



Treball Fi de Màster

Desenvolupament d'aplicacions WEB

Memòria del projecte

Estudiant: Gabriel Estela Hernández
Màster Universitari d'Enginyeria Informàtica

Consultor: Ignasi Lorente Puchades

ÍNDIX

1. Definició del projecte	4
1.1. Idea del projecte	4
1.2. Objectius	4
1.3. Anàlisi de riscos	5
2. Metodologia del projecte	6
2.1. MEAN	6
2.1.1. MongoDB	7
2.1.2. ExpressJS	7
2.1.3. AngularJS	7
2.1.4. NodeJS	8
3. Planificació del projecte	9
3.1. Calendari de planificació	9
3.2. Diagrama de Gantt	9
3.3. Planificació d'entrega	10
3.3.1. PAC1 - Planificació de treball	10
3.3.2. PAC2 - Inici de desenvolupament i documentació	10
3.3.3. PAC3 - Seguiment de desenvolupament i documentació	10
3.3.4. Lliurament final	10
4. Diagrames del projecte	11
4.1. Diagrama de casos d'ús	11
4.2. Diagrama de seqüència	13
5. Pantalles del projecte	14
6. Instal·lació de l'Stack MEAN	21
7. Diagrama de col·leccions de base de dades NoSQL	27
8. Model-Vista-Controlador (MVC)	28
8.1. Model	30
8.2. Controlador (Servidor)	31
8.3. Controlador (Client)	32
8.4. Vista	34
9. Creació de mòduls CRUD	36
9.1. Mòdul Users	36
9.1.1. Roles	39
9.2. Mòdul Pista	39
9.3. Mòdul Reserva	41

10. Proves Unitàries	44
11. Futur del Projecte	48
12. Conclusions del projecte	49

1. DEFINICIÓ DEL PROJECTE

Aquest treball final de màster consisteix en posar en pràctica els coneixements adquirits en les diferents assignatures per a dur a terme un projecte real que podria implementar-se directament a una empresa que demana una aplicació web.

1.1. IDEA DE PROJECTE

A les instal·lacions esportives tenim un nombre determinat de pistes de pàdel, tenis i futbol, on els usuaris fan el llogament de les pistes per un dia i hora determinat. Les empreses tenen la necessitat de gestionar totes aquestes peticions i reserves d'una forma més còmoda i àgil, tant per l'usuari que lloga, com per l'empresari que vol tenir un control de tots els llogaments.

En aquest projecte volem donar una solució a aquest “problema” realitzant una aplicació web on podem realitzar la gestió de reserves d'un àrea esportiva, on els usuaris podran llogar via web les pistes.

L'aplicació tindrà dues parts diferenciades, depenen del tipus d'usuari que accedeixi a la mateixa:

- *Àrea Administrador*: els empleats de les instal·lacions tindran usuari amb accés a aquesta àrea, on podran revisar les reserves realitzades pels clients, confirmar aquestes reserves, indicar si s'han pagat i mantenir un control dels clients registrats a l'aplicació.
- *Àrea Client*: Els usuaris que vulguin llogar una pista tindran un usuari de tipus *client* per realitzar la reserva. Aquest perfil està més limitat, ja que només tindran accés a la gestió del seu perfil, com les dades personals i l'historial de reserves realitzades.

1.2. OBJECTIUS

Els nostres objectius comencen amb una proposta de treball pel desenvolupament de l'aplicació, on definirem el temps estimat amb els possibles errors que puguin sorgir, donant així la informació que creiem que seria la més real possible per aconseguir l'objectiu final, que seria el correcte funcionament de l'aplicació en el temps determinat.

Aquesta aplicació web ha de poder aplicar-se en unes instal·lacions esportives reals, per tant, hem de procurar realitzar una feina lo més eficient possible per donar les facilitats alhora de realitzar consultes de reserves o consultes de dies de llogaments.

El nostre objectiu com estudiant, apart d'aplicar els coneixements comentats abans, és el d'aconseguir realitzar amb èxit un projecte real, amb les dificultats que podem trobar a un lloc de feina com a enginyer i aconseguir solucionar els problemes que ens trobem en un projecte d'aquesta magnitud.

1.3. ANÀLISI DE RISCOS

En la creació d'aquesta aplicació web hem de tenir en compte que farem servir una base de dades per guardar totes les dades, tant dels usuaris com del reserves realitzades. En aquest cas ens podem trobar diferents riscos, que els podem classificar de riscos lleus, greus o molt greus.

Ens podem trobar amb una pèrdua de dades a nivell de taula o a nivell de tota la base de dades, o més greu, una pèrdua total de l'aplicació o de la base de dades per motius diversos. Davant aquest cas hem de prendre mesures, fins i tot abans de començar el disseny i la creació de l'aplicació.

Conjuntament amb la creació de l'aplicació, anirem realitzant diferents còpies de seguretat de la mateixa, aprofitarem diferents discs durs per la realització de còpies, en cas de que el disc dur principal que conté l'aplicació tingui un error que ens sigui impossible accedir, tindrem una còpia en un disc dur totalment diferent al principal, on podrem recuperar l'estructura completa de l'aplicació.

Per obtenir un major grau de seguretat, podem optar també per realitzar una altra còpia en un disc dur extern, connectat per USB, aquesta segona còpia opcional ens pot donar major seguretat en la pèrdua de dades, ja que amb una còpia a la mateixa màquina amb diferents discs no ens dona tota la seguretat que voldríem, ja que hi ha riscos importants en mantenir les dades en la mateixa màquina.

2. METODOLOGIA DEL PROJECTE

Per dur a terme aquest projecte he investigat diferents llenguatges i mètodes de programació web amb els que aconseguir un nivell de qualitat òptim per a que la aplicació funcioni correctament, amb unes ràpides connexions a base de dades i temps mínims de resposta.

Les dues propostes per realitzar aquest projecte eren les següents:

- **LAMP:** (Linux, Apache, MySQL, PHP) era una de les més usades a nivell web per realitzar aplicacions, on tenim *Symphony2* framework de PHP, és bastant potent per la realització d'aquestes tasques. La base de dades és MySQL, i Apache és un servidor HTTP.
- **MEAN:** (MongoDB, ExpressJS, AngularJS, NodeJS) Relativament nova per la realització d'aplicacions web, aquest "stack" fa servir una base de dades "NoSQL" com és MongoDB, i un frontend JavaScript, AngularJS.

Finalment, la metodologia que hem escollit per aquest projecte ha estat **MEAN**, més moderna que l'anterior i, per la investigació que he pogut fer, amb més possibilitats i facilitats per dur a terme el projecte.

2.1. MEAN

MEAN és un grup de tecnologies lliures en Javascript per construir llocs i aplicacions web. És una combinació de MongoDB , Express.js i Angular.js , tot això executant dins Node.js.

Els diferents components se classifiquen de la següent manera:

- Components del servidor
 - o Base de dades (MongoDB)
 - o Servidor/API (Node i Express)
- Components del client
 - o Frontend (Angular)

Aquest és un resum gràfic del funcionament de MEAN:



Des del frontend, amb Angular fem cridades AJAX a la nostra API al servidor Node. Aquest consulta a la base de dades (Mongo) depenent de la trucada realitzada. La BD retorna l'objecte com a resposta a Node i aquest ho serveix com JSON a Angular que ho mostra en el frontal sense necessitat de recarregar la pàgina, creant així una *Single Page Application*.

2.1.1. MongoDB

Mongo és una base de dades no relacional (NoSQL) de codi obert que guarda les dades en documents tipus JSON (JavaScript Object Notation) però en forma binària (BSON) per fer la integració d'una manera més ràpida. Es poden executar operacions en JavaScript en la seva consola en lloc de consultes SQL. A més té una gran integració amb NODE.JS amb els driver propi i amb *Mongoose*. A causa de la seva flexibilitat és molt escalable i ajuda al desenvolupament àgil de projectes web.

2.1.2. ExpressJS

Express és un framework per sobre de NODE.JS que permet crear servidors web i rebre peticions HTTP d'una manera senzilla, el que permet també crear APIs REST de forma ràpida.

2.1.3. AngularJS

Angular és un framework JS per la part client o Frontend d'una aplicació web, que respecta el paradigma MVC i permet crear Single - Page Applications (Aplicacions web que no necessiten recarregar la pàgina), de manera més o menys senzilla. És un projecte mantingut per Google.

2.1.4. NodeJS

Node És un entorn de programació en JavaScript per al Backend basat en el motor V8 de JavaScript del navegador Google Chrome i orientat a esdeveniments, no bloquejant, que el fa molt ràpid a l'hora de crear servidors web i emprar temps real. Va ser creat en 2009 i tot i que encara és jove, les últimes versions el fan més robust a més de la gran comunitat de desenvolupadors que posseeix.

3. PLANIFICACIÓ DEL PROJECTE

En aquest apartat estudiarem el temps estimat per la realització del projecte i les diferents entregues previstes durant el període de desenvolupament.

