés de software, s'ha de dur a terme al mateix temps que les fases anteriors.

Capítol 2

Anàlisi de requisits

2.1 Requeriments

Els requeriments d'un sistema software determinen les condicions o capacitats necessitades per un usuari per tal de solucionar un problema o assolir un objectiu. Per tal d'aconseguir elaborar una llista fiable i el més acurada possible de requeriments, cal entendre els objectius i necessitats de l'usuari, així com definir el conjunt de sistemes que podrien satisfer aquestes necessitats/objectius. Un cop elaborada la llista, cal triar la solució proposada més adient.

En aquesta secció es detalla la llista de requeriments funcionals i no funcionals a què respon l'especificació del nostre sistema software.

2.1.1 Requeriments funcionals

Els requeriments funcionals d'un sistema software especifiquen quines accions ha de poder realitzar el sistema sense tenir en compte restriccions físiques.
Com a estratègia principal d'obtenció dels requisits funcionals del sistema, s'ha optat per analitzar el que en essència resulta un sistema software similar, com ara un agregador RSS per convencional (FeedReader, en concret, un software gratuït i distribuït sota llicència freeware) i ja implementat.

A continuació, es detalla la llista de requisits funcionals de la nostra aplicació:

- Autenticació en el sistema. L'usuari ha de ser capaç d'autenticar-se una única vegada en el terminal mòbil, fent servir les dades tal i com les faria servir en cas de realitzar una acció d'autenticació en moodle. L'usuari també ha de ser capaç d'establir l'adreça del servidor de moodle a on es vol autenticar (sempre que sigui un servidor moodle compatible amb MobileMoodle).

- Visualització de tots els fòrums dels cursos a on l'usuari està subscrit. Donat un usuari autenticat en el sistema, aquest ha de ser capaç de veure els fòrums que pertanyen als seus cursos.

- Visualització dels temes de discussió d'un fòrum concret. Donat un usuari i un fòrum, l'usuari ha de ser capaç d'obtenir una llista amb tots els temes de discussió que conté el fòrum donat.

- Visualització del cos d'un tema de discussió, de forma que l'usuari autenticat en el sistema sigui capaç de llegir-ne el contingut de forma correcta.

- Participació en el tema de debat. Donat un usuari autenticat en el sistema i el cos d'un tema de debat d'un fòrum a on aquest usuari tingui accés, aquest usuari ha de ser capaç de contestar a aquest missatge, de forma que se'n garanteixi la participació des de qualsevol lloc.

- Actualització voluntària de les dades. Un usuari autenticat en el sistema ha de ser capaç d'actualitzar les dades persistents introduïdes al sistema, relatives al seu perfil d'usuari i descarregades del servidor de moodle, així com del contingut dels fòrums (també fets persistents en el terminal mòbil) a voluntat.

- Gestió de les dades. Un usuari autenticat en el sistema, ha de ser capaç tant de visualitzar les dades corresponents al seu perfil d'usuari, així com esborrar-les a voluntat. De la mateixa manera, un usuari autenticat, ha de ser capaç d'esborrar les dades persistents relatives als fòrums dels cursos a on està subscrit.

2.1.2 Requeriments no funcionals

Els requeriments no funcionals, també anomenats factors de qualitat, defineixen tot un seguit de propietats i qualitats desitjables per al nostre sistema software, que descriuen com s'ha de comportar en realitzar les tasques esperades.

En destaquem els següents:

- **Eficiència**. El nostre sistema software ha de ser capaç de realitzar totes les tasques esperades, sense comprometre l'estabilitat del sistema, consumint excessius recursos, com ara memòria, sobretot, tenint en compte que està destinada a executar-se en un terminal de prestacions tècniques força més reduïdes que les de qualsevol ordinador personal. Les funcionalitats requerides també s'han d'executar de forma ràpida, donat que l'usuari no pot esperar tenir ocupat el terminal durant un període de temps molt gran; un procés que trigués massa en executar-se, podria donar a l'usuari la sensació de que l'aplicació està bloquejada, o ha patit algun error.

  És important trobar un equilibri entre l'eficiència en temps i recursos; no obstant, també s'ha de tenir en compte que, l'ample de banda subministrat als terminals mòbils per part de les operadores de telefonia, acostuma a ser reduït i car; per tant , l'eficiència en temps, també ve donada per aquest fet. En la implementació de la aplicació, s'ha prestat especial atenció a aquest fet, fent servir dades lleugeres i minimitzant el tràfic de dades al màxim.

