

Projecte Final de Cicle

Torneig CEIP Joanot Martorell

Alumne:	Vicent Roselló Sanchis
DNI:	20.039.666-L
Tutor Individual:	Nom del tutor individual
Tutor del grup:	Nom del tutor individual

Dades del Projecte

Dades de l'alumne

Nom i cognoms	Vicent Roselló Sanchis
NIF/NIE	20039666L
Curs i CF	2n DAM

Dades del projecte

Títol del projecte	Torneig del mort
Nom del tutor individual	
Nom del tutor del grup	
Resum	<i>Aplicació multiplataforma per a gestionar els distints tornejos anuals d'un campionat esportiu que es realitza el col·legi CEIP Joanot Martorell de Xeraco.</i>
Abstract	<i>Cross-platform application to manage the different annual tournaments of a sports championship held at the CEIP Joanot Martorell school in Xeraco.</i>
Mòduls implicats	<ul style="list-style-type: none">● Accés a dades● Programació multimèdia i dispositius mòbils● Desenvolupament d'interfícies● Sistemes Informàtics
Data de presentació	xx de juny de 2024

Índex

Dades del Projecte.....	1
Dades de l'alumne	1
Dades del projecte	1
Índex.....	2
1. Introducció/Marc del projecte	3
1.1. Descripció del projecte	3
1.2. Objectius	3
1.3. Tipus de projecte	3
1.4. Orientacions per al desenvolupament i recursos	3
2. Anàlisi/Estudi de l'estat actual	4
2.1. Descripció del sistema actual	4
2.2. Viabilitat del sistema actual.....	4
2.3. Requeriments del nou sistema	4
3. Anàlisi de la solució.....	5
3.1. Anàlisi de les possibles solucions	5
3.2. Avaluació de les possibles solucions	5
3.3. Descripció de la solució escollida	5
4. Disseny de la solució.	6
4.1. Requisits.....	6
4.2. Anàlisi de riscos.....	6
4.3. Casos d'us.....	6
4.4. Diagrames	6
5. Desenvolupament de la solució.....	7
5.1. Estimació de cost temporal.....	7
5.2. Dotació de recursos.....	7
5.3. Configuració i/o desenvolupament del sistema	7
5.4. Avaluació del sistema	7
6. Implantació de la solució	8
6.1. Definir les fases d'implantació/migració	8
6.2. Inventari de maquinari i programari	8
6.3. Diagrames de xarxa, estructura,.....	8
6.4. Migració/Implantació de serveis.....	8
6.5. Formació, comunicació i suport a l'usuari	8
6.6. Calendari d'execució	8
7. Conclusions i treballs futurs	9
8. Bibliografia	10

1. Introducció/Marc del projecte

1.1. Descripció del projecte

Aquest projecte consisteix en el desenvolupament d'una aplicació per a la gestió integral d'un torneig esportiu organitzat al col·legi municipal de Xeraco, concretament al Joanot Martorell. L'objectiu principal és proporcionar una eina senzilla però poderosa que permeti als organitzadors gestionar fàcilment els equips, els jugadors i els partits, així com mantenir actualitzats els resultats i les classificacions en temps real. A més, amb la implementació de registres, seguretat i rols, la resta d'usuaris podran seguir el campionat a través de l'aplicació de manera segura, sense poder modificar cap dada, excepte per a simples consultes informatives sobre l'estat del torneig.

L'aplicació consta de quatre components principals: una base de dades desenvolupada i dissenyada amb MySQL per emmagatzemar tota la informació rellevant dels tornejos; una API desenvolupada amb Spring Boot per gestionar la lògica de negoci i la comunicació entre els components; i un client mòbil i web desenvolupat amb Flutter, tecnologia que permet programar per a diferents sistemes amb un únic codi, oferint una interfície intuïtiva i accessible per als usuaris finals. Finalment, el desplegament es realitza mitjançant Docker, creant contenidors per allotjar els diferents serveis i permetent el consum de l'aplicació de manera interna al centre, atès que, per qüestions de protecció de dades, treballarem amb la xarxa local del col·legi.

A més, per a presentar el projecte com una aplicació mòbil i demostrar el funcionament de la mateixa, s'ha realitzat un desplegament paral·lel al núvol d'Amazon AWS, utilitzant dades no oficials però vàlides per demostrar l'ús de l'aplicació.

Amb aquesta solució, esperem millorar l'eficiència en la gestió de tornejos esportius, reduint el temps necessari per a l'administració de dades i oferint una millor experiència als participants i aficionats.

1.2. Objectius

1. **Desenvolupar** una aplicació mòbil i web per a la gestió integral de tornejos esportius al col·legi Joanot Martorell.
2. **Implementar** una base de dades MySQL per emmagatzemar i gestionar tota la informació relacionada amb els equips, jugadors, partits, resultats i classificacions.
3. **Crear** una API amb Spring Boot per gestionar la lògica de negoci i facilitar la comunicació entre la base de dades i els clients mòbil i web.