3.1. CALENDARI DE PLANIFICACIÓ

Recordem les dates importants del projecte, com ara les dates d'entrega de les diferents pacs del projecte:

Part del projecte	Data de lliurament
Pla de treball (PAC1)	08/03/2016
PAC2	06/04/2016
PAC3	08/05/2016
Entrega Final	13/06/2016

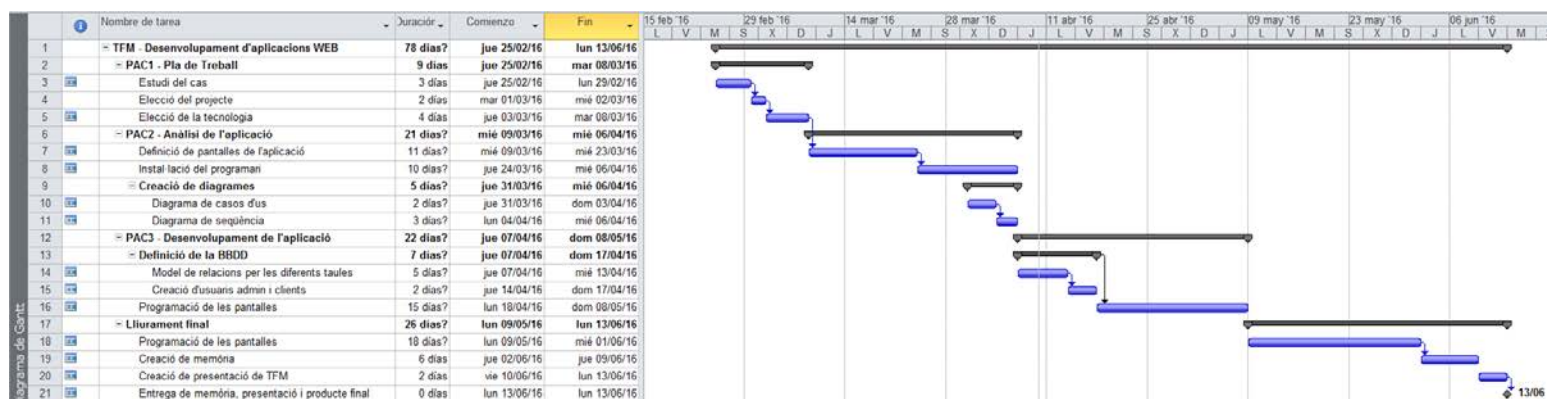
L'inici del projecte és el dia 24/02/2016, per tant, tenim 15 setmanes per realitzar el projecte. La mitjana d'hores setmanals dedicades al projecte estaran compreses entre les 12 y les 14 hores setmanals, aquesta xifra pot variar depenen de les necessitats del projecte.

Partint d'aquestes xifres, podem arribar a la conclusió que dedicarem al voltant d'unes 200 hores a la realització del projecte.

3.2. DIAGRAMA DE GANTT

Hem utilitzat l'aplicació "Microsoft Project 2010" per realitzar aquest gràfic orientatiu de les tasques generals que realitzarem en tot el projecte i el temps que ens pot dur per la seva realització, fent una estimació pensant en els diferents imprevistos que puguin anar sorgint, com el més recent que seria els problemes que hem tingut per la instal·lació de tot l'stack MEAN i la seva execució exitosa que ens ha llevat bastant de dies de desenvolupament.

La proposta de planificació del projecte seria la següent:



3.3. PLANIFICACIÓ D'ENTREGA

El Treball Final de Màster està dividit en quatre parts, les entregues que se realitzaran en cada part són les següents:

3.3.1. PAC1 – Planificació de Treball

- Document amb les previsions del projecte

3.3.2. PAC2 – Inici de desenvolupament i documentació

- Inici de la memòria del projecte.
- Documentació dels avenços en el desenvolupament.
- Codi font de la part desenvolupada.

3.3.3. PAC3 – Seguiment de desenvolupament i documentació

- Memòria del projecte en el punt actual al lliurament.
- Documentació dels avenços en el desenvolupament.
- Codi font de la part desenvolupada al lliurament.

3.3.4. Lliurament final

- Memòria del TFM.
- Presentació del TFM.
- Codi font del projecte final.

4. DIAGRAMES DEL PROJECTE

En aquest projecte hem estudiat les diferents parts on l'usuari ha d'accedir per dur a terme les diferents accions i les passes que ha de seguir per a que l'execució sigui la desitjada.

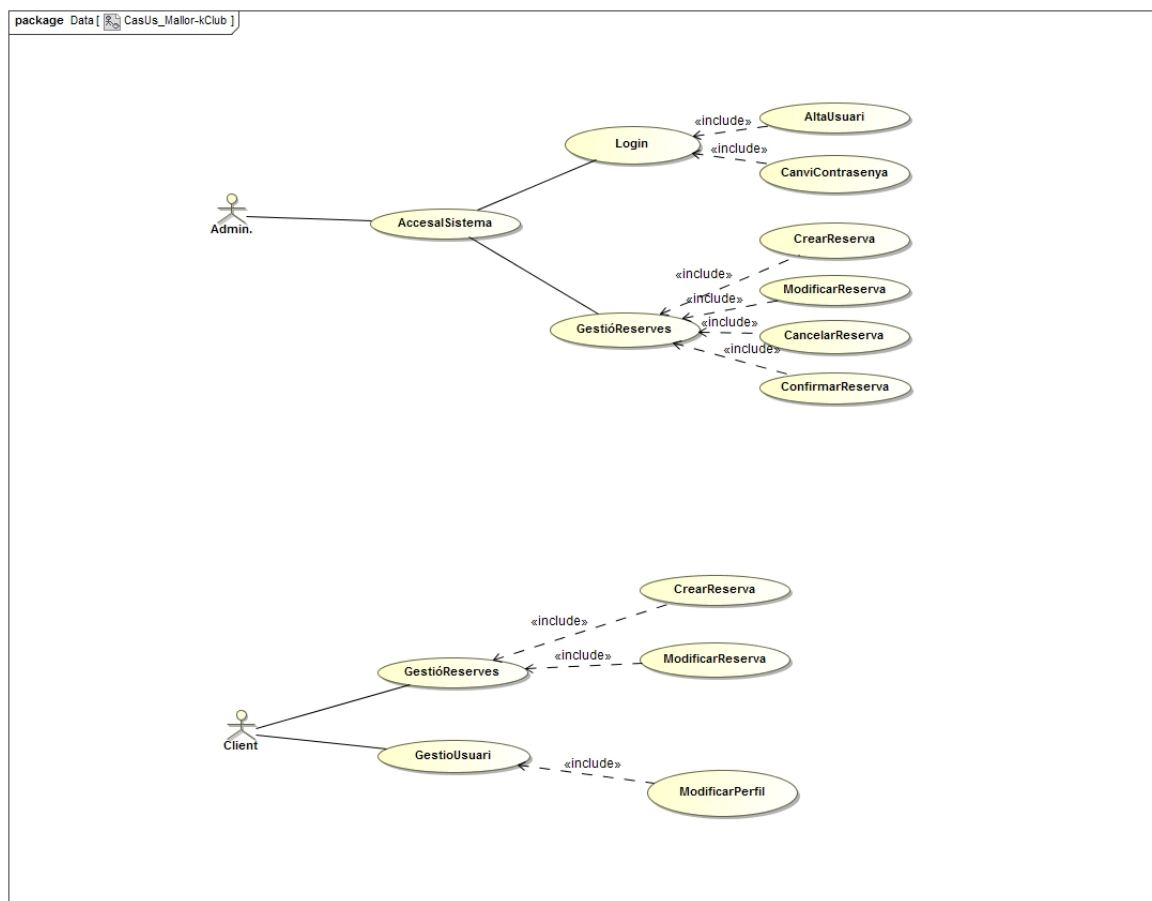
En aquests casos hem definit uns diagrames per seguir les diferents accions que pot fer l'usuari, depenent del perfil de l'usuari, si es un usuari tipus administrador o tipus client, i després les diferents parts de forma seqüencial per a dur a terme una reserva de forma correcta amb l'aplicació.

4.1. DIAGRAMA DE CASOS D'US

En aquest diagrama tenim dos tipus d'usuaris, l'usuari tipus “admin” i l'usuari tipus “client”.

- **Admin:** Aquest tipus d'usuari correspon a l'usuari del centre esportiu, el propietari de les pistes que s'han de reservar. Aquest perfil té els privilegis de gestió de les diferents pistes per reservar, i ha de validar aquelles reserves que estiguin a l'espera de confirmar i assignar pista.
- **Client:** L'usuari client serà l'usuari més comú a l'aplicació, ja que seran totes aquelles persones que volen realitzar l'acció de reservar una pista. Podran revisar les seves reserves i demanar per fer una nova reserva del tipus de pista que vol quedant com reserva pendent per a que sigui assignada la pista i confirmada per l'admin.

El model de casos d'ús s'ha anat actualitzant mentre s'anava realitzant el codi de la pàgina web, una vegada hem determinat quines son les opcions que hem pogut implementar, ens queda el següent diagrama:



- **AltaUsuari**
 - *Actor que realitza:* client/administrador
 - *Precondició:* usuari inexistent al sistema
 - *Estat final del procés:* Usuari creat a la base de dades
- **CanviContrasenya**
 - *Actor que realitza:* client/administrador
 - *Precondició:* usuari creat al sistema
 - *Estat final del procés:* Contrasenya modificada la base de dades
- **CrearReserva**
 - *Actor que realitza:* client
 - *Precondició:* cap
 - *Estat final del procés:* reserva creada al sistema en estat pendent

