



## *Festivals.JS*

Arquitectura d'una aplicació web per trobar gent amb qui gaudir de concerts, amb Node.js

**Miquel Àngel Amorós Alberich**  
Màster d'Enginyeria Informàtica  
Desenvolupament d'aplicacions web.

**Professors responsables:**  
Ignasi Lorente  
César Córcoles

Primer Semestre 2016-2017.



Aquesta obra està subjecta a una llicència de [Reconeixement 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by/3.0/es/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Festivals.JS - Arquitectura d'una aplicació web per trobar gent amb qui gaudir de concerts, amb Node.js</i>
<b>Nom de l'autor:</b>	<i>Miquel Àngel Amorós Alberich</i>
<b>Nom del consultor/a:</b>	<i>César Córcoles</i>
<b>Nom del PRA:</b>	<i>Ignasi Lorente</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>01/2017</i>
<b>Titulació o programa:</b>	Màster d'Enginyeria Informàtica
<b>Àrea del Treball Final:</b>	Desenvolupament d'aplicacions web.
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Màxim 3 paraules clau, validades pel director del treball (donades als estudiants o en base a llistats, thesaurus, etc.)</i>
<p><b>Resum del Treball (màxim 250 paraules):</b> <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i></p>	
<p>En el present TFM s'ha desenvolupat una aplicació web amb una arquitectura basada en Javascript, tant del costat del client, amb HTML5, com per la part del servidor, amb node.js.</p> <p>Aquesta aplicació d'exemple serveix per a que les persones que volen quedar per gaudir festivals i concerts en grup.</p> <p>La motivació pel projecte és més personal que econòmica; simplement, vull consolidar el meu saber fer en aquestes tecnologies que revolucionen el món de la web i compartir aquesta experiència amb altres desenvolupadors; amb una aplicació d'exemple que considero útil.</p> <p>A nivell de tecnologies s'ha utilitzat:</p> <ul style="list-style-type: none"> <li>* Programació servidor: El motor Node.js, el framework Express i el gestor de paquets NPM.</li> <li>* Programació client: HTML5, CSS3, i els frameworks Angular i Bootstrap com a frameworks al costat del client.</li> <li>* Programació comuna entre les dues capes: JavaScript, JSON.</li> </ul> <p>L'aplicació es pot instal·lar en un servidor local, i actualment també està disponible en l'entorn de proves. El codi font està disponible a un repositori Github.</p> <p>En un futur es preveu integrar la aplicació amb una base de dades NoSQL com MongoDB, implementar un sistema d'autenticació d'usuaris registrats i afegir més funcionalitats útils; ja que aquestes tecnologies tenen molt potencial a nivell professional.</p>	
<p><b>Abstract (in English, 250 words or less):</b></p>	

In this project, a web application has been developed with an architecture based on Javascript; on the client side (front-end) with HTML5 and for the server (back-end) with node.js.

This sample application is for people who want to meet new people to enjoy festivals.

The motivation for this project is more personal than economic: I want to consolidate my expertise in these technologies that are improving the world of web and share this experience with other developers;; creating this sample web application that I consider useful.

The technologies which have been used:

- \* Server Development: The engine Node.js, the framework and the Express package manager NPM.

- \* Client Development: HTML5, CSS3, and frameworks and Angular Bootstrap as the client side frameworks.

- \* Shared Development between the two layers, JavaScript, JSON.

- The application can be installed on a local server, and is also available published in a website. The source code is available at a GitHub repository.

In the future it's planned to integrate the application with a database like MongoDB NoSQL; and also implement an authentication system of registered users and add more useful features; because these technologies have great professional possibilities.

## Dedicatòria

*Als meus amics: Dani, Siscu, Pau, Tere, Albert, Neus, Jordi i Laia F., Judit, Laia, penya Pestanya; entre d'altres amistats que m'han acompanyat, són la família que triem.*

*A tots els professionals i seguidors dels projectes i la filosofia del software lliure, i de compartir coneixement i experiències vitals. És d'agrair que bona part dels professionals i membres de la UOC amb qui he coincidit online, apostin per aquesta filosofia.*

*A les bones experiències professionals i companyerisme que he viscut a les empreses dCore, B2B 2000, Linecom, Sopra, S2 Concatel i IN2; i totes les que segurament vindran.*

*Agraeixo a tots els grups de música i cantautors que han tingut i tenen la valentia i talent de compondre música emocionant en català, ja que són referents en autosuperació i innovació disruptiva, qualitats fonamentals en la informàtica.*

*Finalment, gràcies de tot cor a la meva família; en especial als pares, avis, germana i nebodes, que m'han recolzat i estimat incondicionalment.*

# Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball.....	1
1.3 Enfocament i mètode seguit.....	2
1.3.1 Estudi de mercat.....	2
1.3.2 Investigació projectes existents.....	3
1.3.3 Mètode a seguir.....	3
1.4 Planificació del Treball.....	3
1.4.1 Tecnologies de software.....	3
1.4.2 Infraestructura de hardware.....	4
1.4.3 Metodologia a seguir.....	5
1.4.4 Pla de lliuraments.....	6
1.5 Breu sumari de productes obtinguts.....	6
2. Històries d'ús.....	7
2.1 Històries d'ús triades.....	7
2.2 Històries d'ús descartades.....	7
2.3 Criteris d'acceptació.....	8
2.4 Gestió d'incidències i millores.....	9
2.5 Futures versions.....	9
3. Model de dades.....	10
3.1 Entitat festival.....	10
3.2 Entitats per a registrar els usuaris.....	10
3.3 Entitats per a registrar assistents i organitzadors.....	11
3.4 Entitats per a consultar grups de música.....	11
3.5 Entitats per enviar missatges sobre els festivals.....	11
3.6 Model relacional de les entitats.....	12
4. Perfils i rols d'usuari.....	13
4.1 Perfils d'usuari de l'aplicació.....	13
4.2 Rols d'usuari dins del sistema.....	13
5. Prototips i navegació.....	14
5.1 Menú aplicació web.....	14
5.2 PA1 - Pantalla Inici.....	14
5.3 PA2 - Pantalla afegir/editar festivals.....	16
5.4 PA3 - Pantalla informació.....	16
5.5 PA4 - Pantalla autenticació.....	17
5.6 PA5 - Pantalla perfil.....	17
6. Arquitectura.....	18
6.1 Descripció general.....	18
6.2 Client.....	19
6.3 Servidor.....	19
6.4 Base de dades.....	19
6.5 Autenticació i gestió d'usuaris.....	20
6.6 Infraestructura.....	20
6.6.1 Entorn de desenvolupament local.....	20
6.6.2 Entorn de proves remot.....	21
6.6.3 Control de versions de codi.....	21
6.6.4 Infraestructura en un futur.....	21
7. Codi font.....	22
7.1 Estructura d'arxius.....	22
7.2 Model de dades de l'aplicació: fitxers JSON.....	22
7.3 Codi del nucli de node.js.....	23

7.4 Codi de l'API REST amb els serveis web de la capa servidor - carpeta App.....	23
7.5 Codi de la capa client - carpeta Angular.....	24
8. Tests.....	26
8.1 Tests d'acceptació.....	26
8.2 Tests de regressió.....	26
8.3 Tests d'integració de la API REST.....	26
8.4 Tests que falten implantar: unitaris.....	27
9. Seguretat.....	28
9.1 Mesures de seguretat implantades.....	28
9.2 Mesures de seguretat futures.....	28
10. Instal·lació i publicació.....	29
10.1 Instal·lació local.....	29
10.2 Publicació codi font a Github.....	29
10.3 Publicació remota aplicació web a Heroku.....	30
11. Conclusions.....	31
11.1 Encerts i fites assolides.....	31
11.2 Punts de millora i reptes pendents.....	32
12. Glossari.....	33
13. Bibliografia.....	34
13.1 Llibres digitals.....	34
13.2 Pàgines web.....	34
13.3 Articles web (blogs).....	34

## Llista de figures

Imatge 1 Model relacional de les entitats.....	11
Imatge 2 PA1 - Pantalla d'inici.....	13
Imatge 3 PA1 - Pantalla d'inici; festival seleccionat.....	14
Imatge 4 PA1 - Pantalla d'inici, filtre de festivals.....	14
Imatge 5 PA2 - Pantalla de detall.....	15
Imatge 6 PA3 - Pantalla d'informació.....	15
Imatge 7 Esquema arquitectura MEAN.....	17



# 1. Introducció

## 1.1 Context i justificació del Treball

Vull publicar una aplicació web per a que les persones que volen quedar per gaudir festivals i concerts en grup puguin trobar-se fàcilment; amb tecnologies OpenSource.

Sé que hi ha diverses pàgines amb temàtiques similars, entre d'elles destaca meetup, que permet als usuaris organitzar trobades per a diversos events: excursions, conferències, actuacions en bars i pubs, cursos i tallers, etc. Tanmateix, no encara no he trobat cap projecte de software lliure o OpenSource que exposi com aconseguir un producte o servei que permeti a qualsevol usuari crear pàgines amb aquestes possibilitats.

## 1.2 Objectius del Treball

Vull apostar per la simplicitat; per aportar el valor afegit necessari per a que la gent es trobi, i compartir al món els passos que he seguit. A més a més, ho vull desenvolupar amb software lliure, i que el resultat final sigui amb un disseny usable, accessible i multiplataforma.