4. **Assegurar** la privacitat i seguretat de les dades mitjançant la implementació de registres d'usuaris, sistemes de rols i permisos d'accés.
5. **Millorar** l'experiència dels participants i aficionats oferint una interfície intuïtiva i accessible mitjançant Flutter.
6. **Desplegar** l'aplicació utilitzant Docker per a garantir la portabilitat i la facilitat de manteniment en un entorn intern del col·legi.
7. **Permetre** als usuaris seguir en temps real el desenvolupament del torneig, oferint consultes informatives sobre l'estat dels partits i classificacions.
8. **Facilitar** la presentació del projecte desplegant una versió en el núvol a AWS per a demostrar la funcionalitat de l'aplicació en un entorn controlat.

1.3. Tipus de projecte

Tipus de Projecte

Aquest projecte és de tipus **intern**, ja que està destinat a ser utilitzat al col·legi municipal de Xeraco, Joanot Martorell. No obstant això, degut a l'escalabilitat del projecte, amb xicotets ajustos es podria extrapolar a altres centres educatius interessats en la gestió de tornejos esportius. L'objectiu és proporcionar una eina personalitzada per a la gestió del torneig esportiu organitzat per la institució.

Requeriments del projecte

- **Requeriments de programari:**
 - **Llenguatges i frameworks:**
 - El desenvolupament de l'API es realitza en **Java** utilitzant **Spring Boot** per a la lògica de negoci.
 - El client mòbil i web es desenvolupa en **Dart** utilitzant **Flutter** per permetre la programació multiplataforma.
 - La base de dades s'implementa en **MySQL** per gestionar la informació.
 - **Entorn de desenvolupament:**
 - Per al desenvolupament i creació de la base de dades, s'utilitza **MySQL Workbench**.
 - **Eclipse** es fa servir per a desenvolupar l'API, formant conjuntament amb MySQL Workbench la tecnologia backend.

- **Visual Studio Code** s'utilitza per al desenvolupament del frontend amb el framework **Flutter**.
 - Es necessita **Android Studio** per a gestionar els SDK d'Android i emular dispositius mòbils.
 - Per a realitzar proves a l'API, s'utilitza **Swagger**, una potent llibreria d' **Spring** que proporciona una interfície gràfica per realitzar peticions a l'API, evitant la necessitat d'utilitzar altres eines com Postman.
- **Requeriments d'infraestructura:**
 - **Infraestructura interna:** El projecte es desplega en el servidor local del col·legi, utilitzant **Docker** per a la creació de contenidors que allotgen els diferents serveis de l'aplicació.
 - **Xarxa:** Per protegir les dades dels usuaris, l'aplicació s'executa en una xarxa local interna del col·legi, sense exposició a internet.
 - **Infraestructura en el núvol (demostració):** Per a la presentació del projecte fora de l'entorn local, es desplega una versió en el núvol utilitzant **Amazon Web Services (AWS)**, la qual permet la demostració de les funcionalitats de l'aplicació amb dades fictícies.
 - **Requeriments de migració:** No aplicable, ja que es tracta d'una implementació nova sense dades prèvies a migrar.

1.4. Orientacions per al desenvolupament i recursos

Durant el desenvolupament d'aquest projecte, s'ha seguit una metodologia en cascada més coneguda com a waterfall, per a cada fase del projecte. Les fases s'han dividit de la següent forma: construcció de la base de dades, desenvolupament de la api, desenvolupament del client i finalment el desplegament.

Com a bones pràctiques, he fet ús de patrons de disseny per a una estructura de codi clara i mantenible, com ara MVC i la gestió d'usuaris i permisos amb estàndards de seguretat actuals.

Les eines i tecnologies utilitzades inclouen:

- **Git** per al control de versions, facilitant la col·laboració i mantenint un historial de canvis detallat.
- **Swagger** per documentar i provar l'API de manera eficaç.
- **Docker** per crear entorns de desenvolupament i producció consistents i aïllats.

Principal recursos:

- Tot el temari implantat durant el curs

Recursos web:

- **Documentació oficial de Spring Boot:** Ha estat la guia principal per a la configuració i desenvolupament de l'API.
- **Flutter.dev:** La documentació oficial de Flutter, usada per desenvolupar la interfície d'usuari tant per a la web com per a dispositius mòbils.
- **Stack Overflow i fòrums de desenvolupadors:** Han estat recursos valuosos per resoldre dubtes específics durant el desenvolupament.
- “Api Rest con Spring Boot” de l'acadèmia Openwebinars, Un curs que ha estat essencial per dominar les bases del framework Spring.

