



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

## 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)

## 1 Visió General



Un club de billar vol disposar d'un sistema de gestió per facilitar les tasques de control de socis, organització de tornejos, i per donar serveis de valor afegit als seus usuaris.

El billar es juga amb diferents modalitats, potser les més conegudes són la modalitat lliure, a una banda i a tres bandes. Cada modalitat és en si tot un món, i un jugador generalment té nivells de destresa diferents per a cadascuna. Resumint, un potser boníssim en lliure i un patata jugant a 3 bandes.

L'esquema general d'una partida de billar és simple, els jugadors s'alternen en torns o **entrades**. A cada entrada tenen l'opció d'intentar fer carambola (d'acord a la modalitat que es juga). Si aconsegueix fer carambola torna a tirar, i d'aquesta forma pot anar encadenant moltes caramboles seguides fins

que falla. En aquest moment, s'acaba l'entrada i comença l'entrada de l'altre jugador. Les partides s'acaben quan un jugador aconsegueix el nombre de caramboles fixat per l'organització, o quan s'arriba al límit d'entrades. En aquest últim cas, guanya el jugador amb més caramboles. També hi ha la possibilitat que un jugador abandoni la partida.

Per fer les competicions més interessants pels jugadors, és freqüent que en un mateix torneig s'organitzin grups, de forma que els jugadors de nivell similar juguen entre ells, fent la competició més justa i equilibrada.

Els nivells dels jugadors es pot determinar amb el que s'anomena **coeficient**, que és simplement el nombre mig de caramboles per entrada que ha aconseguit.

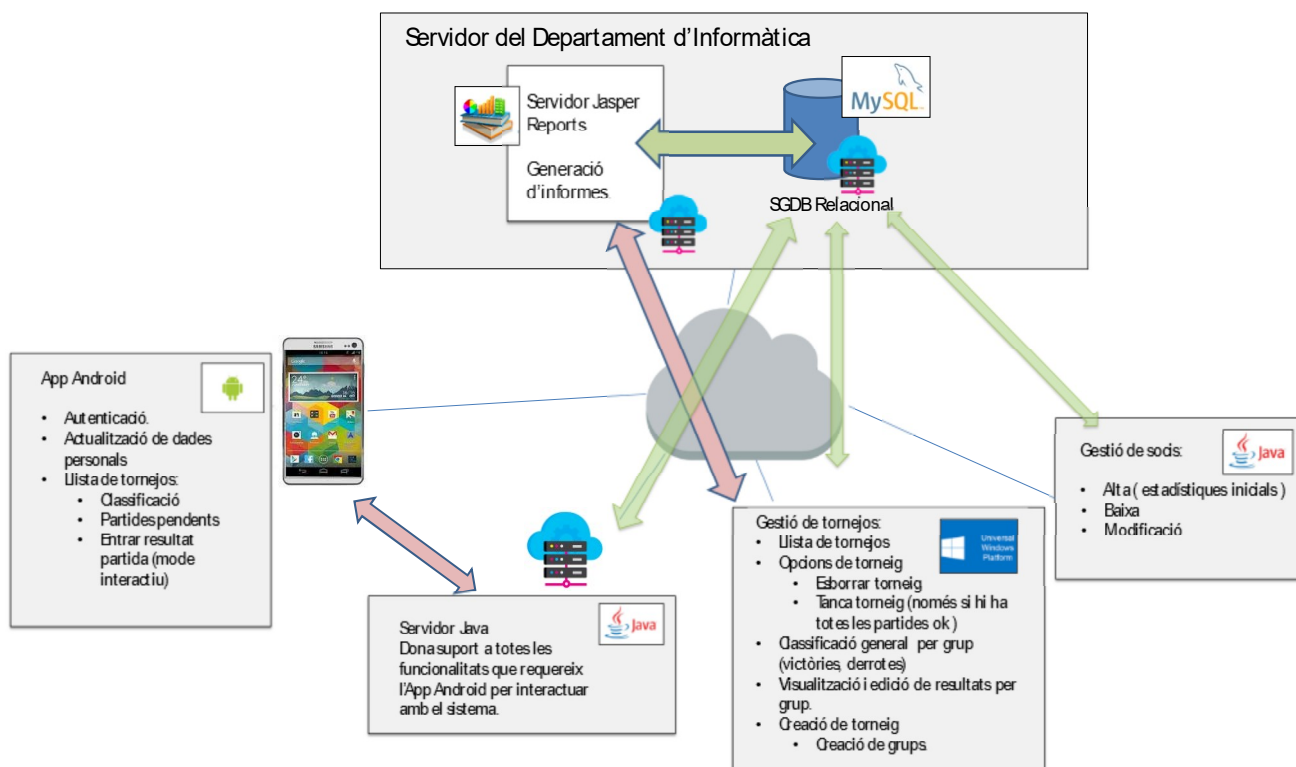
$$C = \frac{\text{total caramboles}}{\text{total entrades}}$$