La part més innovadora a nivell de desenvolupament del projecte és usar Javascript a la part del servidor, gràcies a node.js, que conté el motor V8, per arrencar serveis (i aplicacions) web de manera eficient.

Així, no ens fa falta usar cap tecnologia de servidor com Java EE, PHP, .NET, etc... i ens estalviem temps aprenent aquestes tecnologies i gestionant codi addicional per integrar les capes de l'arquitectura.

Ara mateix, la meva motivació pel projecte és més personal que econòmica; simplement, vull consolidar el meu saber fer aquestes tecnologies que revolucionen el món de la web i compartir aquesta experiència amb altres desenvolupadors; amb una aplicació d'exemple per publicar trobades a festivals.

Més concretament, la llista d'objectius serà:

- Assolir un nivell avançat de desenvolupament en Javascript; suficient per a crear i/o modificar u projecte de software que compleixi amb la funcionalitat descrita en el primer punt.
- Iniciar-me en programació a nivell de servidor amb Javascript i JSON amb l'ecosistema node.js
- Perfeccionar la meva experiència en desenvolupament d'aplicacions web amb serveis i estàndards web (HTTP,HTML,CSS,JSON).
- Tirar endavant el projecte i el posterior manteniment amb una metodologia de desenvolupament àgil, especialment per seguir el manifest àgil i aportar valor pas per pas.
- Utilitzar infraestructura OpenSource actual per a executar i compartir online el projecte; almenys un entorn per desplegar l'aplicació web, i a poder ser, persistència de dades i el versionat de codi.
- A més a més, fer el possible per tal d'exposar el procés que he seguit per construir aquest tipus de software; per a què qualsevol usuari familiaritzat amb la informàtica pugui agafar el software i instal·lar-lo.

La meua experiència professional com a desenvolupador Java EE (tant de front-end com de back-end, en models de dades complexos) serà un avantatge.

D'altra banda, sóc conscient de la complexitat del projecte, i que reconec com a riscos que no estic gaire familiaritzat amb projectes plenament programats amb JavaScript, ni la creació d'API's REST, ni la autenticació i gestió d'usuaris d'un sistema.

Tanmateix, la motivació i la vocació per desenvolupar un projecte útil i tecnològicament interessant farà que tiri endavant amb èxit.

## 1.3 Enfocament i mètode seguit

### 1.3.1 Estudi de mercat

En primer lloc, cal realitzar un estudi del mercat online per descartar que existeix un producte similar que ja cobreixi la necessitat que desitgem.

S'han trobat aquestes populars alternatives per a la creació d'events:

**Meetup** → La més popular arreu del món. Eficax, amb molts usuaris, i gratuït pels usuaris; té moltes activitats d'excursionisme, networking i d'oci nocturn en locals. Els inconvenients que té són l'alt preu de la quota d'organitzador (<https://www.meetup.com/es-ES/pricing/>) i que, tot i estar permès, no hia gairebé cap grup de música ni cap club de fans

**Facebook** → Fa molts anys que es poden crear events des de la xarxa social; però difícilment s'inscriu gent fora dels cercles d'amics/familiars.

Altres serveis, anteriors a meetup són:

**Gruppit**. → <http://www.gruppit.com/eventos/>

**Uolala!** → <http://www.uolala.com/>

**Gentcomtu** → <http://www.gentcomtu.com/> Aplicació web local(catalana), més orientada a trobar parella i amics, que té un apartat per inscriu-re's. Al igual que meetup, és gratuït inscriure's però cal pagar per organitzar-les. Poc a poc cau en desús degut al creixement massiu de meetup a Barcelona.

**Musicalfy** → <https://musical.ly/> És una xarxa social de videos com youtube o vimeo, però orientada a publicar videos musicals, per tal d'apropar els autors amb els fans. Pot ser que a la llarga es pugui usar per unir persones afins a l'assistència de festivals, però el present és que encara no.

**Altaveu.cat** → <http://www.altaveu.cat/> És un portal web on qualsevol usuari, gratuïtament i sense requerir registre, pot publicar concerts arreu dels Països Catalans. Està a mig camí de la idea inicial, perquè li falta la part de reunir a les persones a concerts i festivals.

En definitiva, hi ha tecnologies per cobrir aquesta necessitat, però no de manera prou eficient.

Cobrir necessitats de manera més eficient a la competència ha dut a l'èxit a negocis com: wallapop(compra-venda), blablacar(compartir cotxe), booking(reserva hotels), etc... que en el passat estaven mig cobertes amb portals genèrics com milanuncios o loquo.

### 1.3.2 Investigació projectes existents.

Tot seguit, al comprovar que no hi ha cap producte tan específic com el desitjat, la estratègia és aconseguir obtenir un producte i/o servei que permeti als fans de concerts actuals trobar-se és realitzar un projecte des de zero.

Abans de prendre aquesta decisió; he investigat si hi havia una alternativa existent per a poder adaptar-la.

Fins ara; s'han trobat aquestes solucions OpenSource per a crear software ideals per a negocis i administracions molt madurs:

- **Blogs** (wordpress-PHP, Keystone-Node.js)
- **Gestió documental**(Alfresco-JavaEE)
- **Portals**(Drupal-PHP, Liferay-JavaEE)
- **Tendes online**(Prestashop-PHP, Magento-PHP)
- **Aules obertes**(Moodle-PHP, Reaction-Node.js)
- **Xarxes socials**(Elgg-PHP)

Segur que n'hi ha alguns més; però no existeix cap gestor d'events en una versió estable. Per tant, el projecte serà possible desenvolupant amb les llibreries i frameworks que suporta l'arquitectura de software lliure node.js

### 1.3.3 Mètode a seguir

Per això, considero necessari començar a desenvolupar el software des de zero, i provar l'ecosistema node.js amb el suport de llibreries JavaScript existents: ExpressJS, AngularJS...

## 1.4 Planificació del Treball

### 1.4.1 Tecnologies de software

A nivell de tecnologies, tinc molt clar que em ve de gust experimentar amb l'ecosistema node.js i aprofitar l'ocasió per desenvolupar amb una arquitectura que usi les sigles MEAN (MongoDB-ExpressJS-AngularJS-NodeJS).

Inicialment, la persistència de dades del servidor es realitzarà amb fitxers JSON protegits per a cada entitat. En un futur, està previst implementar un motor de base de dades. S'han estudiat breument aquestes 3 possibilitats OpenSource:

- **MySQL**, gestor relacional molt eficient, basat en SQL i lleuger; especialment popular en aplicacions PHP i Java EE.
- **MongoDB**, gestor NoSQL molt eficient, basat en documents estructurats en JSON, lleuger. Fàcil d'integrar en aplicacions Node.js amb la llibreria mongoose.
- **Apache Cassandra**, gestor NoSQL lleuger i molt eficient. Tanmateix, és el més difícil de configurar i menys madur dels 3.

La decisió final, en un futur, és optar per **MongoDB**, ja que és fàcil d'integrar amb aplicacions node.js, és molt eficient i el tipus d'aplicació a desenvolupar encaixa amb una aplicació NoSQL (que vol dir Not Only SQL), paradigma en que preval la eficiència més que una estructura rígida.

De cara al front-end: optar pel framework Angular.js és opcional, però pot aportar una reducció del temps de desenvolupament. En qualsevol cas, la presentació estarà basada en HTML5:

- **HTML5** per a l'estructura.
- **CSS3** per l'estil.
- **JavaScript** per al tractament de les dades.

En tot cas, també s'usarà el format de dades **JSON** per a transferir les dades de l'aplicació client al servidor.

La conveniència de l'us d'Angular.JS com a framework per a l'estructura i tractament de dades; i de Bootstrap per a facilitar els estils es decideix sobre la marxa.

Finalment serà interessant crear una **API REST** per accedir a totes les consultes necessàries; fins i tot facilitarà el propi desenvolupament. No tinc gaire experiència en la creació de webservices, però una de les majors fortaleses de node.js i express.js és la fàcil creació de serveis web segurs, així que pot ser viable crear serveis web.

#### 1.4.2 Infraestructura de hardware

Està provat que desenvolupar tot el programari en un servidor local en node.js i garantir la persistència de les dades és perfectament viable.

Tot seguit, per exposar públicament el projecte, és necessari d'investigar les múltiples opcions d'allotjament en un servidor online.

Segurament serà necessari inscriure's en un servei que permeti executar una aplicació web desenvolupada en node.js, a poder ser que ofereixi un servei gratuït o assequible per a projectes petits com aquest.

Les 4 opcions que s'han investigat ara per ara són: Heroku, Cloud9, DigitalOcean i Codepen.

- Cloud9 és un projecte interessant per a desenvolupar online, però no permet tenir arrencat cap projecte més d'una hora.
- Heroku està especialitzat en allotjar projectes OpenSource, i pot tenir el projecte arrencat sense restriccions però configurar-lo és una mica més lent que cloud9.
- DigitalOcean; la opció més completa, a un preu assequible, de 5€ al mes. És una màquina virtual de Linux, que permet instal·lar tot el programari necessari per arrencar: node.js, MongoDB, Github...
- Codepen es una opció molt bona per a presentar desenvolupaments perquè té una interfície molt amigable i suporta moltes llibreries JS: JQuery,Angular... però no permet arrencar i gestionar projectes amb node.js, com Heroku o Cloud9.

La decisió que s'ha pres per a presentar aquest projecte al públic és allotjar l'aplicació a de manera gratuïta Heroku.

