



Generalitat de Catalunya

Departament d'Educació

Institut Milà i Fontanals. Igualada

Dossier



Informàtica

ICB0 **Desenvolupament
d'aplicacions multiplataforma**

NF2 Execució d'un projecte

Equip docent de DAM2

Curs 2022/2023



FAMÍLIA: Informàtica i comunicacions	
CICLE: Desenvolupament d'aplicacions web	GS
MÒDUL PROFESSIONAL: Projecte de desenvolupament d'aplicacions web	CODI: M13
HORES TOTALS: 99	HLLD: 0
NOM PROFESSOR/S: Marc Brufau / Daniel Cano / Isidre Guixà / Ester Marsal / Bernat Orellana	

1.- RELACIÓ D'UNITATS FORMATIVES I NUCLIS FORMATIUS

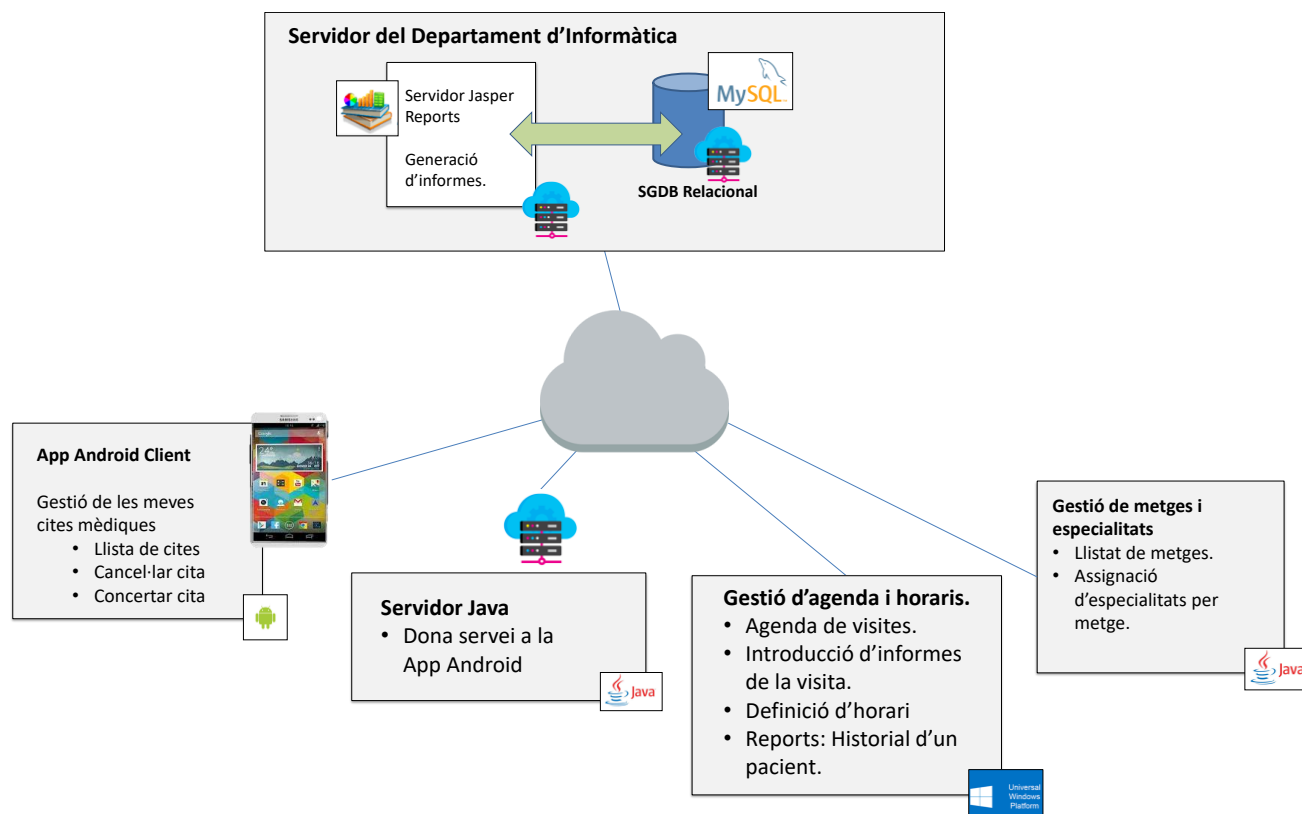
M13 – Projecte de desenvolupament d'aplicacions multiplataforma						
Unitats Formatives	Hores mín. HLLD + NF	Durada	Data d'inici	Data final	H. setmanals	
UF 1: Projecte de desenvolupament d'aplicacions multiplataforma	99 + 0	1	21	10a setm.	30a setm.	1
		2	78	3 darreres setmanes		

