

Kpax Server

Projecte KPax
Kpax Server 2



Miquel A. Muntaner

Treball Final de Màster
19 de juny de 2016



Migració de kPaxServer a NodeJs amb MongoDB

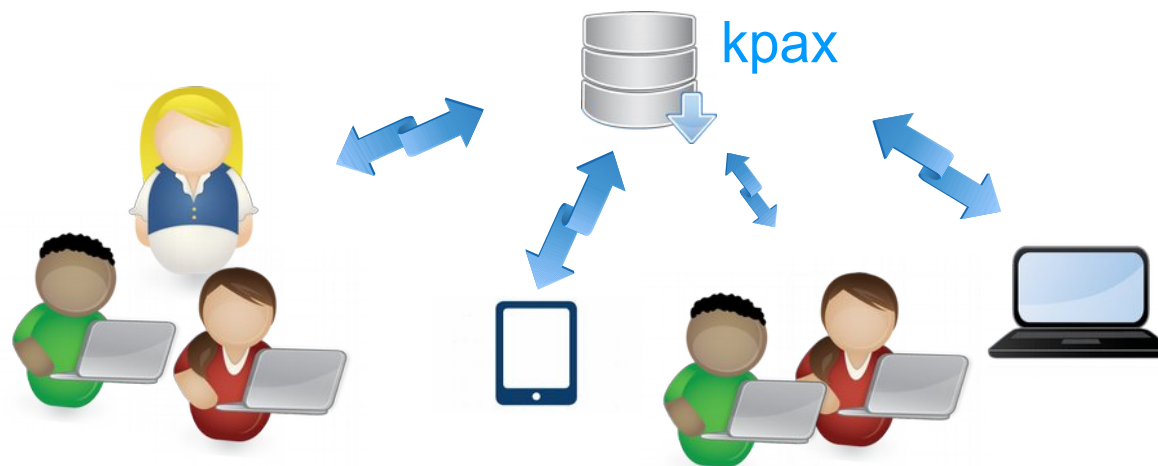


Introducció 1



Kpax neix de l'esforç de conciliar

- Les xarxes socials
- Els jocs seriosos com eina didàctica
- Aprofitar disponibilitat de múltiples dispositius per connectar-se





Versió 1 de Kpax està format per

– Back-End - KpaxServer :

- Capa de Negoci: Servidor API-REST en Java
- Capa de persistència: MySQL

– Front-End – Kpax-FrontEnd:

- Capa de presentació basat en **ELGG** sobre PHP + MySQL + Plugins amb un servidor Apache



Introducció 2



KpaxServer :

- Gestiona usuaris, jocs, partides, desenvolupadors
- Servidor API REST amb operacions **CRUD** sobre la base de dades.



Front-End – Kpax-FrontEnd:

- Suporta la xarxa social
- Comunica amb el nucli a través de cridades a serveis web



Nomes part està implementada.

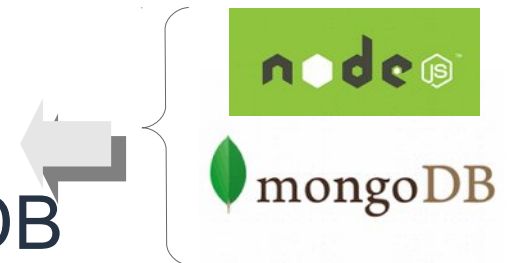
Objectius 1



Projecte Global: doble Objectiu :

Migració del Servidor

- Reescriure el servidor en NodeJS
- Migrar a BDD no Relacional: MongoDB



Actualitzar ELGG.

- Actualitzar la versió d'ELGG
- Adaptar els plugins a la nova versió ELGG



Objectius 2



Projecte específic a desenvolupar al TFM :

Migració del Servidor

- Reescriure el servidor en NodeJS
- Migrar a BDD no Relacional: MongoDB



Actualitzar ELGG.



- Actualitzar la versió d'ELGG
- Adaptar els plugins a la nova versió ELGG



Explicació del projecte 1



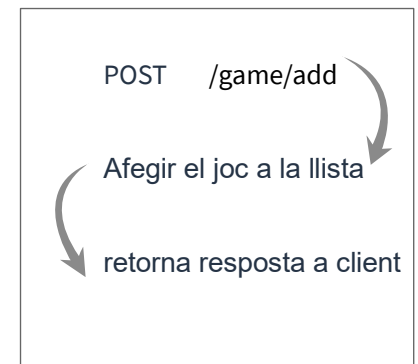
El Nucli del servidor

- Discontinuitat amb l'antic core (Java + MySQL).
- Reescrit completament des de zero.
 - **Node.js**  · servidor orientat a esdeveniments.
 - **MongoDB**  · model BDD no relacional.
- Migració transparent a la resta de kPax
 - Es mantenen URI's de crida, igualment parametritzades.
 - Es serveix la mateixa informació en el mantenint format de sortida.

