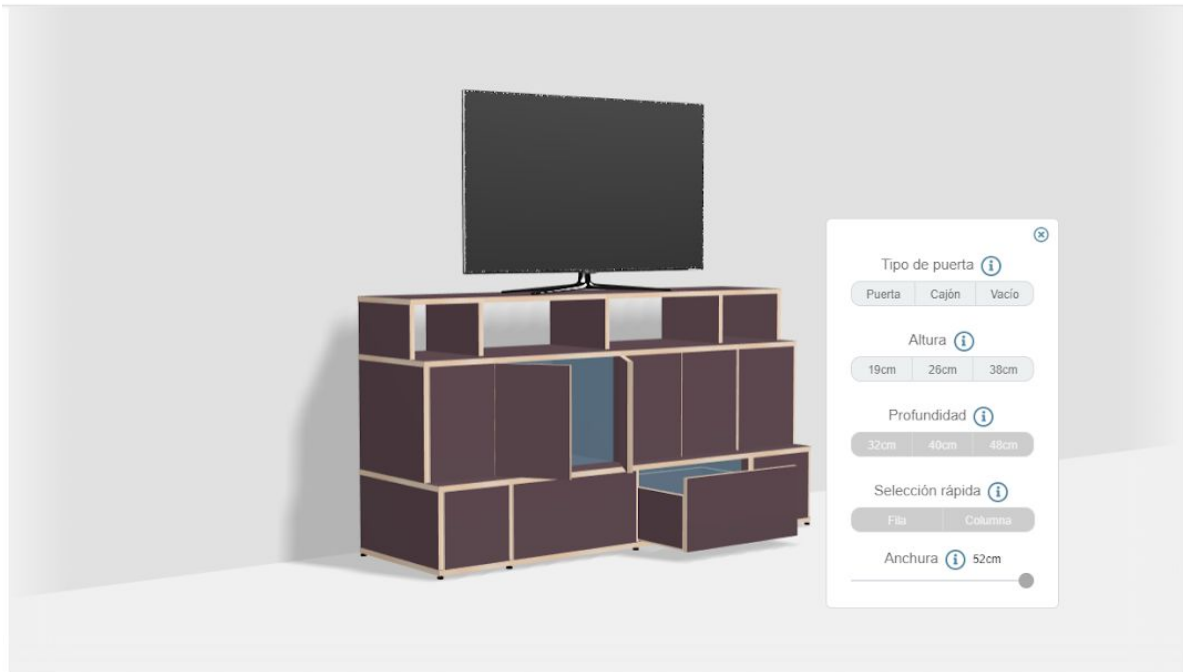


# Desenvolupament d'Aplicacions Web

## Projecte final

# Editor de mobles 3D



Autor: **Marc Pascual Sánchez**

# Índex de Continguts

<b>Introducció</b>	<b>4</b>
Descripció	4
<b>Requisits</b>	<b>4</b>
<b>Tecnologia</b>	<b>5</b>
Client	5
Versions	5
Servidor	5
Versions	6
<b>Arquitectura</b>	<b>6</b>
Client	6
Estructura	6
Funcionament	7
Servidor	8
Estructura	8
Funcionament	8
<b>Diagrames UML</b>	<b>10</b>
Classes	10
Client	10
Servidor	11
Casos d'ús	12
<b>Innovació</b>	<b>13</b>
Llibertat d'edició	13
Guardat i càrrega de models	13
Velocitat	14
Disseny	14
<b>Viabilitat del projecte</b>	<b>17</b>
Competidors	17
Costs	18
Serveis gratuïts	19
Servidor	19
Base de dades	19
Limitacions	20
Profunditat	20
Altura	20
Amplada	20
Portes i calaixos	21
Colors	21
Selecció ràpida	21

<b>Planificació del projecte</b>	<b>21</b>
Dinàmica	21
Eines	21
Organització	21
Control de versions	22
Testing	22
Versions	23
<b>Conclusions</b>	<b>24</b>
Formació DUAL	24
Projecte final	24
Millores de futur i errors	24
<b>Webgrafia</b>	<b>25</b>
Costs	25
Serveis gratuïts	25
Documentació	25
Planificació del projecte	26
Competidors	26
Desplegament	26

# Introducció

Aquest projecte va ser inicialment desenvolupat per encàrrec de l'empresa Edatasoft S.L. per la qual treballava en el seu moment. El projecte ha estat desenvolupat de forma individual des de l'inici fins al final. En cert moment del desenvolupament inicial es va cancel·lar el projecte, quedant a mitges. Així doncs, a partir d'aquesta aplicació prèviament desenvolupada per nosaltres el que farem és reprendre-la, acabar-la i implantar-la.

La pàgina WEB resultant d'aquest projecte es troba en el domini de Heroku, a la URL: <https://furniture-editor.herokuapp.com>.

El repositori en el qual hem treballat és a GitHub, i és el següent: <https://github.com/marcpascualsanchez/Furniture-Editor-3D.git>.

## Descripció

L'objectiu d'aquest projecte és desenvolupar una aplicació **WEB** que crei i mostri mobles renderitzats en 3 dimensions generats a partir dels inputs del client al moment. Els inputs es recullen amb una interfície gràfica del fitxer **HTML**, i s'administren amb els scripts programats en **JavaScript**, que controlaran les dades i les accions a fer amb aquestes dades.

## Requisits

El client que va encarregar aquest projecte va demanar de forma molt vaga un editor de mobles "online", que renderitzés models 3D a temps real, amb llibertat per part de l'usuari de canviar les dimensions, les portes i calaixos. No es va especificar cap tecnologia com a requeriment. Els requisits que ens va proporcionar va ser una llista basada en les opcions que proporciona l'eina d'un dels seus competidors directes, Tylko, dels quals parlarem a l'apartat de competidors d'aquest document.

Els requisits eren poder canviar:

- Lliurement a nivell centimètric de l'amplada i l'altura del moble.
- Controladament d'entre tres nivells de profunditat.
- Controladament d'entre tres tipus de cobertura per cada apartat del moble.
- Controladament els colors del moble.

En aquest projecte s'han complert els requisits i ampliat amb moltes més funcionalitats.

## Tecnologia

Aquest projecte està desenvolupat totalment en base a JavaScript tant per la part de client com de servidor.