UF 1: Projecte de desenvolupament d'aplicacions multiplataforma (99 h)		
Nucli Formatiu	Hores	Resultats d'aprenentatge
NF 1: Gestió de projectes	21	1-4
NF 2: Execució d'una part del projecte	78	5

Aquest document correspon als continguts del NF2

NF2. Execució d'una part del projecte (78h)

Visió General



Es desitja desenvolupar un sistema de gestió de cites mèdiques, accessible des de múltiples dispositius. Es desenvoluparan 3 aplicacions:

- Una aplicació UWP que utilitzarà el metge per gestionar la seva agenda i horaris.
- Una aplicació Java de backoffice per gestionar les assignacions d'especialitats del metges.
- Una aplicació Android que permetra al pacient llistar les cites, cancel·lar-les, o concertar cites noves.

Servidors externs: SGBDR i Jasper Report Sever

A excepció del client mòbil, les aplicacions fan us d'un SGBDR centralitzat per a dur a terme la persistència i compartir dades. El SGBDR serà un MySQL hostatjat al servidor del departament d'informàtica. Per agilitzar el procés de desenvolupament, recomanem usar un SGBDR en local, però sempre tenint en compte que caldrà posteriorment canviar la configuració i usar el SGBDR remot.

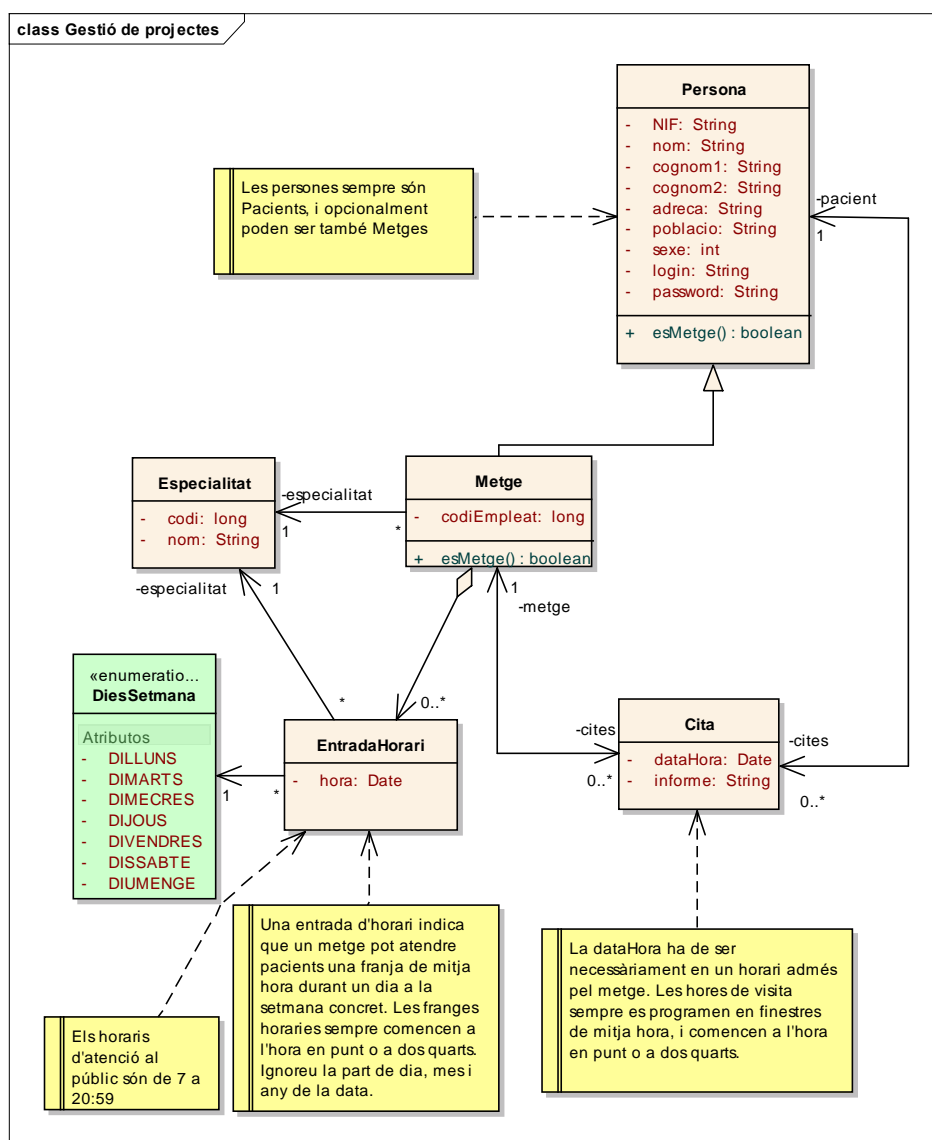


El client Android **no té accés al SGBDR**, per obtenir les dades usarà com a servidor una aplicació Java de consola que es comunicarà amb el mòbil via sockets TCP/IP usant un protocol propi definit per vosaltres. A partir de la petició del mòbil, el servidor es connecta al servidor de BD per obtenir/actualitzar la informació, processa les dades i li dona resposta al mòbil.

El servidor principal del departament proveirà també d'un servei JasperReports, que haureu d'utilitzar adequadament per tal que el client UWP pugui obtenir els informes.

Model genèric de classes

A la figura següent es mostra un diagrama de classes UML amb l'estructura de la informació sobre la que construirem el sistema d'informació.