- **Fiabilitat**. El sistema ha de realitzar les operacions degudes de forma fiable i amb el mínim nombre d'errors possible, informant en tot cas a l'usuari, si es produeix algun error. Aquest punt es crucial, donat que, la nostra aplicació és de tipus distribuïda, i en un determinat instant, és possible que la xarxa o la connexió (de igual manera que el servidor de moodle en sí) no estiguin disponibles per causes alienes a la nostra aplicació, fet que, inevitablement, provocaria la indisponibilitat de la nostra aplicació.

- **Portabilitat**. Donat la gran quantitat de fabricants de terminals mòbils, models de dispositius, sistemes operatius per a aquests, i en definitiva, heterogeneïtat d'aquest mercat, la nostra aplicació, forçosament, ha de ser totalment independent de la plataforma a on s'executi, garantint, d'aquesta manera, que pot arribar al major nombre d'usuaris possible.

- **Mantenibilitat**. El nostre sistema software ha de ser fàcilment mantenible, de forma que, en un futur, sigui possible i senzill corregir errors que es produeixin amb els canvis produïts en els sistemes externs al sistema, així com implementar noves funcionalitats, de forma que se n'allargui el temps de vida.

- **Reusabilitat**. L'aplicació s'ha de construir de forma que se'n puguin aprofitar el major nombre de mòduls en futurs projectes/implementacions de característiques similars.

Capítol 3

Arquitectura

3.1 Proposta d'arquitectura

L'arquitectura del software és una descripció dels subsistemes i components (computacionals) d'un sistema software, i les relacions entre ells.
La determinació de l'arquitectura del software consisteix en la presa de decisions respecte a:

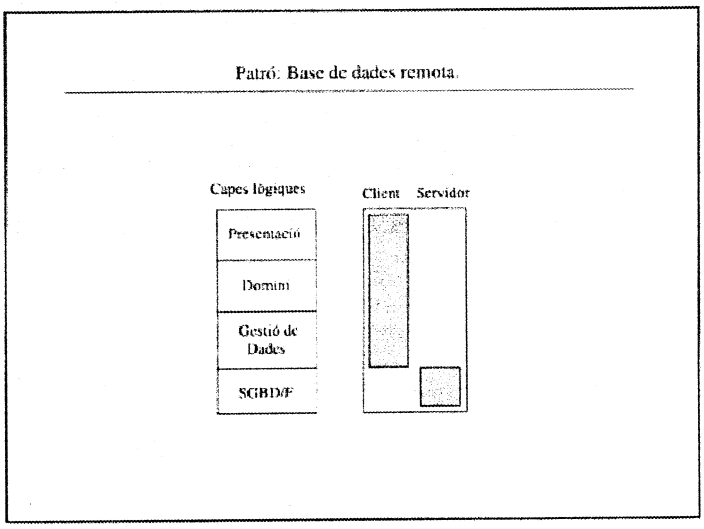
- L'organització del sistema software
- La selecció dels elements estructurals i les seves interfícies
- Comportament d'aquests elements estructurals
- La possible composició dels elements estructurals en subsistemes més grans
- L'estil que guia aquesta organització

I tenint en compte els requeriments no funcionals del sistema software que es volen assolir.

Fonamentant-nos en el fet que el nostre sistema està majoritàriament compost per elements remots, s'ha escollit com a arquitectura del sistema la basada en el mode client/servidor.

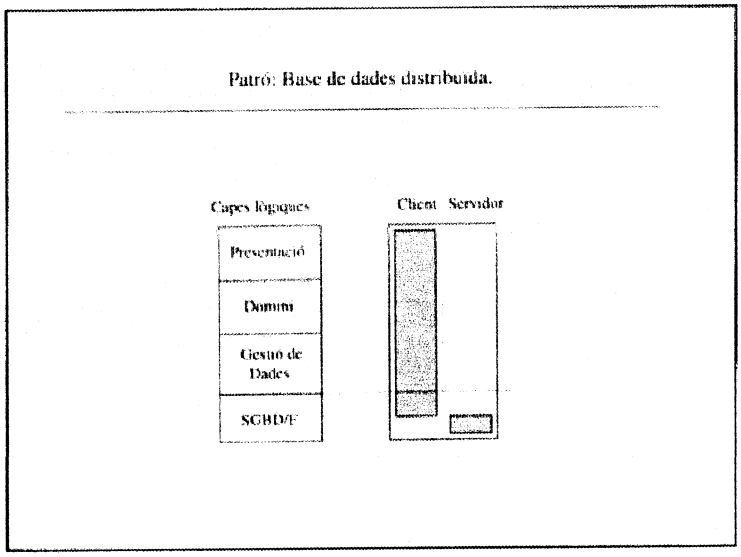
El sistema software serà un client, que es comunicarà amb el servidor a on es troben les dades que es volen consultar mitjançant serveis web, i s'hi comunicarà a la hora d'enviar missatges al servidor de forma directa mitjançant connexions HTTP.