S'ha descartat Digital Ocean a curt, termini, tot i ser una solució més professional. No tant pels diners sinó el temps d'instal·lació i configuració de la màquina.

### 1.4.3 Metodologia a seguir

Durant, la PAC1 i la PAC2, es va planificar definir, prioritzar i dissenyar històries d'usuari rellevants per als objectius de l'aplicació. Més endavant, durant la PAC3, es va revisar la llista d'històries, ajustar la prioritat d'algunes, i implementar les més prioritàries (les 4 primeres, de prioritat 100).

Finalment, la llista d'històries d'usuari en aquest Lliurament Final és la mateixa que la PAC3, i s'han implementat diverses històries de prioritat mitjana (entre 70 i 90); per tant, les funcionalitats que hi haurà implementades seran les següents:

<b>Codi</b>	<b>Prioritat</b>	<b>Descripció</b>
HU1	100	Com a visitant vull veure en que consisteix la pagina, qui l'esta fent, novetats,...
HU2	100	Com a visitant vull veure un llistat de tots els pròxims concerts ordenats per data
HU3	100	Com a visitant/usuari vull crear una trobada a un concert
<u>HU4</u>	100	També vull poder editar el concert abans que comenci.
<u>HU5</u>	90	Fins i tot, anul·lar un concert abans que comenci.
HU8	80	Vull trobar festivals a la localitat que indiqui
HU9	80	Vull filtrar els festivals que siguin gratuïts.
HU16	70	Obtenir el llistat de concerts paginat si és llarg (pex de 20 en 20)

Tanmateix, la llista d'històries d'usuari que no s'ha pogut implementar en aquest Lliurament Final per motius de prioritat i de complexitat:

<b>Codi</b>	<b>Prioritat</b>	<b>Descripció</b>
<u>HU6</u>	70	Com a visitant/usuari vull realitzar operacions de manera privada, amb un sistema d'autenticació per comptes d'usuari, per tal que només jo pugui modificar i eliminar els festivals que he creat.
HU7	70	Com a visitant/usuari vull inscriure'm o desinscriure'm a festivals.
HU10	70	I també vull trobar festivals amb el nom del grup que especifiqui.
HU11	60	Com a usuari organitzador d'un concert vull poder queixar-me sobre la gent que no ha assistit sense avisar
HU12	60	També, com usuari assistent, vull poder queixar-me de si un organitzador ha anul·lat la trobada sense avisar
HU13	60	Com a usuari vull enviar missatges públics sobre el festival creat, per plantejar dubtes.
HU14	50	Com a usuari vull saber quanta gent s'ha inscrit i dades bàsiques dels inscrits.
HU15	40	Com a usuari vull votar i opinar com ha anat l'event posteriorment
HU17	30	Com a usuari, vull que la web es vegi en diversos idiomes.
HU18	70	Com usuari, vull que el llistat de concerts de la pàgina inicial es carregui ràpidament (en menys de 10 segons), fins i tot en connexions lentes.

#### 1.4.4 Pla de lliuraments

A grans trets, una possible pla de lliuraments ajustat a una metodologia àgil (com XP o Scrum) i a les dates d'avaluació de les PAC, pot ser el següent:

**Lliurament PAC 2 - Planificació i disseny.**

**Data: 06-nov-16**

Acabar planificació de l'aplicació.

Entregar disseny de l'aplicació.

**Lliurament PAC 3 - Desenvolupament del software.**

**Data: 06-des-2016**

Completar el disseny de l'aplicació

Lliurar aplicació funcional amb les històries d'usuari de màxima prioritat(100) completes; en un servidor remot, i adjuntar el codi font.

**Lliurament Final – Memòria i manteniment del software.**

**Data: 09-gen-2017**

Corregir errors detectats en l'aplicació.

Implementar una nova versió amb algunes funcionalitats noves; de prioritat mitjana (90-70).

Presentació de la memòria de demostració.

#### 1.5 Breu sumari de productes obtinguts

En resum, s'han desenvolupat els següents lliurables:

Aplicació web festivals.js

<https://festivals-js.herokuapp.com/>

Codi font i documentació accessible en un repositori públic.

<https://github.com/mamorosdev/festivals-js-alpha2>

Llista d'incidències i millores pendents:

<https://github.com/mamorosdev/festivals-js-alpha2/issues>

## 2. Històries d'ús.

### 2.1 Històries d'ús triades.

Aquesta segona versió, la última que serà avaluada al TFM, s'anomenarà alpha-1.0 És més usable que l'anterior, però encara no es pot considerar beta perquè el sistema de persistència de fitxers no és acceptable en un entorn de producció.

Les noves històries d'ús que s'han completat en aquesta fase són aquestes, que tenen una prioritat alta (entre 70 i 90):

1. HU5 Fins i tot, anul·lar un concert abans que comenci.
2. HU8 Vull trobar festivals a la localitat que indiqui
3. HU9 Vull filtrar els festivals que siguin gratuïts.
4. HU16 Obtenir el llistat de concerts paginat si és llarg (pex de 10 en 10)

Que es s'afegiran a les de la primera versió (PAC3); és a dir, les 4 primeres, que tenen una prioritat màxima (100); que són:

1. HU1. Com a visitant vull veure en que consisteix la pagina, qui l'esta fent, novetats,...
2. HU2. Com a visitant vull veure un llistat de tots els pròxims concerts ordenats per data
3. HU3. Com a visitant/usuari vull crear una trobada a un concert
4. HU4. I vull poder editar l'event abans que comenci

Considero que aquestes històries són les més importants; i les històries que s'han afegit aporten força valor i no han estat difícils d'implementar.

### 2.2 Històries d'ús descartades.

Algunes històries, tot i ser fonamentals en una aplicació professional, s'han descartat perquè la seva complexitat excedeix l'abast d'aquest projecte.

Son les relacionades amb l'autenticació dels usuaris al sistema; a destacar les següents (prioritat 70):

- HU6 Com a visitant/usuari vull realitzar operacions de manera privada, amb un sistema d'autenticació per comptes d'usuari, per tal que només jo pugui modificar i eliminar els festivals que he creat.
- HU7 Com a visitant/usuari vull inscriure'm o desinscriure'm a festivals.

Aquestes funcionalitats sí que em preocupa, no pas per tenir control dels usuaris que s'inscriuen, sinó per seguretat dels propis usuaris, ja que no hi ha cap mecanisme eficaç per evitar que qualsevol usuari editi i esborri festivals que han creat altres persones.

En menor mesura, son interessants les següents funcionalitats (prioritat 60):

- HU11 Com a usuari organitzador d'un concert vull poder queixar-me sobre la gent que no ha assistit sense avisar.
- HU12 També, com usuari assistent, vull poder queixar-me de si un organitzador ha anul·lat la trobada sense avisar.
- HU13 Com a usuari vull enviar missatges públics sobre el festival creat, per plantejar dubtes.

Aquestes tres últimes tenen menor prioritat perquè els usuaris tenen alternatives. Per a les històries HU11 i HU12 es pot enviar emails de reclamació als administradors.

En quant a la HU13 es pot aconseguir el mateix feedback mitjançant les dades de contacte que proporcionin els usuaris que creen activitats (correu electrònic, número de mòbil...)

En futures versions serà important aportar aquestes funcions; per guanyar en seguretat i usabilitat.

### 2.3 Criteris d'acceptació.

Les funcionalitats d'aquesta versió beta; que serviran per a provar que l'aplicació web compleix les expectatives son:

- La pàgina inicial mostrarà un llistat amb els pròxims concerts/festivals i des de la qual l'usuari podrà seleccionar-ne un per a veure els seus detalls.
- Des de la pàgina inicial es podran filtrar els concerts per data d'inici, per localitat, i fins i tot filtrar només els que són gratuïts.
- Els resultats de la cerca de concerts no sortiran tots alhora, sinó que es veuran paginats. S'ha decidit que es mostrin els 5 primers, i al final de la taula hi hagi un botó "Mostrar més", que quan l'usuari el pitgi es mostrin els següents 5 concerts, fins que es mostrin tots. És una forma més senzilla i moderna de paginar els resultats.
- Hi haurà una pantalla per a crear nous festivals; que també servirà per mostrar el detall d'un concert seleccionat, poder-lo editar i eliminar.
- Una pantalla amb informació del projecte (pàgina estàtica), de contacte amb l'autor; i sobretot, amb un missatge per aclarir els límits de responsabilitats que també s'indiquen en la llicència de software triada.

Més concretament, cal provar les següents funcionalitats:

- Que les 3 pantalles mostren adequadament tota la informació descrita.
- Que es mostra correctament el llistat de concerts a la pàgina inicial.
- Que es poden cercar els concerts filtrats per data d'inici, per localitat, o filtrar només els gratuïts o tots.
- Que no es mostrin tots els resultats de cop, sinó que es vagin mostrant més quan l'usuari els demani amb un botó "Mostrar més".
- Que es visualitza el detall de cada concert correctament.
- Que es poden afegir alguns nous concerts.
- Que es poden modificar les dades dels concerts seleccionats.
- Que es poden eliminar concerts.

A més a més, es faran proves de regressió (manualment) al llistat de resultats; per verificar que segueix funcionant correctament amb la incorporació dels filtres, paginació i esborrat de festivals:

- En primer lloc verificar el llistat de festivals que es carrega inicialment, el llistat que apareix després d'aplicar un o més filtres, la paginació dels resultats llistat, i sobretot, que el llistat reflecteix correctament les operacions afegir, editar i eliminar festivals.



