

FAMÍLIA: Informàtica i comunicacions	
CICLE: Desenvolupament d'Aplicacions Multiplataforma	GS
MÒDUL PROFESSIONAL: Projecte de desenvolupament d'aplicacions Multiplataforma	CODI: M13
HORES TOTALS: 99	HLLD: 0
NOM PROFESSOR/S: Marc Brufau / 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)



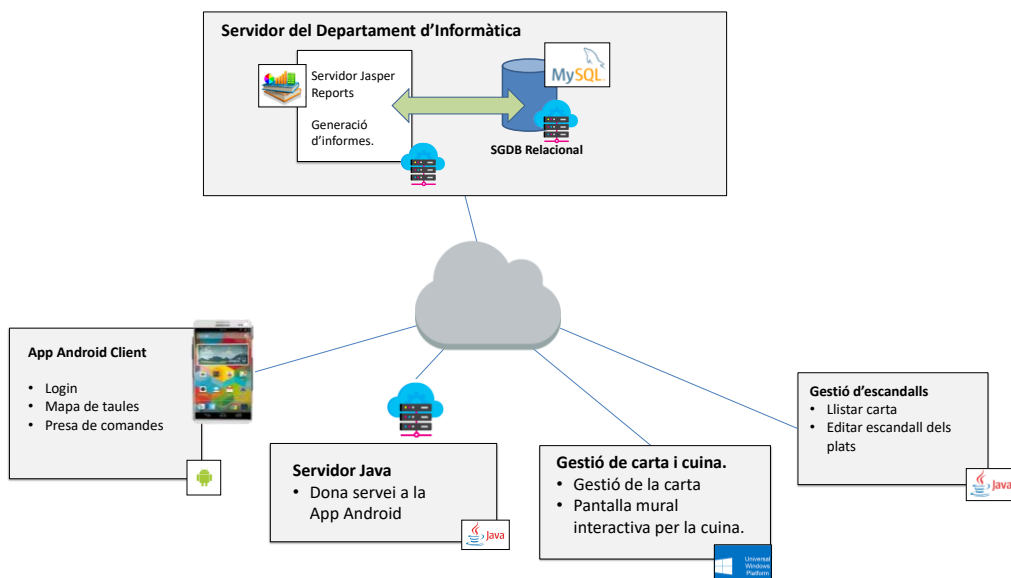
Cook-o-matic

L'empresa cook-o-matic vol llançar al mercat un sistema integral de gestió de restaurants, enfocat a ajudar en el dia a dia d'un establiment de restauració. L'aplicació s'especialitza en la confecció de les cartes, la gestió de les comandes dels clients i la coordinació sala-cuina.

El sistema informàtic que es vol construir estarà format per diversos components que treballen plegats per donar aquest servei. Es programaran 4 petites aplicacions que es comuniquen entre elles usant un SGBD MySQL i connexions directes a través de sockets TCP.

Les aplicacions es descriuen a continuació:

Aplicació	Tipus	Descripció
Gestió d'escandalls	Java / escriptori	Aplicació d'escriptori programada amb Java+MySQL destinada a gestionar l'escandall dels plats de la carta.
Gestió de carta i mural per la cuina.	UWP / escriptori	Aplicació d'escriptori programada amb UWP (.NET Core) que s'encarrega de: <ol style="list-style-type: none"> 1. Gestionar la carta. Creació de nous productes, modificació i baixa. 2. Gestionar les comandes per la cuina
App per prendre nota de les comandes.	Java / Android	És una App per mòbils Android que mostrarà els elements del menú al cambrer i li permetrà prendre comandes a les taules.
Servidor Java	Java / consola	Servei que fa d'interfície entre la base de dades i l'App Android. Totes les operacions de petició de dades i modificació de dades a Android passen per aquest servidor, que posteriorment fa la transacció corresponent amb MySQL.



II-lustració 1:Components del sistema d'informació.

1.1 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, recomanen 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.