### Client

- A la pàgina web hi haurà un fitxer **HTML5** amb estils proporcionats per fitxers **CSS**. Aquest fitxer tindrà enllaçats scripts de **JavaScript** que manegen les dades dels inputs. Els scripts interactuaran canviant en temps real el contingut del document HTML.
- En aquest fitxer hi haurà un element “<canvas>” o “WebGL”, segons la compatibilitat del navegador. Aquest element és controlat pels scripts de **JavaScript**, en el qual mitjançant la llibreria **three.js** es renderitzen els models 3D corresponents a les dades dels inputs de l'usuari.
- Per a realitzar les animacions hem fet servir **TweenJS**. Aquesta eina és per realitzar el càlcul del moviment dels objectes que efectuen les animacions.
- Els esdeveniments del client són gestionats amb **jQuery**.
- Les contrasenyes són convertides en “hash” gràcies a **BCrypt**.
- Els tests es realitzen amb els navegadors **Google Chrome** i **Mozilla Firefox**.

### Versions

- **HTML:** v 5.2
- **CSS:** v 4
- **Three.js:** r 97
- **JavaScript:** ECMAScript 2018
- **jQuery:** v 3.3.1
- **BCrypt:** v 3.0.6
- **Chrome:** v 65.0.33.25.146
- **Firefox:** v 66.0.5
- **TweenJS:** v 16.7.6

## Servidor

- Aquesta part estarà desenvolupada amb el llenguatge **JavaScript**, amb l'entorn que proporciona **NodeJS** junt amb **NPM**. Dins d'aquest entorn s'empren mòduls com:
  - **Express**: per tal d'iniciar el servidor, administrar les rutes i el tractament de les peticions i respostes.
  - **Mongoose**: per a interactuar amb la base de dades de **MongoDB**.
- Per a desenvolupar el projecte hem fet servir **MongoDB** amb **Mongo Compass Community** com a gestor amb interfície gràfica.

## Versions

- **NodeJS**: v 10.15.3
- **NPM**: v 6.4.1
- **Express**: v 4.17.0
- **MongoDB**: v 4.0
- **Mongo CC**: v 1.18.0

# Arquitectura

Tenint en compte el context en el qual vam desenvolupar aquest projecte, sense nocions prèvies de programació orientada a objectes, ni experiència prèvia amb JavaScript, l'arquitectura per la part del client es basa en objectes de JavaScript sense fer servir cap sistema de classes ni herències.

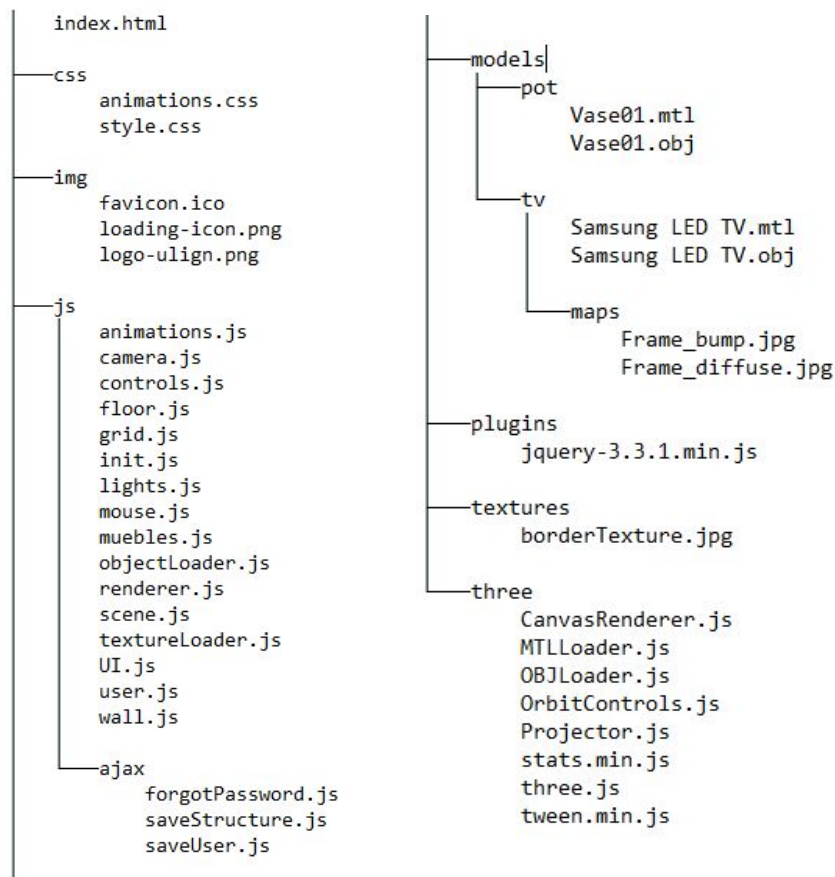
Per la part del servidor tenim una organització molt bàsica, ja que té un pes menor que la part del client per a l'aplicació.

## Client

### Estructura

Tot el codi pertinent a la part del client es troba al directori `"/client"`. Dins d'aquest directori hi ha els subdirectoris següents:

- `"/js"`: per al codi de JavaScript.
  - Dins d'aquest directori hi ha els scripts de l'aplicació i també el subdirectori `"/ajax"`, on es troben els nostres scripts que comuniquen amb el servidor.
- `"/css"`: per als fulls d'estil CSS.
- `"/img"`: per a fitxers d'imatges com icones.
- `"/models"`: per als models 3D externs, on hi ha fitxers d'extensió `".OBJ"` (per a la geometria) i `".MTL"` (per a la textura).
- `"/plugins"`: on trobem scripts externs, com jQuery.
- `"/textures"`: on trobem les textures que fem servir per crear els nostres mobles.
- `"/three"`: amb els scripts necessaris de la llibreria de three.js per a l'aplicació.



*Figura representant l'estructura d'arbre del directori "/client"*

## Funcionament

L'estructura es basa en un document HTML estàtic i principal que reuneix tots els scripts que contenen els objectes necessaris per a fer funcionar l'aplicació. El directori on es troben aquests scripts és "/client/js".

Els noms dels scripts són realment descriptius, ja que per exemple, l'arxiu "muebles.js" conté l'objecte "MUEBLE" que permet crear un nou model 3D amb les dimensions adients, i així amb tots els arxius.

L'script de l'arxiu "init.js" proporciona l'objecte "APP", que conté com a propietats les direccions en memòria dels altres objectes com "MUEBLE", per exemple. Des de "APP" s'inicien i es realitzen les operacions necessàries tant per iniciar l'aplicació com per gestionar els esdeveniments que dispara el client. Al directori "js/ajax" es troben els scripts que comuniquen amb el servidor.