2. Anàlisi/Estudi de l'estat actual

2.1. Descripció del sistema actual

Actualment, la gestió dels tornejos esportius al col·legi Joanot Martorell de Xeraco es realitza de manera manual utilitzant **documents manuscrits i documents impresos**. Els organitzadors dels tornejos són responsables d'inscriure manualment els equips, planificar els partits i actualitzar els resultats i classificacions a mà. Aquesta metodologia presenta diverses limitacions:

1. **Manca d'eficiència:** La gestió manual consumeix molt de temps, especialment quan s'han d'actualitzar els resultats de manera contínua. Això dificulta el treball dels organitzadors, que sovint han de revisar i corregir errors manualment.
2. **Errors i duplicitats:** La introducció manual de dades augmenta el risc d'errors, com ara l'entrada incorrecta de resultats, equips duplicats o la pèrdua d'informació. Això pot afectar la transparència i precisió del torneig.
3. **Manca de seguretat:** L'actual sistema no compta amb cap mesura de seguretat per controlar qui pot accedir i modificar les dades dels tornejos. Això comporta riscos de modificació no autoritzada i pèrdua de confiança en la gestió del torneig.

A causa d'aquestes limitacions, es fa necessària la implementació d'una **solució digital centralitzada** que automatitze aquests processos, garantint la seguretat de les dades i millorar la comunicació amb els participants i aficionats. La nova aplicació permetrà gestionar fàcilment equips, jugadors i resultats, i oferirà una plataforma intuïtiva per seguir el desenvolupament del torneig en temps real.

2.2. Viabilitat del sistema actual

Després de l'anàlisi del sistema actual, s'ha determinat que aquest no és viable per a complir amb els objectius plantejats. Tot i que la gestió manual mitjançant fulls manuscrits permet dur a terme els tornejos, presenta nombroses limitacions que afecten l'eficiència i precisió del procés:

- **Escalabilitat limitada:** A mesura que el nombre d'equips i partits creix, la gestió manual esdevé cada cop més complicada i consumeix molt de temps. Això fa que el sistema no siga escalable.
- **Alta probabilitat d'errors humans:** La dependència de la introducció manual de dades incrementa el risc d'errors, com ara resultats incorrectes o duplicitats, cosa que afecta la confiança en la gestió del torneig.
- **Manca de seguretat:** No hi ha un sistema robust per gestionar l'accés a les dades del torneig, cosa que pot provocar modificacions no autoritzades i posar en risc la integritat de la informació.

Degut a aquestes limitacions, el sistema actual no és una solució viable per a una gestió eficient i escalable dels tornejos. Per tant, es justifica la necessitat de desenvolupar un nou sistema que solucione aquestes deficiències i millore l'experiència dels organitzadors i usuaris finals.

2.3. Requeriments del nou sistema

Per a superar les limitacions del sistema actual i assolir els objectius plantejats, el nou sistema ha de complir amb els següents requeriments:

Requeriments funcionals:

1. **Gestió d'equips i jugadors:** Capacitat per registrar, modificar i eliminar equips i jugadors, incloent-hi informació detallada de cada participant.
2. **Realització automàtica de partits:** Sistema per realitzar els enfrontaments i crear així els partits de manera automàtica simulant el sorteig per a les diferents fases.
3. **Actualització en temps real dels resultats:** Possibilitat d'actualitzar els resultats dels partits en temps real, amb informació immediata per als usuaris.
4. **Accés segur amb rols:** Implementació de rols d'usuari per garantir que només els organitzadors puguin modificar dades, mentre que altres usuaris poden consultar-les.
5. **Interfície d'usuari intuïtiva:** Desenvolupament d'una interfície intuïtiva i accessible per facilitar la navegació als usuaris finals.

Requeriments no funcionals:

1. **Seguretat:** Els usuaris han de poder accedir al sistema de manera segura, amb autenticació i control d'accés robust.
2. **Rendiment:** El sistema ha de poder gestionar múltiples partits i resultats sense desacceleració, fins i tot amb un nombre elevat d'equips.
3. **Portabilitat:** Ha de ser fàcilment desplegable tant en entorns locals com en el núvol, amb configuracions mínimes.
4. **Manteniment i escalabilitat:** El codi ha de ser modular per facilitar el manteniment i la incorporació de noves funcionalitats en el futur.

3. Anàlisi de la solució

3.1. Anàlisi de les possibles solucions

Durant la fase d'anàlisi, es van considerar tres solucions diferents per al desenvolupament del nou sistema de gestió de tornejos esportius:

Solució 1: Desenvolupament des de zero amb Spring Boot i Flutter

- **Característiques:** Crear una aplicació completament nova utilitzant Spring Boot per al backend i Flutter per al frontend mòbil i web. Això permet desenvolupar una solució totalment personalitzada que s'adapta a les necessitats específiques del torneig.
- **Avantatges:** Flexibilitat total en el disseny i funcionalitat; interfícies adaptades tant per a mòbils com per a web; aïllament privat de les dades utilitzades a una base de dades pròpia.
- **Limitacions:** Temps de desenvolupament més llarg.