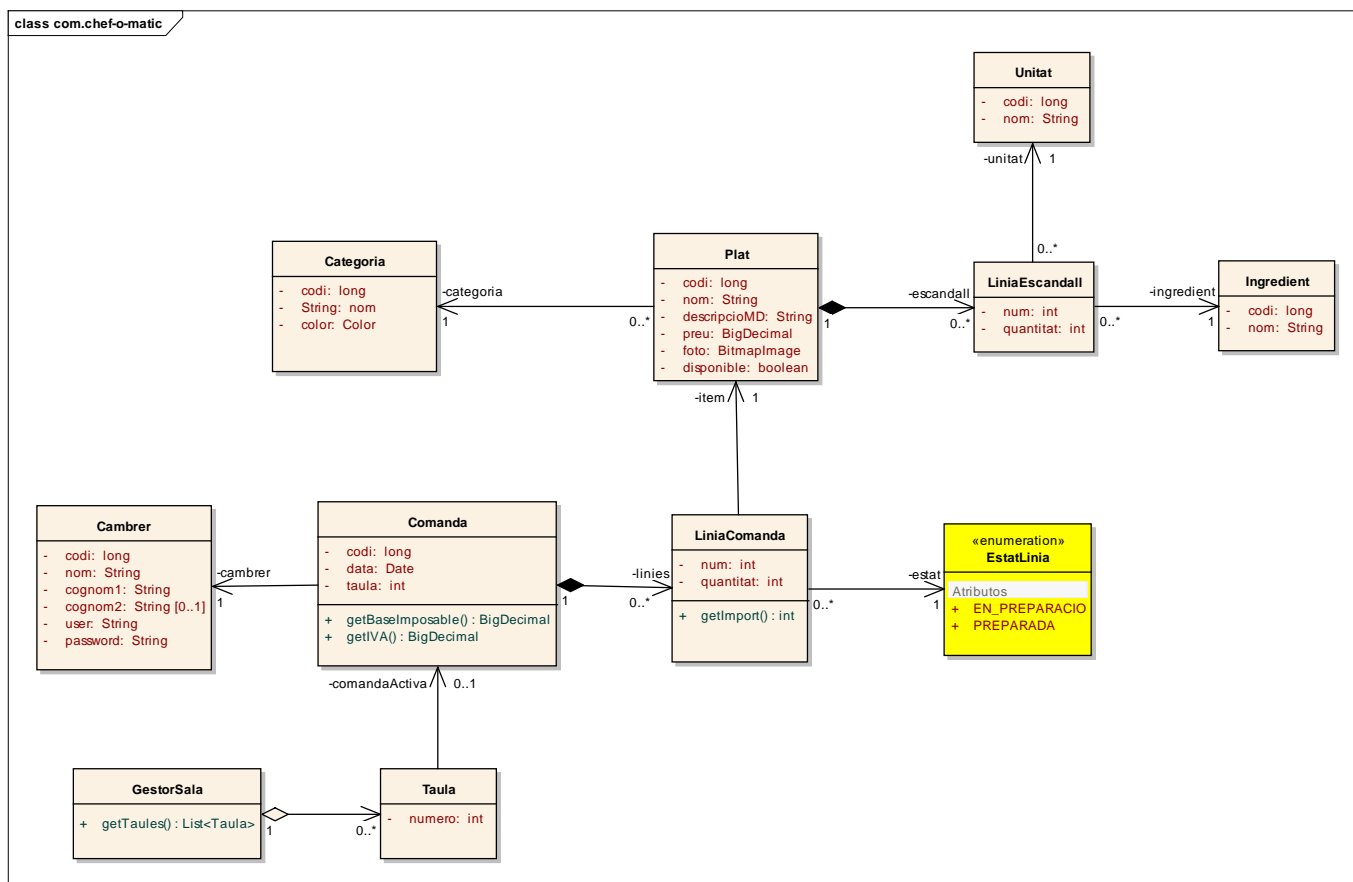
El servidor principal del departament proveirà també d'un servei JasperReports, que haureu d'utilitzar adequadament per tal que el client UWP pugui obtenir informes sobre la carta.