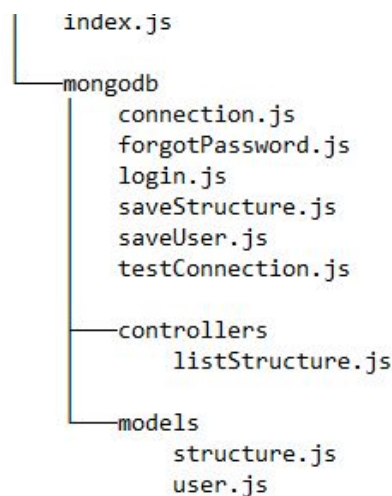


# Servidor

## Estructura

Les tecnologies que hem emprat ens han permès treballar amb una dinàmica de funcions asíncrones. Tot el codi pertinent a la part del client es troba al directori “/server”. Dins d’aquest directori hi ha els subdirectoris següents:

- “./”: al propi directori hi trobem l’arxiu principal “index.js” amb el qual s’inicia el servidor.
- “/mongodb”: que conté els scripts que connecten i interactuen amb la base de dades.



*Figura representant l'estructura d'arbre del directori “/server”*

## Funcionament

El servidor està format per dues parts principals: la gestió de rutes (amb **ExpressJS**) i la comunicació amb la base de dades de MongoDB (amb **Mongoose**).

L’arxiu “index.js” inicia el servidor i administra les rutes, amb les peticions que s’envien a aquestes junt amb les seves respectives respostes.

Quan s’envia una petició a la ruta arrel, el servidor envia l’arxiu HTML principal que conté totes les importacions de scripts que ja hem comentat. Així, quan el client entra a la pàgina hi veurà l’aplicació.

```
app.get('/', function(req, res) {
  app.use(express.static(__dirname + "../client"));
  res.sendFile(path.join(__dirname + "../client/index.html"));
});
```

*Figura extreta del codi de “/server/index.js”, pertinent a la ruta de l’arrel*

Per veure un altre exemple diferent, quan s'envia una petició a la ruta "login" en canvi, s'administren les dades del cos de la petició que se li ha enviat, i segons aquestes s'envia una resposta que permet fer el "login", o que el denega.

```
app.post('/login', function(req, res) {  
  login.login(req.body, res); //responses are handled by /mongodb/login.js  
});
```

*Figura extreta del codi de "/server/index.js", pertinent a la ruta "/login"*

```
exports.login = (user, serverResponse) => {  
  userModel.find({email: user.email}, (err, res) => {  
    if (err) return handleError(err);  
  
    res.length < 1 ? allow(serverResponse, user) : checkPass(serverResponse, user);  
  });  
};
```

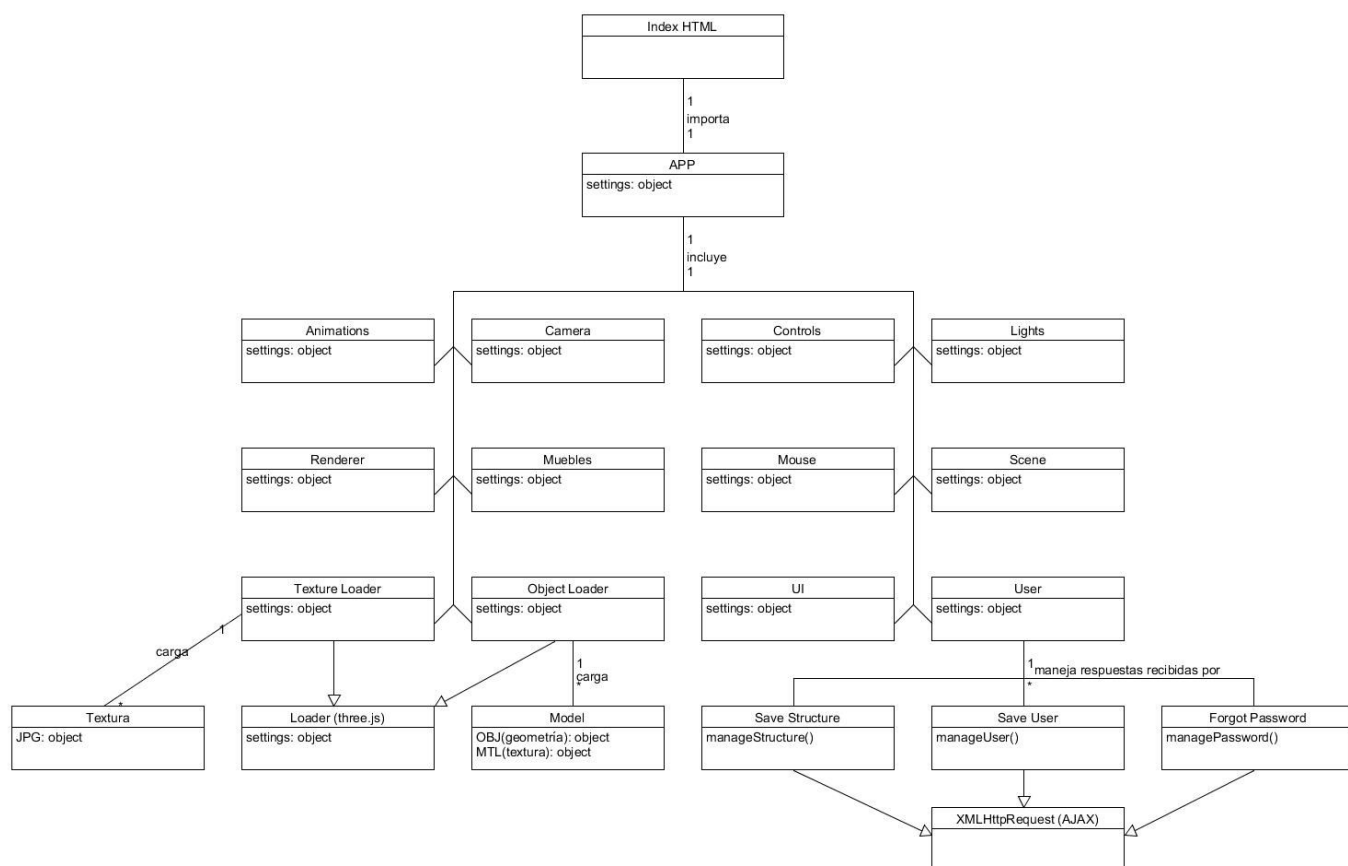
*Figura extreta del codi de "/server/mongodb/login.js", que administra la petició i la resposta*