Control de versions

Tot el codi del projecte es mantindrà **des de l'inici** sota el control d'un GIT, utilitzeu a tal efecte un compte gratuït a **GitHub**, **GitLab** o **BitBucket**,

Persistència

L'intercanvi de dades entre les aplicacions Java i .Net es farà mitjançant un SGBDR, que actuarà com a node central de dades entre els sistemes. L'excepció a aquesta regla serà l'App Android, que **no accedirà en cap cas** a la base de dades.

La base de dades que doni suport a l'aplicació ha de poder residir en SGBD de diverses tecnologies (SGBDR, SGBD-XML, SGBDOR i SGBDOO). Això implica disposar d'una interfície que defineixi els diversos mètodes que ha d'implementar el component concret per a cada cas.

Cal dissenyar la interfície i el(s) component(s) per a UN SGBD concret, en Java i en UWP.

El SGBD triat és MySQL.

IMPORTANT: El programa ha d'anar acompanyat, obligatòriament, de 3 guions:

- [1] Eliminació d'estructures de dades
- [2] Creació d'estructures de dades amb introducció de dades inicials imprescindibles pel funcionament (si és el cas)
- [3] Inserció de dades de demostració.

L'eina JPA usada per les aplicacions Java (escriptori i servidor), en connectar amb el SGBDR, no han d'efectuar cap canvi (res de create, update,... ni similar dins l'arxiu persistence.xml).

Primera fase:

- Cal presentar al professor responsable de M02-M06 el diagrama E-R (en paper és suficient però de forma clara i amb detall de tots els camps), que haurà de ser acceptat.
- L'alumne haurà de repetir la presentació del diagrama E-R fins que li sigui acceptat, el més aviat possible (pot trepitjar els dies següents).

Segona fase:

- Guió amb les instruccions DDL per la creació de l'estructura de la BD en el SGBDR escollit del model relacional corresponent al diagrama E-R acceptat. El guió ha d'incorporar totes les `constraints` adequades amb noms adequats.