1.2 Model genèric de classes

A continuació es mostra el diagrama de classes conceptual de les aplicacions. No implica que totes les aplicacions requereixin totes les classes, o que no en calguin més segons les necessitats de cada tecnologia.

Fixeu-vos que els plats de la carta s'organitzen per categories que, generalment, són bastant estàndards: "Entrants", "Primers", "Segons", "Postres". Però poden ser totalment diferents segons la tipologia de restaurant (imagineu-vos un restaurant japonès...).

La composició d'ingredients de cada plat, el que es coneix com **escandall**, i ens indica senzillament quantes unitats de cada ingredient necessitem per preparar un plat. Això és vital de cara a preveure i planificar les compres de matèria primera per la cuina. La taula d'unitats i la d'ingredients assumirem que són estàndard i que no cal modificar-les. Per tant, a la base de dades haurem de preparar una llista de categories, unitats i ingredients com a part del joc de proves.



1.3 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**,

1.4 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 el component només per a una tecnologia concreta:

- Per l'aplicació d'escriptori Java, via JPA, amb qualsevol proveïdor.
- Per l'aplicació d'escriptori UWP, via Entity Framework, usant consultes preparades.

En qualsevol cas, el SGBDR escollit és MySQL.

IMPORTANT: El programa ha d'anar acompanyat, **obligatòriament**, de 3 guions (eliminació d'estructures de dades, creació d'estructures de dades i inserció de dades de demostració). L'aplicació Java, en connectar amb la BD, mirarà la coherència de la BD amb les classes de l'aplicació i no ha de fer cap canvi (res de *create*, *update*,... ni similar en crear l'EntityManager).

1.5 Gestió d'escandalls. Aplicació d'escriptori (Swing/Java)

L'aplicació es desenvoluparà en tecnologia Swing i ha de permetre les següents operacions:

- Visualitzar la llista de plats, i poder filtrar per:
 - Categoria (amb un desplegable), de manera que si no hi ha categoria seleccionada visualitzi tots els plats. Els plats cal visualitzar-los en el color assignat a la seva categoria.
 - Disponibilitat, amb 3 botons de selecció (Si, No, Totes)L'aplicació ha de facilitar un botó Cercar per a que l'usuari executi la cerca.
En posar en marxa l'aplicació, no hi haurà cap categoria seleccionada i botó Disponibilitat estarà a Totes, però sense efectuar cap cerca fins que l'usuari l'executa via botó Cercar.
- Seleccionar un plat i editar el seu escandall. Les línies de l'escandall no són editables, només podem crear-ne de noves i eliminar-les.
El plat seleccionat i el seu escandall cal mostrar-lo en una nova finestra, on cal mostrar tota la informació del plat (amb disseny acurat) i el seu escandallat. En finalitzar la gestió del plat seleccionat, l'aplicació ha de tornar el control a la pantalla inicial.
- La taula d'unitats i d'ingredients ja està pre-creada a la base de dades i no es editable.

Observació tècnica respecte el component JPA:

- Cal configurar el fitxer persistence.xml amb, com a mínim, dues unitats de persistència (UP): una per Hibernate i una per EclipseLink.
- El fitxer persistence.xml no ha de contenir cap dada de connexió, doncs seria difícil canviar-ne les dades.
- Poder canviar les dades de connexió (SGBD, IP, port, usuari, contrasenya) i la UP a usar ha de ser fàcil.

1.6 Gestió de la carta i cuina. Aplicació d'escriptori (UWP)

S'implementarà una aplicació d'escriptori amb tecnologia UWP amb les següents funcionalitats:

1) Pantalla de gestió de la carta.

- a) Es mostra una llista de categories a la banda esquerra de la pantalla. A la seva dreta es veuen tots els plats. Quan seleccionem una categoria es filtra la llista de plats segons la categoria triada. S'ha de poder desactivar el filtre de categoria i tornar a la llista completa.