Solució 2: Adaptació d'un sistema de codi obert

- **Característiques:** Utilitzar un sistema existent de codi obert per a la gestió d'esdeveniments esportius i adaptar-lo per satisfer els requeriments del torneig.
- **Avantatges:** Temps de desenvolupament més curt i amb solucions ja provades i testades.
- **Limitacions:** Menor flexibilitat per afegir noves funcionalitats personalitzades; possibilitat de complicacions per adaptar el codi existent a les necessitats específiques. Dades a un servidor extern.

Solució 3: Plataforma basada en serveis al núvol (SaaS)

- **Característiques:** Contractar una plataforma ja existent de gestió de tornejos que funcione com un servei al núvol (SaaS), com ara LeagueApps o TournamentSoftware.
- **Avantatges:** Desplegament ràpid; manteniment i actualitzacions gestionades pel proveïdor; baix cost inicial.
- **Limitacions:** Limitacions en la personalització; costos recurrents a llarg termini; dependència del proveïdor per a funcionalitats específiques.

3.2. Avaluació de les possibles solucions

Després d'avaluar les tres solucions considerades, es van definir els següents criteris per a la seva comparació:

1. Cost inicial i de manteniment
2. Flexibilitat i personalització
3. Temps de desenvolupament
4. Escalabilitat
5. Facilitat d'integració i manteniment

Comparativa basada en els criteris:

Solució	Cost Inicial	Flexibilitat	Temps de Desenvolupament	Escala bilitat	Facilitat de Manteniment
Desenvolupament des de zero	Baix	Alt	Llarg	Alt	Alt
Adaptació de codi obert	Baix	Mitjà	Curt	Mitjà	Mitjà
Plataforma SaaS	Baix	Baix	Molt curt	Mitjà	Alt

Justificació: Tot i que la plataforma SaaS té un temps de desplegament més curt i costos inicials més baixos, la seva limitació en la flexibilitat fa que no siga l'opció adequada per adaptar-se a les necessitats específiques del col·legi Joanot Martorell. La solució de desenvolupament des de zero amb Spring Boot i Flutter ofereix la millor combinació de flexibilitat, escalabilitat i facilitat de manteniment, justificant la inversió de temps de desenvolupament més llarg.

3.3. Descripció de la solució escollida

La solució escollida és el desenvolupament des de zero d'una aplicació multiplataforma utilitzant **Spring Boot** per al backend i **Flutter** per al frontend. Aquesta arquitectura permet crear una interfície d'usuari intuïtiva i coherent tant per a dispositius mòbils com per a web, utilitzant un únic codi base per a la major part del desenvolupament.

El backend gestionarà la lògica de negoci, incloent la creació i gestió d'equips, la planificació de partits i l'actualització de resultats en temps real. Així com la gestió de seguretat amb la implementació d'usuaris i contrasenyes, diferenciant el rol d'aquest per al tractament de les dades. **Docker** s'utilitzarà per crear entorns de desenvolupament i producció consistents, i **MySQL** s'encarregarà de la gestió de dades de manera segura i escalable.

Amb aquesta solució, s'aconsegueix la flexibilitat necessària per adaptar-se a les necessitats específiques del col·legi Joanot Martorell, allotjant al mateix centre tota l'aplicació sense haver de dependre de serveis externs garantint la possibilitat futura d'implementació de millores.

4. Disseny de la solució.

4.1. Requisits.

Una vegada sabem el que anem a fer, hem de acotar el que anem a fer, per tant s'han de definir les tasques que realitzarà el programari i com les farà.

Un cop sabem el que volem fer, anem a definir les tasques que el nostre programari ha de fer. La api a desenvolupar serà l'encarregada de gestionar tota la lògica del programari i per tant la que implementarà les següents tasques:

Gestió d'usuaris:

- a. Permetrà la creació d'usuaris amb diferents rols (administrador, usuari).
- b. Autenticació segura mitjançant nom d'usuari i contrasenya.
- c. Assignació de permisos en funció del rol d'usuari.

Gestió d'equips i jugadors:

- d. Registrar equips i jugadors, amb la seva informació detallada.
- e. Modificar i eliminar dades dels equips i jugadors existents.

Planificació i gestió de partits:

- f. Sortejar automàticament les diferents fases del torneig com puga ser la fase de grups i fase eliminatòria.
- g. Fer de forma automàtica els enfrontaments dels partits.
- h. Actualitzar els resultats dels partits.

Visualització i consulta d'informació:

- i. Permetre als usuaris consultar la informació dels jugadors, equips, partits i classificacions.