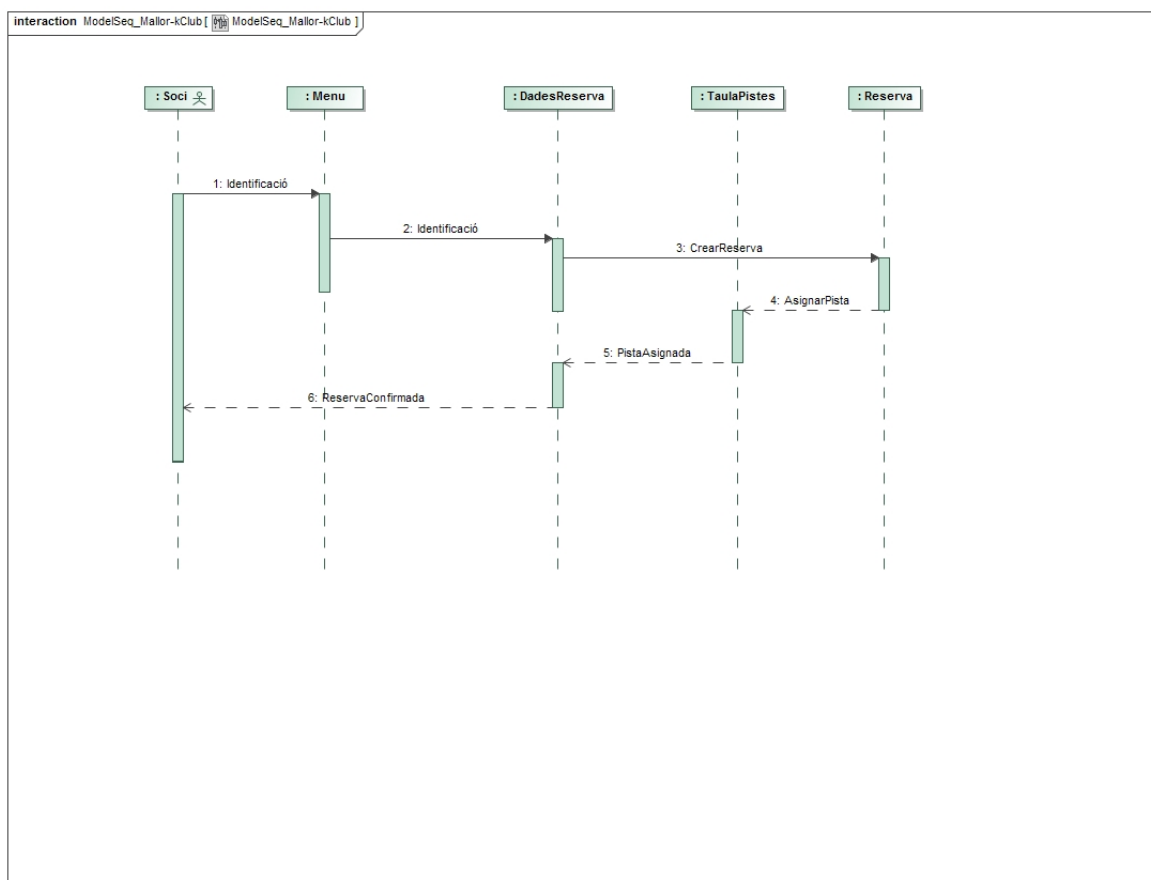
- **ModificarReserva**
 - *Actor que realitza:* administrador
 - *Precondició:* reserva existent
 - *Estat final del sistema:* reserva modificada.
- **ConfirmarReserva**
 - *Actor que realitza:* administrador
 - *Precondició:* reserva existent
 - *Estat final del sistema:* reserva confirmada amb pista assignada.
- **CancelarReserva**
 - *Actor que realitza:* administrador
 - *Precondició:* reserva existent
 - *Estat final del procés:* reserva eliminada del sistema.
- **ModificarPerfil**
 - *Actor que realitza:* administrador
 - *Precondició:* usuari creat al sistema
 - *Estat final del procés:* perfil d'usuari modificat a la base de dades.

4.2. DIAGRAMA DE SEQÜÈNCIA

En el diagrama de seqüència se pretén reflectir les diferents accions que se realitzaran per poder realitzar la tasca de confirmació d'una reserva de pista.

Una vegada hem realitzat tota la codificació de la pàgina web amb les diferents possibilitats que tenim, hem creat un model on l'usuari client realitza una reserva del tipus de pista que desitja (futbol, tenis o pàdel), i selecciona el dia i hora que vol, quedant la reserva pendent de validar i d'assignar pista, aquesta tasca correspon a l'administrador, que s'encarregarà de confirmar la reserva i seleccionar la pista disponible per aquesta reserva.







Un gràfic del diagrama de seqüència seria el següent:









5. PANTALLES DEL PROJECTE

De forma teòrica, analitzarem totes les pantalles que hem pre-definit per a dur a terme aquest projecte de forma exitosa, on puguem realitzar una reserva i poder dur un bon manteniment en aquesta aplicació web.


Pantalla	
Login	
USUARIS	Admin Client
DESCRIPCIÓ	Pantalla inicial de l'aplicació per connectar amb la base de dades i obtenir els valors necessaris per treballar amb l'aplicació
OPCIONS	<ul style="list-style-type: none"> - Camps d'introducció de text <ul style="list-style-type: none"> o Usuari o Password - Botons <ul style="list-style-type: none"> o Entrar a l'aplicació o Nou usuari

PANTALLA	<div> <div>INICIO</div> <div>Sign Up Sign in</div> </div> <div> <div>Mallor-K Club</div> <div>Disfruta de Mallorca en forma</div> </div> <div> <div>Sign in using your social accounts</div> <div>       </div> </div> <div> <div>Or with your account</div> <div> <div>Username</div> <div>gestela</div> </div> <div> <div>Password</div> <div>*****</div> </div> <div> <div>Sign in</div> <div>or Sign up</div> </div> <div>Forgot your password?</div> </div>
-----------------	---



Pantalla	
Nou Usuari	
USUARIS	Admin Client
DESCRIPCIÓ	Pantalla per crear un nou usuari a l'aplicació
OPCIONS	<ul style="list-style-type: none"> - Camps d'introducció de text <ul style="list-style-type: none"> o Nom o Cognom o Nom usuari o Password o Repetició de password o N° Telèfon o E-mail - Botons <ul style="list-style-type: none"> o Confirmar creació o Cancel·lar creació

PANTALLA	<p>Sign in using your social accounts</p> <div style="display: flex; justify-content: space-around; align-items: center;">       </div> <p>Or sign up using your email</p> <p>First Name <input type="text" value="First Name"/></p> <p>Last Name <input type="text" value="Last Name"/></p> <p>Email <input type="text" value="email"/></p> <p>Telephone <input type="text" value="telephone"/></p> <p>Username <input type="text" value="gestela"/></p> <p>Password <input type="password" value="....."/></p> <p>Password Requirements <div style="width: 100%; height: 10px; background-color: green; margin-bottom: 5px;"></div> 100%</p> <p style="text-align: center;"> <input type="button" value="Sign up"/> or <input type="button" value="Sign in"/> </p>
-----------------	---

Pantalla	
Creació de reserva	
USUARIS	Admin Client
DESCRIPCIÓ	Pantalla per realitzar una nova reserva. En aquesta pantalla seleccionarem el dia, hora i el tipus de pista a escollir. Aquesta petició genera una reserva NO confirmada, fins que l'administrador no confirmi aquesta reserva i assigni la pista reservada, només se crea una petició.
OPCIÓNS	<ul style="list-style-type: none"> - Camps d'introducció de text <ul style="list-style-type: none"> o Dia o Hora d'inici o Hora de fi o Tipus de pista (Futbol, tenis, pàdel) - Botons <ul style="list-style-type: none"> o Demanar confirmació o Cancel·lar creació

PANTALLA	<div> INICIO Administrador Pistas Reservas </div> <div>Gabi Estela</div>
	<div>  Distruta de Mallorca en forma </div> <div> <h3>Nueva Reserva</h3> </div> <div> TIPO DE RESERVA: <input checked="" type="radio"/> Futbol <input type="radio"/> Padel <input type="radio"/> Tenis </div> <div> Nombre <input type="text"/> </div> <div> Fecha Inicio <input type="text"/> </div> <div> Fecha Fin <input type="text"/> </div> <div> Estado <input type="text"/> </div> <div> Pista Asignada <input type="text"/> </div> <div> <input type="button" value="Crear"/> </div>

Pantalla	
Confirmació de reserva	
USUARIS	Admin
DESCRIPCIÓ	Pantalla per confirmar les reserves que estan pendents d'assignar pista. L'administrador s'encarrega d'assignar una pista del tipus seleccionat per la reserva per a que
OPCIONS	<ul style="list-style-type: none"> - Camps d'introducció de text (amb la reserva seleccionada) <ul style="list-style-type: none"> o Assignació de pista o Canvi d'estat - Botons <ul style="list-style-type: none"> o Modificar reserva

PANTALLA	INICIO	Administrador ▾	Pistas ▾	Reservas ▾	 Gabi Estela ▾
	<div style="text-align: center;">  <p>Mallor-K Club</p> <p>Disfruta de Mallorca en forma</p> </div> <h3>Editar Reserva</h3> <p>TIPO DE RESERVA: <input type="radio"/> Futbol <input checked="" type="radio"/> Padel <input type="radio"/> Tenis</p> <p>Nombre</p> <input type="text" value="nueva reserva LEO"/> <p>Fecha Inicio</p> <input type="text" value="2016-06-18T08:00:00.000Z"/> <p>Fecha Fin</p> <input type="text" value="2016-06-18T09:00:00.000Z"/> <p>Estado</p> <input type="text" value="Aceptada"/> <p>Pista Asignada</p> <input type="text" value="Pista Padel 2"/> <p><input type="button" value="Actualizar"/></p>				

Pantalla	
Consulta de reservas	
USUARIS	Admin Client
DESCRIPCIÓ	Pantalla per revisar les reserves realitzades
OPCIONES	- Botons <ul style="list-style-type: none"> ○ Consultar ○ Sortir

PANTALLA	<div> INICIO Administrador Pistas Reservas Gabi Estela </div> <div> Mallor-K Club Disfruta de Mallorca en forma </div> <div> Lista de Reservas realizadas </div> <div> Reserva creada en fecha Jun 13, 2016 por Leo Estela nueva reserva LEO tipo: Padel Fecha Inicio: 2016-06-18T08:00:00.000Z Fecha Fin: 2016-06-18T09:00:00.000Z Estado: Aceptada Pista Asignada: Pista Padel 2 </div> <div> Reserva creada en fecha Jun 7, 2016 por Leo Estela Reserva Leo 2 tipo: Futbol Fecha Inicio: 2016-06-18T16:00:00.000Z Fecha Fin: 2016-06-18T17:00:00.000Z Estado: Aceptada Pista Asignada: Pista futbol 1 </div>

Pantalla	
Gestió de perfil	
USUARIS	Admin Client
DESCRIPCIÓ	Pantalla per revisar les dades personals del perfil connectat, on es pot modificar els diferents camps indicats.
OPCIONS	<ul style="list-style-type: none"> - Camps d'introducció de text <ul style="list-style-type: none"> o Nom o Cognom o N° Telèfon o E-mail - Botons <ul style="list-style-type: none"> o Confirmar o Cancel·lar

PANTALLA	<div> INICIO Administrador Pistas Reservas Gabi Estela </div> <div> <h2>Mallor-K Club</h2> <p>Disfruta de Mallorca en forma</p> </div> <div> <h3>Settings</h3> <div> Edit Profile Change Profile Picture Change Password Manage Social Accounts </div> <div> Nombre Gabi Apellido Estela Email gabiestela@hotmail.com Telefono 657121502 Usuario gestela Save Profile </div> </div>
-----------------	---

Pantalla	
Gestió d'usuaris	
USUARIS	Admin
DESCRIPCIÓ	Pantalla únicament visible per l'administrador on pot revisar els diferents usuaris donats d'alta a l'aplicació i poder realitzar eliminacions o bloquejos d'usuaris en cas que sigui necessari.
OPCIONES	<ul style="list-style-type: none"> - Camps d'introducció de text (filtres de consulta) <ul style="list-style-type: none"> o Nom d'usuari - Botons <ul style="list-style-type: none"> o Consultar o Eliminar usuari o Modificar usuari o Sortir
PANTALLA	<div> INICIO Administrador Pistas Reservas Gabi Estela </div> <div> <h2>Mallor-K Club</h2> <p>Disfruta de Mallorca en forma</p> </div> <div> <h3>Users</h3> <div> Search </div> <div> <div> lesteia leosteia@hotmail.com user </div> <div> gestela gabiestela@hotmail.com user:admin </div> </div> <div> First Previous 1 Next Last </div> </div>

6. INSTAL·LACIÓ DE L'STACK MEAN

Una vegada ja tenim una idea clara de com realitzar la nostra aplicació web i la selecció de la tecnologia ja la vàrem definir a l'anterior punt, preparem el sistema per a començar a desenvolupar l'aplicació web.