- b) Addicionalment, es pot filtrar el llistat per nom del plat.
- c) Es pot esborrar un plat seleccionat de la llista (prèvia confirmació), sempre que no tingui referències dins de comandes.
- d) Es permet crear un nou plat. Les fotos es desaran a la base de dades en una columna **MEDIUMBLOB**.
- e) Les descripcions dels plats acceptaran la sintaxi Markdown per tenir l'opció de donar formats rics de text de forma senzilla. Useu el control `MarkdownTextBlock` del nuget `Microsoft.Toolkit.Uwp.UI.Controls` per previsualitzar com queda el text formatat.

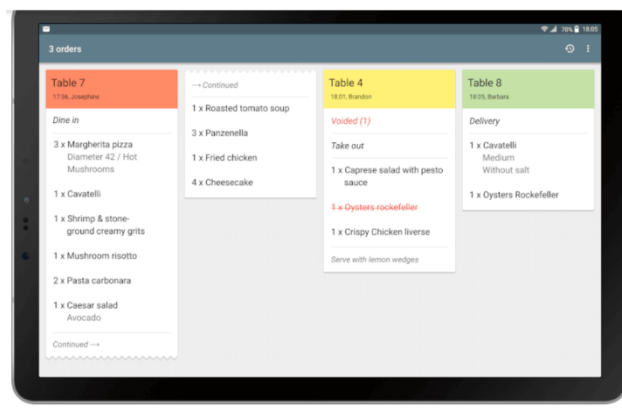
2) Pantalla pel mural de la cuina.

Serà una pantalla de l'aplicació pensada per treballar en un enorme monitor tàctil on els cuiners puguin veure còmodament les comandes que van entrant a cuina.

Es mostrarà una graella amb cada comanda dins d'un requadre, encapçalada amb el títol "Taula X", la hora que s'ha registrat la comanda i el cambrer. Sota la capçalera hi haurà la llista de plats demanats (i comentaris si n'hi ha). Cada plat preparat es podrà clicar i quedarà taxat o marcat amb un check. Els elements marcats "van caient llista avall", deixant els plats pendents a la part superior. Quan s'han marcat tots els ítems d'una comanda, es demana confirmació i aquesta desapareix.

IMPORTANT:

- Les comandes s'ordenen per ordre invers de temps d'espera.
- La pantalla es refrescarà cada 2s per visualitzar les comandes entrants enviades pel cambrer.



1.7 App Android pel cambrer

Caldrà que dissenyeu les funcionalitats següents:

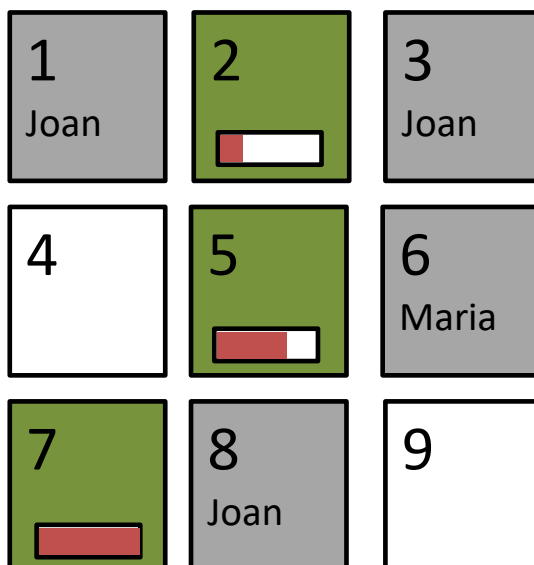
1) Splash screen

2) Login

Permetrà fer login al cambrer. Un cop autenticat, l'aplicació sempre engegarà amb el mateix usuari. (Useu *Shared Preferences*)

3) Activity de taules

- a) Mostra una llista de taules en una quadricula. Dins de cadascuna es veurà:
 - i) El número de taula
 - ii) El nom del cambrer que actualment serveix la taula. Si no és el cambrer que utilitza la App, apareix en gris, altrament surten de color verd. Les taules buides apareixen blanques.
 - iii) Es mostrarà una barra de progrés que és proporcional a la quantitat de plats "checked" a la cuina.
 - iv) La pantalla es refrescarà cada 2s.