# Diagrames UML

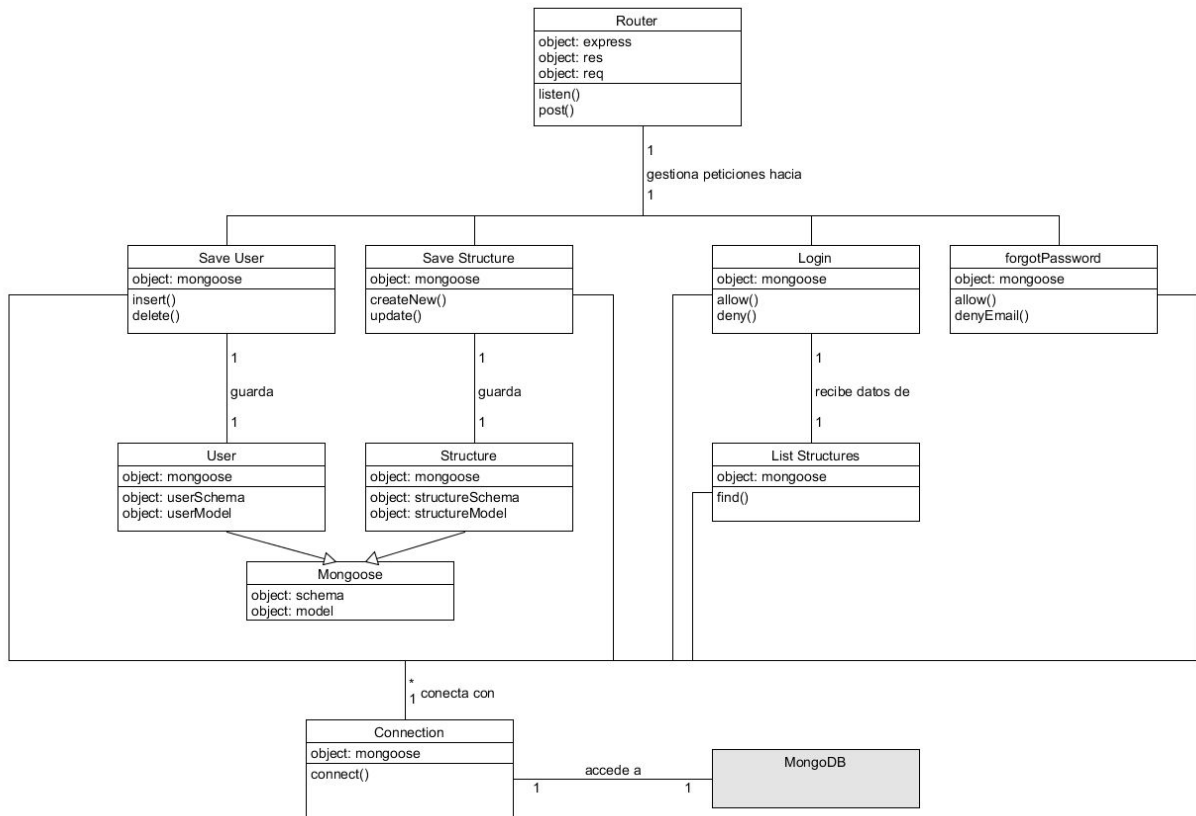
Per acabar el projecte hem fet servir diagrames de classes i casos d'ús d'UML.

## Classes

### Client

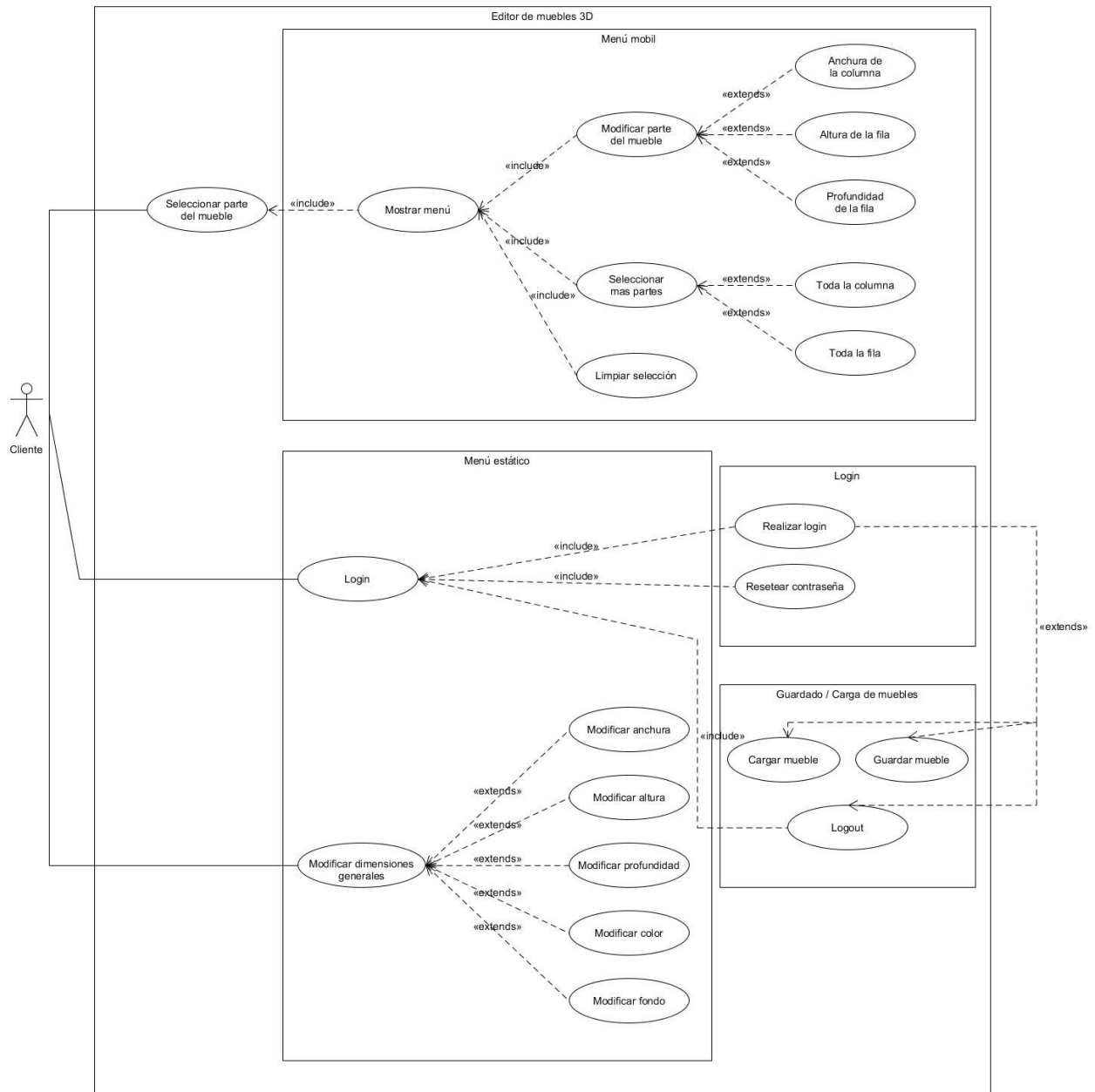


## Servidor



# Casos d'ús

A continuació el diagrama de casos d'ús de l'aplicació:



# Innovació

Els projectes basats en three.js no són molt populars en el món dels negocis, ja que tenen aplicacions molt puntuals i estètiques. Tot i això, són vistosos, innovadors i generalment creen una bona impressió entre els usuaris.