I com a tasques no funcionals, hem de tindre en compte els següents criteris:

Seguretat:

- a. Al tractar-se de dades sensibles com noms d'alumnes, l'aplicació s'allotjarà totalment al centre educatiu de forma local.
- b. No obstant per a divulgar el funcionament amb una demo, és crearà en algun servei de núvol per poder fer us amb dades fictícies del programari.
- c. Gestió robusta d'autenticació i permisos d'accés.

Usabilitat:

- d. La interfície d'usuari ha de ser intuïtiva i fàcil d'utilitzar, amb una corba d'aprenentatge mínima.
- e. Els usuaris han de poder accedir a totes les funcionalitats clau per poder obtenir la informació necessària per seguir el campionat.

Portabilitat:

- f. El sistema ha de ser accessible des de dispositius mòbils i ordinadors de sobretaula, utilitzant navegadors actualitzats i una aplicació nativa en Android/iOS.

Mantenibilitat i escalabilitat:

- g. L'arquitectura ha de permetre una fàcil extensió per afegir noves funcionalitats en el futur.
- h. S'ha de poder escalar per gestionar més usuaris o tornejos amb mínims ajustos tècnics.

4.2. Anàlisi de riscos.

Durant el desenvolupament i implementació del sistema de gestió de tornejos esportius, s'han identificat els següents riscos potencials:

Riscos Tècnics

1. Errors en la integració entre API i frontend.

- a. **Probabilitat:** Mitjana
- b. **Impacte:** Alt
- c. **Mesures preventives:** Realitzar proves freqüents d'integració des del principi del desenvolupament. Utilitzar eines com Postman o Swagger per verificar les respostes de l'API.
- d. **Mesures correctives:** Si es detecten errors, implementar un sistema de logs detallats per identificar i corregir els problemes ràpidament.

2. Problemes de rendiment sota càrregues altes.

- a. **Probabilitat:** Baixa
- b. **Impacte:** Alt
- c. **Mesures preventives:** Realitzar tests de càrrega i rendiment utilitzant eines com JMeter. Optimitzar consultes SQL i assegurar una configuració adequada del servidor.
- d. **Mesures correctives:** Escalar l'arquitectura mitjançant contenidors addicionals amb Docker en moments de càrrega alta.

Riscos Organitzatius

3. Retards en el desenvolupament per falta de temps o recursos.

- Probabilitat:** Mitjana
- Impacte:** Mitjà
- Mesures preventives:** Planificar amb detall cada sprint, establir prioritats clares i reservar temps per a tasques crítiques.
- Mesures correctives:** Redistribuir tasques o simplificar funcionalitats menys prioritàries.

Riscos Legals i de Seguretat

5. Accés no autoritzat a dades sensibles.

- Probabilitat:** Baixa
- Impacte:** Alt
- Mesures preventives:** Implementar autenticació robusta amb JWT i encriptació de dades sensibles.
- Mesures correctives:** Si es detecta un accés no autoritzat, revocar immediatament les credencials afectades i auditar l'activitat recent del sistema.

Risc	Probabilitat	Impacte	Prioritat
Errors en la integració entre API i frontend	Mitjana	Alt	Alta
Problemes de rendiment	Baixa	Alt	Mitjana
Retards en el desenvolupament	Mitjana	Mitjà	Mitjana
Accés no autoritzat a dades sensibles	Baixa	Alt	Alta

4.3. Casos d'ús.

Per detallar els casos d'ús o més coneguts com UML identificarem els diferents actors externs que interactuaran amb el sistema d'informació, i a través de quines funcionalitats es relacionaran amb ell.

Actors

- Administrador:** Responsable de gestionar els usuaris, equips i partits.
- Alumne:** Pot consultar la informació del torneig en temps real.
- Visitant anònim:** Pot accedir a la pàgina principal de benvinguda on es presenta el torneig i les seves regles, però no pot accedir a cap informació més.

Casos d'ús

CU-01: Gestió d'usuaris

- **Descripció:** L'administrador pot gestionar completament els usuaris, incloent-hi la creació, modificació, eliminació, i l'opció de marcar jugadors sancionats.
- **Actor:** Administrador.
- **Flux d'esdeveniments:**
 1. L'administrador inicia sessió amb les seves credencials.
 2. Accedeix a l'apartat "Jugadors".
 3. Té les opcions següents:
 - Crear un nou jugador amb dades com nom, edat, curs i equip.
 - Buscar jugadors utilitzant l'opció de cerca.
 - Editar les dades d'un jugador seleccionant-lo de la llista.
 - Marcar un jugador com a sancionat.
 4. Guarda els canvis realitzats.
- **Precondicions:** L'administrador ha d'haver iniciat sessió.
- **Resultat esperat:** El jugador queda registrat o modificat, i es mostra un missatge de confirmació.