Els requeriments funcionals del nostre sistema software ens obliguen a escollir una solució mixta del patró arquitectònic dades distribuïdes i dades remotes, donat que les dades son persistents en el servidor, però alhora, s'hi fan persistents en el terminal que executa la nostra aplicació.



· Fig. 2. Visió general del patró base de dades remota

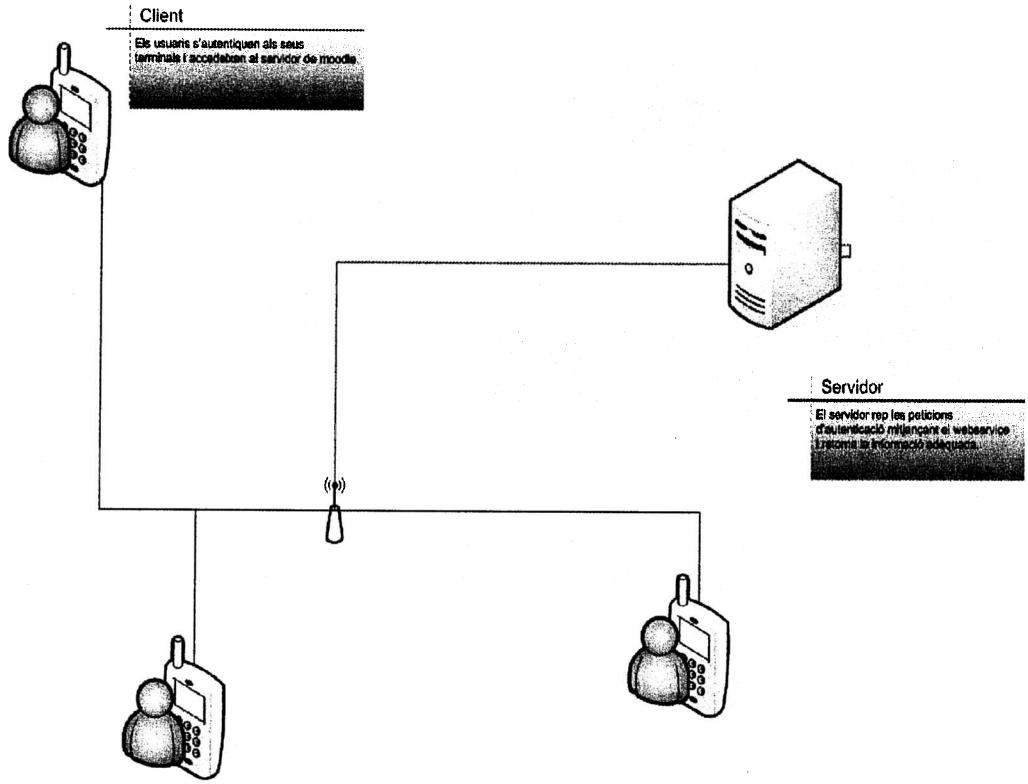
L'avantatja de la solució escollida vers el patró arquitectònic dades remotes rau en el fet que les dades es troben tant al client com al servidor (amb el patró dades remotes ens trobem que les dades les posseeix únicament el servidor).



· Fig. 3. Visió general del patró base de dades distribuïda

La base de dades ja existent al servidor de moodle conté tota la informació necessària per al correcte funcionament del nostre sistema software, de forma que, quan un usuari s'autentiqui de forma efectiva en el servidor a través de la nostra aplicació, aquesta es descarregarà les dades relatives a aquest usuari, i en farà una representació a la BBDD local, de forma que estalviï temps de

connexió (i per tant, cost monetari) a l'usuari en futurs usos del sistema.
Durant la fase de proves, la base de dades estarà allotjada en un servidor de moodle especialment instal·lat en un ordinador personal especialment configurat per a aquesta finalitat, del qual l'autor n'és el propietari.