Com ja he comentat varies vegades, aquest punt a sigut un dels que més problemes m'han donat i on he perdut més temps, que hauré de recuperar aquests dies, tal com he indicat al diagrama de Gantt.


Per començar, hem instal·lat al nostre sistema operatiu la base de dades MongoDB i el servidor NodeJS.

Una vegada hem instal·lat els programes indicats, podem fer servir la instrucció NPM (similar a "apt-get" de linux) per realitzar diferents instal·lacions necessàries per una correcta execució del programari MEAN.

Primer hem de definir una carpeta per ficar el nostre projecte, en aquest cas he definit varis noms per fer proves, algunes captures corresponen a la carpeta C:\MEAN\projects\test, i la carpeta que finalment va donar connexió al port 3000 va ser C:\MEAN\projects\prueba.

Una vegada definida la carpeta, començarem a obtenir tots els fitxers necessaris per una aplicació MEAN.

La instrucció per la instal·lació d'expressJS és la següent:

 C:\Windows\system32\cmd.exe

```
C:\MEAN\projects\test>npm install express --save
npm WARN saveError ENOENT: no such file or directory, open 'C:\MEAN\projects\test\package.json'
C:\MEAN\projects\test
~-- express@4.13.4
  +-- accepts@1.2.13
    | +-- mime-types@2.1.10
    | | `-- mime-db@1.22.0
    | `-- negotiator@0.5.3
  +-- array-flatten@1.1.1
  +-- content-disposition@0.5.1
  +-- content-type@1.0.1
  +-- cookie@0.1.5
  +-- cookie-signature@1.0.6
  +-- debug@2.2.0
    | `-- ms@0.7.1
  +-- depd@1.1.0
  +-- escape-html@1.0.3
  +-- etag@1.7.0
  +-- finalhandler@0.4.1
    | `-- unpipe@1.0.0
  +-- fresh@0.3.0
  +-- merge-descriptors@1.0.1
  +-- methods@1.1.2
  +-- on-finished@2.3.0
    | `-- ee-first@1.1.1
  +-- parseurl@1.3.1
  +-- path-to-regexp@0.1.7
  +-- proxy-addr@1.0.10
    | +-- forwarded@0.1.0
```

Tenim ara instal·lat expressJS amb nodeJS i mongoDB, ara hem d'obtenir l'estructura MEAN, per això farem la crida a "Yeoman" que ens crearà tota l'estructura de fitxers:

```
C:\Windows\system32\cmd.exe
C:\MEAN\projects\test>npm install -g yo
npm WARN deprecated npmconf@2.1.2: this package has been reintegrated into npm and is now out of date with respect to npm
C:\Users\Gabi Estela\AppData\Roaming\npm\yo -> C:\Users\Gabi Estela\AppData\Roaming\npm\node_modules\yo\lib\cli.js
> yo@1.7.0 postinstall C:\Users\Gabi Estela\AppData\Roaming\npm\node_modules\yo
> yodoctor

Yeoman Doctor
Running sanity checks on your system

✓ Global configuration file is valid
✓ NODE_PATH matches the npm root
✓ Node.js version
✓ No .bowerrc file in home directory
✓ No .yo-rc.json file in home directory
✓ npm version

Everything looks all right!
C:\Users\Gabi Estela\AppData\Roaming\npm
~-- yo@1.7.0
  +-- async@1.5.2
  +-- chalk@1.1.1
    | +-- ansi-styles@2.2.0
    | | `-- color-convert@1.0.0
    | +-- escape-string-regexp@1.0.5
    | +-- has-ansi@2.0.0
```

En aquesta part hem d'indicar el nom del nostre projecte i el programa ja s'encarrega d'assignar aquest nom als fitxers.

Ara comprovarem la connexió de la base de dades de MongoDB, veiem que ens indica la versió i la connexió al Shell amb usuari "Test".

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.10586]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Gabi Estela>cd c:\

c:\>cd mongodb

c:\MongoDB>cd bin

c:\MongoDB\bin>mongo
MongoDB shell version: 3.2.4
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user

> bye
2016-03-23T00:18:37.761+0100 E QUERY    [thread1] ReferenceError: bye is not defined :
@ (shell):1:1
```

Per poder treballar amb la base de dades, hem d'obrir abans el listener de la base de dades, MongoDB ens connecta al port 27017:

```
Microsoft Windows [Versión 10.0.10586]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>cd c:\mongodb

c:\MongoDB>cd bin

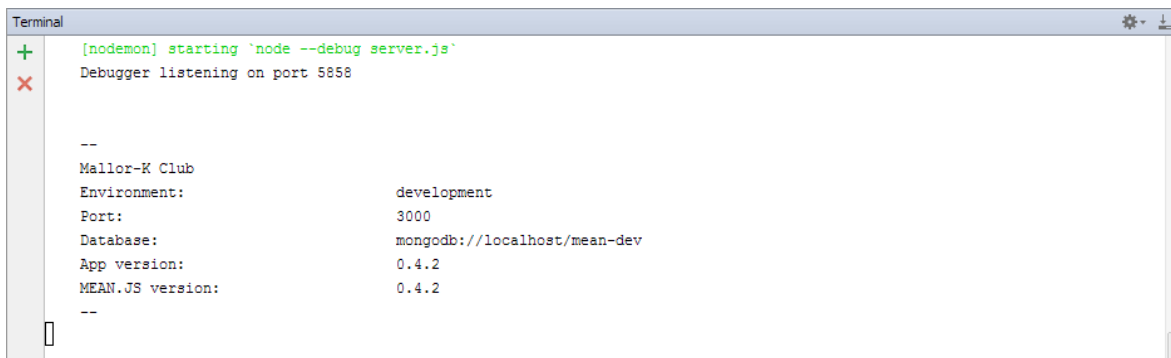
c:\MongoDB\bin>mongod
2016-04-07T18:21:32.676+0200 I CONTROL [initandlisten] MongoDB starting : pid=9140 port=27017 dbpath=c:\data\db\ 64-bit
host=DESKTOP-L85MMI4
2016-04-07T18:21:32.682+0200 I CONTROL [initandlisten] targetMinOS: Windows Vista/Windows Server 2008
2016-04-07T18:21:32.687+0200 I CONTROL [initandlisten] db version v3.2.4
2016-04-07T18:21:32.689+0200 I CONTROL [initandlisten] git version: e2ee9ffcf9f5a94fad76802e28cc978718bb7a30
2016-04-07T18:21:32.693+0200 I CONTROL [initandlisten] allocator: tcmalloc
2016-04-07T18:21:32.695+0200 I CONTROL [initandlisten] modules: none
2016-04-07T18:21:32.697+0200 I CONTROL [initandlisten] build environment:
2016-04-07T18:21:32.699+0200 I CONTROL [initandlisten] distarch: x86_64
2016-04-07T18:21:32.701+0200 I CONTROL [initandlisten] target_arch: x86_64
2016-04-07T18:21:32.704+0200 I CONTROL [initandlisten] options: {}
2016-04-07T18:21:32.710+0200 I - [initandlisten] Detected data files in c:\data\db\ created by the 'wiredTiger' s
storage engine, so setting the active storage engine to 'wiredTiger'.
2016-04-07T18:21:32.716+0200 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=4G,session_max=20000,e
viction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snapp
y),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2016-04-07T18:21:34.437+0200 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2016-04-07T18:21:34.438+0200 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'c
:\data\db\diagnostic.data'
2016-04-07T18:21:34.478+0200 I NETWORK [initandlisten] waiting for connections on port 27017
```

Així queda l'estructura inicial de l'aplicació MEAN:

Nombre

- app
- config
- node_modules
- public
- .bowerrc
- .csslintrc
- .editorconfig
- .gitignore
- .jshintrc
- .slugignore
- .travis.yml
- bower.json
- Dockerfile
- fig.yml
- generate-ssl-certs.sh
- gruntfile.js
- karma.conf.js
- LICENSE.md
- npm-debug.log
- package.json
- Procfile
- README.md
- server.js

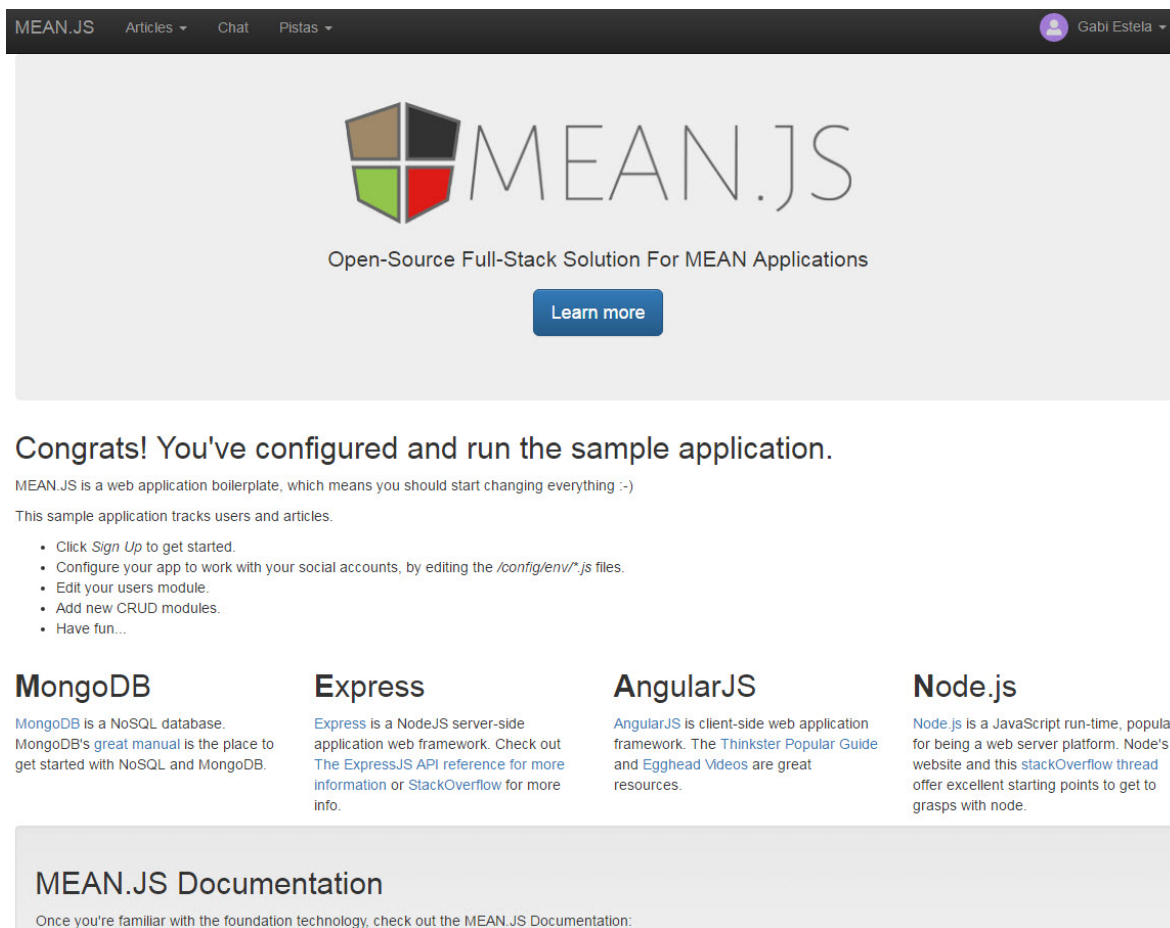
Ara hem d'executar “grunt” per poder realitzar la compilació i que ens comuniqui amb la base de dades i el servidor, obtenint un port de connexió des de localhost, que sol ser 3000:



```
Terminal
+ [nodeemon] starting `node --debug server.js`
x Debugger listening on port 5858

--
Mallor-K Club
Environment:      development
Port:             3000
Database:         mongodb://localhost/mean-dev
App version:      0.4.2
MEAN.JS version:  0.4.2
--
```

Amb una aplicació base de MEAN tindrem una pantalla d'inici com la següent:



Una vegada hem tingut accés a l'aplicació, hem creat els mòduls, aquests mòduls funcionen com les diferents taules que tenim a la base de dades MongoDB.

Les principals taules (o entitats) que tindrem en aquesta aplicació seran “Pistes”, “Usuaris” i “Reserves”. Tindrem un manteniment de cadascuna de les entitats, partint d'aquest manteniment realitzarem tota la part *front-end* de l'aplicació, per ara, tenim el manteniment *back-end* per aquestes entitats esmentades.

Una vegada hem realitzat les modificacions per a la nostra aplicació, s'ha definit una pantalla d'inici per a l'aplicació "Mallor-K Club", l'aspecte final és el següent:



La pantalla principal de MEAN ens demana crear un usuari inicial per tenir una connexió a l'aplicació:

Aquest usuari que hem creat ens permet controlar els mòduls que anem creant, podent realitzar insercions directes d'un registre a la base de dades des de el mòdul.

Les pantalles que anirem creant ara mateix consisteixen en mòduls de manteniment, el pas previ per entendre el funcionament d'aquest stack i veure com se fan les diferents connexions entre el servidor i el client.

Hem començat amb el mòdul de pistes, per crear el seu manteniment i realitzar diferents proves per poder usar el mòdul en futures reserves de pistes per part d'un usuari.

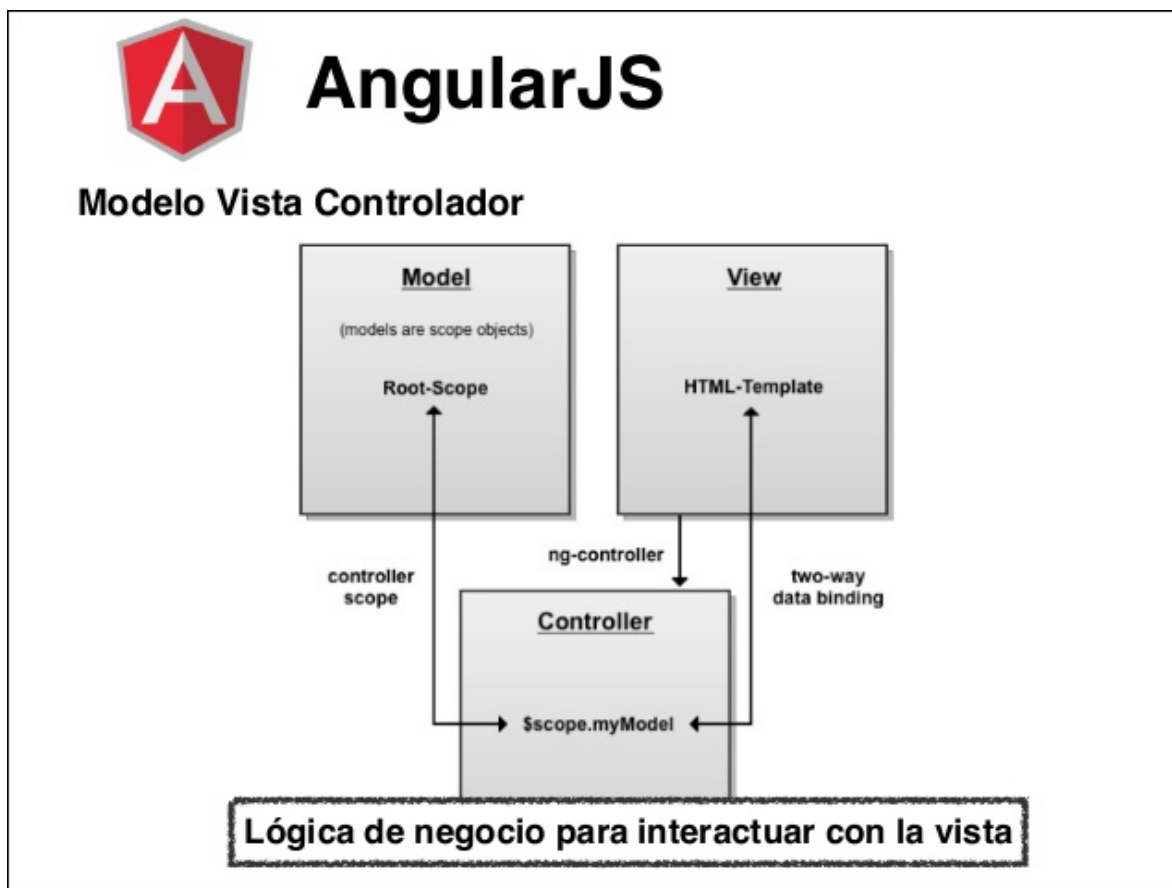
7. DIAGRAMA DE COL·LECCIONS DE BASE DE DADES NOSQL

Al nostre projecte MEAN farem servir una base de dades NOSQL, on se tindrem col·leccions, que seria el equivalent a les taules de les BBDD relacionals. Hem definit per aquest projecte 3 col·leccions, "pistes", "reserves" i "users".

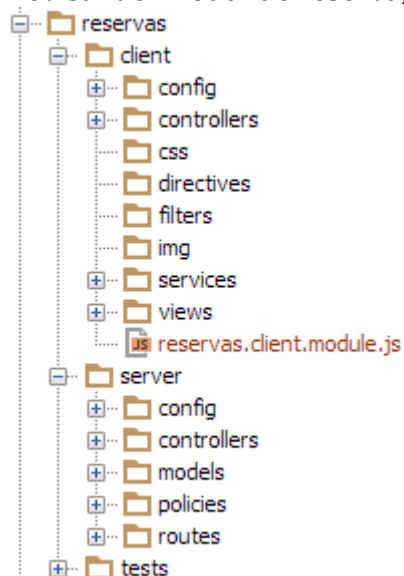


8. MODEL-VISTA-CONTROLADOR (MVC)

El model-vista-controlador (MVC) és un patró d'arquitectura de programari que separa les dades i la lògica de negoci d'una aplicació de la interfície d'usuari i el mòdul encarregat de gestionar els esdeveniments i les comunicacions. Per a això MVC proposa la construcció de tres components diferents que són el model, la vista i el controlador, és a dir, d'una banda defineix components per a la representació de la informació, i d'altra banda per a la interacció de l'usuari. Aquest patró d'arquitectura de programari es basa en les idees de reutilització de codi i la separació de conceptes, característiques que busquen facilitar la tasca de desenvolupament d'aplicacions i el seu posterior manteniment.



Revisant el mòdul de reserva, tenim la següent estructura:

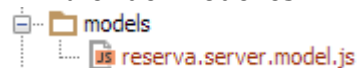


Tenim controlador en la part de client i en la part del servidor, on recuperarem les diferents accions de la base de dades a la part del servidor, que després prepararem en la part del client per a que la view pugui interactuar amb les dades.

8.1. MODEL

En la part del model és on indiquem quins son els camps que constitueixen la col·lecció que guardarem després a la base de dades.

El fitxer del model és:



En el cas de reserves, tenim el següent codi en el model:

```
/**
 * Reserva Schema
 */
var ReservaSchema = new Schema({
  name: {
    type: String,
    default: '',
    required: 'Please fill Reserva name',
    trim: true
  },
  tipo: {
    type: String,
    default: '',
    required: 'Selecciona tipo de Reserva',
    trim: true
  },
  fec_ini: {
    type: Date,
    default: '',
    required: 'Selecciona hora de inicio',
    trim: true
  },
  fec_fin: {
    type: Date,
    default: '',
    required: 'Selecciona hora de fin',
    trim: true
  },
  estado: {
    type: String,
    default: 'Pendiente',
    trim: true
  },
  pista_asig: {
    type: String,
    default: '',
    trim: true
  },
  coment: {
    type: String,
    default: '',
    trim: true
  },
  created: {
    type: Date,
```

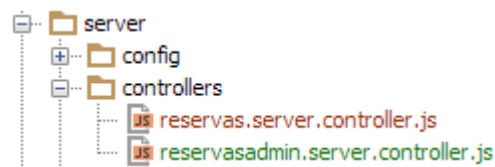
```

    default: Date.now
  },
  user: {
    type: Schema.ObjectId,
    ref: 'User'
  }
});

```

8.2. CONTROLADOR (SERVIDOR)

En la part del servidor tenim el controlador del mòdul de reserves per obtenir les diferents dades en format json com a resultat de les cridades a la base de dades mongodb.



En aquest *controller* indiquem els diferents “exports” que utilitzem amb MongoDB, fent les diferents funcions de consulta, actualització i eliminació de dades.