Els nostres objectius són crear un producte amb les següents característiques:

- Llibertat d'edició
- Guardat i càrrega de models
- Velocitat
- Disseny simple i modern

## Llibertat d'edició

El que busquem amb aquest projecte és un editor amb més llibertat de disseny que els que ja existeixen, referent a dimensions, colors, materials, etcètera, i amb addicions estètiques de models 3D afegibles (com una televisió per exemple) per a millorar l'escena.

Respecte a les dimensions es poden determinar de la següent manera:

- Altura:
  - Des del menú estàtic de la pàgina s'altera l'altura general del moble.
  - Des del menú dinàmic es pot alterar l'altura de la fila que hem seleccionat.
- Amplada:
  - Des del menú estàtic de la pàgina s'altera l'amplada general del moble.
  - Des del menú dinàmic es pot alterar l'amplada de la columna que hem seleccionat.
- Profunditat:
  - Des del menú estàtic de la pàgina s'altera la profunditat general del moble.
  - Des del menú dinàmic es pot alterar la profunditat de la fila que hem seleccionat.

## **Guardat i càrrega de models**

Hem afegit també un sistema de “login” que permet als clients guardar i carregar de forma ràpida i amb llibertat els mobles que han dissenyat fins a un màxim de 5 models. En les finestres per administrar aquestes funcions apareixen les dimensions, hora d’últim guardat i color. Internament, es carreguen totes les dades necessàries per recrear el model guardat.

## **Velocitat**

És important que sigui ràpid i els canvis es vegin amb claredat. Aconseguirem que es generi i es renderitzi el model que s’està editant del moble a temps real, responent als canvis de dimensions del client de forma aparentment instantània.

La velocitat de càrrega també és important, i tot i que depèn de la velocitat de descàrrega del client, la gestió de càrregues és important. En aquest projecte hem afegit una capa amb una animació de “loading” mentre es prepara l’escena 3D. Hem fet que aquest “loading” duri només fins que es carrega el moble, i no els accessoris de models 3D externs com són la televisió. Aquests, es poden afegir posteriorment, de manera que no incrementen el primer temps de càrrega visible per a l’usuari.

## Disseny

Per a l'estètica d'aquest projecte hem fet servir interfícies gràfiques d'usuari senzilles i intuïtives. Hem emprat pocs colors, i tots els que són presents tenen una funció clara. Molta part del disseny està inspirada en la web de Tylko.

El color general és el blanc i tons de grisos molt clars.

*Figures dels menús per a editar el moble per part de l'usuari (a l'esquerra el menú de l'apartat seleccionat, a la dreta el menú del moble en general).*

*Figures de les finestres de "login" (esquerra) i del menú de gestió de models (dreta).*

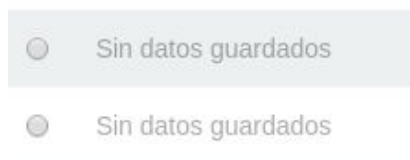


Fem servir grisos clars amb poca opacitat per deixar clar que una opció no està permesa temporalment, en contraposició fem servir grisos foscos per indicar que l'opció està seleccionada.



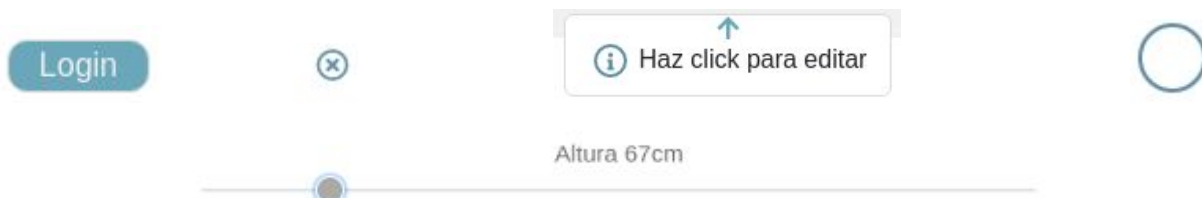
*Figura d'un apartat del menú d'edició de l'apartat seleccionat del moble*

També fem servir grisos pàl·lids amb poca opacitat per a mostrar que encara no hi ha dades per a carregar.



*Figura del missatge del menú de gestió de models*

Fem servir el color blau per ressaltar elements d'ajuda, o altres icones. També el fem servir en els botons al registrar l'esdeveniment "hover" o amb un "border" de seleccionat (per a les rodones de colors). El blau (#6ba7ba) és el color corporatiu i ajuda a aconseguir una estètica pròpia.



*Figures de les icones, elements d'input, etc.*

També s'empra aquest blau per marcar el quadrant del moble pel que s'està passant el ratolí, de forma molt transparent. Quan es clica a sobre, queda seleccionat amb el mateix color amb més opacitat.



*Figura d'apartat de moble en "hover"*



*Figura d'apartat de moble seleccionat*

Fem servir el color vermell només per a mostrar un error de validació.



*Figura de la finestra de “login” anunciant un error per part de la contrasenya*

Els colors variables en rodones d'un tamany més gran corresponen al color del moble (seleccionat amb el “border” del color blau corporatiu com hem indicat prèviament).



*Figura dels colors del menú d'edició del moble en general*

Per a diferenciar entre diferents files de la taula de guardat i càrrega de models, o el propi “login” fem servir fons de color gris molt lleu.