## 2.4 Gestió d'incidències i millores

El servei on s'allotja el codi font, Github, del qual es parlarà més endavant; disposa d'un sistema de control d'incidències inclòs.

El d'aquest projecte està ubicat a:

<https://github.com/mamorosdev/festivals-js-alpha2/issues>

## 2.5 Futures versions

La versió actual és la **alpha-2**. La versió inicial, de la PAC 3 és la alpha-1. Quan hagi millorat una mica el disseny, la usabilitat, i corregit algún error menor es convertirà en la versió **beta-2**.

Quan es publiqui una versió amb més **històries d'usuari implementades** i la persistència en **una base de dades** es nombrarà com a **alpha-3**. Quan funcioni de manera estàble es nombrarà com a **beta-3**.

La base de dades, apart d'aportar seguretat i alt rendiment, és necessària per a implementar les històries per a la autenticació i gestió d'usuaris.

## 3. Model de dades.

### 3.1 Entitat festival.

La entitat bàsica des de la qual gira la aplicació és: festival.

#### **Camps entitat FESTIVAL.**

<b>Id(*)</b>	Identificador intern únic.
<b>Títol(*)</b>	Un títol atractiu del festival.
<b>Concerts(*)</b>	Camp de text lliure amb la llista de concerts.
<b>Data inici(*)</b>	Data que comença.
<b>Hora inici(*)</b>	Hora que comença.
<b>Data de fi</b>	Si és un festival de diversos dies indicar-ho.
<b>Població(*)</b>	Població on començarà el festival.
<b>Ubicació(*)</b>	Adreça on començarà el festival.
<b>Lloc</b>	Lloc on començarà el festival.
<b>Informació</b>	Camp on inserir un enllaç amb informació oficial del festival/concert.
<b>Descripció(*)</b>	Camp on l'organitzador explica amb més detalls on, quan i com trobar-se amb els assistents.
<b>Contacte.(*)</b>	Camp per a què l'organitzador indiqui el mitjà que li va millor per contactar amb els assistents (whatsapp, email, ...)
<b>Preu(*)</b>	Camp text lliure on s'indica el preu, si és gratuït o no.

Representar aquesta entitat és suficient per a la fase beta de l'aplicació.

Per a futures versions, s'ha previst modelar més entitats, que es plantegen en els apartats posteriors.

### 3.2 Entitats per a registrar els usuaris.

USUARI idusuari,nom,contrassenya,mail

En futures versions, és previsible comptar amb aquestes entitats per a implementar les històries d'usuari; i per registrar l'activitat dels usuaris: festivals on s'ha inscrit i que organitza, informació del perfil, missatges i opinions de cada concert... així que moltes entitats estaran relacionades a l'usuari.

És especialment important per garantir que els festivals creats no podran ser editats o eliminats per altres usuaris. Els altres usuaris en principi només és necessari que s'inscriguin o desinscriguin i es posin en contacte amb el organitzador amb les dades que ha facilitat.

Opcionalment, i més endavant; es podrien demanar dades personals com el gènere, la data de neixament, població, música preferida...; per animar els usuaris a inscriure's amb gent afí.

### 3.3 Entitats per a registrar assistents i organitzadors.

Un cop creada l'entitat usuari, es podran crear taules per a registrar els usuaris assistents

ASISTENT idfestival,idusuari,asistira(S/N),compliment(S/N)

ORGANITZADOR idfestival,idusuari,compliment(S/N)

El camp compliment serà un booleà per a registrar si algun usuari o organitzador no ha complert amb el seu compromís.

És a dir, si un usuari o l'organitzador no s'han presentat sense justificar-ho degudament amb antelació.

### 3.4 Entitats per a consultar grups de música.

Per a poder establir la cerca per grups de música haurem de crear la següent entitat Grup

GRUP id,nom,web

Com que la relació entre les entitats festival i grup és M:N, caldrà crear aquesta taula intermitja:

GRUPS\_FESTIVAL

id\_grup,id\_festival

### 3.5 Entitats per enviar missatges sobre els festivals.

Per a poder emmagatzemar missatges per preguntar dubtes de cada festival, necessitem la següent entitat:

MISSATGE

id,contingut,data

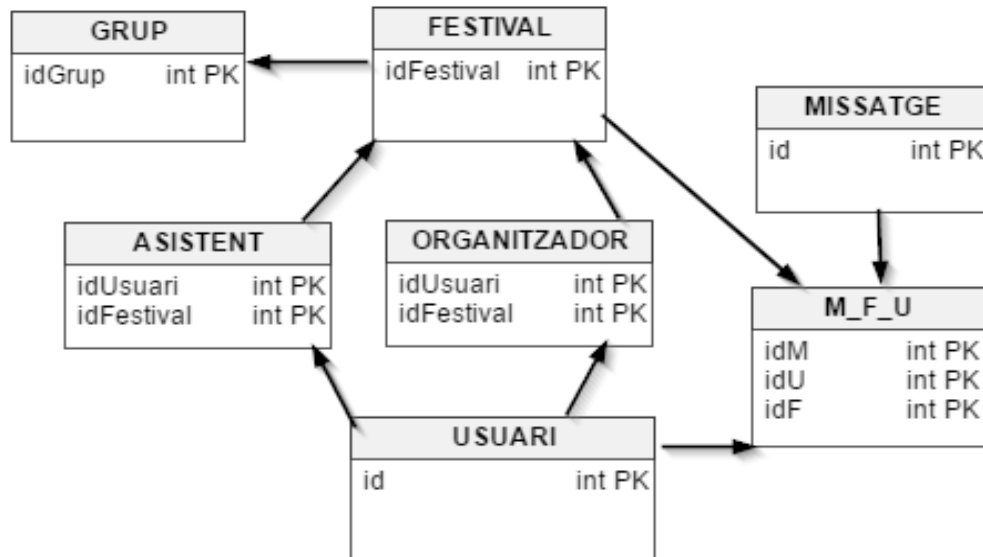
Com que la relació entre les entitats missatge,usuari i festival i grup és 1:M:N, caldrà crear una aquesta taula intermitja; similar a:

MISSATGES\_FESTIVAL

id\_mis,id\_fest,id\_usuari

### 3.6 Model relacional de les entitats.

El model relacional de totes les entitats definides en el llistat d'històries d'usuari complet quedarà de la següent manera.



Imatge 1 Model relacional de les entitats

## 4. Perfils i rols d'usuari.

### 4.1 Perfils d'usuari de l'aplicació.

Els perfils d'usuari que trobarem a la nostra aplicació, a grans trets, seran els següents:

**- Visitant observador.**

Aquest usuari només consultarà els festivals per curiositat, però no té previst inscriure's.

**- Usuari que s'inscriu a festivals.**

Aquest usuari, apart de consultar els festivals, voldrà apuntar-se en algún.

**- Usuari que organitza festivals.**

Aquest usuari crearà festivals, apart d'inscriure's en algún i consultar altres festivals.

Si hagués realitzat un estudi etnogràfic segurament hagués identificat més perfils d'usuari i més ben definits; però per a les primeres fase d'aquest projecte identificar aquests 3 perfils és suficient.

### 4.2 Rols d'usuari dins del sistema.

Per altra banda, a nivell més tècnic (rol dins l'aplicació), només tindrem un únic rol d'**usuari per ara serà el visitant**, que tindrà accés a totes les pàgines.

Per a les versions alpha i beta, només es podrà disposar del perfil d'usuari visitant, que podrà crear (organitzar) festivals i els que vulguin assistir la aplicació ja sigui vàlida;

Tanmateix, en futures versions serà important aconseguir l'autenticació i registre d'usuaris, per oferir un servei més fiable i professional. Per tant, podriem comptar amb el rol d'**usuari registrat**.

Fins i tot, si volguéssim fer negoci, tindriem l'usuari registrat gratuït i l'usuari registrat de pagament.

Està previst que només es puguin inscriure i organitzar festivals els usuaris registrats.

Tot i que inicialment pot semblar senzill de desenvolupar (de fet, probablement sigui una mica més senzill que amb altres plataformes com PHP,JavaEE o .NET), cal indicar que, desgraciadament, aquest objectiu supera d l'abast d'aquest projecte, tant a nivell de temps, d'investigació, com a nivell de consideracions legals.

Per tant, es tindrà en compte en el disseny però no es desenvoluparà l'autenticació.

## 5. Prototips i navegació.

### 5.1 Menú aplicació web.

La aplicació web tindrà un menú a la part superior de la pantalla on es podran accedir a totes les seccions:

PA1. Pantalla inici – Llistat i detall dels festivals

PA2. Pantalla afegir/editar festivals

PA3. Pantalla informació

PA4. Pantalla autenticació

PA5. Pantalla perfil

A la fase beta de l'aplicació només es podrà accedir a les 3 primeres.

### 5.2 PA1 - Pantalla Inici.

És la pàgina que es carregarà inicialment per defecte.

A l'esquerra mostrarà un llistat (en forma de taula) dels pròxims festivals ordenats per data d'inici. També hi haurà un filtre per mostrar el llistat per criteris definits.

De cada festival, apareixeran les dades més importants:

- Títol
- Ubicació
- Concerts
- Data d'inici
- Gratuït?