```

/**
 * Update a Reserva
 */
exports.update = function(req, res) {
  var reserva = req.reserva ;

  reserva = _.extend(reserva , req.body);

  reserva.save(function(err) {
    if (err) {
      return res.status(400).send({
        message: errorHandler.getErrorMessage(err)
      });
    } else {
      res.jsonp(reserva);
    }
  });
};

/**
 * Delete an Reserva
 */
exports.delete = function(req, res) {
  var reserva = req.reserva ;

  reserva.remove(function(err) {
    if (err) {
      return res.status(400).send({
        message: errorHandler.getErrorMessage(err)
      });
    } else {

```

```

        res.jsonp(reserva);
    }
    });
};

/**
 * List of Reservas (user)
 */
exports.list = function(req, res) {
    Reserva.find().sort('-created').populate('user',
'displayName').exec(function(err, reservas) {
        if (err) {
            return res.status(400).send({
                message: errorHandler.getErrorMessage(err)
            });
        } else {
            res.jsonp(reservas);
        }
    });
};

/**
 * Reserva middleware
 */
exports.reservaByID = function(req, res, next, id) {

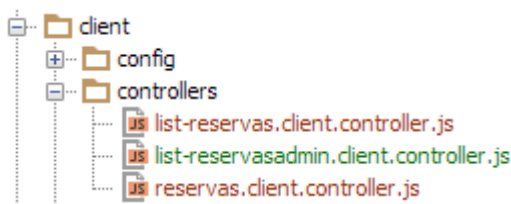
    if (!mongoose.Types.ObjectId.isValid(id)) {
        return res.status(400).send({
            message: 'Reserva is invalid'
        });
    }

    Reserva.findById(id).populate('user', 'displayName').exec(function
(err, reserva) {
        if (err) {
            return next(err);
        } else if (!reserva) {
            return res.status(404).send({
                message: 'No Reserva with that identifier has been found'
            });
        }
        req.reserva = reserva;
        next();
    });
};

```

8.3. CONTROLADOR (CLIENT)

En el cas del client, s'indica quin és el *controller* del servidor per poder utilitzar les diferents funcions definides, en el client farem diferents comprovacions, com enviar missatge d'avis abans de realitzar la funció d'eliminació de registre.



El codi del controller de reserves de client seria aquest:

```

(function () {
  'use strict';

  // Reservas controller
  angular
    .module('reservas')
    .controller('ReservasController', ReservasController);

  ReservasController.$inject = ['$scope', '$state', 'Authentication',
    'reservaResolve'];

  function ReservasController ($scope, $state, Authentication, reserva) {
    var vm = this;

    vm.authentication = Authentication;
    vm.reserva = reserva;
    vm.error = null;
    vm.form = {};
    vm.remove = remove;
    vm.save = save;

    // Remove existing Reserva
    function remove() {
      if (confirm('Are you sure you want to delete?')) {
        vm.reserva.$remove($state.go('reservas.list'));
      }
    }

    // Save Reserva
    function save(isValid) {
      if (!isValid) {
        $scope.$broadcast('show-errors-check-validity',
          'vm.form.reservaForm');
        return false;
      }

      // TODO: move create/update logic to service
      if (vm.reserva._id) {
        vm.reserva.$update(successCallback, errorCallback);
      } else {
        vm.reserva.$save(successCallback, errorCallback);
      }

      function successCallback(res) {
        $state.go('reservas.view', {
          reservaId: res._id
        });
      }
    }
  }
}

```

```

    }

    function errorCallback(res) {
        vm.error = res.data.message;
    }
}
}
}
})();

```

Podem veure a l'inici del codi les declaracions necessàries per enllaçar el controlador del client amb el controlador del servidor:

```

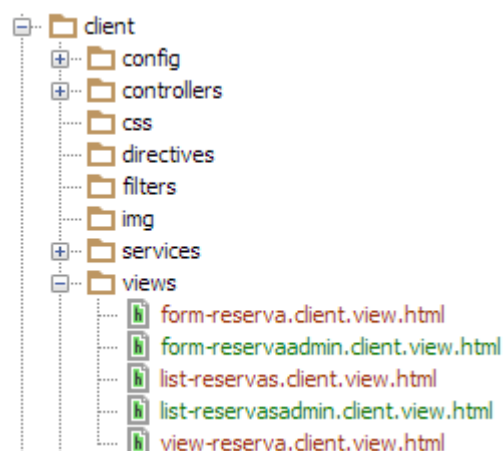
.module('reservas')
.controller('ReservasController', ReservasController);

ReservasController.$inject = ['$scope', '$state', 'Authentication',
'reservaResolve'];

```

8.4. VISTA

La vista correspon a la part visual de la web, amb html i codi angularJS afegit a les etiquetes, anirem afegint els diferents resultats obtinguts dels controladors.



Tenim diferents vistes per al llistat de reserves i modificació de les mateixes, separant la vista preparada pel client de la vista preparada per l'administrador. El codi de la pantalla de creació-modificació de reserva és el següent:

```

<section>
  <div class="jumbotron text-center">
    <div class="row">
      <div class="col-md-6 col-md-offset-3 col-sm-6 col-sm-offset-3 col-
xs-12">
        
      </div>
    </div>
  </div>
  <div class="page-header">

```

```

    <h2>{{vm.reserva._id ? 'Editar Reserva' : 'Nueva Reserva'}}</h2>
  </div>
  <div class="col-md-12">
    <form name="vm.form.reservaForm" class="form-horizontal" ng-
submit="vm.save(vm.form.reservaForm.$valid)" novalidate>
      <fieldset>
        <p>
          <label>TIPO DE RESERVA: </label>
          <input type = "radio"
            name = "tipo"
            id = "Futbol"
            value = "Futbol"
            checked = "checked"
            ng-model="vm.reserva.tipo" />
          <label for = "Futbol">Futbol</label>
          <input type = "radio"
            name = "tipo"
            id = "Padel"
            value = "Padel"
            ng-model="vm.reserva.tipo"/>
          <label for = "Padel">Padel</label>
          <input type = "radio"
            name = "tipo"
            id = "Tenis"
            value = "Tenis"
            ng-model="vm.reserva.tipo"/>
          <label for = "Tenis">Tenis</label>
        </p>
        <div class="form-group" show-errors>
          <label class="control-label" for="name">Nombre</label>
          <input name="name" type="text" ng-model="vm.reserva.name"
id="name" class="form-control" placeholder="Nombre" required>
          <div ng-messages="vm.form.reservaForm.name.$error"
role="alert">
            <p class="help-block error-text" ng-message="required">El
nombre de la Reserva es obligatorio.</p>
          </div>
        </div>
        <div class="form-group" show-errors>
          <label class="control-label" for="name">Fecha Inicio</label>
          <input name="fec_ini" type="datetime-local" ng-
model="vm.reserva.fec_ini" id="fec_ini" class="form-control"
placeholder="Introduzca fecha inicio" required>
          <div ng-messages="vm.form.reservaForm.fec_ini.$error"
role="alert">
            <p class="help-block error-text" ng-message="required">La
fecha de inicio de la Reserva es obligatorio.</p>
          </div>
        </div>
        <div class="form-group" show-errors>
          <label class="control-label" for="name">Fecha Fin</label>
          <input name="fec_fin" type="datetime-local" ng-
model="vm.reserva.fec_fin" id="fec_fin" class="form-control"
placeholder="Introduzca fecha fin" required>
          <div ng-messages="vm.form.reservaForm.fec_fin.$error"
role="alert">
            <p class="help-block error-text" ng-message="required">La

```