*Figura del menú de gestió de models*

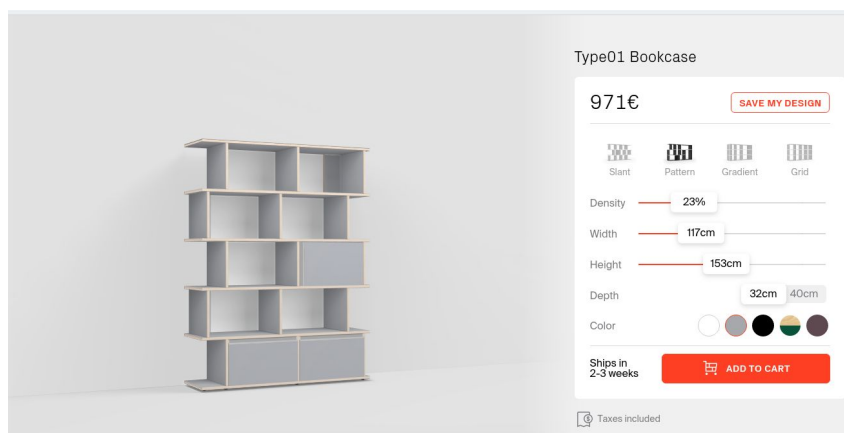
# Viabilitat del projecte

El servei que volem oferir té punts forts, però també s'ha de tenir en compte els costos i l'estat actual dels competidors per tal de realitzar un producte competitiu.

## Competidors

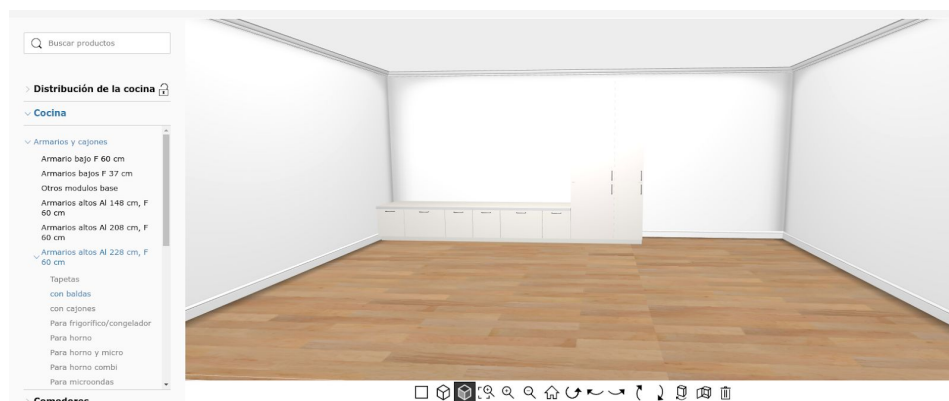
Altres projectes del mateix tipus que aquest ja es troben en el mercat, com a la plana WEB de **Tylko**, per exemple, o el visualitzador de mobles d'**IKEA**.

- **Tylko**: té animacions dinàmiques però té moltes limitacions a l'hora de dissenyar un moble, com no poder canviar el tipus de porta o calaix d'un apartat concret, o l'amplada d'una columna concreta.



*Figura extreta de l'editor en línia de Tylko*

- **IKEA**: l'eina que tenen és bona per a dissenyar una habitació, o part d'una habitació, de forma general. Tenen molts models en "stock", diferents i amb materials ben aconseguits. Els temps de càrrega són llargs. La part negativa que veiem i que nosaltres aportem és la llibertat de disseny del moble en si que es vol comprar. Tampoc té animacions dinàmiques, que nosaltres sí que aportem.



*Figura extreta de l'editor en línia d'Ikea*

## Costs

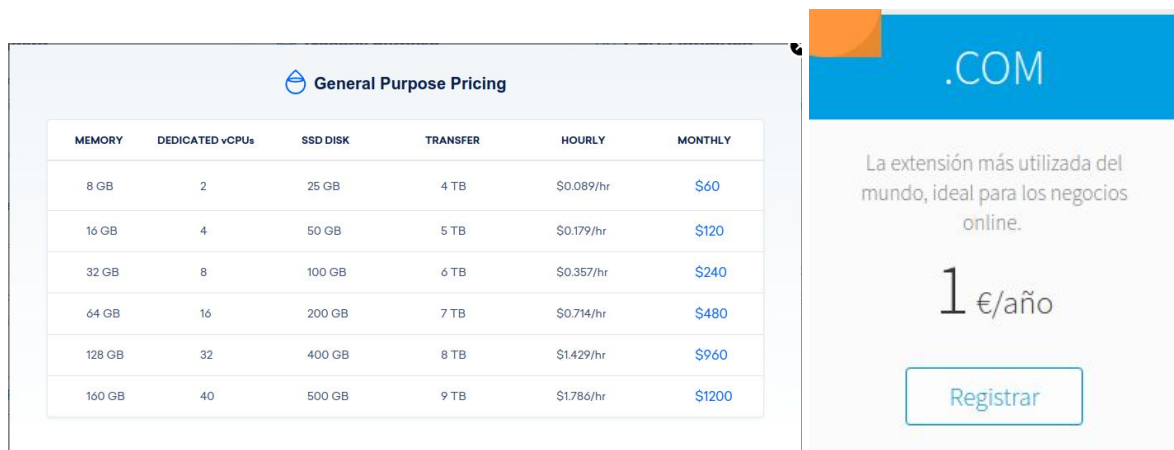
Com a cost variable tenim les hores de feina que implica el desenvolupament de l'aplicació. Dins d'aquestes hores no incloem les hores de recerca i aprenentatge de noves tecnologies com per exemple three.js i també les hores de creació de documentació. Estimem que aquest projecte ha portat uns dos mesos de desenvolupament, i que el nostre sou és el sou mitjà d'un programador júnior.



*Figura extreta de "indeed.es"*

Entenent que aquest sou anual s'ha de dividir entre les 14 pagues de l'any, queden **1.395'89€** mensuals bruts. Això implica que en dos mesos, el cost puja a **2.791'71€**.

Per altra banda, els costs fixes que comporta tenir l'aplicació en servei, com són la quota del servidor o pagar els drets d'un domini els calculem de forma aproximada, i són d'uns **53'8€** al mes.



*Figures extretes de "try.digitalocean.com"*

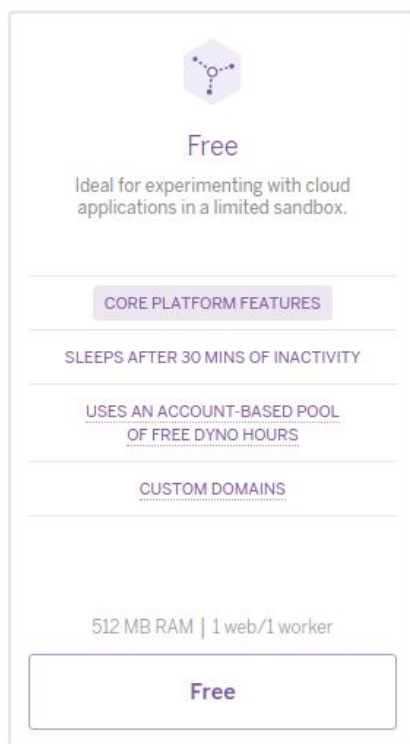
Tot i això, sempre hi ha la possibilitat de fer servir alternatives gratuïtes amb prestacions inferiors. Si no féssim servir aquests serveis gratuïts, el cost total del projecte serà d'aproximadament:

- **2791.71€** bruts d'entrada.
- **53'8€** mensuals.

