

Internet Game Data Base



Felipe Murillo Serrano

Rafael Morena Orellana

Mario Arranz Águeda

2º Desarrollo de Aplicaciones Web

I.E.S. Gaspar Melchor de Jovellanos

M
E
A
N



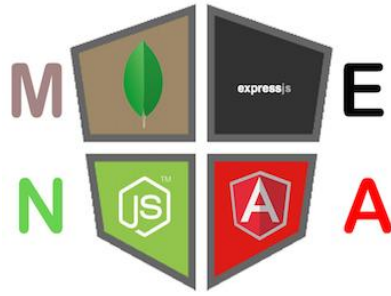
Índice

Tabla de contenido

Índice	2
Introducción	3
Instalación de Mean JS	6
Poniendo en marcha la aplicación en Linux	9
GitHub	14
Casos de uso	15
Plantillas de casos de uso.....	17
Diseño de Navegación	26
Diseño de Roles en la Web	27
Diseño del Desarrollo	28
Manual de Usuario.....	33
Bibliografía	36

Introducción

En los últimos años se han puesto muy de moda tecnologías como MongoDB, Node.js y AngularJS, lo que ha hecho que haya surgido una nueva corriente para la construcción de aplicaciones web rápidas, robustas y de fácil mantenimiento usando las tecnologías de MongoDB, Express, AngularJS y Node.js. El uso de estas 4 tecnologías juntas conforman el llamado "Stack MEAN" (MEAN = **M**ongo-**E**xpress-**A**ngular-**N**ode) que tienen como nexo común la utilización del mismo lenguaje de programación (JavaScript), que por otro lado no es un lenguaje muy complejo. Por tanto con estas 4 tecnologías tenemos todo lo necesario para desarrollar aplicaciones web: Frontend, Backend y Base de Datos.



A continuación explicaremos cada una de estas tecnologías:

MongoDB



Es un sistema de base de datos documental no relacional de código abierto. Este sistema de base de datos, guarda los datos en documentos en formato JSON (o BSON). Cabe destacar su flexibilidad y escalabilidad, lo que ayuda al desarrollo ágil de proyectos software. Son muchos los lenguajes que pueden trabajar con MongoDB (Java, PHP, Python, etc.) y por supuesto NodeJs es uno de ellos.

Express



Express es un framework realizado en NodeJs muy ligero y flexible que proporciona un conjunto muy robusto de facilidades para crear fácilmente servidores web y recibir peticiones HTTP. Permite desarrollar API REST de forma muy rápida.

Angular



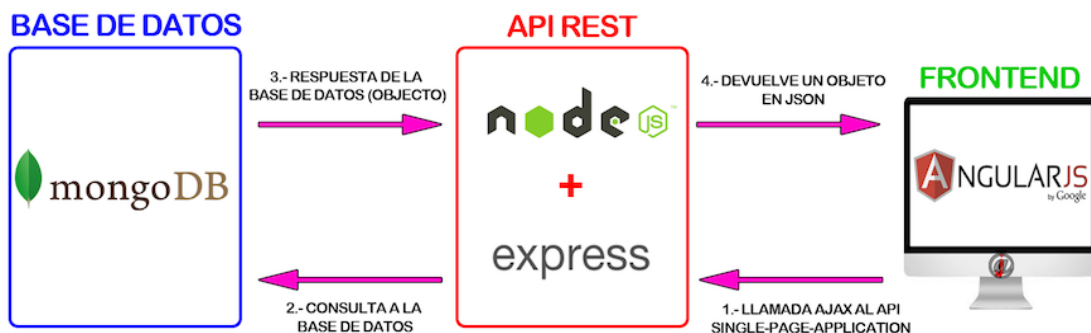
AngularJS es un framework para Frontend desarrollado por Google) que sigue el patrón MVC (Modelo-Vista-Controlador). AngularJS permite crear lo que se llama Single-Page-Applications, que son aplicaciones web que no necesitan recargar la página. Para terminar decir que AngularJS es un framework muy potente y que facilita muchísimo la labor de los desarrolladores de Frontend.

Node



NodeJS es un entorno de programación en JavaScript en la capa del servidor basado en el motor V8 de JavaScript, orientado a eventos no bloqueantes, lo que lo hace muy rápido y permite crear programas de red altamente escalables.

Con el uso de estas 4 tecnologías tenemos todo lo necesario (Frontend, Backend y Base de Datos) para desarrollar una aplicación web. En la siguiente imagen podemos ver cuál es el flujo de las aplicaciones web realizadas con MEAN.



Como se observa en la imagen, el Frontend lo desarrollamos con AngularJS, que es el encargado de hacer llamadas al API REST (Post, Put, Get y Delete) desarrollado en NodeJS que utiliza el framework de Express. El API podrá hacer un CRUD (Create-Read-Update-Delete) a la base de datos MongoDB y cuando el API tenga los datos que se le han pedido en la llamada los devolverá a AngularJS en formato JSON y este los mostrará en pantalla sin necesidad de recargar la página ya que AngularJS mantiene el modelo de datos actualizado sin necesidad de recargar la página.

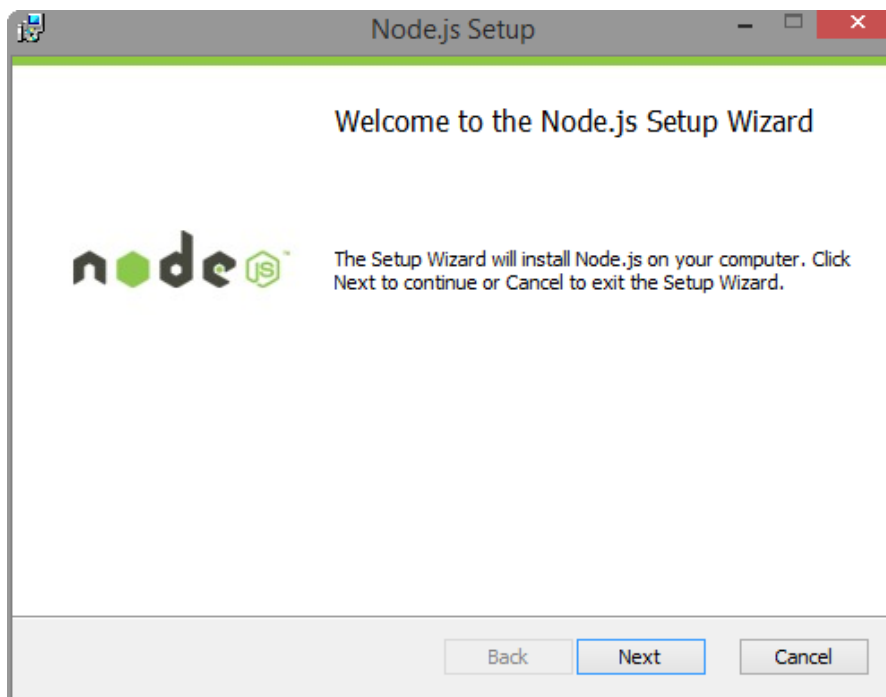
Como vemos esta es la arquitectura y el flujo de datos que hay en las aplicaciones desarrolladas con estas tecnologías. Dependiendo del proyecto podremos meter más lógica en la parte de Frontend o en la de Backend, pero eso ya son cosas que hay que valorar según las características de los proyecto.

Instalación de Mean JS

Lo primero que debemos hacer es descargar e instalar lo siguiente:

- Node js.
- Express
- Angular js
- Mongo js.

Para instalar NodeJS en Windows descargamos desde su página web <https://nodejs.org/> el instalador:



En el caso de instalarlo en Linux, tenemos que escribir el siguiente comando:

sudo apt-get install nodejs

Internet Game Data Base

Para instalar Express en Windows escribimos el siguiente comando en la carpeta de trabajo donde tenemos nuestra aplicación:

npm install express

En el caso de Linux, debemos escribir el siguiente comando en la carpeta donde tengamos el proyecto montado:

npm install

En el caso de AngularJS descargamos la librería de su página oficial <https://angularjs.org/>. Este caso vale tanto para Windows como para Linux.



Y por último, para instalar MongoDB en Windows descargamos el instalador en su página web <https://www.mongodb.org/>.

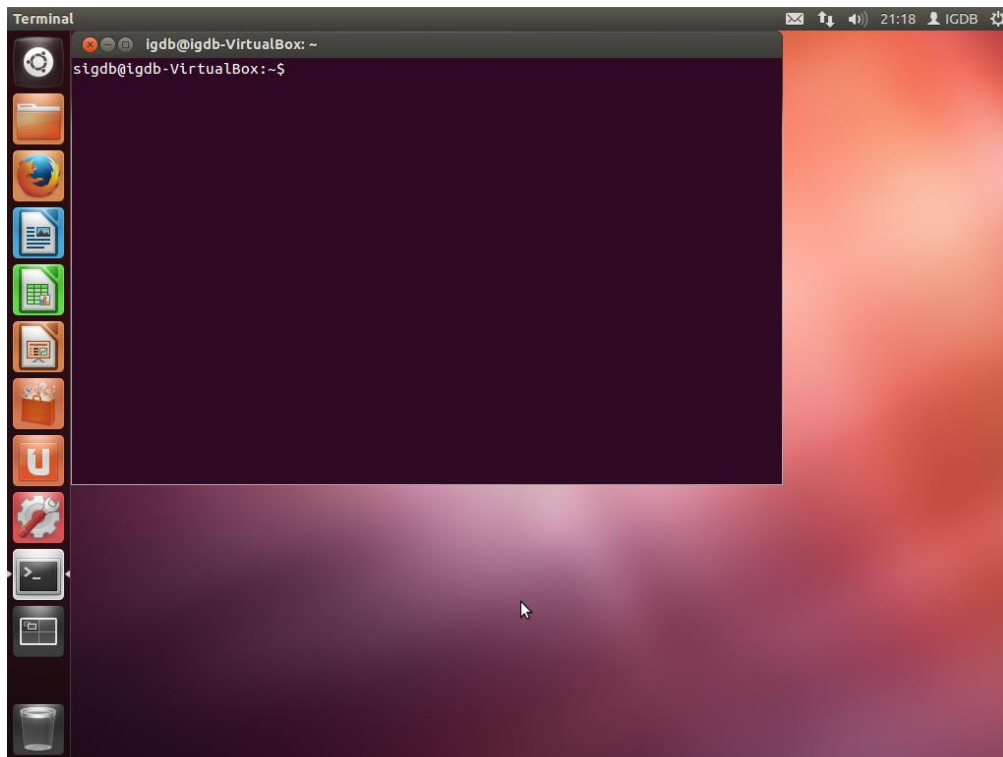


Y en el caso de Linux escribimos el siguiente comando:

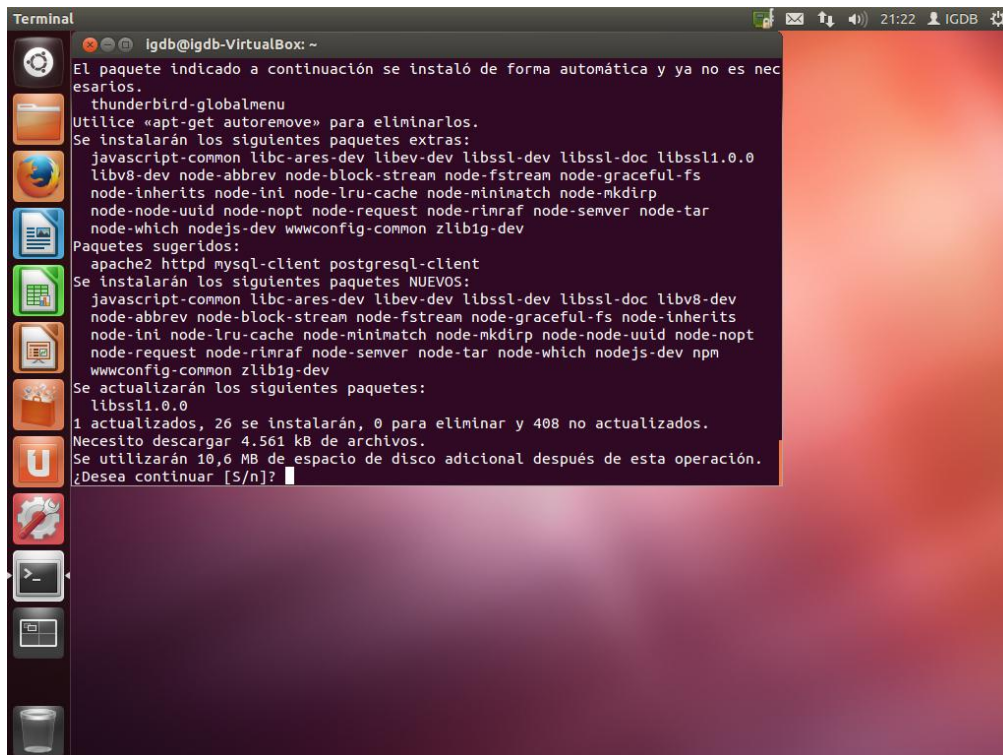
```
sudo apt-get install mongodb
```


Poniendo en marcha la aplicación en Linux

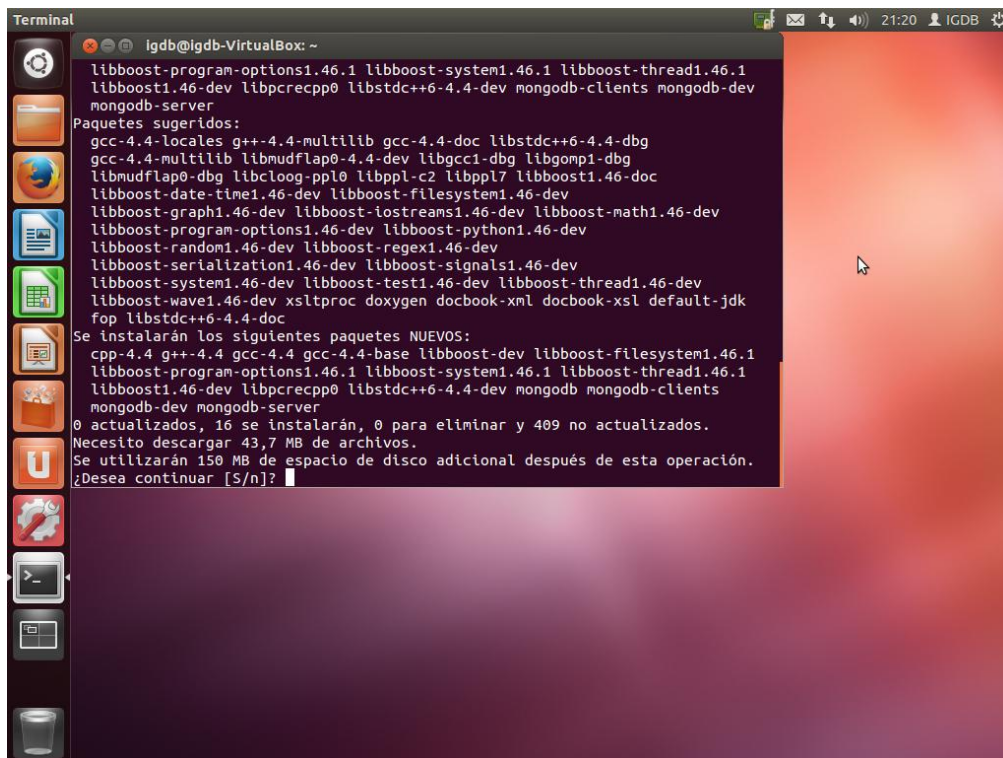
Abrimos el terminal en nuestro Linux.



Instalamos Node JS con el comando “*sudo apt-get install nodejs*”.

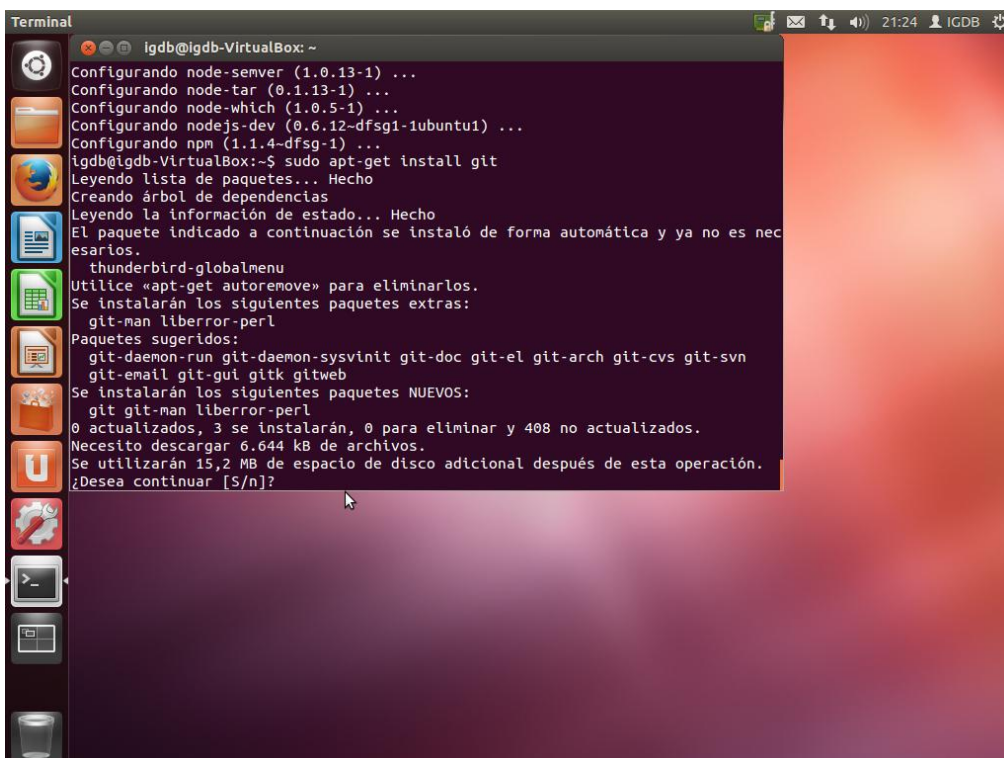


Instalamos MongoDB con el comando “*sudo apt-get install mongodb*”.



```
Terminal
igdb@igdb-VirtualBox: ~
libboost-program-options1.46.1 libboost-system1.46.1 libboost-thread1.46.1
libboost1.46-dev libpcrecpp0 libstdc++6-4.4-dev mongodb-clients mongodb-dev
mongodb-server
Paquetes sugeridos:
gcc-4.4-locales g++-4.4-multilib gcc-4.4-doc libstdc++6-4.4-dbg
gcc-4.4-multilib libmudflap0-4.4-dev libgcc1-dbg libgomp1-dbg
libmudflap0-dbg libcloog-ppl0 libppl-c2 libppl7 libboost1.46-doc
libboost-date-time1.46-dev libboost-filesystem1.46-dev
libboost-graph1.46-dev libboost-iostreams1.46-dev libboost-math1.46-dev
libboost-program-options1.46-dev libboost-python1.46-dev
libboost-random1.46-dev libboost-regex1.46-dev
libboost-serialization1.46-dev libboost-signals1.46-dev
libboost-system1.46-dev libboost-test1.46-dev libboost-thread1.46-dev
libboost-wave1.46-dev xsltproc doxygen docbook-xml docbook-xsl default-jdk
fop libstdc++6-4.4-doc
Se instalarán los siguientes paquetes NUEVOS:
cpp-4.4 g++-4.4 gcc-4.4 gcc-4.4-base libboost-dev libboost-filesystem1.46.1
libboost-program-options1.46.1 libboost-system1.46.1 libboost-thread1.46.1
libboost1.46-dev libpcrecpp0 libstdc++6-4.4-dev mongodb mongodb-clients
mongodb-dev mongodb-server
0 actualizados, 16 se instalarán, 0 para eliminar y 409 no actualizados.
Necesito descargar 43,7 MB de archivos.
Se utilizarán 150 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]:
```