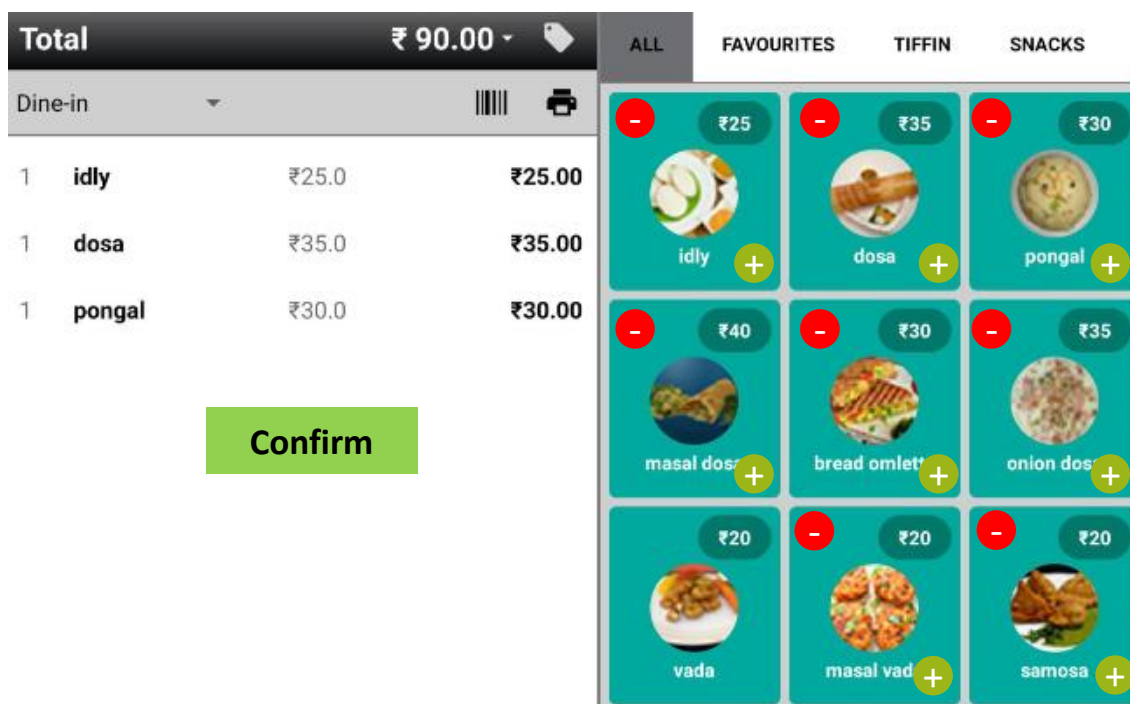
- v) Si cliquem sobre una taula, anirem a l'activity de presa de comandes (veure punt següent). Si la taula estava buida (blanca) es crea una nova comanda, si la taula estava amb comanda activa, es passarà a **visualitzar** la comanda existent.
- vi) Si fem long-click sobre una taula amb comanda activa, preguntarem amb un diàleg si es vol "buidar" la taula (s'entén que els clients que hi havia a la taula ja han marxat).

4) Activity de presa de comandes

Mostrarà una graella amb tots els plats, on es veu la foto, el nom i el preu. Podem afegir o treure plats a la comanda usant botons sobre la imatge. Els plats estan organitzats per categories (només les de primer nivell de l'arbre).

Només es poden fer canvis quan creem per primer cop la comanda. Si tornem a aquesta pantalla més endavant només permetrà visualitzar, sense editar res.

Els canvis realitzats no s'apliquen fins que es prem el botó confirmar. Si es prem "back" es perden els canvis.