## Serveis gratuïts

### Servidor

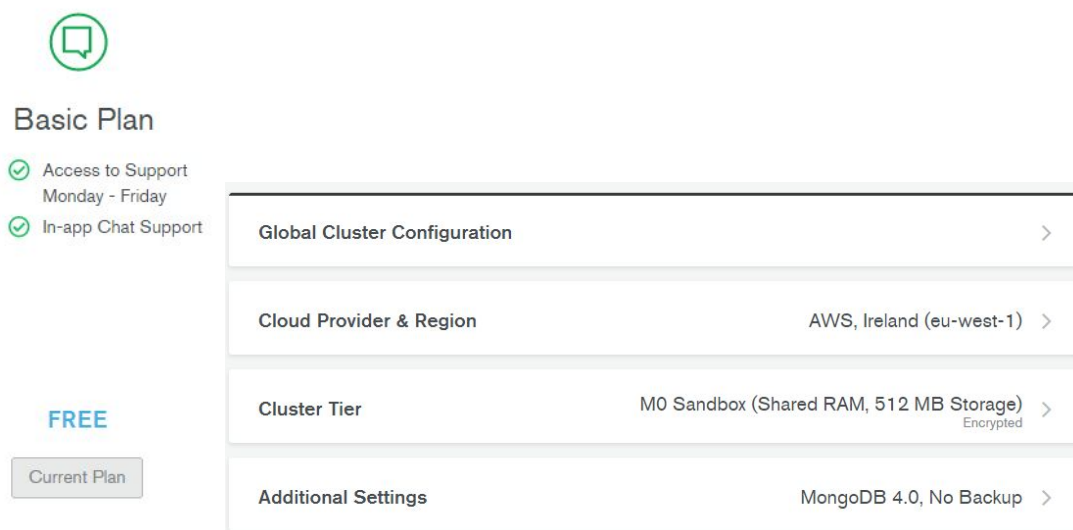
Sense evitar el cost del desenvolupament, podem evitar el cost inicial i mensual del "hosting" del servidor si fem servir un servei gratuït. Nosaltres hem escollit el que ens ofereix Heroku, amb 512MB de RAM i els límits i característiques següents:



*Figura extreta dels plans oferts a "heroku.com"*

## Base de dades

Heroku només ofereix un servidor gratuït sense bases de dades, de manera que hem de fer servir un altra opció gratuïta com és MongoDB Atlas, que ens garanteix una base de dades de com a màxim 0.5GB de memòria. Hi ha diferents opcions quant a la localització del servidor, hem escollit Irlanda, ja que era l'opció més pròxima a nosaltres.



The screenshot shows the 'Basic Plan' configuration for MongoDB Atlas. It includes a green chat icon, the text 'Basic Plan', and two green checkmarks indicating 'Access to Support Monday - Friday' and 'In-app Chat Support'. A blue 'FREE' label is present, along with a 'Current Plan' button. The configuration table below lists the following settings:

Global Cluster Configuration	>
Cloud Provider & Region	AWS, Ireland (eu-west-1) >
Cluster Tier	M0 Sandbox (Shared RAM, 512 MB Storage) Encrypted >
Additional Settings	MongoDB 4.0, No Backup >

*Oferta gratuïta de MongoDB Atlas*

## Limitacions

Prèviament hem parlat de la llibertat que volem aportar amb aquest editor. Tota llibertat ha de tenir uns límits per tal d'afavorir la claredat i l'experiència de l'usuari alhora que una millor gestió per part del nostre programa.

Hem d'aclarir que totes les dimensions limitades que veurem a continuació es poden alterar de manera que per un altre tipus de moble, es pot fer que les altures predeterminades fossin de 30cm, 35cm i 45cm, per exemple.

## Profunditat

Les profunditats d'una fila sempre dependran de la fila inferior i superior, ja que mai poden tenir una major profunditat que les inferiors, ni tampoc inferior que la de les superiors, fet que comprometria l'estabilitat del moble.

## **Altura**

Les altures estan limitades a ser de tres tipus, o bé de 19cm, de 26cm o 38cm, per demanda del client. L'altura màxima total del moble és de 250cm, també per demanda del client.

## **Amplada**

L'amplada màxima total del moble és de 262cm, per demanda del client.

## **Portes i calaixos**

En una altura mínima com és la de 19cm no poden haver-hi portes, però sí calaixos, per demanda del client. Tampoc poden haver-hi calaixos a més de 160cm d'altura, també per demanda del client i per assegurar la seguretat dels usuaris del moble.

## **Colors**

Hem limitat el nombre de colors disponibles a 5. Aquests són variables segons l'"stock" del client i dependrà de la seva decisió. Com a demostració, en posem 5 de possibles.

## **Selecció ràpida**

La selecció ràpida es basa en seleccionar la fila corresponent a l'apartat que ja hem seleccionat. De la mateixa manera es pot seleccionar tota la columna. Si ja hi ha més d'un apartat seleccionat, es deshabilita aquesta opció.

# **Planificació del projecte**

Amb raó del desenvolupament del projecte de forma individual, la planificació i organització ha estat més simple i seqüencial.

## **Dinàmica**

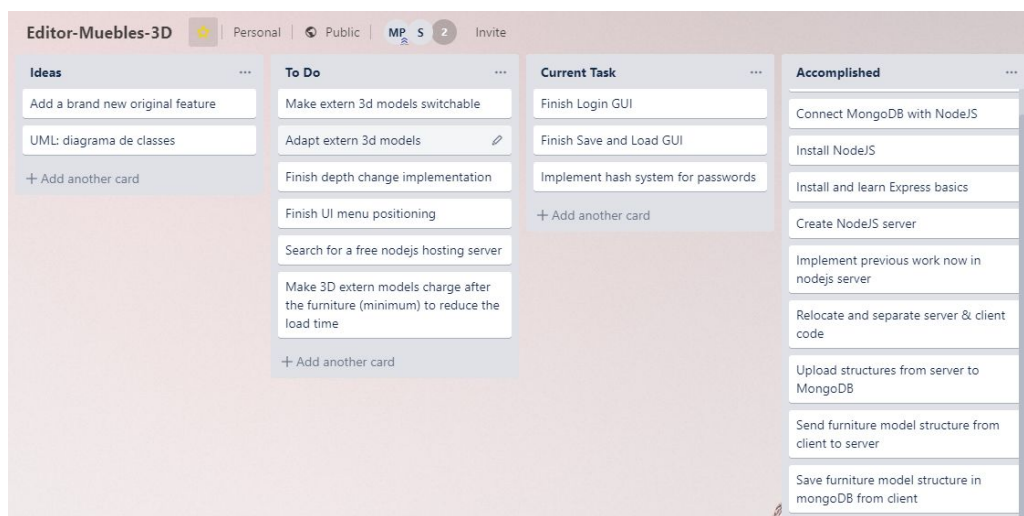
La planificació i organització que hem seguit per a fer aquest projecte ha estat la dinàmica àgil de SCRUM. Els "sprints" han estat setmanals. En total han estat quatre "sprints" on l'últim és l'entrega final del projecte. Per a cada "sprint" s'han elaborat informes amb els mètodes per a iniciar l'aplicació, les noves característiques de la versió que s'ha presentat, les característiques que es presentaran en la pròxima versió, les conclusions, mostres de la versió actual i annexos.