· Fig. 4. Esquema de funcionament de MobileMoodle

Donat que l'aplicació en sí mateixa és un client, s'encarrega d'obtenir la informació adequada del servidor mitjançant crides a webservices. En concret, s'encarrega de cridar a un webservice implementat per Oriol Nieto i Alex Moreno, antics alumnes de la FIB, i creadors del projecte final de carrera "Plugin per a exportar/importar Nwikis de moodle a tiddlyWikis", dirigit per Marc Alier i Forment.

Aquest webservice (en endavant **webservicelib**) s'encarrega de facilitar vàries funcionalitats a la nostra aplicació:

- Una primera fase en que el sistema software envia les dades de l'usuari al webservice, i aquest determina si és un usuari registrat a la base de dades del servidor o no. Donat l'usuari, si està registrat a la base de dades, genera una resposta en llenguatge XML (detallada més endavant), amb tota la informació necessària.

- Més endavant, quan un usuari vol contestar a un missatge concret d'un fòrum, el sistema software s'encarrega d'obtenir informació crítica del missatge en qüestió, per tal de construir un objecte conceptual que representi un missatge, per tal d'enviar-lo al servidor.

Cal dir que, l'especificació i disseny referent als mètodes que cal implementar al webservice per tal d'obtenir la informació crítica per al correcte funcionament de la nostra aplicació han estat elaborats per l'autor del present projecte. A més, per definir l'esquema d'organització estructural del sistema, s'ha escollit el patró arquitectònic en tres capes, que estructura el sistema en un nombre apropiat de capes verticals, de forma que totes les capes que pertanyen a un mateix nivell horitzontal han de treballar al mateix nivell d'abstracció. D'aquesta manera, els serveis que proporciona la capa j, son utilitzats per la capa superior (j+1). Tanmateix els serveis de la capa j, poden dependre d'altres serveis d'aquesta mateixa capa.

Així doncs, aquesta seria la visió abstracta del nostre sistema software:

- **Capa de presentació:** S'encarrega de la interacció amb l'usuari, tant ajudant-lo a introduir al sistema les dades requerides, com mostrant la informació rellevant i desitjada per l'usuari. Desencadena tota la sèrie d'esdeveniments que s'executen a les capes inferiors (treballa a un alt nivell d'abstracció).

- **Capa de domini:** Responsable de la implementació de les funcionalitats del sistema. Rep esdeveniments de la capa de presentació, en controla la validesa, canvia l'estat del domini, executa les operacions encomanades, en rep els resultats i els envia cap a la capa de presentació. Treballa a un nivell mig d'abstracció, amb objectes conceptuals (tals com usuaris, missatges, etc....)

- **Capa de dades:** S'encarrega d'interactuar amb el sistema gestor de bases de dades, i de gestionar les comunicacions externes: efectua les crides als webservices per obtenir les dades, es comunica amb el servidor de moodle a fi d'enviar dades. Comunica els resultats obtinguts de les crides remotes a la capa de domini.

3.2 Tecnologies

En aquesta secció es descriuen les tecnologies emprades en el procés de desenvolupament del projecte. De igual forma es detallen les eines que s'han fet servir, així com mòduls externs de tercers ja existents.