CU-02: Crear una nova temporada

- **Descripció:** L'administrador pot registrar una nova temporada relativa al torneig d'eixe curs escolar.
- **Actor:** Administrador
- **Flux d'esdeveniments:**
 1. L'administrador es registra amb les seues credencials.
 2. Posa sobre el botó "+" i afegeix en nom de la nova temporada de la que penjarà tot el Torneig d'un curs escolar.
 3. Fa clic a "Crear" per registrar la Temporada.
- **Precondicions:** L'administrador ha d'haver iniciat sessió.
- **Resultat esperat:** La Temporada queda registrada per poder començar a gestionar el Torneig corresponent al mateix curs escolar anual.

CU-03: Crear un nou equip

- **Descripció:** L'administrador pot registrar un nou equip amb la seva informació.
- **Actor:** Administrador
- **Flux d'esdeveniments:**
 1. L'administrador es registra amb les seues credencials.
 2. Accedeix a la temporada corresponent.
 3. Selecciona l'opció "Crear Equip" des de la barra de navegació.
 4. Introdueix el nom de l'equip, els jugadors i el curs al que van.
 5. Fa clic a "Guardar" per registrar l'equip.
- **Precondicions:** L'administrador ha d'haver iniciat sessió.
- **Resultat esperat:** L'equip es registra correctament i queda disponible per a la planificació de partits.

CU-04: Creació de grups i partits

- **Descripció:** L'administrador organitza la fase de grups mitjançant un sorteig i crea automàticament els partits de la fase inicial.
- **Actor:** Administrador.
- **Flux d'esdeveniments:**
 1. L'administrador inicia sessió amb les seves credencials.
 2. Accedeix a la temporada corresponent.
 3. Navega a l'apartat "Classificació" i fa clic a "Sortejar grups".
 4. Els equips es distribueixen aleatòriament en 4 grups.
 5. Navega a l'apartat "Partits".
 6. Fa clic a "Crear partits" per generar automàticament els enfrontaments entre els equips.
 7. Per a les següents fases, crea manualment els partits.
- **Precondicions:**
 - Han d'haver-se registrat com a mínim 16 equips.
 - El nombre total d'equips ha de ser múltiple de 4.
- **Resultat esperat:** Els grups i partits es creen correctament, i es mostra una notificació de confirmació.

CU-05: Consultar resultats dels partits

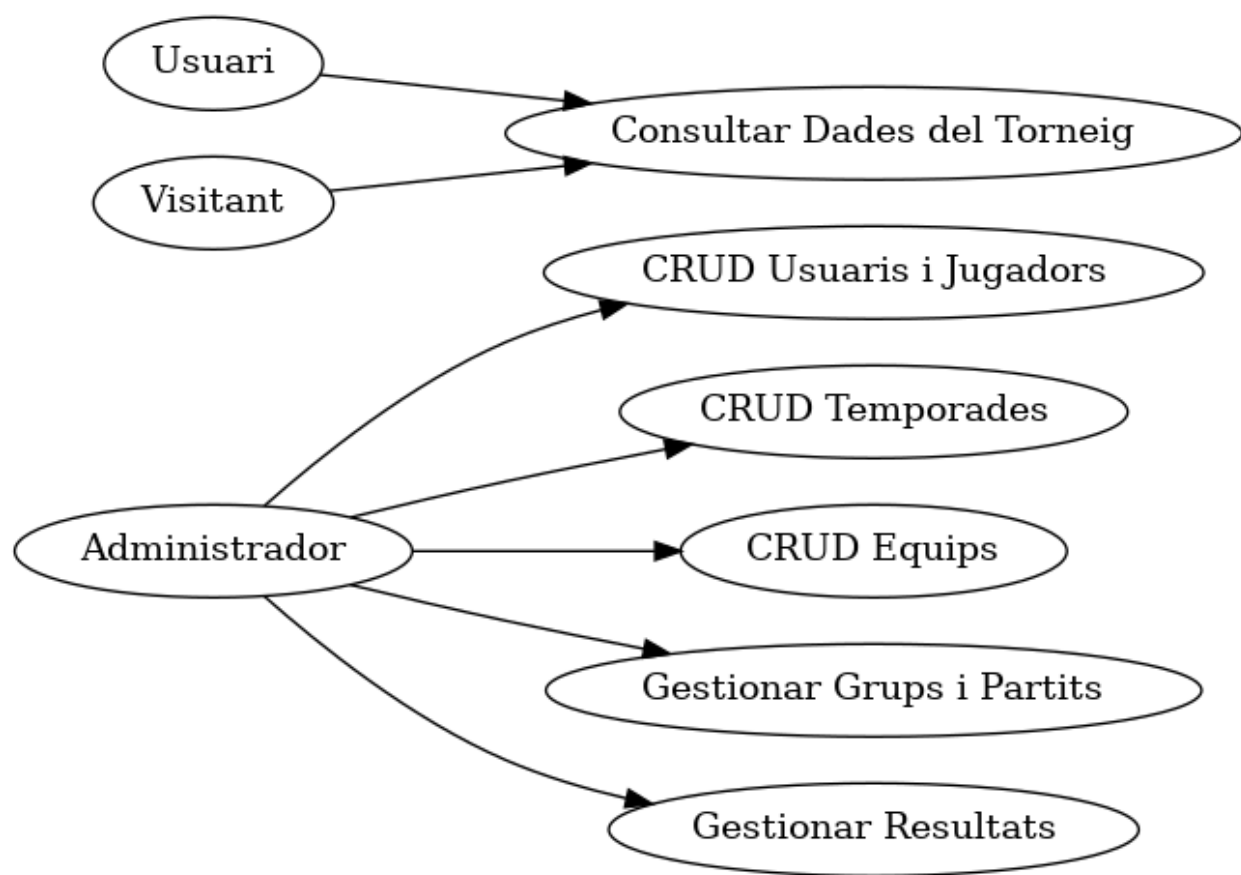
- **Descripció:** Qualsevol usuari amb credencials pot consultar qualsevol dada del Torneig. Partits, classificacions, grups, equips, jugadors, etc.
- **Actor:** Alumne
- **Flux d'esdeveniments:**
 1. L'usuari accedeix a la pàgina principal.
 2. Fa login
 3. Navega per tota l'aplicació consultant qualsevol informació.
- **Precondicions:** Ser membre del col·legi.
- **Resultat esperat:** L'usuari pot veure els resultats, classificacions, partits, equips, etc, correctament.