### Festivals JS - Llista de concerts

[Nou concert](#) [Com funciona?](#)

Població:

Només concerts gratuïts

Que comencin a partir del dia...

Títol	Ubicació	Concerts	Data inici	Gratuït?
Sant Jordi 2017 a Barcelona, Palau Sant Jordi	Barcelona	Txarango, Buhos, Roba Estesa	24/04/2017	✘
Concerts Girona Temps de Flors 2017	Girona	Oques Grasses, Sense Sal	15/05/2017	✘
Patum de Berga 2017	Berga	Brams, Ebri Knight	26/05/2017	✔
Festa Medieval Montblanc	Montblanc	Roba Estesa, Ebri Knight	04/05/2017	✔
Festes Majors de Maig de Lleida 2017	Lleida	Gossa Sorda, Oques Grasses, Amigues de les ciències	05/05/2017	✔

Imatge 2 PA1 - Pantalla d'inici

Clicant en qualsevol punt de la fila d'un concert, automàticament es mostrarà amb detall més dades sobre el concert, a la part dreta de la pantalla.

A la part baixa del formulari l'usuari pot prémer el botó de **Detall** per a visualitzar, editar o esborrar un festival creat.

Població:

Només concerts gratuïts

Que comencin a partir del dia...

Titul	Ubicacio	Concerts	Data inici	Gratuït?
Sant Jordi 2017 a Barcelona, Palau Sant Jordi	Barcelona	Txarango,Buhos,Roba Estesa	24/04/2017	✗
Concerts Girona Temps de Flors 2017	Girona	Oques Grasses,Sense Sal	15/05/2017	✗
Patum de Berga 2017	Berga	Brams,Ebri Knight	26/05/2017	✓
Festa Medieval Montblanc	Montblanc	Roba Estesa,Ebri Knight	04/05/2017	✓
Festes Majors de Maig de Lleida 2017	Lleida	Gossa Sorda, Oques Grasses,Amigues de les ciències	05/05/2017	✓

Sant Jordi 2017 a Barcelona, Palau Sant Jordi

Barcelona

Barcelona, Palau Sant Jordi

24/04/2017

Txarango,Buhos,Roba Estesa

Qui vol apuntar-se a passarsho de conya? Per ara som 4 inscrits; de 25-35 anys.

Si teniu mes dubtes escriviu al email [emaildeprova@email.com](mailto:emaildeprova@email.com) Si voleu apuntarvos escrivime un missatge al whatsapp i us agrego al grup. 653313567.

20€

Imatge 3 PA1 - Pantalla d'inici; festival seleccionat.

D'altra banda, també es pot filtrar la llista de festivals pels criteris de població.

Per exemple, podem filtrar els concerts que es faran a Barcelona a partir del dia 25/02/2017, no necessàriament gratuïts i ens apareixeran aquests 3, que estan al fitxer de dades:

Població:

Només concerts gratuïts

Que comencin a partir del dia...

Titul	Ubicacio	Concerts	Data inici	Gratuït?
Sant Jordi 2017 a Barcelona, Palau Sant Jordi	Barcelona	Txarango,Buhos,Roba Estesa	24/04/2017	✗
Concert Roba Estesa - Festes de Sagrada Familia 2017	Barcelona	Roba Estesa, Dj's	12/04/2017	✓
Concerts Nova Sala Barcelona 26-02-2016	Barcelona	Amigues de les ciències, Sense Pebre	26/02/2017	✗

Imatge 4 PA1 - Pantalla d'inici, filtre de festivals.

### 5.3 PA2 - Pantalla afegir/editar festivals.

Des d'aquesta pantalla l'usuari podrà crear una convocatòria a un nou festival. Podrà omplir diversos camps.

Aquesta pantalla es reutilitzarà amb la funcionalitat de consultar i editar un festival.

**Festivals JS - Afegir nou festivals**

[Lista concerts](#)   [Com funciona?](#)

---

**Títol**

Títol motivador per apuntar-se del festival

**Població**

Població del festival

**Ubicació**

Lloc del festival

**Concerts**

Llista de concerts

**Data inici**

24/06/2016

**Hora inici**

23:00

**Data fi**

25/06/2016

**Informació**

Inserir un enllaç amb informació oficial del festival/concert.

**Descripció**

Explica amb més detall on, quan i com trobar-te amb els assistents abans del festival.

**Contacte**

Si vols, indica el mitjà que et vagi millor per contactar amb els assistents (whatsapp, email, facebook...)

**Preu**

Preu

Crear

Imatge 5 PA2 - Pantalla de detall.

### 5.4 PA3 - Pantalla informació.

En aquesta pantalla hi haurà tota la informació del projecte festivals-js, i s'anirà actualitzant a mesura que evolucioni.

## Festivals JS - Com funciona festivals.js?

[Lista concerts](#)   [Nou concert](#)

---

Festivals.js és un projecte de final del Màster d'Enginyeria Informàtica de la UOC.

Consisteix en crear i desenvolupar l'arquitectura d'una aplicació web per trobar gent amb qui gaudir de concerts, amb Node.js

Es un projecte creat per Miquel Àngel Amorós Alberich.

Imatge 6 PA3 - Pantalla d'informació.



#### 5.5 PA4 - Pantalla autenticació.

En aquesta pantalla el visitant podrà registrar-se per primer cop o bé, si ja està registrat, entrar al seu compte; i també una secció on poder recuperar la contrasenya.

#### 5.6 PA5 - Pantalla perfil.

En aquesta pantalla l'usuari registrat podrà veure i editar les seves dades personals; i podrà veure els festivals que s'ha apuntat i està organitzant.

## 6. Arquitectura.

### 6.1 Descripció general

L'arquitectura que s'ha implantat és d'una aplicació client(web) servidor. Concretament, s'ha utilitzat un stack de tecnologies molt popular des de 2012: **MEAN** (MongoDB,Express,Angular,NodeJS).

Els objectes que representen les entitats estaran representats en format **JSON** (JavaScript Object Notation).

És un format estàndard, lleuger, fàcil d'escriure i d'interpretar pels ordinadors, basat en JavaScript.

Més informació a: <http://www.json.org/>

La transferència de dades entre el client(el navegador), el servidor web (Node.js i Express) i servidor de base de dades (MongoDB) es realitzarà amb objectes JSON.

Aquest és l'esquema que segueix l'arquitectura MEAN:



Imatge 7 Esquema arquitectura MEAN.

En el projecte, he aplicat totes les tecnologies descrites excepte la persistència de dades en MongoDB, perquè queda fora de l'abast del projecte a nivell de temps i utilitat de cara a una primera versió.

Els serveis web, en comptes d'obtenir les dades de una base de dades MongoDB les obtenen de fitxers de text en format JSON.

Un gran avantatge de totes les llibreries utilitzades és que són Open Source (per a poder consultar el codi font), són utilitzades en molts portals, i estan molt ben documentades.

## 6.2 Client

Inicialment estava previst crear la presentació únicament amb les tecnologies natives del navegador, els estàndards web **HTML**, **CSS** i **JavaScript**. Opcionalment, usar la llibreria JQuery per a realitzar crides AJAX (crides asíncrones) i algun full d'estils extern.

Tanmateix, cada vegada existeixen més recursos útils per a implementar aplicacions MEAN, i són més fàcils d'implementar i potents que els que es poden trobar sense utilitzar aquestes tecnologies. Per això, m'he decantat per usar el framework **Angular.js** de Google.

Angular facilita la creació de **SPA** (Single Page Applications), aplicacions web que no necessiten recarregar la pàgina, i en el cas d'aquest projecte ens ve bastant bé, ja que el sistema requereix recarregar sovint llistats de concerts extensos en una mateixa pàgina.

També he usat la llibreria d'estils **Bootstrap**, creada inicialment per Twitter; que facilita adaptar la presentació de les pàgines web al tamany de la pantalla del dispositiu (mòbil, ordinador...) i aporta icones i estils predeterminats bastant atractius.

## 6.3 Servidor

**Node.js** és un entorn de programació d'aplicacions multiplataforma de Codi Obert; a la capa del servidor. Utilitza el motor de Javascript de Google V8; permet manejar esdeveniments (com els navegadors), inclou en sistema d'integració de llibreries JavaScript **npm**, i permet usar **API's** no bloquejants (asíncrones). Per tot això, és ideal com a servidor d'aplicacions web escalables. És una alternativa al popular servidor Apache.

D'altra banda, la llibreria **Express.js** serveix per detectar i tractar peticions **HTTP** i **HTTPS** fàcilment (mètodes GET, POST, PUT, DELETE...)

Gràcies a aquesta facilitat, express.js pot servir perfectament per a programar API's **REST**.

En el cas que ens ocupa, s'han creat serveis web REST per a llistar, crear i actualitzar festivals; així com usuaris, assistents...

## 6.4 Base de dades

Tal i com he comentat prèviament, en el desenvolupament de les primeres versions del projecte no està previst implementar cap base de dades les dades persistents.

S'ha optat per la alternativa d'emmagatzemar les entitats en fitxers JSON; i usar la llibreria nativa de node.js FS(file system) per a tractar els fitxers.

Tanmateix, si el projecte creix, serà molt bona idea optar per MongoDB com a base de dades; ja que és ideal per a carregar entitats que ocupen força espai(com Festival), amb molts registres(com Usuaris) i és senzilla de connectar amb Node.js ja que utilitza fitxers JSON (en realitat es diuen BSON, però són molt similars).