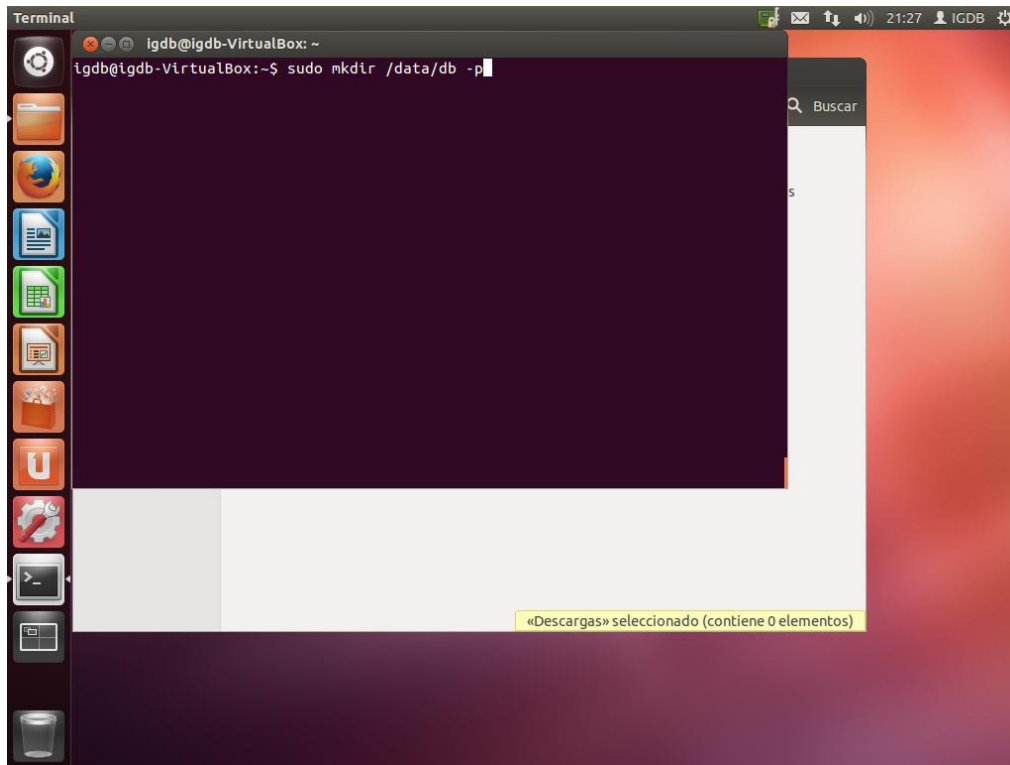
Instalamos GIT con el comando “*sudo apt-get install git*” y recuperamos el código que tenemos guardado en el repositorio de GitHub.



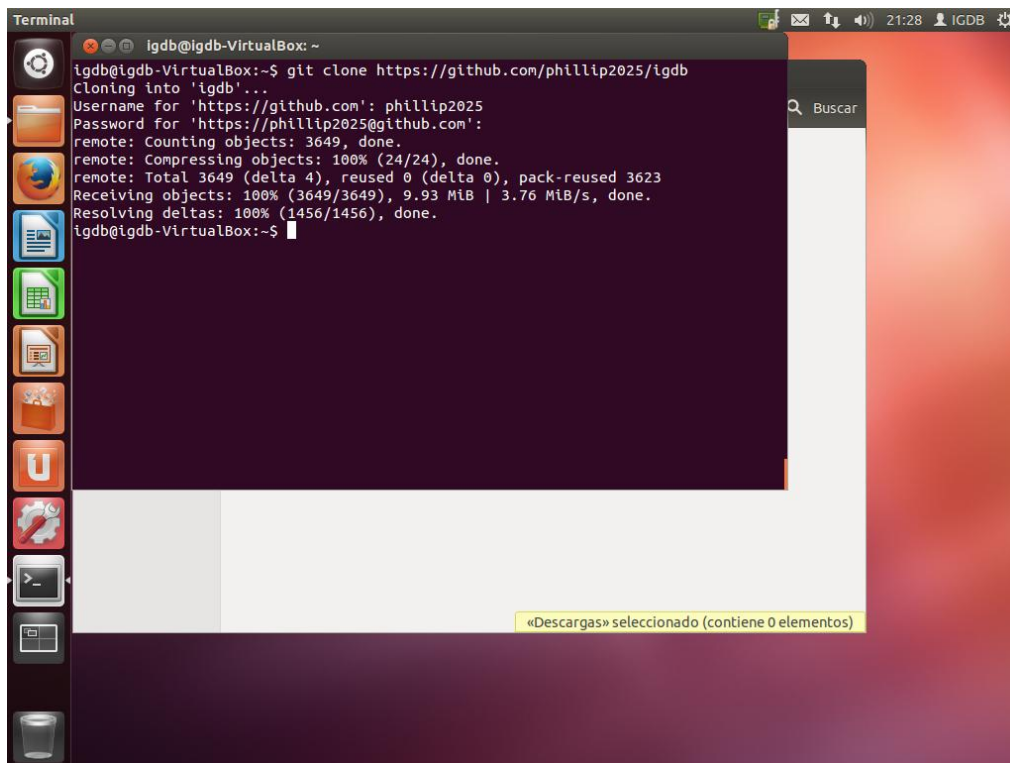
```
Terminal
igdb@igdb-VirtualBox: ~
Configurando node-semver (1.0.13-1) ...
Configurando node-tar (0.1.13-1) ...
Configurando node-which (1.0.5-1) ...
Configurando nodejs-dev (0.6.12-dfsg1-1ubuntu1) ...
Configurando npm (1.1.4-dfsg-1) ...
igdb@igdb-VirtualBox:~$ sudo apt-get install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es nec
esarios.
thunderbird-globalmenu
Utilice «apt-get autoremove» para eliminarlos.
Se instalarán los siguientes paquetes extras:
git-man liberror-perl
Paquetes sugeridos:
git-daemon-run git-daemon-sysvinit git-doc git-el git-arch git-cvs git-svn
git-email git-gui gitk gitweb
Se instalarán los siguientes paquetes NUEVOS:
git git-man liberror-perl
0 actualizados, 3 se instalarán, 0 para eliminar y 408 no actualizados.
Necesito descargar 6.644 kB de archivos.
Se utilizarán 15,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]:
```

Internet Game Data Base

Creamos el directorio necesario para guardar MongoDB.

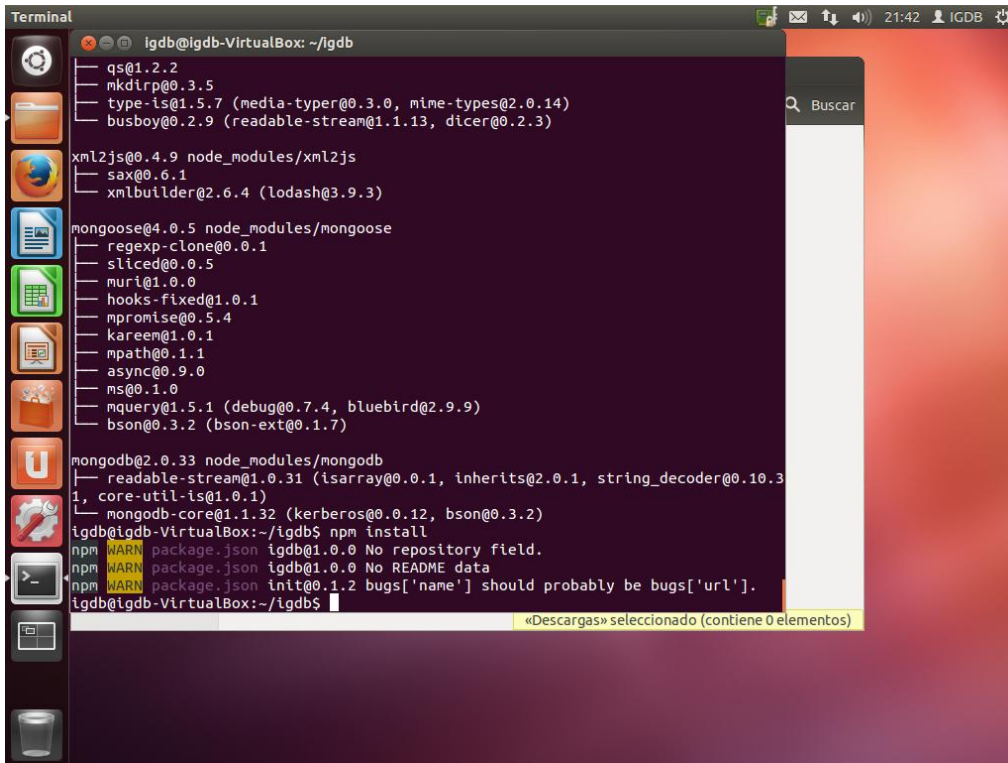


Descargamos el proyecto que tenemos guardado en el GitHub con *git clone* “url del repositorio de github”.



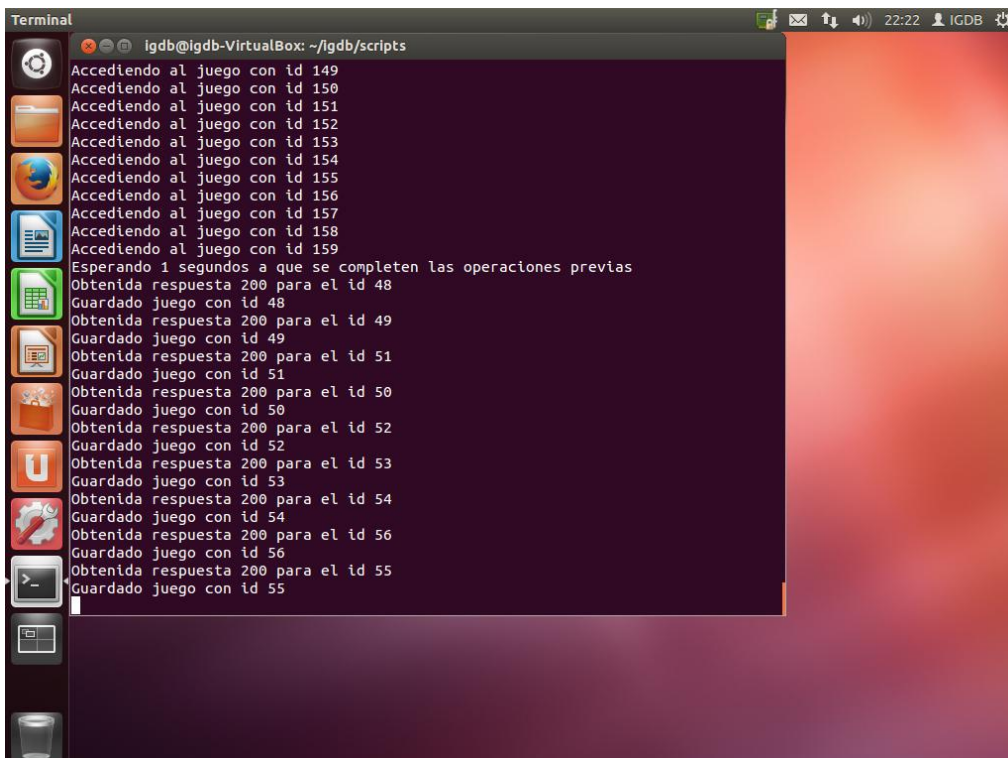
Internet Game Data Base

Dentro de la carpeta del proyecto que acabamos de clonar, descargamos los módulos de NodeJS como ExpressJS con el comando “npm install”.