- Diagrama del model relacional corresponent al guió, creat amb qualsevol dels programes existents que generen diagrama a partir de la BD (per exemple, DBVisualizer)

Concrecions per les aplicacions Java

- El model de classes ha de ser únic per l'aplicació escriptori i per l'aplicació servidor Java.
Les classes persistents no han d'incorporar cap marcatge, però sí han d'incorporar les característiques habituals per poder usar-les en persistència (*Serializable*, constructor sense paràmetres amb l'accés que correspongui,...)
- La interfície ha de ser única, utilitzable per l'aplicació escriptori i per l'aplicació servidor Java. Per tant, ha de contenir tots els mètodes susceptibles de ser invocats per qualsevol de les dues aplicacions.
- Cal implementar component JPA d'obligat us per l'aplicació escriptori. Els mètodes d'aquest component no poden usar cap instrucció directa JDBC ni cap consulta nativa SQL. El marcatge ha de ser extern al codi de les classes (marcatge XML). Si no ho fem així podem tenir problemes amb Android.
- Cal implementar component JDBC d'obligat us per l'aplicació servidor Java.
- Els 2 components només cal que incorporin els mètodes necessaris per l'aplicació que usará cada component. Per la resta de mètodes que no s'usin del component, incorporar-los amb el codi:

```
throw new UnsupportedOperationException("Not supported yet.");
```

Aplicació d'escriptori , client complet (UWP)

S'implementarà una aplicació d'escriptori amb tecnologia UWP de forma que suporti les operacions següents:

- 1) Agenda de visites, en forma de graella en format setmanal (horari de 7 a 21 hores). Al fer click sobre una visita es mostra a la dreta informació (nom i edat del pacient, i l'informe si està escrit). Permet escriure l'informe de la visita i tancar la visita només si són visites del mateix dia (o d'un dia anterior) i encara no s'han tancat.

	DL 10/04/23	DM 11/04/23	DX 13/04/23	DJ 14/04/23	DV 15/04/23	DS 16/04/23	DG 17/04/23	
7:00								Pacient: Josep Benet i Jornet Edat: 78 anys Informe: <div></div>
7:30								
8:00								
8:30								
9:00								
9:30								
10:00								
	<			Avui		>		Desar

**Institut Milà i Fontanals. Igualada**

La interfície permetrà avançar i retrocedir a la setmana posterior/anterior, i tornar a la setmana actual. El dia actual estarà destacat amb un color de fons.

- 2) Gestió d'horari: defineix les franges horaris que té disponibles. De dilluns a diumenge, en franges de 1/2 hora, en horari de 7 a 21 hores. La finestra serà una taula de 7 columnes x 28 files, representant l'agenda. A cada casella hi haurà un desplegable on inicialment hi haurà seleccionada l'opció "N/D" (no disponible), i les altres opcions del desplegable seran les especialitats que té el metge assignades (p.ex. "Pediatría", "Cardiologia").

NOTA: Les especialitats de cada metge es gestionen des de l'aplicació Java (veure apartat següent)

	DL	DM	DX	DJ	DV	DS	DG
7:00	N/D	N/D	N/D	N/D	N/D	N/D	N/D
7:30	N/D	N/D	N/D	N/D	N/D	N/D	N/D
8:00	N/D	N/D	Cardiologia	N/D	Cardiologia	N/D	Pediatría
8:30	N/D	N/D	Cardiologia	N/D	Cardiologia	N/D	Pediatría
9:00	N/D	N/D	Cardiologia	N/D	Cardiologia	N/D	Cardiologia
9:30	N/D	N/D	Cardiologia	N/D	Cardiologia	N/D	Pediatría
10:00	N/D	N/D	N/D	N/D	N/D	N/D	N/D

L'accés a les dades es farà mitjançant el component de dades adequat al SGBD escollit.

Aplicació d'escriptori - Java/Swing

S'implementarà una petita aplicació d'escriptori amb tecnologia JAVA SWING, de forma que suporti les operacions següents:

- 1) Llistat de metges: Es mostra la llista de metges, podem filtrar per especialitat.
- 2) Quan cliquem sobre un metge es mostren les seves dades (només consulta) i la seva llista d'especialitats.
- 3) Podrem eliminar especialitats de la llista del metge i afegir-ne de noves (triant-les en un desplegable)

L'accés a les dades es farà mitjançant el component JPA de dades.