Per tot això, és bona idea usar **MongoDB** en comptes d'una base de dades relacional lleugera com MySQL o SQLite.

Per a connectar la base de dades MongoDB amb el servidor Node.js seria necessària la llibreria mongoose.

Aquests articles són un bon punt de partida per dissenyar i codificar la futura solució:

<http://speakinbytes.com/2014/03/api-rest-con-node-js-express-mongodb-y-mongoose-js/>

<https://scotch.io/tutorials/build-a-restful-api-using-node-and-express-4>

## 6.5 Autenticació i gestió d'usuaris

Tal i com s'ha vist a l'apartat 2, disposar d'un motor de base de dades com MongoDB és un requisit previ necessari per a implementar les històries per a la autenticació i gestió d'usuaris.

Aquests articles són un bon punt de partida per dissenyar i codificar la futura solució:

<https://carlosazaustre.es/blog/autenticacion-con-token-en-node-js/>

<https://devdactic.com/restful-api-user-authentication-1/>

<https://stormpath.com/blog/tutorial-build-rest-api-mobile-apps-using-node-js>

<https://code.tutsplus.com/tutorials/token-based-authentication-with-angularjs-nodejs--cms-22543>

## 6.6 Infraestructura.

### 6.6.1 Entorn de desenvolupament local.

És força senzill instal·lar, programar i executar l'aplicació en un ordinador en qualsevol sistema operatiu d'escriptori; i que el servidor node.js estigui actiu a la màquina local (localhost).

Un dels avantatges de node.js és que no requereix un hardware gaire potent ni programari pesat. En el meu cas, he utilitzat:

- PC portàtil Intel Core i3 amb 4 Gb de RAM. S.O.**Windows 7**
- Programari **node.js**, inclou **NPM** per baixar llibreries de tercers.
- Llibreries **Express 4, Angular 1 i Bootstrap 3**.
- Editor de text **Sublime Text 3**
- **Libre Office 4** per crear la documentació
- **Navegadors** Mozilla Firefox i Google Chrome, últimes versions.
- Client **Git-Scm** per pujar el codi a Github i Heroku.
- Connexió a Internet

Per això, he decidit desenvolupar l'aplicació des del meu entorn local instal·lat.

### 6.6.2 Entorn de proves remot.

Com que crear un servidor remot casolà amb arrencat no és, en general, una bona idea; usaré un servidor remot extern per allotjar i arrencar l'aplicació. Ha de tenir un cost assequible per a particulars, ser compatible amb node.js i navegadors web, i que garanteixi disponibilitat del servei i mesures de seguretat.

A poder ser, que també sigui compatible amb MongoDB per un futur pròxim.

Després d'estudiar diverses opcions; s'ha triat, provisionalment, la versió gratuïta de heroku:

<https://dashboard.heroku.com/>

Promet instal·lar node.js i mantenir aplicacions arrencades, encara que estigui en mode d'hibernació.

La capacitat de processament és baixa per a un entorn productiu, però com a portald de proves és perfectament vàlid.

Apart de les prestacions limitades; té l'inconvenient que no permet allotjar cap base de dades. Amb fitxers de dades JSON cap problema, però si volem disposar de MongoDB hem de contractar un altre servei gratuït o de pagament.

La aplicació està pujada i disponible a la URL:

<https://festivals-js.herokuapp.com/>

El nom és provional.

Si el servidor detecta una petició HTTP quan està en estat d'hibernació, la primera petició triga en arrencar per despertar, però permet tenir operativa la aplicació web remotament, que és la funcionalitat més interessant

### 6.6.3 Control de versions de codi.

S'ha decidit usar un repositori públic de Github, basat en el sistema distribuït de control de versions Git; per tal de compartir el codi font, i especialment per portar el control de versions de l'aplicació.

La URL del repositori és:

<https://github.com/mamorosdev/festivals-js-alpha2>

### 6.6.4 Infraestructura en un futur.

En el futur, em plantejo seguir usant el compte de Github; per gestionar millor les versions del codi font; i a poder ser, implantar Integració Contínua.

Per abordar el projecte de manera més professional, contractar un domini i allotjament com el de digitalocean: <https://www.digitalocean.com/solutions/>

Pot allotjar node.js i la base de dades MongoDB; tot un preu de 5€ al mes.

La capacitat de processament, tot i ser superior a Heroku, no és gaire alta; però pel tamany de l'aplicació pot valer perfectament en un entorn de producció. Permet contractar prestacions superiors sota demanda (per 10,20 euros mes), que és una solució interessant.

## 7. Codi font.

### 7.1 Estructura d'arxius.

L'estructura d'arxius que s'ha seguit és la següent:

```
/node_modules
/app
  /model
    festival.js
    controller.js
    routes.js
  /angular
    core.js
    index.html
festivals.json
server.js
package.json
```

A continuació s'explicarà breument quin codi conté cada arxiu dividit per capes.

### 7.2 Model de dades de l'aplicació: fitxers JSON.

Tal i com hem comentat anteriorment, la persistència de les dades la realitzarem amb arxius de dades JSON (en comptes d'una base de dades). En el nostre cas disposem del fitxer **festivals.json**, amb 5 concerts d'exemple.

L'estructura de dades d'un festival la trobem al fitxer `/app/model/festival.js`

```
function Concert(json) {
  this.id=json.id;
  this.titol=json.titol;
  this.concerts=json.concerts;
  this.dataInici=json.dataInici;
  this.horaInici=json.horaInici;
  this.dataFi=json.dataFi;
  this.poblacio=json.poblacio;
  this.ubicacio=json.ubicacio;
  this.informacio=json.informacio;
  this.descripcio=json.descripcio;
  this.contacte=json.contacte;
  this.preu=json.preu;
}
```

### 7.3 Codi del nucli de node.js.

**Package.json** és un arxiu comú en tots els projectes node.js que usen el gestor de paquets npm, i conté informació rellevant del projecte i les seves dependències (per exemple, express.js).

En el nostre cas, totes les dependències s'han descarregat cap la carpeta **/node\_modules** amb la comanda:

```
npm install
```

Server.js és l'arxiu on s'inicia el servidor web. És molt senzill arrencar-lo amb la instrucció:

```
node server.js
```

El codi del fitxer server.js:

```
1 // Inicialització serveis web amb express
2 var express = require('express');
3 var app = express();
4
5 // Configuració mongoose per connectar amb mongodb
6 // var mongoose = require('mongoose');
7 // mongoose.connect('mongodb://localhost:27017/Festivalsis');
8
9 // Configuració servidor.
10 app.configure(function() {
11 // Permet l'accés al codi del client.
12 app.use(express.static(__dirname + '/angular'));
13 // Activació log
14 app.use(express.logger('dev'));
15 // Mètodes per facilitar
16 app.use(express.bodyParser());
17 app.use(express.methodOverride());
18 });
19
20 // Carregar els endpoints pels serveis web.
21 require('./app/routes.js')(app);
22
23 // Inicialitzar el servidor web node.js al port indicat
24 var port = process.env.PORT || 8080;
25 var domain = process.env.DOMAIN || '127.0.0.1';
26 // Inicialitzar el servidor web node.js al port indicat
27 app.listen(port, domain);
28 console.log("Servidor web iniciat pel domini i port " + domain + ":" + port);
```

Més informació sobre node.js i npm al manual d'instal·lació (apartat 8 de la memòria), i a la pàgina oficial:

<https://nodejs.org/es/>

### 7.4 Codi de l'API REST amb els serveis web de la capa servidor - carpeta App.

El fitxer de **routes.js** recull les crides realitzades al client (navegador dels usuaris) i les encamina als mètodes del servidor indicats.

```
var Controller = require('./controller');
module.exports = function(app) {
  // Obtenir tots els Festivals
```

```

    app.get('/api/festivals', Controller.getFestivals);

    // Crear un nou Festival
    app.post('/api/festivals', Controller.saveFestival);
    // Modificar les dades d'un Festival
    app.put('/api/festivals/:festivals_id',
Controller.updateFestival);

    // Carreguem la vista des del domini.
    app.get('*', function(req, res) {
        res.sendFile('./angular/index.html');
    });
};

```

El fitxer **controller.js** conté la implementació dels mètodes que accediran al model de dades. En el nostre cas, un extracte del fitxer és així:

```

// var Persona = require('./modelo/persona');
var fs = require("fs");
var festivalsData = fs.readFileSync('festivals.json');
var festivalsJSON = JSON.parse(festivalsData);

// Obté tots els festivals de la base de dades
exports.getFestivals = function (req, res){
    // console.log(festivalsJSON);
    res.json(festivalsJSON);
}

// Guarda un festival a base de dades
exports.saveFestival = function(req, res) {
    console.log("saveFestival="+req.body.titol);
    var nouFestival = req.body;
    festivalsJSON.push(nouFestival);
    var festivalsString = JSON.stringify(festivalsJSON);
    fs.writeFileSync('./festivals.json',
festivalsString);
    res.json(festivalsJSON);
}

```

## 7.5 Codi de la capa client - carpeta Angular.

A **index.html** podem trobar tot el codi en HTML5 de la pàgina; i les invocacions a la llibreria Angular.JS, al full d'estils Bootstrap.css i al codi que s'ha escrit per a tractar les dades a **core.js**

La peculiaritat d'Angular.js és que el podem integrar a les etiquetes d'HTML5 fàcilment; que es transformaran en el codi HTML, CSS i JS necessari.