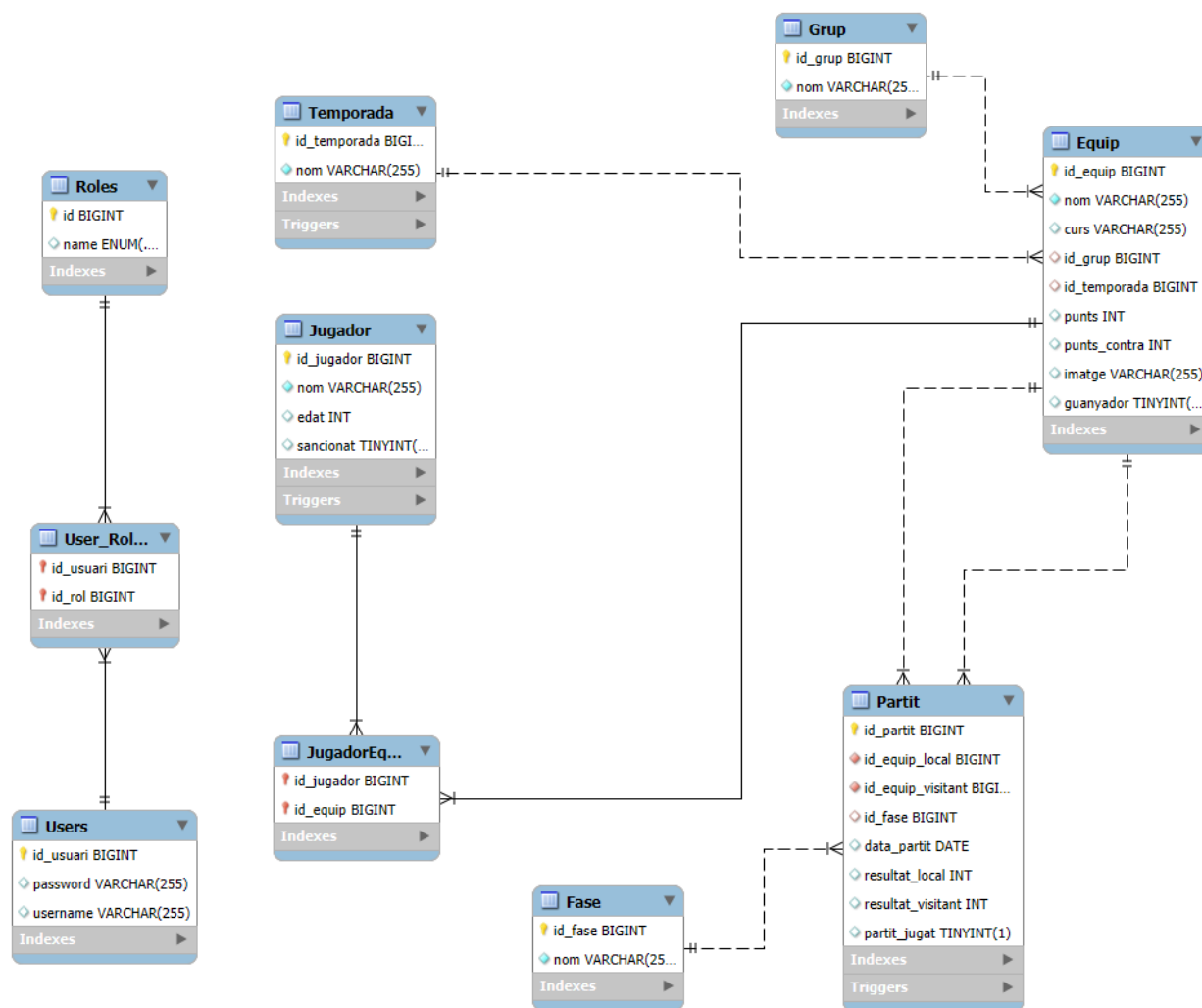
CU-06: Actualitzar resultats d'un partit