Explicació del projecte 2

El Nucli del servidor ... +

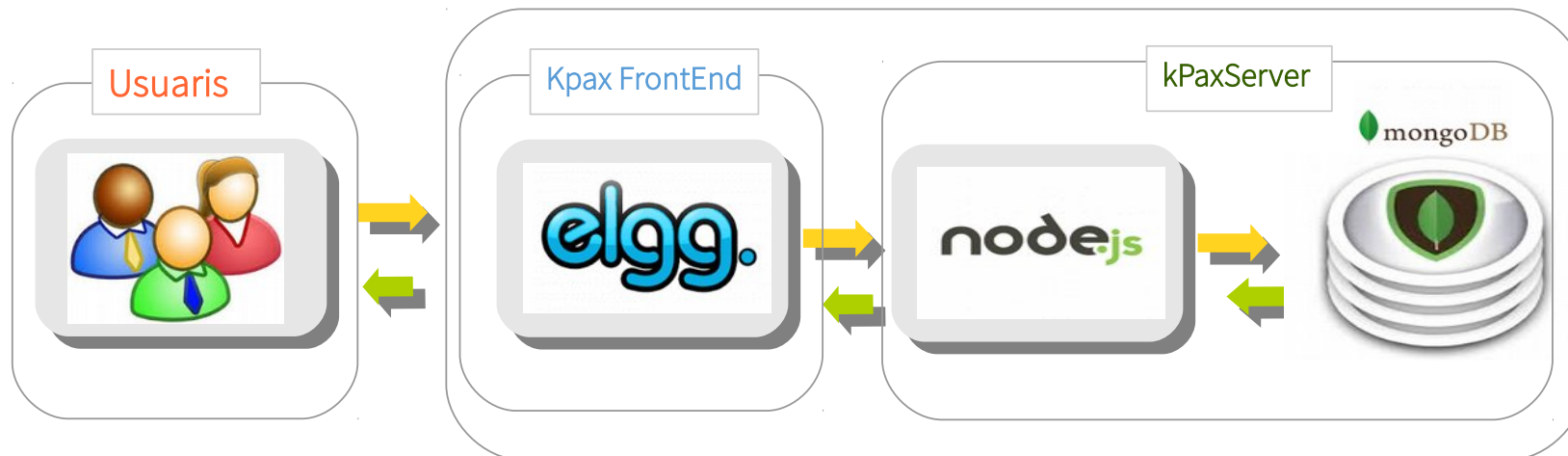
- Basat en serveis que proporciona la API REST del core.
- Anàlisi de Kpax.1 reutilitzable per Kpax.2
- Migració del Server :
 - Identificar peticions al servidor
 - Identificar resposta esperada
 - Identificar el format per servir la resposta



El projecte 1

Contrucció d'un servidor REST en NodeJs i MongoDB

- No hi ha interfaces d'usuari.
- KpaxServer es una API destinada a donar servei a peticions externes.
- No interactua amb usuaris





Mecanisme de les peticions REST

- Cada recurs s'identifica per una URI única.
- El mètode HTTP : GET, POST, DELETE (~CRUD)
- El servidor identifica la petició :
 - Realitza accions amb la BDD
 - Envia resposta al client del Codi d'Estat HTTP
 - Envia al client Dades JSON, quan s'ha sol·licitat amb GET.





Exemple de Petició: creació de nou usuari

- **Petició:**

```
$ curl 'http://localhost:3000/user/' --data "name=user1 Uoc01 &login=user01@uoc.edu" -v
```

- URI: /user Mètode : **POST**

- Paràmetres: 'name': 'user1' , 'login': 'user1@uoc.edu'

- **Resposta**

- Status code: 200

- Creació de usuari
a la BDD:

```
Time: 1466276196773  
POST /user/ 200 141.685 ms - 172
```

```
> db.users.find().pretty()  
{  
  "_id" : ObjectId("576599647a821fda07dd052a"),  
  "login" : "user01@uoc.edu",  
  "name" : "user1 Uoc01",  
  "created_at" : ISODate("2016-06-18T18:56:36.779Z"),  
  "updated_at" : ISODate("2016-06-18T18:56:36.779Z"),  
  "status" : 1  
}
```

El projecte 4



Exemple de funcionament del servidor

Petició REST al servidor

```
$ curl 'http://localhost:3000/user/' --data "name=user1 Uoc01 &login=user01@uoc.edu" -v
```