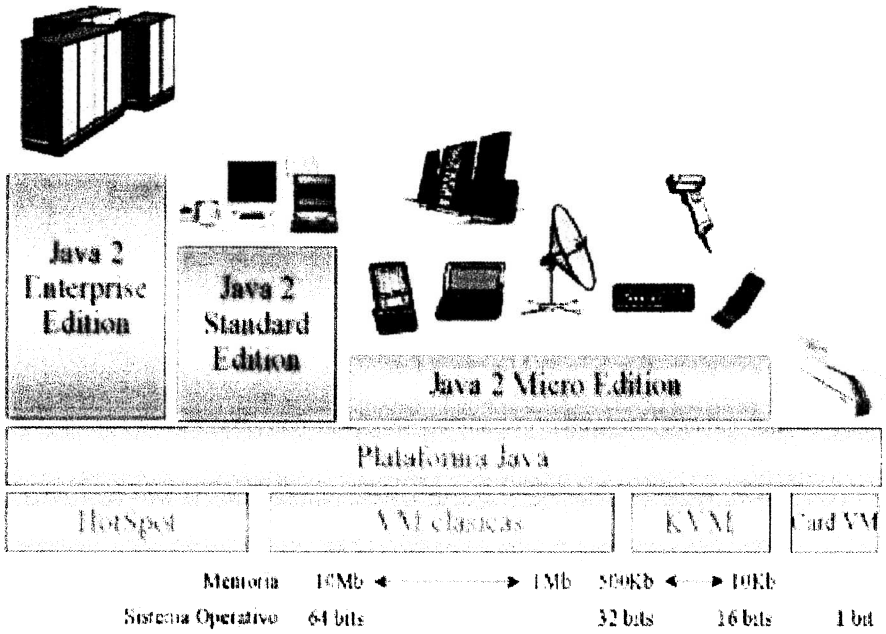
3.2.1 Java: J2ME

Java es un llenguatge de programació orientat a objectes i multi tasca amb el que
obtenim portabilitat dels programes desenvolupats.
La portabilitat s'aconsegueix gracies a la seva principal característica : de la compilació no s'obté cap fitxer executable. El compilador de Java genera un fitxer .class, que conte els anomenats bytecodes (que son un representació intermèdia del programa compilat independents de la plataforma). Els bytecodes seran interpretats per una màquina virtual al moment de l'execució del programa. Aquesta màquina virtual es dependent de la plataforma a la qual s'executi el programa, però se'n disposen per a la majoria de plataformes.

Com a resultat s'obté que el programador només necessita disposar de l'entorn de desenvolupament i no s'ha de preocupar de per a quina plataforma esta destinat el programa (això no sempre es completament cert, p.e. quan els programes han d'accedir a disc o a altres aplicacions del sistema).
D'altra banda, qui executi el programa necessitarà la maquina virtual corresponent a la plataforma que faci servir. Tot i que avui dia, quasi totes les companyies proveïdores de sistemes operatius i navegadors web han optat per incloure de forma implícita una màquina virtual de Java, segons els especificacions de Sun Microsystems, companyia propietària del llenguatge.

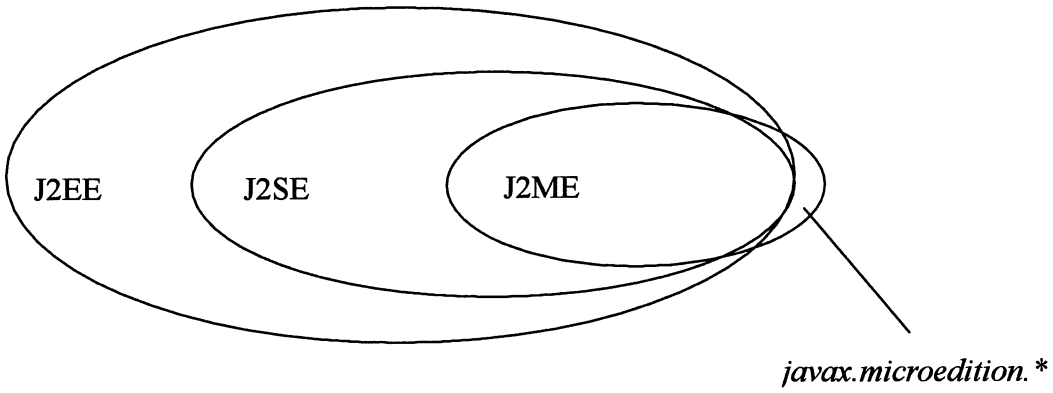
Dins del context de Java, apareix J2ME, que es defineix com un subconjunt de totes les llibreries que ofereix la versió estàndard del llenguatge, i destinat a desenvolupadors de software per a dispositius mòbils (tals com terminals de telefonia mòbil, PDAs, etc...). J2ME està preparat per a executar-se en dispositius de prestacions tècniques reduïdes, i, segons s'ha mencionat amb

anterioritat, disposa de la seva pròpia màquina virtual de Java, anomenada, KVM (Kilobyte Virtual Machine, donat que només necessita uns quants kilobytes per funcionar). També s'implementa un ràpid recol·lector de deixalles, destinat a purgar la memòria del dispositiu.