El coeficient d'un jugador és diferent segons la modalitat, p.ex. en Paco Taco té un 4.3 en lliure, i un 1.3 a una banda. Quan es registra un soci nou al club, se li demana quin és el seu coeficient base en cadascuna de les modalitats. Així mateix, per interès estadístic es fa un recompte de caràcter anual de les caramboles i entrades de cada soci per cada modalitat en els diferents tornejos en els que participa. Amb aquestes dades podem calcular el coeficient de l'any actual.

## 2 Sistema d'informació

Des del punt de vista de l'usuari, el sistema consta de tres **aplicacions**:

- 1) **Gestió de socis**: És una aplicació Java-Swing que permet dur el manteniment de socis.
- 2) **Gestió de tornejos**: Aplicació d'escriptori programada amb UWP (.NET Core) destinada a gestionar tot el cicle de vida d'un torneig.
- 3) **App pels socis**: És una App per mòbils Android que permetrà tenir informació actualitzada dels tornejos on participa i introduir resultats de partides interactivament.



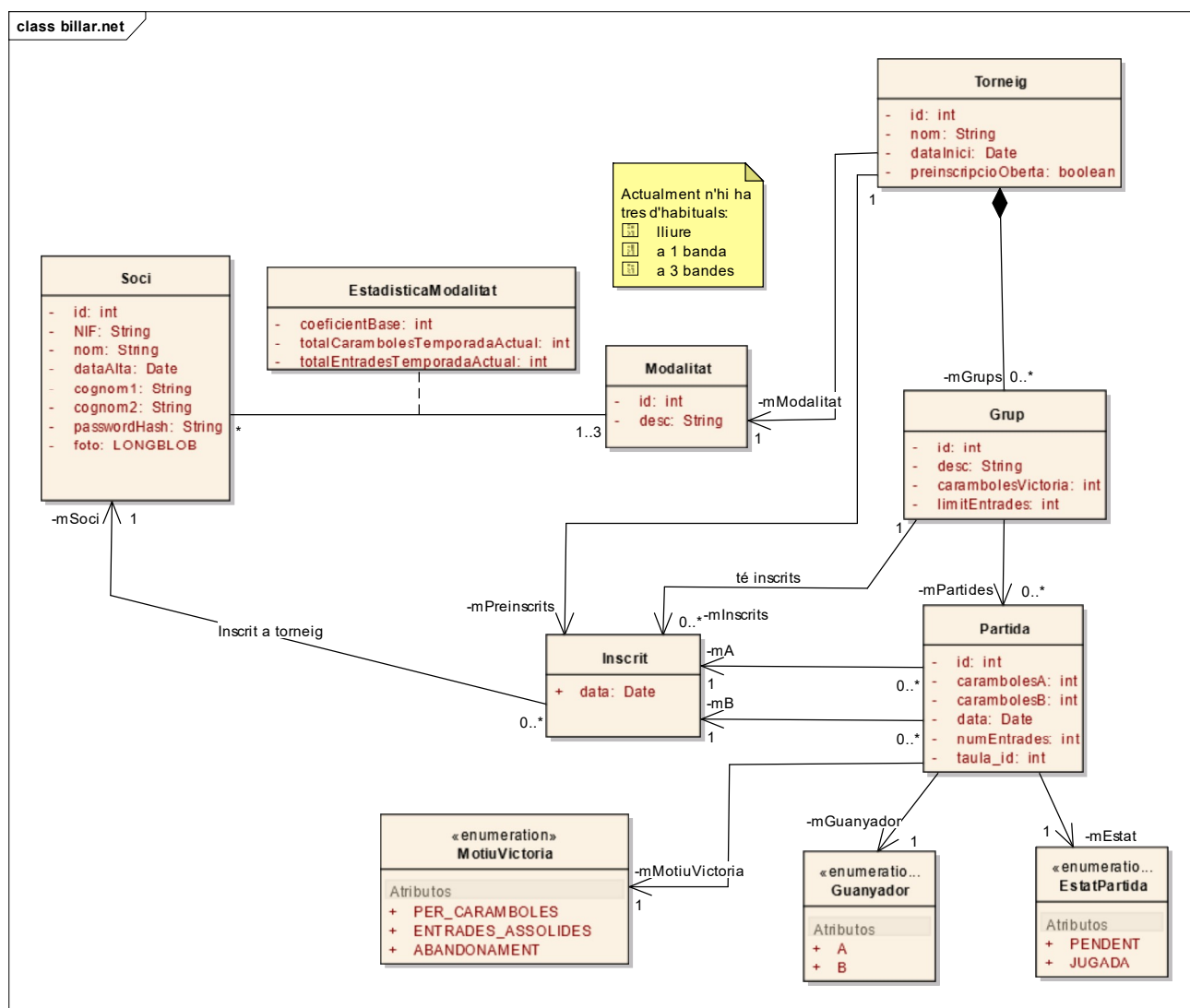
### 2.1 Components del sistema: Servidors