Resposta del servidor



```
miquel@ubuntu1510:~/src/node/kPAX2_server/server/routes$ curl 'http://localhost:3000/user/' --data "name=user1 Uoc01 &login=user01@uoc.edu" -v
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 3000 (#0)
> POST /user/ HTTP/1.1
> Host: localhost:3000
> User-Agent: curl/7.43.0
> Accept: */*
> Content-Length: 38
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 38 out of 38 bytes
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept
< X-Content-Type-Options: nosniff
< Content-Type: application/json; charset=utf-8
< Content-Length: 172
< ETag: W/"ac-A7T+Sb1pHMtBXKT9a0XC4g"
< Date: Sat, 18 Jun 2016 18:56:36 GMT
< Connection: keep-alive
* Connection #0 to host localhost left intact
{"login":"user01@uoc.edu","name":"user1 Uoc01 ","created_at":"2016-06-18T18:56:36.779Z","updated_at":"2016-06-18T18:56:36.779Z","status":1,"_id":"576599647a821fda07dd052a"}miquel@ubuntu1510:~/src/node/kPAX2_server/server/routes$
```



```
> db.users.find().pretty()
{
  "_id" : ObjectId("576599647a821fda07dd052a"),
  "login" : "user01@uoc.edu",
  "name" : "user1 Uoc01 ",
  "created_at" : ISODate("2016-06-18T18:56:36.779Z"),
  "updated_at" : ISODate("2016-06-18T18:56:36.779Z"),
  "status" : 1
}
```



Endpoints usuari :

- POST /user Creació d'usuari
- GET /user/listall Llista de tots els usuaris
- GET /user/list?q={'JSON param'} Consulta usuari
- GET /user/:user Info d'un usuari :user
- POST /user/del Esborrat d'un usuari



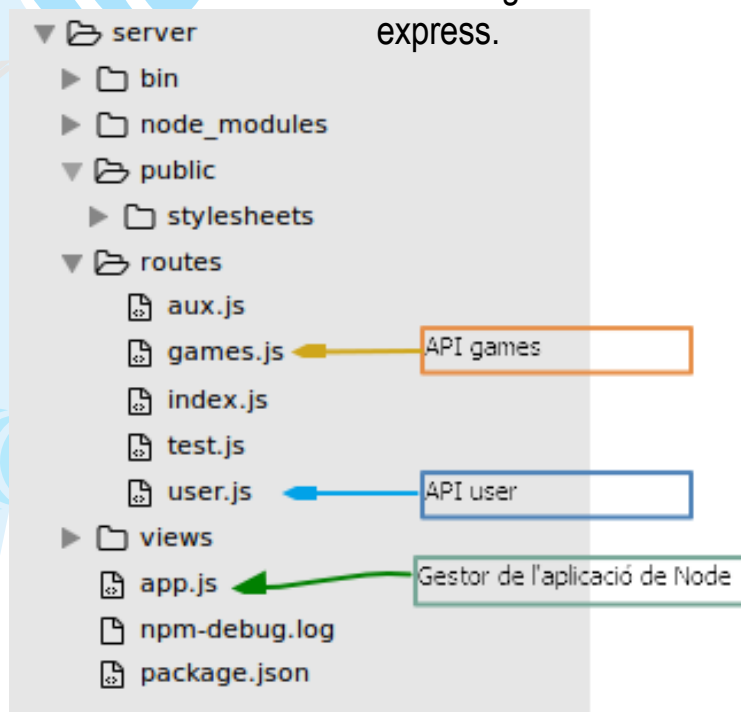
Endpoints jocs

- POST /game Creació d'un joc
- GET /game/listall Llista de tots els jocs
- GET /game/list?q={'JSON param'} Consulta joc
- GET /game/:game Info del joc :game
- POST /game/del Esborrat d'un joc
- POST /game/:game/like Marca el joc :game amb 'like'
- POST /game/:game/unlike Esborra marca de 'like' si la tenia

El projecte 7

Estructura del servidor

Típica estructura del servidor generat amb express.



Estructura del Fitxer de configuració del servidor.

Important : dependències del servidor.

```
package.json
{
  "name": "KPAX2_server",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "body-parser": "^1.13.3",
    "cookie-parser": "^1.3.5",
    "debug": "^2.2.0",
    "express": "~4.13.1",
    "jade": "^1.11.0",
    "mongodb": "^2.1.18",
    "morgan": "^1.6.1",
    "pug": "*",
    "serve-favicon": "^2.3.0"
  }
}
```




Canvi d'arquitectura del servidor

Nou Servidor → modern, flexible i d'altres prestacions.

- **Node.js** → Minimitza el temps de sistema i maximitza escalabilitat.
 - Dissenyat per escriure servidors i aplicacions WEB escalables.
 - Javascript.
 - Orientat a esdeveniments
 - Entrada/Sortida asíncrona.
- **MongoDB.**
 - Informació organitzada en documents
 - Format similar a JSON
 - Esquema Lliure, No SQL



Migració a Node.js + MongoDB

- Codificació del servidor des de zero.
- Més eficient en la seva execució.
- Més flexible en la programació dels nous mòduls.
- Entorn ideal per servidor com kpaxServer.
- Canvis interns del servidor dràstics.
- Externament les cridades REST al servidor han de mantenir la mateixa estructura tant d'entrada com de sortida



Demostració funcionament del servidor