- **Descripció:** L'administrador introdueix els resultats d'un partit acabat i el sistema recalcula automàticament la classificació.
- **Actor:** Administrador.
- **Flux d'esdeveniments:**
 1. L'administrador inicia sessió amb les seves credencials.
 2. Navega a l'apartat "Partits" i selecciona el partit corresponent.
 3. Introdueix el resultat final (ex. 3-1).
 4. Fa clic a "Guardar".
- **Precondicions:** El partit ha d'haver estat prèviament programat.

- **Resultat esperat:** El resultat s'actualitza al sistema, i la classificació es recalcula automàticament. Es mostra un missatge de confirmació a l'administrador.

4.4. Diagrames





5. Desenvolupament de la solució

Ara ens centrem ja en la part grossa del projecte, el desenvolupament de la solució. Una vegada tenim clar el què volem, anem a detallar les diferents fases i què anem fent en cadascuna d'elles.

5.1. Estimació de cost temporal

Fase	Tasques	Temps Estimat
Anàlisi i Disseny	Disseny del sistema, ERD i casos d'ús	1 setmana
Base de Dades	Creació i configuració de MySQL	2 setmana
Desenvolupament API	Implementació del backend amb Spring	3 setmanes
Desenvolupament Front	Disseny i programació amb Flutter	4 setmanes
Proves i Optimització	Correcció d'errors	2 setmanes
Desplegament	Configuració de Docker i AWS	1 setmana

Durada total estimada: 13 setmanes.

5.2. Dotació de recursos

Els Recursos emprats han sigut a nivell humà 1 desenvolupador. A nivells tècnics he utilitzat un ordinador amb les eines de desenvolupament com:

- Workbench per a la base de dades.
- SpringToolSuite4 per a desenvolupar la API amb llenguatge Java.
- Visual Studio Code i Android Studio per a la part Front-end amb Flutter.
- Docker i AWS per el desplegament.
- Swagger i Postman per a fer les diverses proves de la API.

5.3. Configuració i/o desenvolupament del sistema

La part central del projecte és aquesta, on establireu com s'ha desenvolupat i configurat el nou sistema: automatització de processos, scripts, empaquetat, configuracions, etc.

5.4. Avaluació del sistema

Una vegada finalitzat el desenvolupament, es realitzarà una avaluació per veure si compleix amb els requisits que ens havíem plantejat i millora el sistema anterior.

6. Implantació de la solució

Una vegada tenim la solució, passem a implantar aquesta en l'organització o entitat per a la què s'haja realitzat. En cas que es tracte d'un projecte propi, caldria indicar com podrien posar-lo en marxa en qualsevol organització, mitjançant guies d'instal·lació i manuals d'usuari.

6.1. Definir les fases d'implantació/migració

Indicarem en quines fases es dividirà la implantació del sistema nou. Generalment, en organitzacions mitjanes i grans, no s'implanta tot el sistema d'una, sinò que es fa per departaments o seccions, de manera que cal planificar aquesta migració.

6.2. Inventari de maquinari i programari

Indicarem el maquinari i programari de què disposarem.

6.3. Diagrames de xarxa, estructura,

Com queda la xarxa estructurada després de la posta en funcionament de la solució.

6.4. Migració/Implantació de serveis

Indicarem els passos per tal de realitzar la migració o implantació dels diferents serveis.

6.5. Formació, comunicació i suport a l'usuari

Caldrà pensar en la formació que es dona als usuaris i el suport que aquests tindran.

6.6. Calendari d'execució

Finalment, caldrà indicar quin serà el calendari d'implantació de les diferents fases de la solució.

7. Conclusions i treballs futurs

Com a últim apartat, deixarem un apartat per indicar, a mode de resum què s'ha fet en el projecte, així com possibilitats de millora o futures ampliacions que aquest podrà tindre.

8. Bibliografia