A terminal window titled 'Terminal' with the prompt 'igdb@igdb-VirtualBox: ~/igdb'. It shows the output of 'npm install', listing various dependencies like qs, mkdirp, type-is, busboy, xml2js, sax, xmlbuilder, mongoose, regexp-clone, sliced, muri, hooks-fixed, mpmise, kareem, mpath, async, ms, mquery, and bson. It also shows warnings about missing repository fields and README data for some packages. At the bottom, a yellow status bar says '«Descargas» seleccionado (contiene 0 elementos)'.

```
igdb@igdb-VirtualBox: ~/igdb
├─ qs@1.2.2
├─ mkdirp@0.3.5
├─ type-is@1.5.7 (media-typer@0.3.0, mime-types@2.0.14)
├─ busboy@0.2.9 (readable-stream@1.1.13, dicer@0.2.3)
├─ xml2js@0.4.9 node_modules/xml2js
├─ sax@0.6.1
├─ xmlbuilder@2.6.4 (lodash@3.9.3)
├─ mongoose@4.0.5 node_modules/mongoose
├─ regexp-clone@0.0.1
├─ sliced@0.0.5
├─ muri@1.0.0
├─ hooks-fixed@1.0.1
├─ mpmise@0.5.4
├─ kareem@1.0.1
├─ mpath@0.1.1
├─ async@0.9.0
├─ ms@0.1.0
├─ mquery@1.5.1 (debug@0.7.4, bluebird@2.9.9)
├─ bson@0.3.2 (bson-ext@0.1.7)
├─ mongodb@2.0.33 node_modules/mongodb
├─ readable-stream@1.0.31 (isarray@0.0.1, inherits@2.0.1, string_decoder@0.10.31, core-util-is@1.0.1)
├─ mongodb-core@1.1.32 (kerberos@0.0.12, bson@0.3.2)
igdb@igdb-VirtualBox:~/igdb$ npm install
npm WARN package.json igdb@1.0.0 No repository field.
npm WARN package.json igdb@1.0.0 No README data
npm WARN package.json init@0.1.2 bugs['name'] should probably be bugs['url'].
igdb@igdb-VirtualBox:~/igdb$
```

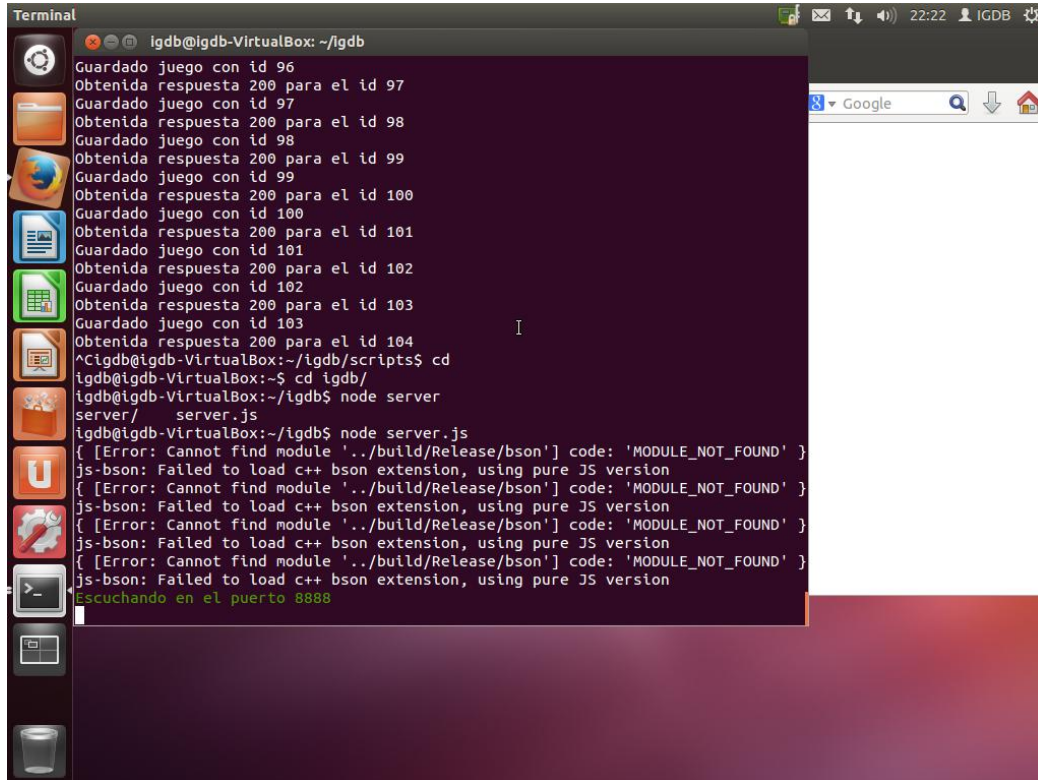
Usamos el script que tenemos creado para convertir los juegos en XML de la página thegamedb.net a nuestra base de datos en JSON usando el comando XmlToJsonHttpGet.js

A terminal window titled 'Terminal' with the prompt 'igdb@igdb-VirtualBox: ~/igdb/scripts'. It shows the execution of a script that processes game data. It lists game IDs from 149 to 159, followed by a 1-second wait, and then shows a loop of sending requests to thegamedb.net and saving the results to a JSON database. The status bar at the bottom is empty.

```
igdb@igdb-VirtualBox: ~/igdb/scripts
Accediendo al juego con id 149
Accediendo al juego con id 150
Accediendo al juego con id 151
Accediendo al juego con id 152
Accediendo al juego con id 153
Accediendo al juego con id 154
Accediendo al juego con id 155
Accediendo al juego con id 156
Accediendo al juego con id 157
Accediendo al juego con id 158
Accediendo al juego con id 159
Esperando 1 segundos a que se completen las operaciones previas
Obtenida respuesta 200 para el id 48
Guardado juego con id 48
Obtenida respuesta 200 para el id 49
Guardado juego con id 49
Obtenida respuesta 200 para el id 51
Guardado juego con id 51
Obtenida respuesta 200 para el id 50
Guardado juego con id 50
Obtenida respuesta 200 para el id 52
Guardado juego con id 52
Obtenida respuesta 200 para el id 53
Guardado juego con id 53
Obtenida respuesta 200 para el id 54
Guardado juego con id 54
Obtenida respuesta 200 para el id 56
Guardado juego con id 56
Obtenida respuesta 200 para el id 55
Guardado juego con id 55
```

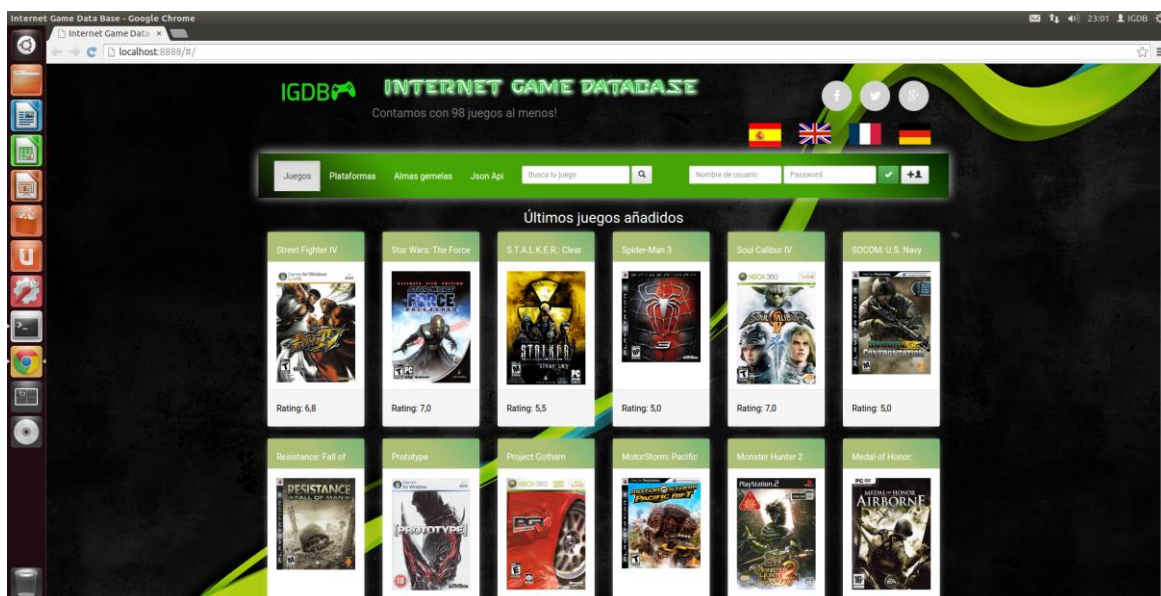
Internet Game Data Base

Arrancamos el servidor de la aplicación escribiendo `node server.js` en la carpeta donde se encuentre el archivo `server.js`. También debemos arrancar MongoDB con el comando `"mongo igdb"`. Si no tuviéramos creada la carpeta `/data/db` para mongo, tendríamos que poner el comando `"mongod"` para arrancar el servicio y después en otro terminal escribir `"mongo igdb"`.



```
igdb@igdb-VirtualBox: ~/igdb
Guardado juego con id 96
Obtenida respuesta 200 para el id 97
Guardado juego con id 97
Obtenida respuesta 200 para el id 98
Guardado juego con id 98
Obtenida respuesta 200 para el id 99
Guardado juego con id 99
Obtenida respuesta 200 para el id 100
Guardado juego con id 100
Obtenida respuesta 200 para el id 101
Guardado juego con id 101
Obtenida respuesta 200 para el id 102
Guardado juego con id 102
Obtenida respuesta 200 para el id 103
Guardado juego con id 103
Obtenida respuesta 200 para el id 104
^Cigdb@igdb-VirtualBox:~/igdb/scripts$ cd
igdb@igdb-VirtualBox:~$ cd igdb/
igdb@igdb-VirtualBox:~/igdb$ node server
server/ server.js
igdb@igdb-VirtualBox:~/igdb$ node server.js
{ [Error: Cannot find module '../build/Release/bson'] code: 'MODULE_NOT_FOUND' }
js-bson: Failed to load c++ bson extension, using pure JS version
{ [Error: Cannot find module '../build/Release/bson'] code: 'MODULE_NOT_FOUND' }
js-bson: Failed to load c++ bson extension, using pure JS version
{ [Error: Cannot find module '../build/Release/bson'] code: 'MODULE_NOT_FOUND' }
js-bson: Failed to load c++ bson extension, using pure JS version
{ [Error: Cannot find module '../build/Release/bson'] code: 'MODULE_NOT_FOUND' }
js-bson: Failed to load c++ bson extension, using pure JS version
{ [Error: Cannot find module '../build/Release/bson'] code: 'MODULE_NOT_FOUND' }
js-bson: Failed to load c++ bson extension, using pure JS version
Escuchando en el puerto 8888
```

Escribimos en la url de nuestro navegador `localhost:8888` y aparecerá nuestra página.