```

fecha de fin de la Reserva es obligatorio.</p>
</div>
</div>
<div class="form-group" show-errors>
  <label class="control-label" for="name">Estado</label>
  <input name="estado" type="text" ng-model="vm.reserva.estado"
id="estado" class="form-control" placeholder="Pendiente" disabled>
</div>
<div class="form-group" show-errors>
  <label class="control-label" for="name">Pista Asignada</label>
  <input name="pista_asig" type="text" ng-
model="vm.reserva.pista_Asig" id="pista_asig" class="form-control"
placeholder="Pendiente asignar pista" disabled>
</div>
<div class="form-group">
  <button type="submit" class="btn btn-default">{{ vm.reserva._id
? 'Actualizar' : 'Crear' }}</button>
</div>
<div ng-show="vm.error" class="text-danger">
  <strong ng-bind="vm.error"></strong>
</div>
</fieldset>
</form>
</div>
</section>

```

Podem veure com hem indicat els diferents camps per afegir les dades, indicant en cada dada a quin camp de base de dades pertany. En el cas del botó de creació-actualització, aquest ens envia a la funció “save”, depenent de si té un `_id`, ens actualitza el registre, sinó, ens crea un de nou.

9. CREACIÓ DE MÒDULS CRUD

Per poder interactuar amb la base de dades, els diferents mòduls que crearem son creats com mòduls CRUD.

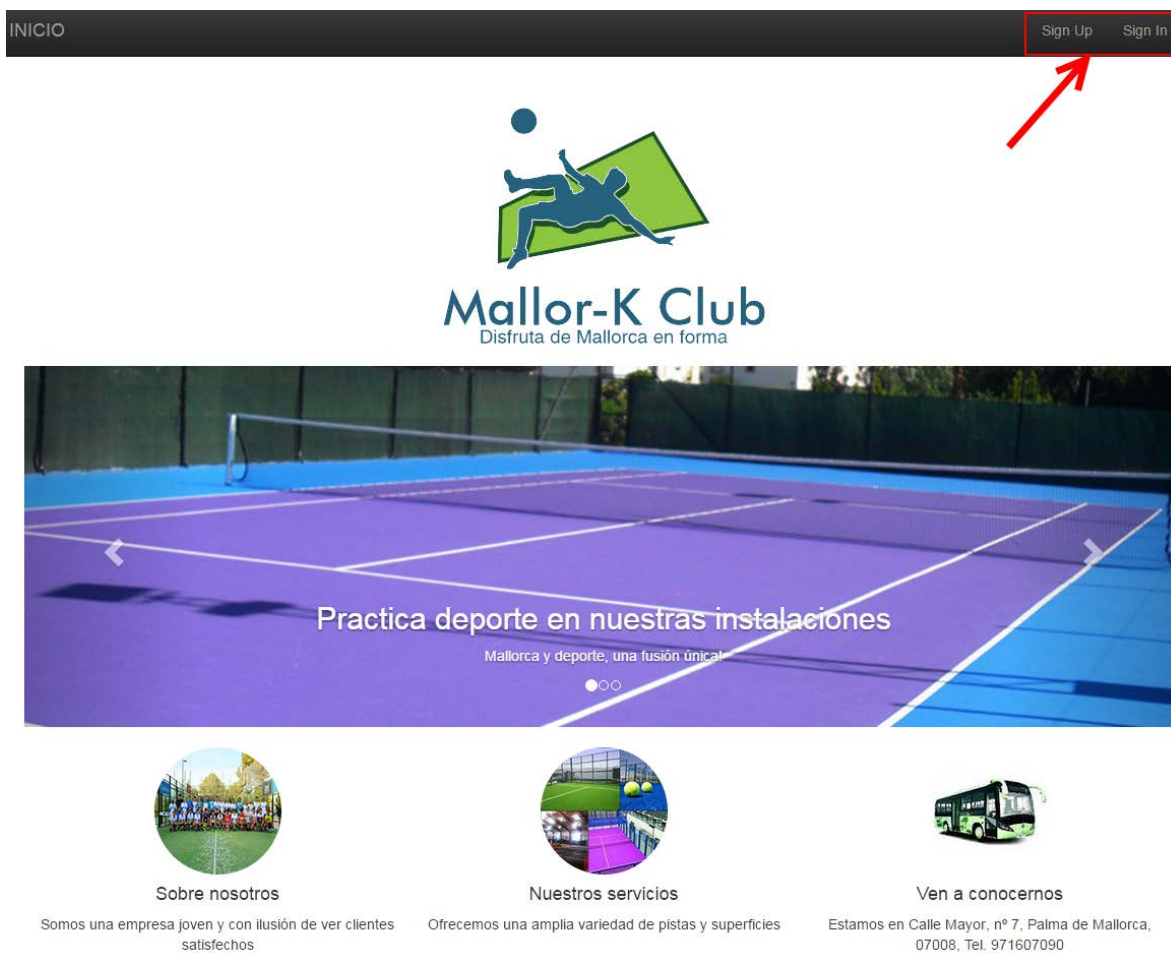
CRUD és l'acrònim de *Create, Read, Update, Delete*, com es pot veure, les funcions més bàsiques a una base de dades que se poden realitzar.

Hem creat aquests mòduls CRUD per les pistes i les reserves, ja que aquests mòduls estaran directament lligats a les col·leccions creades a la base de dades MongoDB.

9.1. MÒDUL USERS

Aquest mòdul el tenim pre-definit amb la creació del projecte MEAN, es genera la col·lecció “users” per mantenir el control d'usuaris i el rol que té cadascú en l'aplicació (admin o user).

Una vegada hem connectat a la pàgina, tenim l'opció de crear usuari o accedir amb un usuari existent



Amb l'opció de creació d'usuari, ens demanen els diferents camps necessaris per a crear un nou usuari al sistema.

Sign in using your social accounts



Or sign up using your email

First Name

Last Name

Email

Telephone

Username

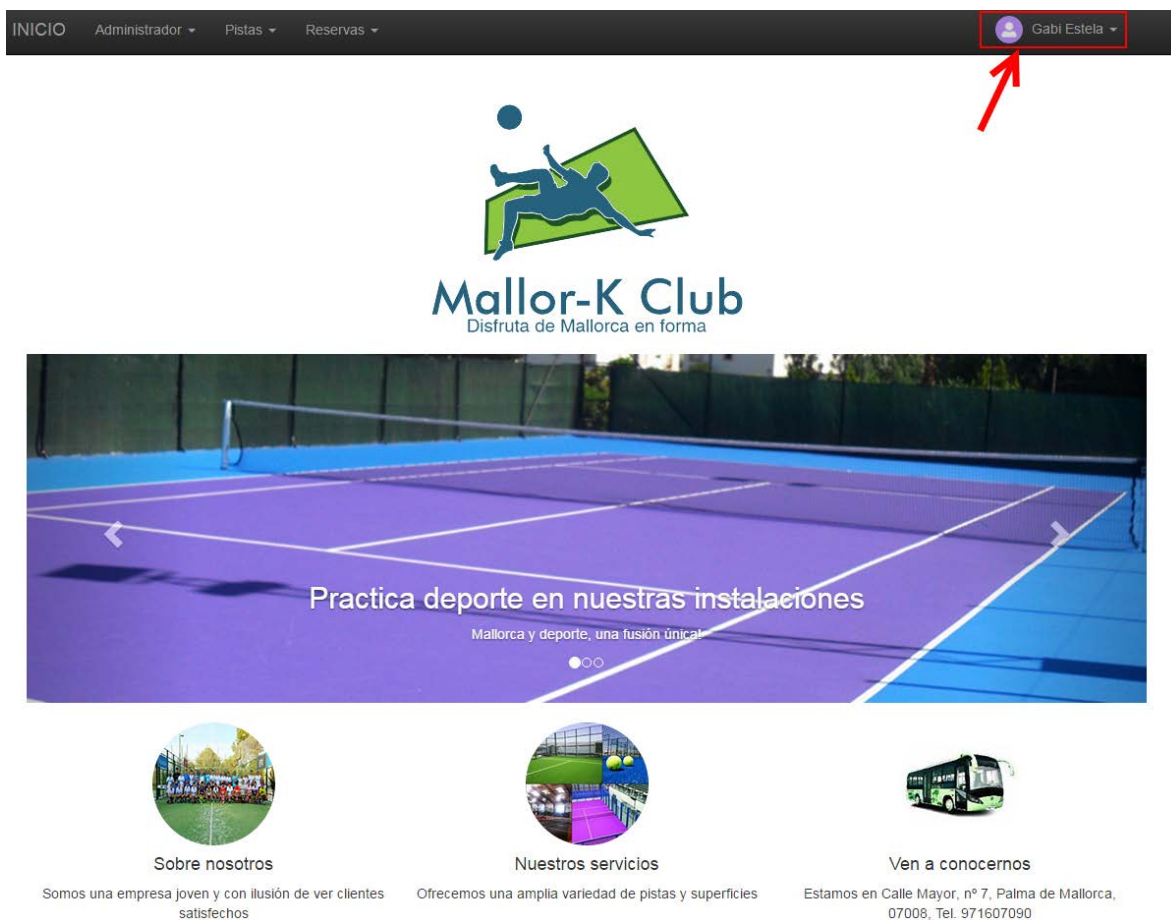
Password

Password Requirements

100%

[Sign up](#) or [Sign in](#)

Amb l'usuari creat, realitzarem connexió amb aquest usuari, el sistema ens informa de qui usuari està connectat:



Hem realitzat diferents proves d'usuaris, on hem vist com funcionen els rols i estan ben aplicats al sistema, ja que el menú de l'aplicació filtra les opcions segons el rol que tingui l'usuari connectat.

9.1.1. Roles

Hem definit dos tipus de rols:

- *Admin*: Aquest rol és el que té accés complet a l'aplicació, té els permisos per confirmar una reserva, assignar la pista, crear i modificar pistes.
- *User*: Aquest correspon a l'usuari client, aquest usuari té accés a la creació de reserva, una vegada crea la reserva, queda pendent esperant a que un usuari administrador la confirmi, aquesta acció no està disponible pel rol user.

9.2. MÒDUL PISTA










El mòdul "pista" ens permet realitzar les accions "Crear pista" i "Llista pistes" on podem veure totes les pistes que hem anat creant.

Per defecte, només se crea un únic camp de la taula, el camp "Name", hem afegit el camp "Tipo" per diferenciar les pistes de pàdel, les pistes de futbol sala i les pistes de tenis.

Mallor-K Club

Disfruta de Mallorca en forma

Pistas Disponibles

Nombre	Tipo	
Pista Tennis 3	Tenis	
pista tenis 2	Tenis	
pista tenis 1	Tenis	
Pista Futbol 3	Futbol	
Pista Futbol 2	Futbol	
Pista Futbol 1	Futbol	
Pista Padel 3	Padel	
Pista Padel 2	Padel	
Pista Padel 1	Padel	

Les diferents pistes les podem veure amb el detall que tenim indicat, ja que amb la creació de la pista, indicarem una ruta de base de dades on tenim emmagatzemada la imatge de la pista.

Mallor-K Club

Disfruta de Mallorca en forma

Pista Tenis 3



Creado con fecha Jun 7, 2016 por Gabi Estela

9.3. MÒDUL RESERVA

El mòdul “reserva” està preparat per poder crear una reserva i confirmar les diferents reserves realitzades, on l’administrador assignarà la pista per confirmar la reserva.

INICIO Administrador Pistas Reservas Gabi Estela

Mallor-K Club

Disfruta de Mallorca en forma

Lista de Reservas realizadas

Reserva creada en fecha Jun 7, 2016 por Leo Estela Reserva Leo 2 tipo: Futbol Fecha Inicio: 2016-06-18T16:00:00.000Z Fecha Fin: 2016-06-18T17:00:00.000Z Estado: Aceptada Pista Asignada: Pista futbol 1
Reserva creada en fecha Jun 7, 2016 por Leo Estela reserva padel leo 1 tipo: Padel Fecha Inicio: 2016-06-12T10:00:00.000Z Fecha Fin: 2016-06-12T11:00:00.000Z Estado: Pendiente Pista Asignada:
Reserva creada en fecha Jun 7, 2016 por Leo Estela Reserva Leo 1 tipo: Futbol Fecha Inicio: 2016-06-11T16:00:00.000Z Fecha Fin: 2016-06-11T17:00:00.000Z Estado: Pendiente Pista Asignada:

Per crear una reserva tenim l'opció al menú de reserves, aquesta ens portarà a la pantalla següent:

INICIO Pistas Reservas Leo Estela

Mallor-K Club

Disfruta de Mallorca en forma

Nueva Reserva

TIPO DE RESERVA: ☒ Futbol ☐ Padel ☐ Tennis

Nombre

Fecha Inicio

Fecha Fin

Estado

Pista Asignada

Hem d'indicar un nom identificatiu, el tipus de pista i les dates d'inici i fi de la reserva, aquesta queda en estat pendent, ja que l'ha de validar l'administrador i assignar una pista.

L'administrador revisa les reserves pendents i les indica una pista i les marca com acceptada per a reservar.

INICIO Administrador Pistas Reservas Gabi Estela

Mallor-K Club

Disfruta de Mallorca en forma

Editar Reserva

TIPO DE RESERVA: ☒ Futbol ☐ Padel ☐ Tennis

Nombre

Reserva Leo 2

Fecha Inicio

2016-06-18T16:00:00.000Z

Fecha Fin

2016-06-18T17:00:00.000Z

Estado

Aceptada

Pista Asignada

Pista futbol 1

Actualizar

Totes aquestes dades les guardem a la col·lecció de reserves:

```
/* 3 */
{
  "_id" : ObjectId("5756b5b5bb30f5ec189ba353"),
  "user" : ObjectId("5755bf2f51a5c6781d9472cb"),
  "created" : ISODate("2016-06-07T11:53:25.379Z"),
  "coment" : "",
  "pista_asig" : "Pista futbol 1",
  "estado" : "Aceptada",
  "fec_fin" : ISODate("2016-06-18T17:00:00.000Z"),
  "fec_ini" : ISODate("2016-06-18T16:00:00.000Z"),
  "tipo" : "Futbol",
  "name" : "Reserva Leo 2",
  "__v" : 0
}
```

Per tant, ja tindriem una reserva realitzada correctament i acceptada per l'administrador.

10. PROVES UNITÀRIES

Per comprovar el funcionament de la web hem anat realitzant diferents proves unitàries:

- *Creació d'una pista:*

Realitzarem la prova de creació de pista per veure si funciona la crida a base de dades i ens guarda correctament la pista.



Nueva Pista

Nombre

Tipo

Tenis

Imagen (nombre del fichero)

Crear

Nueva Pista

Nombre

El nombre de la Pista es obligatorio.

En cas de no indicar el nom o el tipus de pista, donarà error ja que aquests camps són obligatoris.

Tipo

El tipo de la Pista es obligatorio.

pista prueba unitaria

Pista creada satisfactoriament



Pista creada en MongoDB:

ObjectId("5...	ObjectId("57287c297a...	2016-06-22 22:07:36.836Z	pista prueba unitaria	0	Tennis	pistatenis2.jpg
----------------	-------------------------	--------------------------	-----------------------	---	--------	-----------------

- Creació d'una reserva:

Es crea una reserva per veure si se realitza correctament i es guarden les dades en la base de dades.

Nueva Reserva

TIPO DE RESERVA: ☐ Futbol ☒ Padel ☐ Tennis

Nombre
reserva padel leo 2

Fecha Inicio
26/06/2016 17:00

Fecha Fin
26/06/2016 --:--

junio de 2016

lun	mar	mié	jue	vie	sáb	dom
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Crear

El format de dates fa servir un calendari

Comprovarem si el sistema revisa les dades obligatòries:

Nueva Reserva

TIPO DE RESERVA: ☐ Futbol ☐ Padel ☐ Tenis ← No hem indicat cap tipus de reserva

Nombre

Fecha Inicio

Fecha Fin

Estado

Pista Asignada

← Al intentar crear la reserva, dona error per no indicar el tipus de reserva

Selecciona tipo de Reserva ←

Una vegada afegits tots els camps, crearem la reserva, aquesta pantalla ens confirma que s'ha creat correctament.



reserva padel leo 2

tipo: Padel
Fecha Inicio: 2016-06-26T15:00:00.000Z
Fecha Fin: 2016-06-26T16:00:00.000Z
Estado: Pendiente
Pista Asignada: ← No tenim cap pista assignada ja que se crea com una reserva pendent de confirmar

Reserva creada en fecha Jun 23, 2016 por Leo Estela

A la base de dades quedarà el registre pendent:

_id	user	created	coment	pista_asig	estado	fec_fin	fec_ini	tipo	name	V
ObjectId("5...)	ObjectId("5...)	2016-06-22 22:44:01.476Z			Pendiente	2016-06-26 16:00:00.000Z	2016-06-26 15:00:00.000Z	Padel	reserva padel leo 2	0

Realitzant la confirmació de la reserva, canviarem els valors dels camps d'estat i pista assignada:

Mallor-K Club

Disfruta de Mallorca en forma

Editar Reserva

TIPO DE RESERVA: ☐ Futbol ☒ Padel ☐ Tenis

Nombre

reserva padel leo 2

Fecha Inicio

2016-06-26T15:00:00.000Z

Fecha Fin

2016-06-26T16:00:00.000Z

Estado

Aceptada

Pista Asignada

Pista Padel 3

Actualizar

Actualitzarem les dades de la reserva i revisarem la llista de reserves:

Lista de Reservas realizadas

Reserva creada en fecha Jun 23, 2016 por Leo Estela

reserva padel leo 2

tipo: Padel

Fecha Inicio: 2016-06-26T15:00:00.000Z

Fecha Fin: 2016-06-26T16:00:00.000Z

Estado: Aceptada

Pista Asignada: Pista Padel 3

Podem veure com ara tenim actualitzat l'estat i la pista.

A la base de dades queda el registre de la següent manera:

_id	user	created	coment	pista_asig	estado	fec_fin	fec_ini	tipo	name	_v
Objectid("5...	Objectid("5...	2016-06-22 22:44:01.476Z		Pista Padel 3	Aceptada	2016-06-26 16:00:00.000Z	2016-06-26 15:00:00.000Z	Padel	reserva padel leo 2	0

11. FUTUR DEL PROJECTE

Aquest treball final de màster és un inici d'un projecte pensat en un futur per explotar en diferents centres esportius, aprofitant tots els centres que tenim per l'illa. Aquest treball és una versió primitiva de la idea final. Anirem indicant diferents punts analitzats que no s'han pogut implementar en aquesta versió i serien necessaris en una versió final.

- *Revisió de base de dades:* La base de dades que tenim ara mateix és una base de dades bàsica, amb les taules més principals per poder realitzar una reserva de forma satisfactoria. En aquest punt hem de realitzar relacions entre les col·leccions, aquest punt bàsic no és tan ràpid d'implementar en una base de dades NoSQL, on treballem amb col·leccions que per realitzar una relació d'una col·lecció a una altra, hem d'utilitzar tècniques com *MapReduce*, programant aquesta relació en el controlador de la col·lecció principal.
- *Modificació de reserves:* la modificació de reserva la duu directament l'administrador. El correcte funcionament seria la modificació directa de l'usuari per a que vagi modificant la reserva que ha creat, verificant en tot moment que la reserva no té solapament amb una altra reserva.
- *Control de reserves:* Les reserves les gestiona l'administrador per a que seleccioni les pistes depenent de les que vegi lliures, aquest control s'ha de fer automàticament, aquí entraria el primer punt de relacions de pistes amb reserves. Si hi ha 3 reserves de pistes de pàdel pel mateix dia i mateixa hora, el sistema detectarà que ja hi ha reserves fetes i directament informarà que la reserva no se pot realitzar a l'hora indicada ja que no hi ha pistes disponibles.
- *Disseny de pantalles:* Les pantalles queden ara mateix una mica buides de contingut, una idea per implementar és afegir una llista d'hores en el dia seleccionat per reservar una pista. Aquest punt estaria lligat a l'anterior, ja que amb les dades rebudes del sistema, tindrem la informació necessària per crear aquest calendari d'hores, deixant en vermell les hores sense pistes disponibles i en verd les hores amb pistes.

12. CONCLUSIONS DEL PROJECTE

Aquest projecte ha sigut totalment innovador per mi, he realitzat un estudi de les diferents opcions disponibles en tecnologia web, agafant finalment una solució tecnològica que considero bastant interessant de cara al futur, ja que les empreses estan començant a aplicar “AngularJS” a les seves aplicacions web, i com a inici a nivell professional, és una passa important que m’ha permès l’estudi i desenvolupament d’aquest projecte per aprofitar els coneixements adquirits en un àmbit de treball innovador.

Cal dir també que he tingut molts problemes al llarg del quadrimestre per arribar a fer funcionar tota l’eina MEAN, aquest temps “perdut” ha provocat una manca de funcionalitats que volia implementar i que finalment no han estat desenvolupades. El punt important és la finalització d’una eina web que sigui l’inici d’una bona pàgina per un futur on els professionals del món esportiu puguin gaudir d’aquest sistema.

Amb aquest projecte he pogut veure les grans possibilitats que ofereixen els nous sistemes de programació web, on una idea la pots implementar de forma ràpida, en comparació amb altres sistemes coneguts.