No pot usar cap accés a la BD que no sigui via component JPA.



App Android: Gestió de visites

Caldrà que dissenyeu les pantalles necessàries per a:

- Fer login (autenticació, només el primer cop)
- Llista de consultes concertades (presentes i futures, no tancades pel metge), ordenades per data (ASC). Mostra data, hora, especialitat, nom del metge. Cada cita té un botó cancel·lar que, prèvia confirmació, permet anul·lar la cita.
- Com a opció de menú (a la barra d'aplicació) podem concertar una nova visita. Això ens portarà a una nova pantalla amb el següent fluxe:
 - 1) Seleccionar especialitat des d'un **Spinner**
 - 2) Seleccionar metges per l'especialitat triada d'un altre **Spinner**
 - 3) Triar el dia en un **DatePickerDialog**
 - 4) Mostra una llista de hores lliures del metge.
 - 5) Selecciona l'hora desitjada i confirma.
 - 6) Torna a la pàgina de llista de cites.

Pel que fa a la compatibilitat, l'aplicació cal que suporti versions d'Android a partir de l'API level 26 (Android 8.0).

Servidor (Java)

Cal dissenyar un servidor basat en sockets TCP/IP usant llenguatge Java, responsabilitzant-se de donar servei a les transaccions amb clients remots (Android). El servidor no requerirà cap tipus d'interfície gràfica, tot i que es recomana desenvolupar una interfície mínima que incorpori una finestra de depuració i algun botó de control (p.ex. aturar/engegar).

L'accés a les dades es farà mitjançant el component JDBC de dades.

No pot usar cap accés a la BD que no sigui via component JDBC.

Podeu definir un protocol al vostre gust que doni sortida a les següents operacions (l'intercanvi d'informació pot ser en binari, json o XML):

Notació:

tuples	{camp1, camp2,...}
arrays	[item]*



Codi	Petició	Paràmetres	Retorn
1	Login	{user, password}	{session_id, tupla_dades_usuari} o {0} si no és correcte
2	GetCites	{session_id}	[cita]*
3	GetEspecialitats	{session_id}	[especialitat]*
4	GetMetgesPerEspecialitat	{session_id, especialitat_id}	[metge]*
5	GetForatsDisponibles	{session_id, metge_id,especialitat_id,data}	[hora_disponible]*
6	ConcertaVisita	{session_id, metge_id,especialitat_id,data, hora_disponible}	{0} si error, {1} si ok.

JasperReport

Cal elaborar el següent informe i publicar-lo en el JRS del departament:

Historial d'un pacient (NIF del pacient com a paràmetre). Es mostrarà una tarja amb la informació bàsica del pacient, i una llista de les seves visites ordenades per data descendent. Cada visita tindrà la data, nom del metge i especialitat, i l'informe que el metge ha escrit.

Aquest informe s'ha de poder executar des de:

- Servidor JasperReport, on caldrà facilitar un desplegable que mostri **NIF - Cognoms, Nom** dels pacients, ordenats per Cognoms, Nom en ascendent, per a que l'usuari en seleccioni un i es mostri el corresponent informe.
- L'aplicació UWP mitjançant un control **WebView**, passant el NIF del pacient.

Algunes indicacions a considerar:

- Useu una capçalera amb logotip empresarial, un títol adequat.
- Cal també un peu que mostri la data i hora d'elaboració, paginació, etc.

Entrega i avaluació

El lliurament i presentació del projecte es farà entre els dies 29 i 31 de maig (segons volum de lliuraments) en horari comunicat la setmana anterior i amb avaluació col·legiada.

Caldrà lliurar el projecte en suport digital. Hi haurà demostració del funcionament i justificació de les eleccions preses per part de l'alumne

Cada mòdul s'avaluarà per separat proporcionant una nota única que serà la que es lliurarà a l'alumne. En el cas que alguna de les notes dels diferents MP sigui menor a 4, la nota final del projecte serà insuficient fins que es recuperi la part/parts suspeses en horari indicat segons el calendari de 2a convocatòria.