A excepció del client mòbil, la resta d'aplicacions utilitzaran 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.

Per tal que el client Android funcioni és necessari que pugui obtenir les dades relatives a l'usuari i als tornejos des d'un servidor que no sigui el SGBD. Aquest servidor serà una aplicació Java de consola que es comunicarà via sockets TCP/IP amb la App amb un protocol definit per vosaltres. El servidor es connecta al servidor de BD per obtenir la informació i li envia al mòbil. Així mateix, les operacions realitzades pel mòbil són gestionades pel servidor, i les dades del SGBD són actualitzades per aquest.

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 els tornejos.

## 2.2 Model genèric de classes



## 2.3 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**,

## 2.4 Persistència

L'intercanvi de dades entre les aplicacions Java ( servidor i client ), 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 component per a UN SGBD concret, en Java (pels accessos del servidor i el client) i en .NET (pels accessos des de l'aplicació d'escriptori).


La tria d'un SGBD concret queda restringida a MySQL. El component en Java ha d'utilitzar l'api JPA amb qualsevol fabricant que la suporti (Hibernate, EclipseLink,...)



**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 *update* dins l'arxiu *persistence.xml*).

## 2.5 Aplicació d'escriptori per a gestió de socis - Java

S'implementarà una aplicació d'escriptori per a fer la gestió de socis del club usant la tecnologia Swing de Java. Les funcionalitats inclouen totes les operacions d'alta, baixa i modificació de dades relacionades amb els socis.

Per donar d'alta un soci cal enregistrar totes les dades personals (la fotografia és un camp opcional – programar-lo és obligatori ) , i es poden especificar els coeficients base per a cada modalitat de billar existent.

NOTES IMPORTANTS:

- a) no es pot assumir que hi ha 3 modalitats, el nombre de modalitats pot canviar amb el temps.
- b) Les fotografies es desen a la base de dades.
- c) Per eliminar un soci caldrà verificar abans que no tingui referències.

Finalment es prepararà un **programa d'instal·lació** per la distribució de l'aplicació Java.

## 2.6 Gestió de tornejos. Aplicació d'escriptori UWP o WPF

S'implementarà una aplicació d'escriptori amb tecnologia .NET o .NET Core per gestionar els tornejos. L'aplicació ens ofereix les següents funcionalitats:

- 1) **Creació de tornejos.** La creació de tornejos consisteix en tres etapes: entrada de dades bàsiques, creació de grups i generació d'encreuaments
  - a) **Dades bàsiques:** molt simple: entrem la data d'inici, títol de la competició i modalitat. El torneig queda obert a inscripcions fins que es creen els grups (b).
  - b) **Creació de grups:** Es poden afegir tants grups com es vulgui. Per cada grup creat, l'administrador ha d'indicar:
    - i) Nom del grup. ( P.ex, "Campionat Festa Major. Categoria A" )
    - ii) Objectiu de caramboles . (p.ex 50)
    - iii) Limit d'entrades. (p.ex. 40)
    - iv) Assignar els socis inscrits al grup. Els socis es podran triar d'una llista d'inscrits, que es mostrarà ordenada segons coeficient base descendent. Els socis només poden estar assignat a UN ÚNIC grup dins d'un torneig.
  - c) **Generació d'encreuaments:** Un cop tancats els grups, el sistema genera partides per a tots els encreuaments dels jugadors dins de cada grup. Les partides inicialment no tenen data assignada i el seu estat és PENDENT. Aquest resultat no es visualitza, només es desa a la BD.
- 2) **Llista de tornejos.** Es mostren en una única llista tots els tornejos, actius i tancats. Pels tornejos actius les dades visibles són: data d'inici, total de partides programades, total de grups, i per cada grup partides jugades/total. Pels tornejos tancats només mostrem la data de finalització i el guanyador per a cada grup.
  - a) Podem filtrar per data d'inici i/o per estat del torneig ( obert / tancat )
  - b) Podem esborrar un torneig ( prèvia confirmació si té partides definides )
  - c) Podem tancar el torneig ( si s'han acabat totes les partides ).Al costat de cada torneig posarem icones per donar accés a dues funcionalitats : "classificació" (3) i "entrada de resultats"(4)