· **Fig.5. Esquema de J2ME en comparació a les altres versions del llenguatge.**

J2ME és, de fet, un subconjunt de J2EE i J2SE, que aporta algunes noves APIs úniques i exclusives dels dispositius mòbils als que està dirigit (relatives a les comunicacions remots, emmagatzemat de dades, i visualització de dades per pantalla), totes englobades dins de la api *javax.microedition.**. Això és d'aquesta manera degut a les anteriorment esmentades limitacions físiques que comporta el fet d'executar software en dispositius d'aquestes característiques.



· Fig. 6. Ubicació de J2ME dins del context de Java

3.2.3. XML

XML, o eXtensible Markup Language és un metallenguatge que permet definir llenguatges de "marcat" segons les nostres necessitats. XML no té només aplicació a Internet, sinó que es proposa com llenguatge de baix nivell (a nivell d'aplicació) per a l'intercanvi d'informació estructurada entre diferents plataformes. Això és degut a la facilitat per estructurar la informació definint nous llenguatges dins l'estructura bàsica del llenguatge.
La informació en XML s'estructura en documents. En podem tenir de 2 tipus:

- Documents ben formats: són aquells que segueixen les especificacions sintàctiques del llenguatge i no estan subjectes a l'estructura de cap DTD

- Documents vàlids: son documents ben formats que a més segueixen l'estructura definida per un DTD. Un DTD (Document Type Definition) defineix una sintaxi i unes restriccions sobre l'estructura d'un document XML.

Al projecte, XML s'empra com a format de comunicació entre el sistema software i el webservicelib, i per definir tant l'estructura dels continguts com les seves descripcions semàntiques i possibles atributs.
S'empren, per tant, 3 tipus d'estructures XML: les referents a les dades de l'usuari, les referents a un RSS d'usuari, i les referents a les dades crítiques d'un missatge del fòrum.

A continuació, es detalla el format d'aquestes tres estructures (documents) XML:

- Estructura XML referent al contingut del perfil d'un usuari:

```
<?xml version="1.0" encoding="UTF-8"?>
    <profile>
        <person>
            <name>Nom d'usuari</name>
            <surname>Cognoms</surname>
            <email>email</email>
            <town>localitat</town>
            <username>username</username>
            <password>password</password>
            <userID>id d'usuari</userID>
            <rss>
                <link name="nom del fòrum" course="curs a que
                pertany el fòrum" forumid="id d'instància de
                fòrum">adreça</link>
                <!-- pot tenir-ne més d'un -->
            </rss>
        </person>
    </profile>
```

Aquesta estructura XML és la que el webservicelib s'encarrega de construir una vegada ha validat l'usuari que el sistema software li envia mitjançant una pipe HTTP.
La informació necessària s'ha triat seguint el criteri de tria d'informació necessària per el funcionament de l'aplicació, és a dir, crítica per a poder garantir el compliment dels requisits funcionals especificats, afegint algunes propietats escollides arbitràriament, tals com el correu electrònic de l'usuari, la seva localitat, el seu nom i cognom, etc....
En cas de no existir cap usuari a la base de dades de moodle amb les credencials facilitades, el webservice construeix una resposta sense contingut, que respon a la següent estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
    <profile>
    </profile>
```

És a dir, un document sense continguts (indicant que la cerca a la base de dades no ha donat cap resultat.

- Estructura XML obtinguda en fer una petició al webservice per a obtenir la informació crítica necessària per construir un objecte conceptual que representi un missatge de fòrum ben format.
  En aquest cas, la informació necessària, impossible d'obtenir de cap altre manera que no sigui realitzant una crida al webservice (que genera una consulta a la BBDD de moodle, per després generar un document XML com el que segueix), ha resultat ser d'identificador relatiu al que és el primer missatge que obre un tema de discussió en un fòrum, és a dir, el pare del que serà el missatge que volem enviar al fòrum.
  D'aquesta manera, quan volem contestar a un missatge d'un fòrum, realitzem una crida al webservicelib, tot indicant-li quin és d'identificador del tema de discussió al que volem contestar des de la nostra aplicació.