Per exemple, en aquesta etiqueta:

```
<body ng-controller="mainController">
```

S'invocaran totes les variables i funcions disponibles a l'objecte «mainController» que es troba a core.js

Un altre exemple és la etiqueta `{{titolVista}}`

```
<h2> Festivals JS - {{ titolVista }} </h2>
```



Que es transformarà en el text definit al camp `titolVista` de l'objecte `mainController`; és a dir, en un text com el següent:

```
<h2> Festivals JS - Llista de concerts </h2>
```

D'altra banda, a continuació es mostra un extracte del codi del fitxer `core.js`

```
function mainController($scope, $http) {
    // Variables de navegació
    $scope.vista = "llista";
    $scope.titolVista = "Llista de concerts";

    // Variables amb dades d'un festival.
    $scope.newFestival = {};
    $scope.detailFestival = {};
    $scope.saveOrUpdate = "save";

    // Variables amb llistes de festivals.
    $scope.festivalsInici = {};
    $scope.festivals = {};

    $scope.selected = false;

    // Mètode per carregar tots els festivals de la base de
    // dades a l'iniciar la sessió.

    $http.get('/api/festivals').success(function(data) {
        $scope.festivals = data;
        $scope.festivalsInici = data;
    })
    .error(function(data) {
        console.log('Error: ' + data);
    });
}
```

Com podem veure, amb l'objecte **\$scope** es defineixen les variables que podrà agafar i transformar la llibreria Angular automàticament (pot ser un element(o un array d'elements) de tipus String, numèric, booleà, objecte/s JSON...)

També és important l'objecte `$http`, que ens permet invocar peticions HTTP al servidor (en el nostre cas una `get`) i tractar la resposta amb els mètodes `success`(si tot ha anat bé) o `error`.

Aquesta invocació anirà a parar als fitxers del servidor que hem vist abans: `routes.js` i `controller.js`

## 8. Tests.

### 8.1 Tests d'acceptació.

Els tests d'acceptació són consisteixen en verificar que es compleixen els criteris d'acceptació definits a l'apartat 2.3

En aquest cas, s'ha provat criteri per criteri que s'han complert, tant en el servidor local com el remot de desenvolupament.

### 8.2 Tests de regressió.

A continuació, es proven els tests de regressió definits prèviament. És a dir, que les funcionalitats que ja funcionaven en la versió prèvia segueixen funcionant de manera consistent.

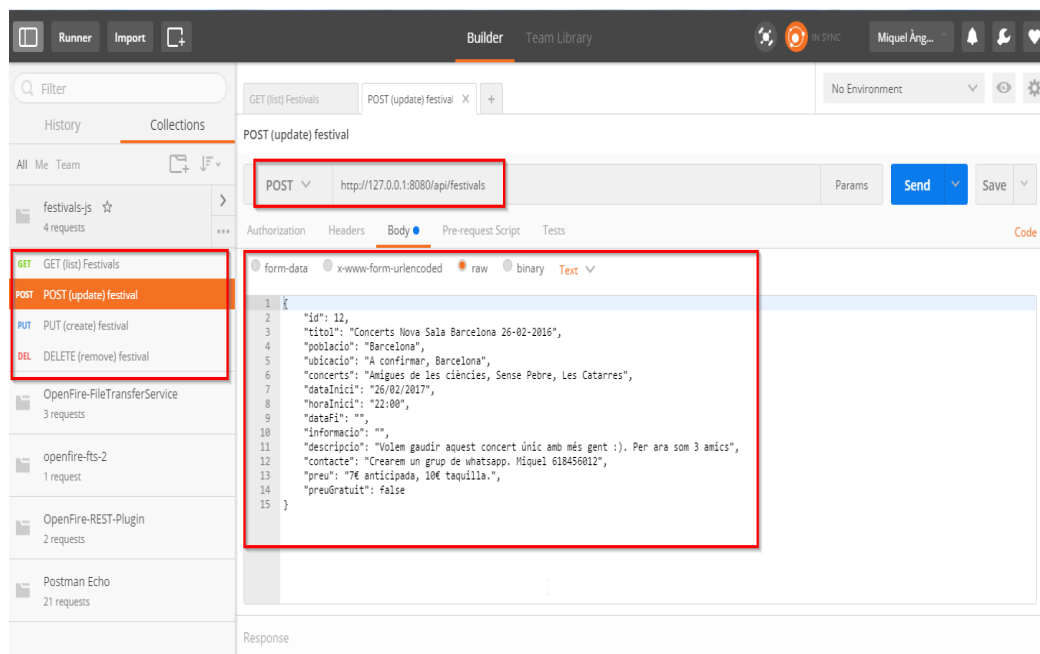
En el nostre cas, s'ha provat que els testos de regressió respecte els llistats de festivals, definits a l'apartat 2.3, funcionen correctament.

### 8.3 Tests d'integració de la API REST.

A partir d'aquesta versió beta he considerat útil crear una bateria de tests per provar les crides als web services de la API REST de festivals que s'ha creat.

El software que s'ha escollit per a provar els web services és Postman, creat per Google. Una altra opció vàlida seria SoapUI però s'ha descartat perquè no és tan lleugera i intuïtiva com Postman.

Amb Postman, podem crear crides a web services REST fàcilment, a qualsevol domini HTTP/S i mètodes GET,POST,PUT... i organitzar-les en col·leccions. Les que s'han creat tenen el següent aspecte:



#### 8.4 Tests que falten implantar: unitaris.

Donada la simplicitat del codi del servidor, s'ha prescindit de tests unitaris automàtics provisionalment.

Tanmateix, caldria incorporar-los en futures versions, perquè són bàsics per detectar errors durant el desenvolupament.

Hi ha moltes llibreries en Javascript per a crear tests unitaris al servidor(Jasmine,Karma,Chai...); encara cal decidir les més adients i escriure els tests.

D'altra banda, tot i que hi ha fonts que recomanen la creació de tests d'interfície d'usuari; no considero que siguin tant importants; especialment en fases inicials com la actual, ja que si canvia la interfície gràfica cal canviar els tests generats i es requereix molt temps. En cas que es canviï d'opinió, la opció més madura per crear els tests és Selenium RC.

## 9. Seguretat.

Durant el disseny i desenvolupament he tingut molt en compte seguir bones pràctiques per a oferir les garanties fonamentals de la seguretat de la informació del sistema: confidencialitat, disponibilitat i integritat.

En futures versions, també caldrà oferir les altres dues característiques de seguretat: autenticació i traçabilitat.

### 9.1 Mesures de seguretat implantades.

En concret, les mesures de seguretat que s'han implantat són les següents:

#### **1. Creació de testos del software.**

Tots els testos que s'han implementat (explicats a l'apartat anterior, el 8) protegeixen la informació d'errors deguts a talls de connexió, a errors d'integritat de les dades, a un ús negligent o malintencionat... entre d'altres.

#### **2. Validacions de la informació tant al codi del client(Angular) com al del servidor(Express)**

Ara per ara, com a mesura de seguretat, he implantat validacions de tots els camps; tant a nivell de client com a nivell de servidor.

Aquesta mesura és imprescindible en qualsevol aplicació i/o servei web; per evitar que s'injecti codi Javascript i objectes JSON amb contingut maliciós.

#### **3. Ús de tecnologies de servidor segures.**

Express.js, la llibreria que manega les peticions HTTP, es tracta d'una llibreria segura i fiable que prevé d'altres atacs al contingut del servidor, i traça qualsevol moviment; al igual que el servidor node.js

#### **4. Definició de Web Services.**

Crear una API de Web Services, ja siguin SOAP o REST(com el nostre cas) i limitar l'accés a les dades a aquests mètodes, ha facilitat molt la implementació de validacions i controls de seguretat al costat del servidor.

### 9.2 Mesures de seguretat futures.

Les mesures de seguretat més importants a implantar en el futur seran:

1. Gestionar les dades en un motor de base de dades. Tal i com he indicat a l'apartat d'arquitectura, MongoDB pot ser una bona solució, apart de ser eficient pels programadors i usuaris.
2. Implantar el protocol HTTPS.
3. Ofuscar el codi Javascript de l'aplicació.

# 10. Instal·lació i publicació.

## 10.1 Instal·lació local.

Per a instal·lar i executar el servidor i l'aplicació localment, a partir del codi font, proporcionat cal seguir els següents passos:

1. Instal·lar node.js des de la pàgina oficial. La recomanada per la majoria d'usuaris és suficient.

<https://nodejs.org/en/>

2. Un cop instal·lat, accedir a la consola de Node.js, anomenada «Node.js command prompt» i accedir a la carpeta on es troba el projecte festivals-js

3. Des de la mateixa carpeta, instal·lar les dependències (llibreries externes; pex express) amb la comanda:

```
npm install
```

4. Després, executar el servidor node.js amb la comanda:

```
node server.js
```

Hauria de sortir per la consola el missatge:

Servidor iniciat pel domini i port 127.0.0.1:8080

5. Finalment, obrir l'adreça:

<http://localhost:8080/>

Per parar el servidor simplement pitjar Ctrl.+C dues vegades.