GitHub

Para que el trabajo en equipo fuera más eficiente, decidimos utilizar un repositorio en la página de GitHub, con esto conseguimos que cuando el equipo quisiera continuar el trabajo en la aplicación tenga la última versión del proyecto. De forma sencilla, el miembro que quisiera continuar el trabajo, tecleando un simple comando, tenga el proyecto en su ordenador local.

Los comandos que utilizamos eran los siguientes:

Para poder clonar el repositorio entero que teníamos en la página <https://github.com> escribimos el siguiente comando:

Git clone <https://github.com/Phillip2025/igdb>

Para poder traer al ordenador local los cambios que ha hecho otro compañero, escribimos el comando:

Git pull

Para ver los ficheros que han sido creados o modificados, escribimos el comando:

Git status

Para ver que archivos han cambiado o se han añadido al proyecto escribimos el comando:

Git add -all

Para añadir todos los cambios y archivos nuevos escribimos:

Git commit -m "comentario del cambio"

Para guardarlo en el repositorio de github escribimos:

Git push

Casos de uso

Hay tres tipos de usuarios:

- Usuario Administrador.
- Usuario Logeado.
- Usuario Anónimo.

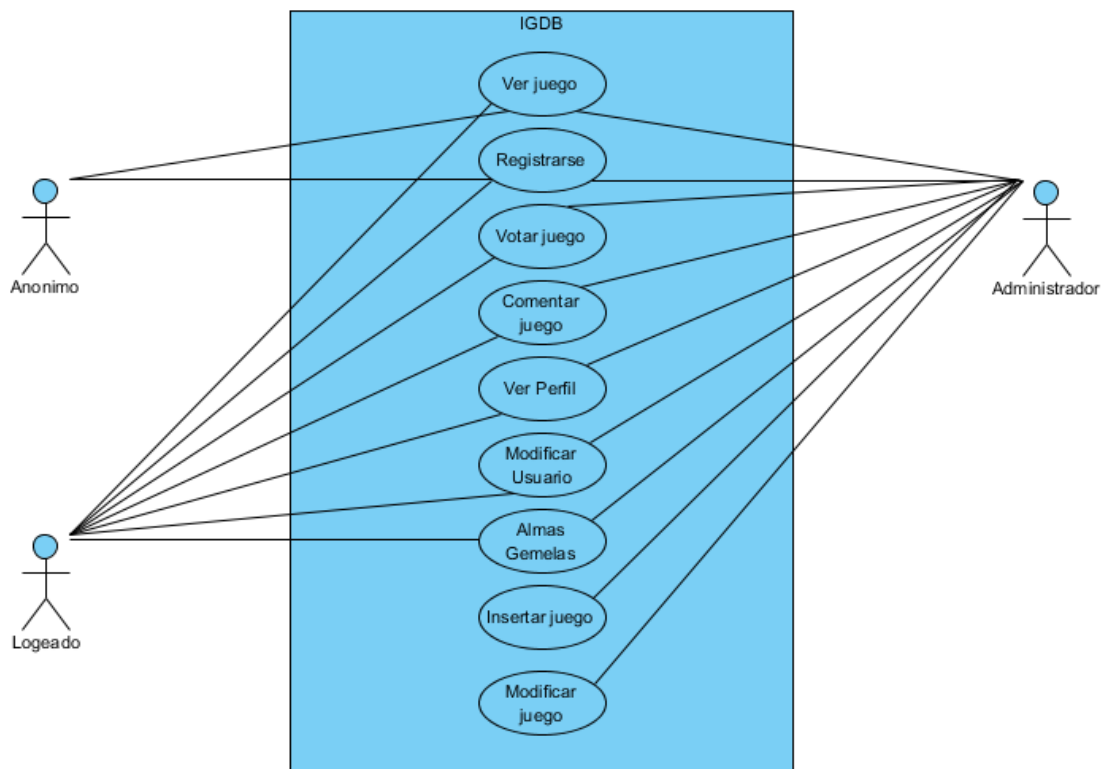
No todos los tipos de usuarios pueden hacer todos los casos de uso.

ID	Nombre	Descripción
1	Ver un juego	Permite ver las características de un videojuego, como por ejemplo, portada, nombre, sinopsis....
2	Registrarse	Permite registrarnos en la página web como usuario para poder realizar acciones que solo un usuario registrado podría realizar, como votar y comentar juegos.
3	Votar un juego	Permite poder votar juegos para así poder dejar constancia de tu opinión del juego para que los demás usuarios puedan ver tu valoración.
4	Comentar un juego	Permite poder dejar comentarios en el juego que quieras para que los demás usuarios vean tu opinión.
5	Ver perfil	Permite al usuario logeado ver sus datos y poder visualizar los comentarios y votos de sus juegos.
6	Modificar usuario	Permite modificar campos de tu perfil de usuario como tu nombre, apellido, foto de perfil, etc.
7	Almas gemelas	Permite visualizar los usuarios que tienen gustos similares a los tuyos y recomiendan juegos.
8	Insertar juego	Permite al usuario Administrador insertar un nuevo juego en la base de datos.
9	Modificar juego	Permite al usuario Administrador modificar los campos de un juego que este en la base de datos.

Lista de casos de uso

Actores principales	Casos de uso
Usuario Administrador Usuario Logeado Usuario Anónimo	Caso de uso 1 y 2
Usuario Administrador Usuario Logeado	Casos de uso 1, 2, 3, 4, 5, 6 y 7
Usuario Administrador	Casos de uso 1, 2, 3, 4, 5, 6, 7, 8 y 9

Diagrama de casos de uso



Plantillas de casos de uso

ID:	1		
Nombre:	Ver Juego		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	4 mayo de 2015

Actores:	Anónimo, Logeado y Administrador
Descripción:	Ver la información de un videojuego: Nombre, consola, fecha en la que salió al mercado, descripción, imágenes y votos.
Desencadenantes:	Pulsando encima de cualquier videojuego.
Precondiciones:	No es necesario estar logeado, con lo que cualquier usuario puede acceder a esta página.
Postcondiciones:	Se mostrara la página de ese videojuego en concreto.
Flujo Normal:	Va a la página donde está toda la información del juego en el que se ha clicado.
Flujos alternativos:	No se ven algunas imágenes
Frecuencia de uso	50 por minuto.
Suposiciones y motivos:	Incluimos este caso de uso para que cualquier usuario pueda visualizar videojuegos pero que un usuario no logeado no pueda votar dentro de la comunidad.
Notas:	Sin incidencias
Incidencias y TBDs:	Todas las incidencias han sido resueltas.

Internet Game Data Base

ID:	2		
Nombre:	Registrarse		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	19 mayo de 2015

Actores:	Anónimo, Logeado y Administrador
Descripción:	Permite registrarnos en la página web como usuario para poder realizar acciones que solo un usuario registrado podría realizar, como votar y comentar juegos.
Desencadenantes:	Clicando en el icono con forma de persona con un símbolo más (+).
Precondiciones:	La persona debe clicar en el icono.
Postcondiciones:	Sale un formulario para rellenar con los datos de la persona.
Flujo Normal:	La persona se registra y una vez enviado los campos, vuelve a la página principal con su usuario logeado.
Flujos alternativos:	Falta algún campo que rellenar del formulario o que le nombre de usuario ya exista.
Frecuencia de uso	10 por minuto
Suposiciones y motivos:	Incluimos este caso de uso para que las personas puedan ser parte de la comunidad de la página y así poder votar y comentar en un futuro sus juegos favoritos.
Notas:	Sin incidencias
Incidencias y TBDs:	Todas las incidencias han sido resueltas.

Internet Game Data Base

ID:	3		
Nombre:	Votar un juego		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	19 mayo de 2015

Actores:	Logeado y Administrador
Descripción:	Permite votar un videojuego en un rango entre 1 y 10.
Desencadenantes:	Clicando en número de mandos que quiera ponerle, significando un mando un 1 y diez mandos un 10.
Precondiciones:	El usuario debe estar logeado.
Postcondiciones:	La media del juego aumentara o disminuirá dependiendo de la nota que le ponga al juego.
Flujo Normal:	El usuario vota el juego, la media sube o disminuye y se añade a su historial de videojuegos votados.
Flujos alternativos:	El juego ya ha sido votado y cambia el voto.
Frecuencia de uso	17 por minuto
Suposiciones y motivos:	Incluimos este caso de uso para que los usuarios puedan dejar constancia de su opinión del juego con una nota representativa entre 0 y 10.
Notas:	Sin incidencias
Incidencias y TBDs:	Teníamos el problema de que un usuario podía votar varias veces el mismo juego.

Internet Game Data Base

ID:	4		
Nombre:	Comentar juego		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	4 mayo de 2015

Actores:	Logeado y Administrador
Descripción:	Permite dejar un comentario en el juego que quieras, para poder expresar tu opinión del juego.
Desencadenantes:	Una vez hayas escrito el comentario, clicas en el icono de publicar y tu comentario estará en la página hasta que decidas borrarlo.
Precondiciones:	El usuario debe de estar logeado.
Postcondiciones:	Una vez hayas publicado el comentario, en la vista del juego que has comentado quedara reflejado el comentario que has publicado.
Flujo Normal:	El comentario se publica con total normalidad en la página del juego.
Flujos alternativos:	Haya un error en la publicación del comentario.
Frecuencia de uso	12 por minuto.
Suposiciones y motivos:	Suponemos que a los usuarios les gusta dejar sus opiniones acerca de los juego, tanto comentarios buenos como malos, sin llegar a ofender a nadie.
Notas:	Sin incidencias
Incidencias y TBDs:	Ninguna incidencia.