```
<?xml version="1.0" encoding="UTF-8"?>
    <firstpost>id del primer missatge del tema de discussió indicat
```

```
</firstpost>
```

- Estructura XML relativa als documents RSS. RSS és un format d'arxiu de la família XML desenvolupat específicament per a llocs de notícies i weblogs que s'actualitzen amb freqüència i per mitjà del qual es pot compartir la informació i usar-la en altres llocs web o programes. És en essència una redifusió de continguts. Per influència de la llengua anglesa s'acostuma a referir-s'hi utilitzant el barbarisme sindicació.

  L'acrònim s'usa per als següents sistemes de comunicació estàndard:

  - Rich Site Summary (RSS 0.91)
  - RDF Site Summary (RSS 0.9 i 1.0)
  - Really Simple Syndication (RSS 2.0)

  Els programes que llegeixen i presenten fonts RSS de diferents procedències es denominen agregadors. Moodle utilitza documents RSS versió 2.0. L'estructura estàndard d'un document RSS és la següent:

```
<?xml version="1.0"?>
    <rss version="2.0">
        <channel>
            <title>titol del canal que genera el RSS</title>
            <link>enllaç al lloc web que dona suport al
            RSS</link>
            <description>breu descripció del RSS</description>
            <language>idioma per defecte del RSS</language>
            <pubDate>data de publicació</pubDate>
            <lastBuildDate>data de l'ultima actualització
            </lastBuildDate>
            <docs>enllaç al RSS</docs>
            <generator>software creador del RSS</generator>
```