I per reiniciar-lo, introduir de nou:

```
node server.js
```

## 10.2 Publicació codi font a Github.

Github és un servei web molt popular creat per Linus Torvalds que proporciona el sistema de control de versions distribuït Git de manera gratuïta a través de la creació d'un compte a la web:

<https://github.com/join?source=login>

Un cop donats d'alta, només cal seguir les instruccions per a crear un nou repositori i pujar el codi font des de la mateixa web:

<https://help.github.com/articles/create-a-repo/>

### 10.3 Publicació remota aplicació web a Heroku.

Per a instal·lar la aplicació en l'entorn remot de proves he seguit pas per pas les indicacions que recomana el proveïdor, Heroku:

<https://devcenter.heroku.com/articles/getting-started-with-nodejs#scale-the-app>

Igualment, cal exposar un petit resum d'aquestes instruccions:

1. Com és habitual, per tal d'optar al compte gratuït (Free). Amb 512Mb de RAM i un espai web, cal registrar-se al web oficial:

<https://www.heroku.com/>

2. Un cop registrats, seguir pas per pas les indicacions que recomanaden per a aplicacions en node.js (també permet ruby, java, php...):

<https://devcenter.heroku.com/articles/getting-started-with-nodejs>

Concretament, he seguit els passos des de «Introduction» fins a «Push local Changes»

3. Disposar d'un repositori a Github facilita encara més la publicació:

<https://devcenter.heroku.com/articles/getting-started-with-nodejs#prepare-the-app>

4. Després de seguir els passos, tindrem la aplicació arrencada i funcionant. Actualment, la aplicació publicada està disponible a la URL:

<https://festivals-js.herokuapp.com/>

5. Operacions comuns.

Consultar els logs:

<https://devcenter.heroku.com/articles/getting-started-with-nodejs#view-logs>

Iniciar i parar:

<https://devcenter.heroku.com/articles/getting-started-with-nodejs#scale-the-app>

# 11. Conclusions

Considero que he superat adequadament l'objectiu principal que m'havia proposat l'agost del 2016, que és el de desenvolupar una aplicació d'exemple en la arquitectura node.js, OpenSource i basada en JavaScript tant en el servidor com en el client. A més a més, he gaudit molt amb el procés.

## 11.1 Encerts i fites assolides.

**He pogut consolidar les meves habilitats de programació, i he adquirit experiència en:**

- **Programar en JavaScript tant a la capa del client com a la de servidor**, o dit més coloquialment, per darrera i per davant; que ha sigut més senzill del que esperava. La manera de treballar costa una mica d'entendre al principi però val molt la pena estudiar-la, perquè **es guanya temps i confiança al programar amb Node i Express**.
- També estic molt satisfet perquè he aconseguit crear serveis web amb facilitat, i ha estat còmode **l'ús d'objectes JSON**; que són dos aspectes molt útils que he aconseguit millorar com a professional gràcies al projecte.
- He quedat gratament sorprès amb la facilitat que he pogut crear un prototip d'aplicació amb **Angular 1 i Bootstrap 3**. A nivell de crear coses més complexes (llista filtrable i paginable) també he pogut solucionar-ho a mida; i segurament que la comunitat OpenSource té solucions encara més potents (pex, Polymer, Angular 2...)
- M'ha costat decidir-me per un proveïdor per publicar l'aplicació, però un cop après em resulta molt còmode el desenvolupament local, **la publicació a Heroku i l'ús de GitHub** per desplegar millores i correccions.
- El funcionament de l'aplicació **compleix els criteris d'acceptació** que havia definit, ja que es poden cercar, crear, eliminar i editar festivals amb facilitat. La major part de les històries d'usuari han estat realistes.

**Les assignatures que han estat més útils per completar el projecte son:**

- Desenvolupament d'Aplicacions per a Dispositius Mòbils
- Sistemes Distribuïts a Gran Escala
- Tècniques Avançades d'Enginyeria del Programari.

**I en alguns punts de la memòria també m'han anat bé els apunts de:**

- Enginyeria de la usabilitat
- Seguretat de la informació
- Gestió avançada de projectes

## 11.2 Punts de millora i reptes pendents.

### Tanmateix, els punts que podrien haver anat millor són:

- La **previsió de temps** a dedicar i compliment de dates no ha estat adequada. Tant en el disseny com en el desenvolupament he acabat tot a última hora. Tot i la passió que sento per la programació web porto malament dedicar-hi temps addicional. No tant com amb les assignatures teòriques, però necessito més disciplina. Sense aquesta disciplina, no he pogut seguir cap metodologia àgil, tot i entendre perfectament els seus principis.
- Tot i que ja ho havia previst a temps, em sap **molt greu no haver implantat la base de dades MongoDB i un sistema per gestionar els usuaris**. En el primer cas ha estat per falta de temps; i el segon per falta d'aptituds (que puc adquirir amb estudi i pràctica) ja que ni tan sols en Java (amb 5 anys d'experiència) he tingut ocasió de crear/adaptar un mòdul d'autenticació. Crec que és més greu el tema de la base de dades, però ho he tingut present en tot el projecte.
- En menor mesura, també he trobat en falta dedicar més temps al **disseny de l'interfície**, que no és gaire atractiu; tant a nivell gràfic(icones, imatges, colors...) com en arquitectura (usar el nou Angular 2, optimitzar el temps de càrrega...)
- M'ha quedat pendent **crear més web services**; mètodes més avançats i altres entitats; utilitzar una arquitectura de microserveis (pex un microservei per entitat); és a dir, treure més partit al valor afegit de node.js respecte els servidors d'aplicacions tradicionals.

En definitiva, estic **molt satisfet** d'haver realitzat aquest projecte; i puc assegurar que **hi haurà pròximes versions operatives (beta-2 i beta-3)**, i que quan acabi aquest PFM seguiré fent créixer aquest projecte, i probablement d'altres; amb aquesta arquitectura MEAN. Segur que sortirà bé si hi puc **dedicar més hores**, que és el punt prioritari que cal millorar.

En els punts 6.4 i 6.5 exposo una **aproximació a nivell d'arquitectura** de la futura solució.

Node.js, Express, Bootstrap, Angular... és **el programari basat en JavaScript ha superat les bones expectatives** que tenia; i em queden força ganes de descobrir-lo amb més profunditat, **professionalment**.



## 12. Glossari

Definició dels termes i acrònims més rellevants utilitzats dins la Memòria.

Angular.js	Framework que facilita la creació d'aplicacions HTML5.
API	Applications Programming Interface
Bootstrap	Llibreria que facilita l'estil de les pàgines, independent del tamany de la pantalla del dispositiu (disseny responsive)
BD/DB	Base de dades/Data base
CSS	Cascade Style Sheets, estil pàgines web.
Express	Llibreria que inclou Node.js que <u>facilita</u> la creació de serveis web.
HTML	Hiper Text Markup Language, estructura pàgines web.
HTTP	Hypertext Transfer Protocol.
HTTPS	Secure Hypertext Transfer Protocol
JS	Java Script, llenguatge de programació clients i servidors web.
JSON	JavaScript Object Notation
Node.js	La tecnologia que permet executar JavaScript en servidors.
NoSQL	Not Only SQL. Bases de dades que no usen el llenguatge i/o model de dades relacional, basat en SQL.
NPM	Node Package Manager; inclòs a Node.js per instal·lar llibreries com Express.
REST	Representational State Transfer, tots els web services creats són API's REST.
Serveis Web	Web services.
SPA	Single Page Application

# 13. Bibliografía

## 13.1 Llibres digitals.

Llibres digitals:

Eloquent Javascript

Marijn Haverbeke

2n Edition, 2015.

<http://eloquentjavascript.net/>

Arquitecturas web y Angular.js

Cecilio Álvarez.

1a Edición, 2014.

<http://www.arquitecturajava.com/nuevo-libro-arquitecturas-web-y-angular-js/>

Introducció a Node.js a través de Koans.

Arturo Muñoz de la Torre

Edición online, última actualización en 2016.

<http://nodejskoans.com/>

## 13.2 Pàgines web.

Node.js docs, agost-deseembre 2016

<https://nodejs.org/en/docs/>

Guia Express.js, setembre-deseembre 2016

<http://expressjs.com/en/guide/routing.html>

Tutorial Angular 1, setembre 2016 - gener 2017

<https://docs.angularjs.org/tutorial/>

Getting started Bootstrap.js, octubre 2016 - gener 2017

<http://getbootstrap.com/getting-started/>

Documentación de MongoDB, Novembre-deseembre 2016

<https://docs.mongodb.com/manual/>

Getting started on Heroku with Node.js, novembre 2016 - gener 2017.

<https://devcenter.heroku.com/articles/getting-started-with-nodejs>

HTML Demos and Tutorials Agost - Deseembre 2016.

<http://www.tutorialspoint.com/html5/>

## 13.3 Articles web (blogs).

Tipos de pruebas, creat el 04/07/2014.

<http://www.javiergarzas.com/2014/07/tipos-de-pruebas-10-min.html>

Tutorial per a principiants en node.js; novembre-Deseembre 2016, 2016.

<http://www.nodebeginner.org/index-es.html>

Tutorial ejemplo aplicación web con Angular.js y API REST con Node.13/01/14

<https://carlosaustre.es/blog/tutorial-ejemplo-de-aplicacion-web-con-angular-js-y-api-rest-con-node/>

MEAN Mongo Express Angular Node, desarrollo Full Stack Javascript.09/07/14

<http://jarroba.com/mean-mongo-express-angular-node-desarrollo-full-stack-javascript-parte-i/>