- 3) **Classificació:** Per a cada torneig podem veure les taules de classificació. Hi haurà tantes taules com grups s'hagin creat. Cada classificació de grup s'encapçalada pel codi i el nom del grup, conté una llista ordenada de millor a pitjor puntuació on es mostren els camps següents:
- i) Posició
  - ii) Nom del jugador
  - iii) Partides jugades
  - iv) Partides guanyades
  - v) Partides perdudes
  - vi) Coeficient d'entrades.
- 4) **Entrada de resultats:** En aquesta pantalla es mostra una graella amb tots els creuaments, i l'administrador podrà seleccionar qualsevol d'ells i entrar-ne el resultat indicant qui guanya, el nombre de caramboles de cada jugador, les entrades jugades i el motiu de victòria.
- 5) **Instal·lador:** Es prepararà amb WIX un **programa d'instal·lació** per la distribució del programari.

## 2.7 App Android per Socis

*Caldrà que dissenyeu les funcionalitats següents:*

1) **Splash screen**

- 2) **Autenticació:** *el login és el NIF del soci, i el password el que estigui especificat a la BD. Suposarem que algú li ha fet arribar al soci.*

IMPORTANT: El login només es fa un cop, les dades es desen localment al mòbil, i no es tornaran a demanar de nou. L'usuari podrà en tot moment tornar a la pantalla d'edició per modificar les seves dades d'accés.

3) **Activity** d'inici: *Es mostra:*

- a) *Part superior:* un resum de les estadístiques de l'usuari ( el coeficient base i *resum de caramboles/entrades del temporada actual per a cada modalitat* )
- b) *Part inferior:*
  - i) *La llista de tornejos oberts, amb un botó d'inscripció. Quan ens inscrivim (previ diàleg de confirmació), el torneig passa la llista (ii)*
  - ii) *La llista de tornejos actius als que actualment està participant.*

- 4) **Activity de perfil:** S'hi accedeix mitjançant una opció de menú. Permet modificar les dades personals de l'usuari, foto inclosa!

- 5) **Activity de torneig:** S'hi accedeix fent clic sobre un torneig. Usant una única Activity amb dos Tabs (<https://www.androidhive.info/2015/09/android-material-design-working-with-tabs/> )

- a) **TAB 1: Classificació:** *mostra la classificació del grup al que pertany el soci. És la llista ordenada de:*

- ☐ Posició
- ☐ Partides jugades
- ☐ Partides guanyades
- ☐ Partides perdudes
- ☐ Coeficient d'entrades.

L'ordre es determina per nombre de partides guanyades en primer lloc, i per coeficient en cas d'empat.

- b) **TAB 2: Partides:** *mostra la llista de partides pendents:*

- ☐ Contrincant
- ☐ Botó "Jugar"









6) **Activity** de “partida en joc”: Si en el TAB2, es prem el botó “Jugar” sobre una partida, s’inicia el mode *partida en joc*. L’App demana el número de taula on es jugarà, i identifica *als dos jugadors, indicant qui és l’A i el B*.

*El mode de joc permet anar anotant les entrades que fa cada jugador. La partida acaba quan un dels jugadors assoleix el nombre de caramboles indicat pel grup dins del torneig, o quan s’arriba al màxim nombre d’entrades estipulat. En aquest últim cas, guanya el jugador que ja fet més caramboles. Si hi ha empat, es deixa fer una entrada extra que es marcarà clarament com a “DESEMPAT”*

El funcionament d’aquesta pantalla ha de ser simple:

-  La pantalla és horitzontal, dividida en dos panells: A l’esquerra hi ha el comptador de caramboles de l’entrada actual i a la dreta la llista resum de totes les entrades anteriors.
  - (ESQUERRA: panell d’entrada) L’aplicació mostra el nom del jugador que està fent l’entrada actual, i indica si és A o B. Useu un background diferenciat que identifiqui clarament de qui és el torn. Es disposa d’un polsador per anar incrementant / decrementant el nombre de caramboles. Quan acaba de tirar prem el botó “final d’entrada”, i alternem a l’altre jugador.  
*Adicionalment hi ha dos botons més:*
    - a. **Abandonar:** botó que permet abandonar al jugador actual, donant la victòria immediata a la contrapart. Cal demanar confirmació per permetre un abandonament.
    - b. **Cancel·lar:** Cancel·lar entrada i tornar a l’anterior.
  - (DRETA: panell resum d’entrades) Es mostra a la part superior el nombre de caramboles totals de cada jugador i el número de l’entrada actual. A sota es mostra el detall de la llista d’entrades. Per cadascuna simplement indiquem el número d’entrada, hora, el jugador (A o B) i el nombre de caramboles.
-  Quan s’assoleix l’objectiu de joc (per caramboles, per entrades, o per abandonament) es mostra un missatge amb el guanyador i s’envia aquesta informació al servidor. A partir d’aquest moment, la partida ja no estarà a la llista de partides pendents i la llista de resultats cal que s’actualitzi adequadament. *També cal refrescar les estadístiques.*

**NOTA IMPORTANT:** El servidor només desarà el resultat final de la partida, els resultats parcials només són mostrats durant la partida. Tanmateix caldrà tenir algun tipus de persistència en el mòbil doncs no volem en cap cas perdre les dades si ens truquen a mitja partida !!!!!!!!!!!

## 2.8 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 log i algun botó de control ( p.ex. aturar/engegar )

L’accés a les dades s’implementarà mitjançant el component de dades adequat al SGBD escollit.

Podeu definir un protocol al vostre gust que doni sortida a les següents operacions:

### 2.8.1 Operacions del protocol de xarxa que suporta el servidor

#### Peticions client Android Servidor Java

##### Notació:

tuples	{camp1, camp2,...}
arrays	[item](cardinalitat)





Codi	Petició	Paràmetres	Retorn
1	Login	{user,password}	{session_id, tupla_dades_usuari, [tupla_estadistica_modalitat]*} o {-1} si no és correcte  tupla_estadistica_modalitat := {modalitat, coeficient base, entrades_any_actual, caramboles_any_actual}
2	ActualitzarDades	{session_id, tupla_dades_usuari}	{codi_resultat}  codi_resultat:= {-1: error   0: ok, resultat desat }
3	GetTornejosActiusOnParticipo	{session_id}	[tupla_info_torneig]*  tupla_info_torneig:= {torneig_id, grup_id, torneig_nom, grup_desc , modalitat }
4	GetTornejosObertsInscripció	{session_id}	[tupla_info_inscripcio]*  tupla_info_inscripcio:= {torneig_id, torneig_nom, modalitat, dataInici}
5	FerInscripció	{session_id, torneig_id}	{codi_resultat}  codi_resultat:= {-1: error   0: ok }
6	GetInfoTorneig	{session_id, grup_id, torneig_id }	{ [tupla_partida]* , [tupla_classificació]*}  tupla_partida := {data, {A B}, tupla_contrincant, id_taula}  tupla_contrincant:= {NIF,cognom1, cognom2, nom}  tupla_classificació:={num_ordre, cognom1, cognom2, nom, jugats,guanyats, perduts, coeficient torneig }
7	SendResultatPartida	{session_id, partida_id, entrades_A, entrades_B}	{codi_resultat}  codi_resultat:= {-1: error   0: ok, resultat desat }

## 2.9 JasperReport

De cara a obtenir versions per impressió de la informació de socis i tornejos, utilitzarem JasperReport Community del servidor del departament des de l'aplicació UWP, usant controls WebView. Concretament es demana que es pugui aconseguir:

- Les classificacions d'un torneig per cada grup ( paràmetre: torneig ) L'informe contindrà llistats diferenciats per grup, encapçalant cada grup amb la seva descripció i el nom del guanyador ( si el torneig s'ha tancat ). Així mateix, l'informe estarà encapçalat per les dades generals del torneig ( data, nom, modalitat).
- Llistat de socis amb detall de participacions. És un llistat de resum de tots els socis, on per cadascun es mostra una petita fitxa amb les seves dades personals i un quadre resum on es llisten els tornejos on ha participat i la posició on ha quedat.

Algunes indicacions a considerar:







- *Useu una capçalera amb logo i dades del club, títol adequat, dades introduïdes en els filtres*
- *Les zones de línies amb columnes han de ser clarament identificades*
- *Peu, amb alguna informació (data-hora d'elaboració, paginació,...)*

### 3 Entrega i avaluació

*El lliurament del projecte es farà el dia 1 de juny en horari comunicat al dia anterior i amb avaluació col·legiada.*

*Caldrà lliurar el projecte en format 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 d'exàmens extraordinaris.*