```
                    <webMaster>
                    email del webmaster del lloc web que dona suport al
                    RSS
                    </webMaster>
                    <item>
                        <title>títol d'un element del RSS</title>
                        <link>enllaç a l'element en qüestió</link>
                        <description>cos del text de
l'element</description>
                        <pubDate>
                        data de publicació de l'element
                        </pubDate>
                    </item>
                </channel>
            </rss>
```

A moodle, els documents RSS generats per donar suport de sindicació als fòrums, es generen d'acord amb l'execució d'un script, que es dedica a realitzar tasques de manteniment. L'execució regular d'aquest script, és predeterminada per l'administrador del sistema, fet que també condiciona la freqüència d'actualització dels RSS dels fòrums de moodle.

3.2.4 KXML

Donat un document XML, cal fer servir el que s'anomena un parser, una aplicació (o conjunt de llibreries) externes que s'encarregui de buscar en el document XML les dades desitjades.

De parsers XML, en trobem de 3 tipus:

- Els de tipus push. Els parsers de tipus push llegeixen tot el document XML sencer, i notifiquen a l'aplicació que els controla mitjançant objectes de tipus *listener*, és a dir, observadors.

- Els de tipus pull. Els parsers pull processen fragments del document XML, i notifiquen a l'aplicació que els controla, de forma que cada vegada que es processa un fragment de document, l'aplicació decideix què n'ha de fer.

- Els de tipus model. Aquesta mena de parser llegeix el document XML sencer, i en crea una representació conceptual a la memòria del dispositiu, resultant lleugerament més ineficients en consum de recursos.

Fins la data actual, la especificació MIDP 2.0 de J2ME no incloïa cap mena de parser XML; en la última implementació, s'ha inclòs un parser (del tipus DOM) per defecte.
Amb la finalitat d'interpretar els resultats obtinguts a cada crida del webservicelib, s'han inclòs les llibreries de la API kXML, un parser amb llicència EPL (Enhydra Public Licence, semblant a les especificacions de la GPL), de tipus pull, és a dir, llegeix fragments del document d'entrada i avança a petició de l'aplicació que el fa servir.
Cal dir que segons les necessitats del nostre sistema software, és aquesta mena de parser la més adequada, donat que, aquest model de parser (vers els de tipus push i model) ens permet mantenir de forma efectiva el control de l'aplicació, i alhora monitoritzar totes les activitats que es realitzen, així com el seu progrés, evitant d'aquesta manera que l'usuari tingui la sensació que el terminal mòbil s'ha penjat.

Com a consideracions finals sobre kXML, i comparant-lo amb altres alternatives de característiques similars, s'ha perfilat com a la millor, donat que es tracta d'un projecte madur i estable (tot i que actualment ja es troba tancat; els autors, en la actualitat, han implementat una versió 2.0 estable, en la data de realització del projecte, la versió 2.0 es trobava en fase de proves).

3.2.5 MD5

MD5 és un dels algoritmes de reducció criptogràfica dissenyats pel professor Ronald Rives, del *Massachusetts Institute of Technology*, desenvolupat al 1991, i destinat a substituir al MD4 després que Hans Dobbertin en descobrís la seva debilitat.
La codificació d'un resultat obtingut després de tractar una cadena de text amb un MD5 es representa típicament com una cadena de 32 bits hexadecimals. L'algoritme de codificació es redueix a 5 fases:

- (1). Pas 1: El missatge s'extén fins que la seva longitud en bits, sigui congruent amb 448 mòdul 512, fins i tot si inicialment ja ho era. La extensió, es realitza segons el següent criteri: un bit 1 s'afegeix al missatge, i després bits 0 s'afegeixen fins que la longitud en bits del missatge sigui congruent amb 448 mòdul 512.
  Com a mínim, sempre s'afegeix un bit, i com a màxim, 512.

- (2). Pas 2: S'afegeix al final del missatge obtingut a (1) una representació en 64 bits de la longitud inicial abans de (1). Si la longitud del missatge inicial és major a 2 elevat a 64, aleshores només es fan servir els 64 bits de menor pes.
  En aquest punt, hem obtingut un missatge amb longitud múltiple de 512, i que té exactament un múltiple de 16 paraules (32 bits per paraula).

- (3). Pas 3: S'inicialitza un buffer de 4 paraules (A, B, C i D) destinat a calcular el resum del missatge, cadascuna de 32 bits, inicialitzades amb uns valors predeterminats, i considerant els bits de menor pes primer.

- (4). Pas 4: En primer lloc, definim quatre funcions auxiliars que prenen com a entrada les paraules de 32 bits A, B, C i D, de (3), i la sortida és una paraula de 32 bits.

    - **F(X,Y,Z) = (X ∧ Y) v (¬X ∧ Z)**
    - **G(X,Y,Z) = (X ∧ Z) v (Y ∧ ¬Z)**
    - **H(X,Y,Z) = X ⊕ Y ⊕ Z**
    - **I(X,Y,Z) = Y ⊕ (X v ¬Z)**

  Els operadors **v**, ∧, ⊕, i ¬ són els operadors matemàtics OR, AND, XOR i NOT, respectivament.

  En aquest pas es fa servir una taula de 64 elements T[1...64] construïda amb la funció sinus. L'element i-èssim de T és la part entera del valor absolut del sinus de i 4294967296 vegades, siguent la unitat de i els radians.

- (5). Pas 5: Es recull la sortida en 4 paraules de 32 bits cadascuna (A, B, C i D), ordenades des de el bit de menor pes de A, fins al bit més pesant de D.

Donat el fet que el nostre sistema software intercanvia informació confidencial referent a l'usuari que el fa servir, és pràcticament obligat garantir-ne la privacitat de les dades. Es per aquest motiu, i pel fet que les dades a la BBDD de moodle estan codificades mitjançant aquest algorisme, que s'ha pres la decisió d'encriptar el tràfic de dades existent entre el nostre sistema software i el servidor de moodle mitjançant l'algorisme MD5.

Per a dur a terme aquesta tasca, s'han inclòs les llibreries corresponents a la implementació de l'algorisme MD5 anomenada Fast MD5, una implementació específica per J2ME, realitzada per Timothy W. Macinta, i distribuïda sota llicència GNU LGPL versió 2.1
Cal destacar-ne l'eficiència, fins i tot vers les llibreries nadiues de Java.

3.2.6 Serveis web

Un servei web (Web Service) és un recull de protocols i estàndards que permeten intercanviar dades entre aplicacions. Aplicacions desenvolupades sobre llenguatges de programació diferents i executant-se sobre qualsevol plataforma poden utilitzar serveis web per intercanviar dades.

La interoperabilitat s'aconsegueix mitjançant l'adopció d'estàndards
oberts. Entre aquests estàndards podríem destacar l'ús del XML com a format de les dades a retornar per part del servei web.

L'ús dels serveis web aporten els següents beneficis:

- Es basen en HTTP sobre TCP al port 80. Això permet que les comunicacions via serveis web no es vegin bloquejades pels tallafocs que puguin tenir instal·lats les empreses

- Aporten gran independència entre l'aplicació que crida al servei i el servei en sí. Només és necessari que el client faci una crida al servidor a on es troba allotjat el servei web, fent servir el protocol d'intercanvi escollit, i com a resposta n'obtindrà un XML amb les dades esperades.