1.8 Servidor (Java)

El servidor Java compleix la funció de proveir de dades a l'aplicació Android. Tota la informació que necessita el mòbil li demana via un socket TCP al servidor Java, que residirà a una IP fixa. Teniu molta informació a la xarxa de com implementar un client-servidor TCP amb Java, però us recomanem aquesta [referència](#).

Consells:

- Creeu objectes "contenidors de dades" amb la informació precisa que cal enviar del servidor al mòbil. **No useu objectes que tinguin marcadors de persistència.**
- Aquests objectes són els mateixos tant pel server Java com per Android, per tant els podeu "compartir".
- Useu `ObjectOutputStream` i `ObjectInputStream` per enviar i rebre els objectes.
- És preferible no serialitzar tots els objectes sencers, envieu-los camp per camp.

El servidor no requerirà cap tipus d'interfície gràfica, tot i que es recomana desenvolupar una interfície mínima que incorpori una finestra de log i algun botó de control (p.ex. aturar/engegar)

L'accés a les dades s'implementarà directament usant JDBC contra el SGBD.

Podeu definir un protocol al vostre gust que doni sortida a les operacions descrites en el punt següent. Generalment s'envia primer pel socket el codi de la operació que es vol fer i posteriorment els paràmetres associats a aquesta operació.

1.8.1 Operacions del protocol de xarxa que suporta el servidor

Notació:

Tuples	{camp1, camp2....}
Arrays	[item]*

Peticions Aplicació client (Android) → Servidor Java

Codi	Petició	Paràmetres	Retorn
1	Login	{user, password}	{session_id, tupla_dades_usuari} o {-1} si no és correcte
2	GetTaules	{session_id}	{ num_taulas, [info_taula]*} num_taulas:= número sencer que es correspon al número de taules info_taula:={numero, codi_comanda, es_meva, plats_totals,plats_preparats, nom_cambrer, }, on codi_comanda és -1 si no té comandes actives i es_meva és cert si la taula té comanda i la porta el cambrer autenticat.
3	GetCarta	{session_id}	{ num_categories, [categoria]*, num_plats, [plat]*} Categoria:={codi ,nom} Plat:={codi ,nom, preu, mida_bytes_foto, stream_foto}
4	GetComanda	{session_id, codi_comanda}	{num_linies_comanda, [línia_comanda]*} Línia_comanda:={num, quantitat, codi_plat}

5	CreateComanda	{session_id, taula, num_linies_comanda, [línia_comanda]*}	{codi_comanda} de la nova comanda creada si tot va bé, {-1} si hi ha errors
6	BuidarTaula	{session_id, taula}	0 si tot va bé, -1 si hi ha errors

1.9 JasperReport

De cara a obtenir una versió imprimible de la carta, utilitzarem JasperReport Community del servidor del departament des de l'aplicació UWP, usant un control **WebView**.

L'informe ha d'estar parametritzat per Categoria. Hi haurà un desplegable que ens permetrà seleccionar la categoria dels plats que volem veure, i es farà l'informe exclusivament de la categoria triada. També hi haurà l'opció de no filtrar i veure tota la carta sencera.

- a) Per a cada plat de la carta
 - i) Es mostren totes les dades i la foto en una fitxa.
 - ii) Es mostra el desglossament de l'escandall en un format semblant a una taula.

Algunes indicacions a considerar:

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

2 Entrega i avaluació

Cada setmana l'alumne informará del seu progrés fent un reunió on mostri la feina feta durant la setmana.

El lliurament del projecte es farà els dies 3 i 4 de juny en l'horari comunicat el dia anterior i amb avaluació col·legiada.

Caldrà lliurar el projecte en format digital. Hi haurà demostració del funcionament de les aplicacions i es podran demanar explicacions sobre els detalls d'implementació o de les decisions preses al llarg del desenvolupament.

IMPORTANT:

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.

Hi haurà l'opció de recuperar les parts suspeses a la convocatòria extraordinària.