Internet Game Data Base

ID:	5		
Nombre:	Ver perfil		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	4 mayo de 2015

Actores:	Logeado y Administrador
Descripción:	Permite ver los datos del perfil del usuario registrado y ver sus comentarios y votaciones de los juegos.
Desencadenantes:	Una vez el usuario este logeado, aparecerá una pestaña de “ver perfil”, clicando en ella, puede ver su información
Precondiciones:	El usuario debe de estar logeado y debe de haber clicado en la pestaña “ver perfil”.
Postcondiciones:	Vera toda su información y vera enlaces que llevan a sus comentarios y a sus votos (en orden de mas nuevo a más antiguo).
Flujo Normal:	El usuario ve todos sus datos con toda normalidad.
Flujos alternativos:	El usuario no puede ver algunos campos de su información.
Frecuencia de uso	2 por minuto.
Suposiciones y motivos:	Cuando no saben exactamente que juegos han votado o comentado, este caso de uso les será perfecto, ya que pueden revisar que nota tienen a sus juegos y también les sirve como diario de juegos.
Notas:	Sin incidencias
Incidencias y TBDs:	Ninguna incidencia.

Internet Game Data Base

ID:	6		
Nombre:	Modificar usuario		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	4 mayo de 2015

Actores:	Logeado y Administrador
Descripción:	Permite modificar los datos de usuario.
Desencadenantes:	Una vez el usuario este logeado, aparecerá una pestaña de “modificar datos”. Si clicas en esa pestaña, podrá modificar los campos que quiera.
Precondiciones:	El usuario debe de estar logeado y debe de haber clicado en la pestaña “modificar usuario”.
Postcondiciones:	Una vez haya rellenado los campos que quiera modificar y haya dado a modificar, volverá a la página de inicio con sus datos modificados.
Flujo Normal:	El usuario modifica los datos que quiera y se guarda.
Flujos alternativos:	El usuario cambia su nombre de usuario por uno que ya exista.
Frecuencia de uso	1 por minuto.
Suposiciones y motivos:	Muchas veces, un usuario se equivoca al hacer el registro poniendo mal su nombre o eligiendo una foto que no le convence, al añadir este caso de uso, el usuario puede modificar lo que no le haya gustado en cualquier momento.
Notas:	Sin incidencias
Incidencias y TBDs:	Ninguna incidencia.

Internet Game Data Base

ID:	7		
Nombre:	Almas Gemelas		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	8 de junio de 2015	Fecha última modificación:	8 de Junio de 2015

Actores:	Logeado y Administrador
Descripción:	Permite ver los juegos recomendados por otros usuarios.
Desencadenantes:	Una vez el usuario este logeado, aparecerá una pestaña de “Almas gemelas”, si clicamos en ella, podemos ver los usuarios que tienen gustos similares a los nuestros y nos pueden recomendar juegos.
Precondiciones:	El usuario debe de estar logeado y debe de haber clicado en la pestaña “Almas gemelas”.
Postcondiciones:	Una vez haya clicado veremos los juegos que nos recomienda un usuario con nuestros mismos gustos.
Flujo Normal:	El usuario clica en las almas gemelas, aparece un usuario con nuestros mismos gustos y nos recomienda juegos
Flujos alternativos:	Que no haya usuarios que tengan nuestros mismos gustos, lo que haría que la pantalla apareciera sin juegos recomendados.
Frecuencia de uso	1 por minuto.
Suposiciones y motivos:	Añadimos este caso porque es una buena forma de saber que juegos podemos jugar ya que al recomendárnoslo una persona con nuestros mismos gustos, nos fiaremos más de ella.
Notas:	Sin incidencias
Incidencias y TBDs:	Ninguna incidencia.

Internet Game Data Base

ID:	8		
Nombre:	Insertar juego		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	4 mayo de 2015

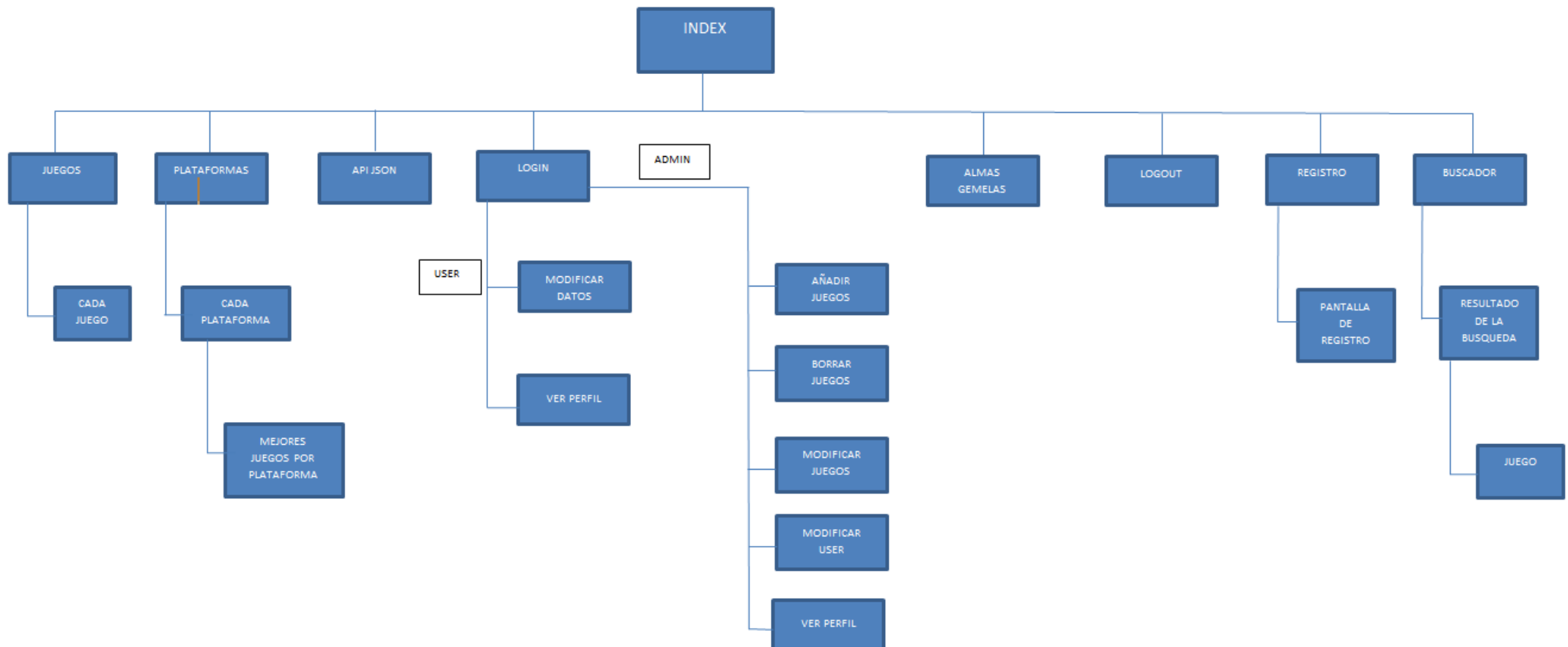
Actores:	Administrador
Descripción:	Permite añadir un juego a la base de datos.
Desencadenantes:	Una vez el usuario este logeado como administrador, podrá insertar un juego nuevo a la base de datos.
Precondiciones:	El usuario debe de estar logeado como administrador y clicar en la pestaña de “insertar juego”.
Postcondiciones:	Una vez haya rellenado todos los campos del juego y haya clicado en añadir, el juego estará disponible para ser visualizado en la página.
Flujo Normal:	El administrador rellena todos los campos y el juego se añade a la base de datos.
Flujos alternativos:	El administrador no rellena todos los campos, el juego se añade pero algunos campos no tienen información.
Frecuencia de uso	1 por hora.
Suposiciones y motivos:	Cada semana salen juegos nuevos al mercado y los usuarios quieren ver que nuevos juegos existen, de esta forma la base de datos no queda desactualizada.
Notas:	Sin incidencias
Incidencias y TBDs:	Ninguna incidencia.

Internet Game Data Base

ID:	9		
Nombre:	Modificar juego		
Creador:	Mario Arranz Águeda	Última modificación por:	Mario Arranz Águeda
Fecha creación:	19 abril de 2015	Fecha última modificación:	4 mayo de 2015

Actores:	Administrador
Descripción:	Permite modificar los datos de un juego.
Desencadenantes:	Una vez el usuario este logeado como administrador, podrá modificar un juego de la base de datos.
Precondiciones:	El usuario debe de estar logeado como administrador y clicar en la pestaña de “modificar juego” en la pantalla del juego que quiera modificar.
Postcondiciones:	La información del juego se modifica.
Flujo Normal:	La información del juego se ha modificado correctamente.
Flujos alternativos:	
Frecuencia de uso	1 por hora.
Suposiciones y motivos:	Con las nuevas generación de consolas, la información de hoy, puede que no sea la misma que la de mañana con la inclusión de DLCs en los juegos, por eso, decidimos poder modificar un juego, por si algún juego necesita ser modificado y tener siempre la base de datos lo más actualizada posible.
Notas:	Sin incidencias
Incidencias y TBDs:	Ninguna incidencia.

Diseño de Navegación



Diseño de Roles en la Web

Nombre: Angelita.
Edad: 82 años
Oficio: Ama de casa.
Residencia: Fuenlabrada.
Género: Mujer.
Educación: Primaria.
Ingresos: Pensión.
Estado civil: Viuda:
Hijos: 3.



Intereses: Su nieto le ha pedido un juego para su cumpleaños, pero Angelita quiere saber si es un juego apto para su nieto, por eso busca el juego para saber si es un juego apto para él.

Nombre: Francisco.
Edad: 50 años
Oficio: Administrativo.
Residencia: Madrid.
Género: Hombre.
Educación: EGB.
Ingresos: Trabajo.
Estado civil: Casado:
Hijos: 2.



Intereses: Francisco quiere retomar su antigua afición por jugar a los juegos y entra en la página para ver qué novedades hay y poder valorar lo que le ha gustado con una nota. Además quiere que le recomienden juegos que pasarse y así poder jugar juegos nuevos que le recomiende los usuarios con sus mismos gustos.

Nombre: Nacho.
Edad: 27 años.
Oficio: Ingeniero.
Residencia: Madrid.
Género: Hombre.
Educación: Ingeniería Telecomunicaciones.
Ingresos: Trabajo de Ingeniero.
Estado civil: Soltero:
Hijos: Ninguno.