# Eines

## Organització

L'eina d'organització que hem fet servir en aquest treball és Trello. Aquesta eina serveix per tenir un “dashboard” visual personalitzable. Abans de cada jornada de treball, mirem les tasques que estem fent, i si no n'hi ha perquè prèviament les hem acabat, n'afegim de les tasques que queden per començar.

Per a afegir les tasques que s'han de començar, reflexionem sobre els defectes o mancances que té l'aplicació després de fer un test general de la vista client. Sovint una tasca que presenta dificultats necessita desglossar-se en diferents tasques.



*Figura extreta de “trello.com” del nostre propi “dashboard” per al projecte*

A vegades també hem fet pluges d'idees per afegir noves tasques més enllà de solucionar les que ja tenim plantejades.

## Control de versions

Per al control i gestió de versions hem fet servir **Git**. L'hem fet servir tant a nivell de línia de comandes com per interfície gràfica amb **GitHub**. És una eina excel·lent per a mantenir un projecte a l'abast de qualsevol màquina amb el repositori al núvol, amb un control de cada canvi en el projecte amb comentaris, i gestió de versions amb “tags”.



## Testing

Després d'implementar qualsevol nova funcionalitat o correcció, hem dedicat una tasca de "testing" de l'aplicació per tal de trobar possibles errors i correccions a fer. A través de la vista client hem provat les opcions que un usuari té i que poden afectar el funcionament de parts prèviament implementades. Si veiem que s'han d'efectuar correccions, les afegim com a noves tasques; si no pugem els canvis al repositori del projecte.

Durant els últims estadis del desenvolupament, un cop l'aplicació estava desplegada, vam compartir-la amb persones properes per tal d'obtenir "feedback" d'usuaris que no estaven familiaritzats amb l'aplicació. A partir d'aquest tipus de proves també hem pogut corregir defectes.

## Versions

- **Git:** v 2.17.1

# Conclusions

## Formació DUAL

Durant el període de feina a l'empresa eDataseft he treballat amb diferents tecnologies i llenguatges, aprenent de forma totalment autodidacta. Vaig aprendre a programar per part del client amb **JavaScript**, complementant amb el "framework" de **jQuery**. Amb aquesta base vaig aprendre a fer servir la llibreria de **three.js** i **TweenJS**. Per part del servidor vaig realitzar tasques amb el llenguatge **PHP**. Pel que fa als entorns i editors de text vaig practicar amb **Visual Studio Code** i **Sublime Text 3**. En el projecte que presento amb aquest document he fet servir també altres tecnologies noves i diferents a part de les que he comentat en aquest apartat.

De forma més general també he après a com funcionar dins d'una petita empresa més enllà del desenvolupament de software, realitzant tasques com mantenir el contacte i atenant a clients de forma directa.

## Projecte final

Gràcies a aquest projecte he tingut la oportunitat d'explorar tecnologies punteres i explotar el potencial que aquestes ofereixen. Això m'ha ajudat a aprendre i optimitzar millor el treball amb eines potents que avui en dia són demandades en el món laboral. També m'ha obligat a mantenir un ritme de treball constant amb la dinàmica àgil SCRUM, per tal de poder tenir una entrega sòlida després de cada "sprint" setmanal. Aquesta dinàmica ha estat realment productiva i és una pràctica que mantindré.

## Millores de futur i errors

Com he comentat prèviament en aquest document, aquest projecte ha estat una mena d'experiment amb el que he après i practicat molts tipus de tecnologies, totes relacionades amb **JavaScript**. Aquesta inexperiència ha comportat problemes com la falta d'organització en projectes a una escala més gran de la que estic acostumat a realitzar. La organització i planificació de l'arquitectura és el principal camp on es veu una necessitat de millora en el futur. Òbviament, també en l'optimització del codi escrit durant l'inici d'aprenentatge del llenguatge i tecnologies.

Tot i la gran quantitat de coneixements assolits en diferents camps, és ben clar que he de seguir aprenent i millorant en molts aspectes, aprofitant els errors comesos com a base per a construir millors projectes en el futur.

# Webgrafia

## Costs

Comprar un domini:

<https://www.nominalia.com/dominios/>

Contractar un servidor amb NodeJS:

<https://try.digitalocean.com/cloud-hosting/>

Sou mitjà:

<https://www.indeed.es/salaries/Programador/a-junior-Salaries>

## Serveis gratuïts

Heroku:

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

MongoDB.Atlas:

<https://cloud.mongodb.com/>

TurboSquid:

<https://www.turbosquid.com/Search/3D-Models/free>

## Documentació

Three.js:

<https://threejs.org/docs/index.html>

TweenJS:

<https://github.com/tweenjs/tween.js/>

MongoDB:

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

NodeJS:

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

ExpressJS:

<https://expressjs.com/>

BCrypt:

<https://www.npmjs.com/package/bcrypt>

Heroku NodeJS:

<https://devcenter.heroku.com/categories/nodejs-support>

## **Planificació del projecte**

SCRUM:

<https://www.scrum.org/resources/what-is-scrum>

Trello:

<https://trello.com/b/qpcFNCUU/editor-muebles-3d>

## **Competidors**

Tylko:

<https://tylko.com/shelf/bookcases>

Ikea:

<https://www.ikea.com/addon-app/platsa/latest/#/standard/es/es/?>

## **Desplegament**

Desplegament de l'aplicació d'aquest projecte:

<https://furniture-editor.herokuapp.com>

Repositori GIT:

<https://github.com/marcpascualsanchez/Furniture-Editor-3D.git>