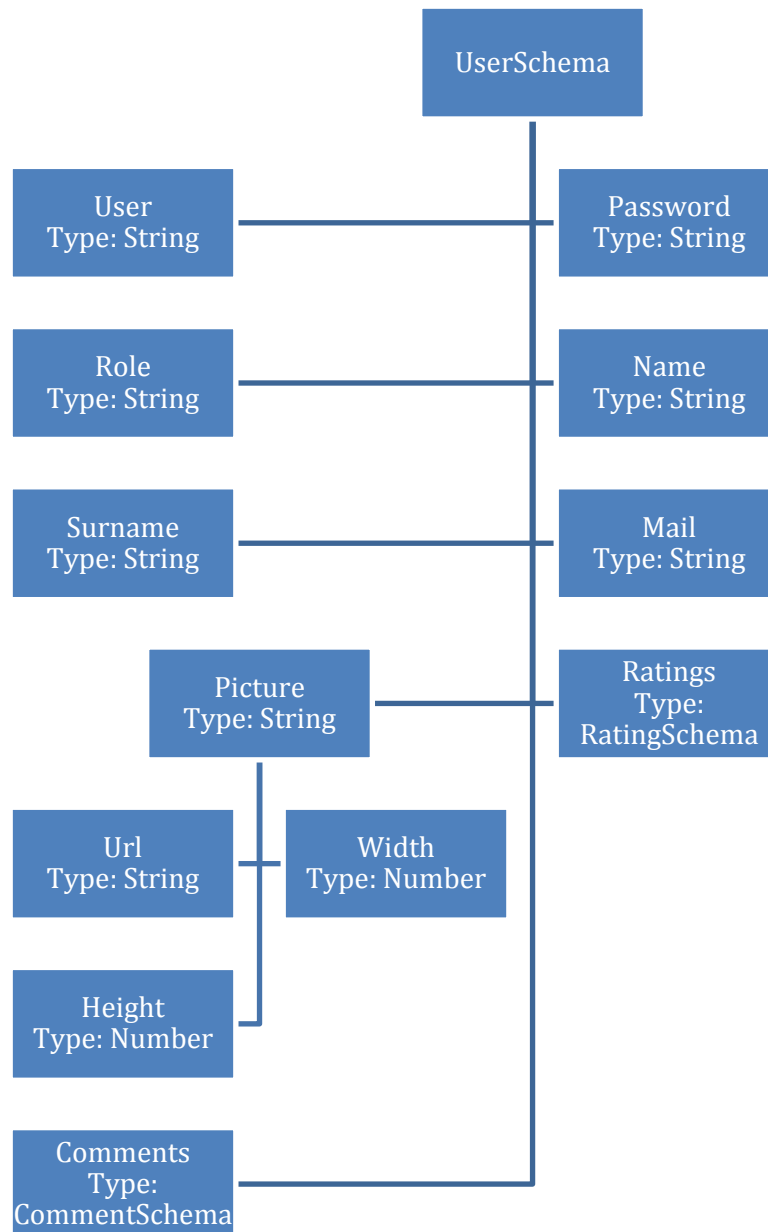


Intereses: A Nacho le encantan los videojuegos, por eso quiere seguir cuales son las últimas novedades en la industria de los videojuegos. También le gusta poder dejar su opinión respecto a los juegos que se ha pasado. Quiere saber qué juegos se ha pasado, por eso en cuanto se pasa un juego, lo que quiere es dejar constancia de lo que le ha parecido, poniéndole nota.

Diseño del Desarrollo

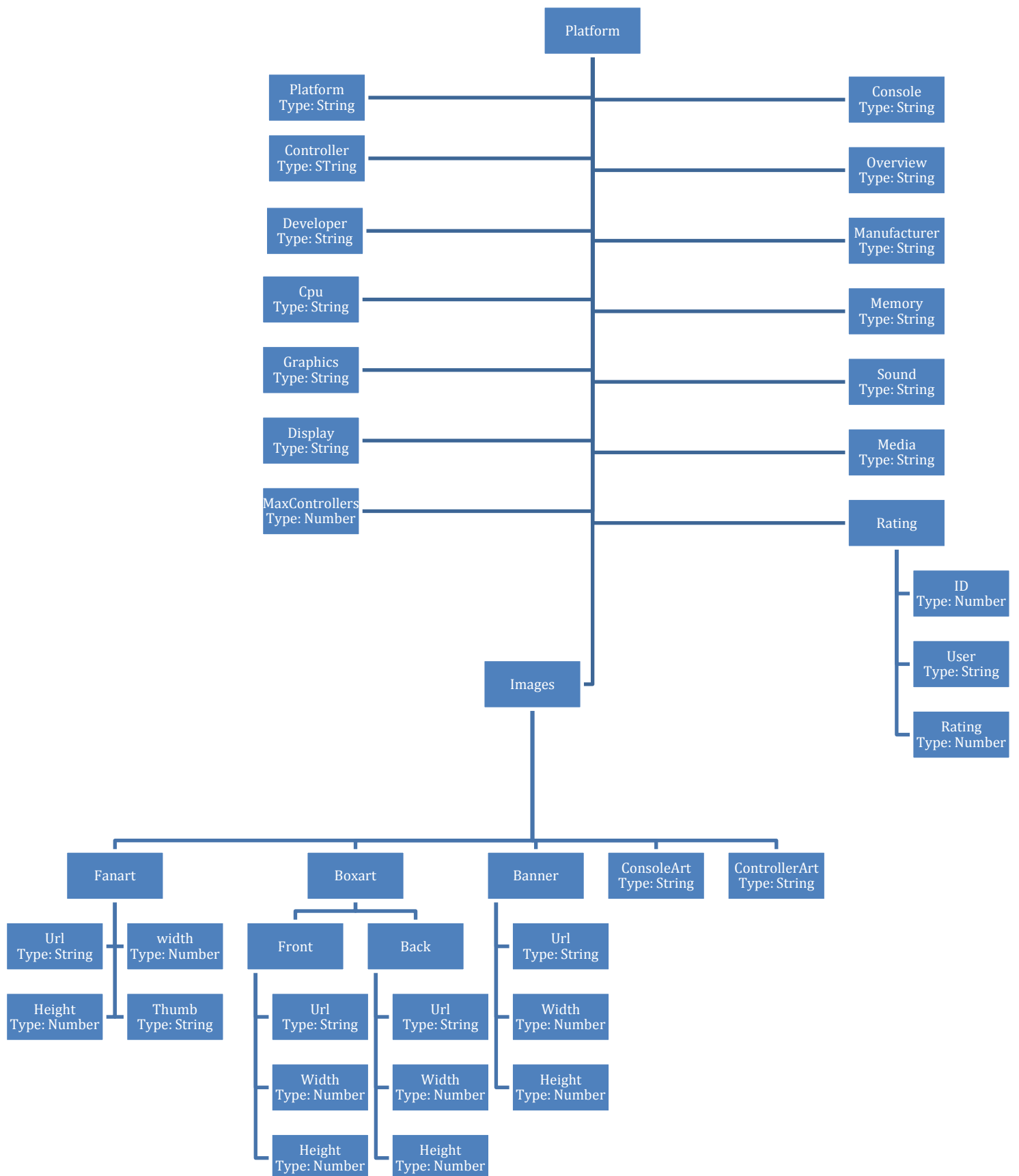
Los modelos que hemos utilizado son esquemas de JSON. A continuación están todos los modelos utilizados en la página.

Modelo de Usuario:



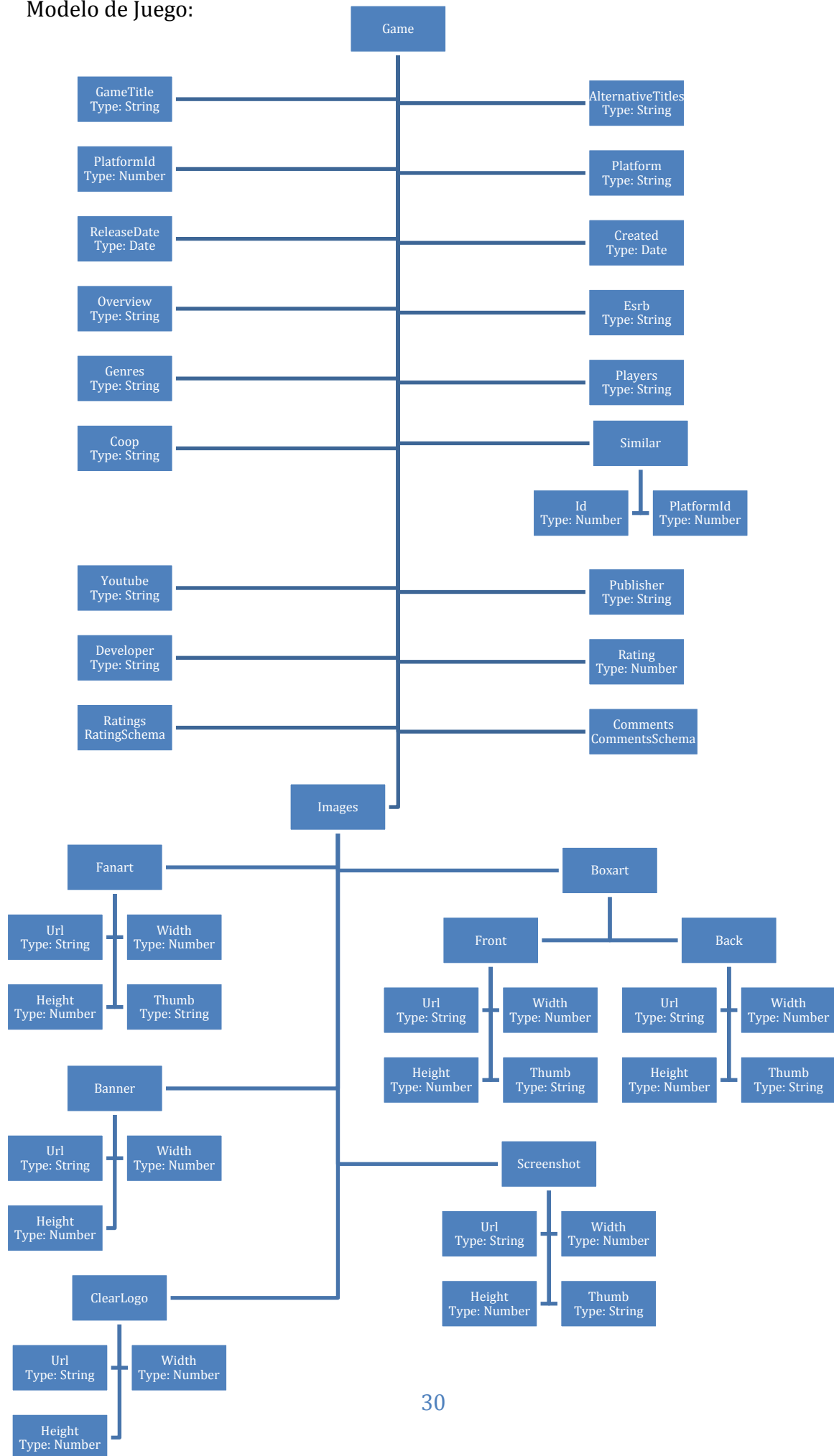
Internet Game Data Base

Modelo de Plataforma:



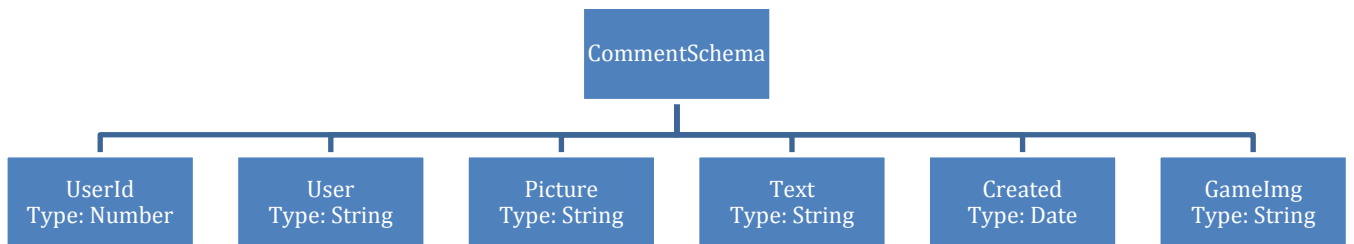
Internet Game Data Base

Modelo de Juego:

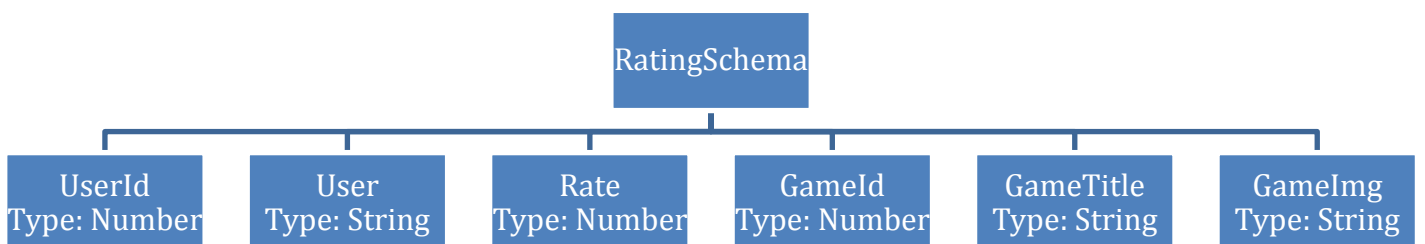


Internet Game Data Base

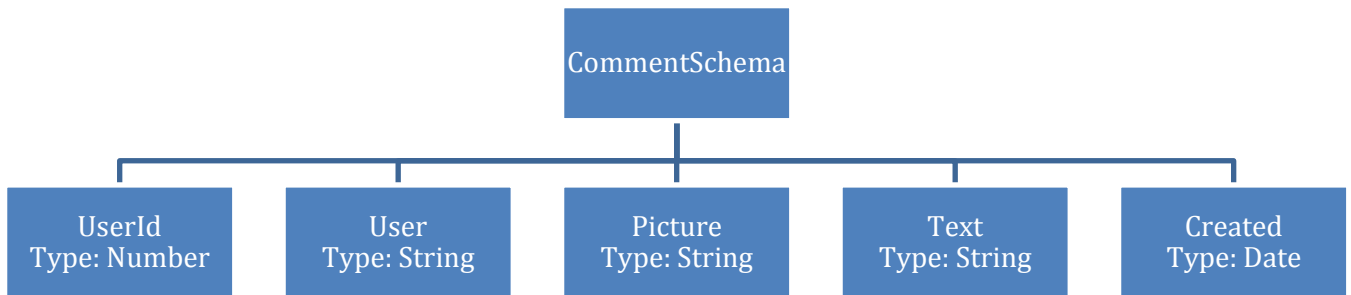
Modelo de Comentarios para usuario:



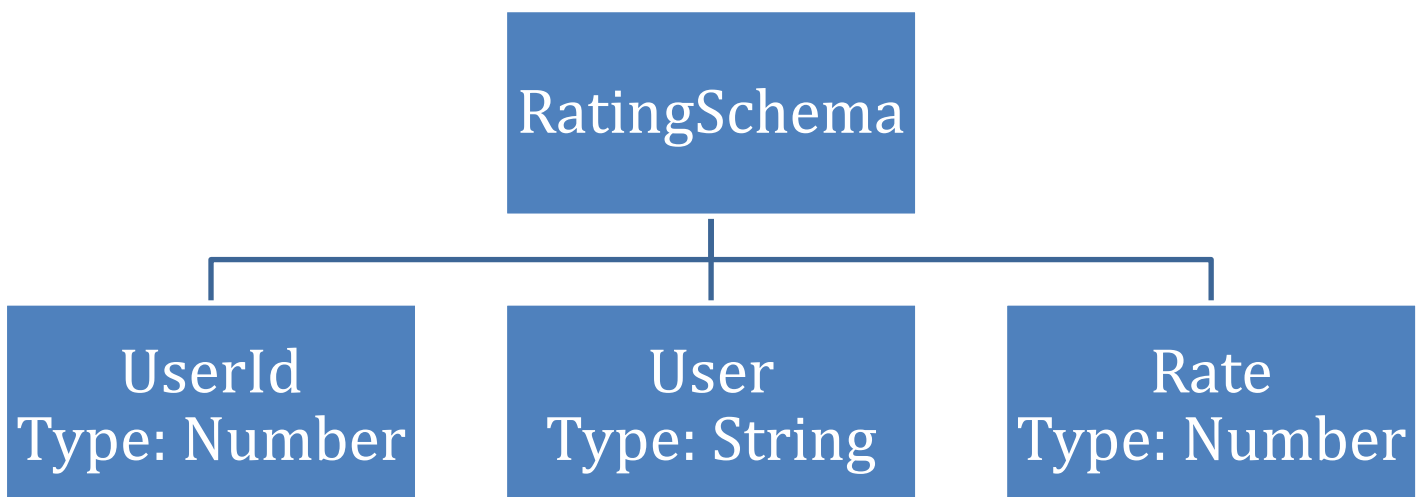
Modelo de Votos para usuario:



Modelo de Comentarios para juego:



Modelo de Votos para juego:



Manual de Usuario

A continuación explicaremos como interactuar en las páginas de navegación.

Página Inicio

Para poder ver un juego podemos clicar en un juego de los últimos añadidos o ir a la pestaña de “Juegos”.

Para poder ver las plataformas clicamos en la pestaña “Plataformas”.

Para poder ver los juegos que nos recomienda un usuario con gustos afines a los nuestros clicamos “Almas gemelas”.

Para poder ver el Api de la página clicamos en la pestaña “Api Json”.

Para poder registrarse clicamos en el icono de la silueta.

Para poder logearse escribimos nuestro nombre de usuario y contraseña en los campos indicados.

Para buscar el juego que queramos, escribimos el nombre del juego que queramos en el campo del buscador.

Página Juego

En la página del juego podemos ver toda la información del videojuego en cuestión.

Para poder comentar, necesitamos estar logeados. Una vez logeados escribimos nuestro comentario en el campo de la parte inferior de la página y clicamos en el botón de Publicar comentario (botón verde).

Para poder puntuar, necesitamos estar logeados. Una vez logeados clicamos en la cantidad de mandos que queramos poner al juego, siendo 1 mando un 1 y 10 mandos un 10.

También podemos ver varias imágenes, capturas de pantalla del juego y videos del juego. Si no se dispone de video o imagen, aparecerá “Imagen no disponible”.

Página Plataformas

Una vez clicamos en la pestaña de “Plataformas”, iremos a ver todas las plataformas que tiene la página.

Para ver una plataforma en concreto clicamos sobre la plataforma que queramos ver.

Podemos ver imágenes de los usuarios o imágenes promocionales de la plataforma. Si no se dispone de la imagen aparecerá “Imagen no disponible”.

Página Plataforma

En la página de la plataforma podremos ver toda la información de la consola, además de poder ver los mejores juegos valorados por los usuarios de la plataforma en la que estemos.

Página de Registro

Para poder registrarse clicamos sobre el icono con la silueta de una persona de color gris. Una hallamos clicado, nos saldrá una ventana donde nos pedirá nuestros datos. Una vez estén todos los datos rellenos clicamos sobre el tick verde y el registro se habrá completado con éxito.

El registro no se habrá completado con éxito si pasa uno de los siguientes casos:

- El formato de nuestro email no es el correcto (el formato correcto sería [nombre@dominio.Tld](#) correspondiente)
- Las password no coinciden.
- Falta algún campo por rellenar.
- El nombre de usuario ya exista.

Página de Modificación

Podemos modificar los datos de nuestro perfil, como nuestro nombre, apellido, contraseña, e incluir una foto de perfil.

Ver perfil

En esta página podemos ver nuestra información del perfil, además de ver los jugos que hemos votado y la nota que le hemos puesto

Página Json Api

Podemos ver como se utiliza nuestra app internamente par a poder descargarse de forma libre y gratuita.

Buscar un juego

Si queremos buscar un juego en la página, basta con escribir el nombre en el campo de “Buscar tu juego”. Aparecerán todos los juegos que coincidan con el nombre que has escrito.

Almas gemelas

En esta página, una vez estemos logeados, veremos a los usuarios que tienen gustos similares a los nuestros. Nos aparecerán los juegos que nos recomienda y que no hemos jugado.

Bibliografía

La información tanto de las tecnologías MEAN como la api las hemos sacado de las siguientes páginas:

Api:

- <http://devdocs.io/>
- <https://nodejs.org/api/>
- <https://docs.angularjs.org/api?PHPSESSID=cae8e98e7ca559b4605d75c813b358ee>
- <http://expressjs.com/api.html>
- <http://docs.mongodb.org/manual/crud/>

Información sobre MEAN JS:

- <https://carlosazaustre.es/blog/>
- <http://www.genbetadev.com/frameworks/el-auge-del-stack-mean-basado-en-javascript-para-crear-aplicaciones-web-completas>
- <http://mean.io/#!/>
- <http://meanjs.org/>