

# Editor de Mobles 3D - Sprint 1/4

Marc Pascual Sánchez

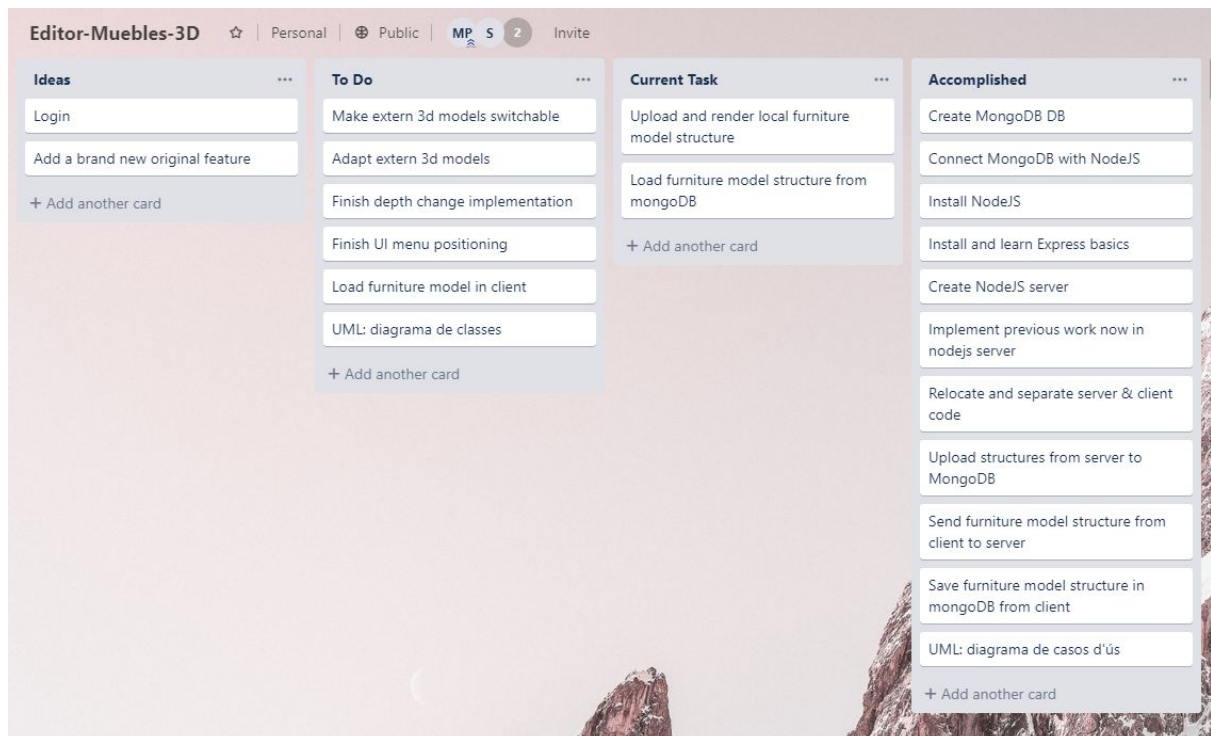
## Guía d'implementació

Partint de disposar del repositori del projecte (link a la Webgrafia) i descarregar-lo, indiquem els passos a seguir a continuació. Aquest projecte necessita tenir instal·lat NodeJS i MongoDB (servei aixecat al port per defecte, el 27017).

1. Disposar-nos al directori principal de l'aplicació "Furniture-Editor-3D".
2. Fer servir la comanda "node server/index.js" per iniciar el servidor.
3. Al cercador web escriure la URL "localhost" (port 80, que és el es troba per defecte).

## SCRUM

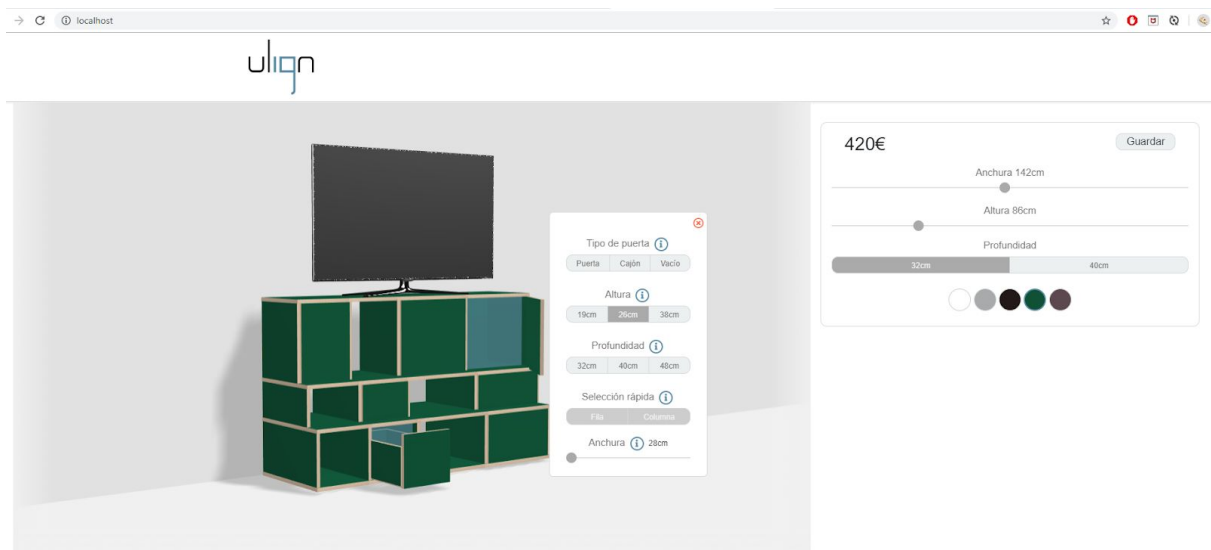
Aquest projecte es realitza de forma individual de manera que les reunions no són possibles però si les reflexions i organització pròpia abans de cada jornada de treball. Per aconseguir un hàbit eficient hem fet servir Trello com a pissarra per a gestionar tasques:



## Versió actual

Les implementacions en aquesta versió són bàsiques i enfocades a la part del servidor. També hem realitzat un diagrama de casos UML. A continuació el llistat d'implementacions:

- El muntatge del servidor amb NodeJS, fent servir els paquets d'Express per a gestionar les rutes i també les peticions HTTP, de moment de guardat dels models, per exemple.
- També el muntatge de la base de dades de MongoDB i l'estructura dels objectes que guardarem a les col·leccions.
- Les dades del programa han estat editades per adaptar-s'hi.
- Per a la connexió amb MongoDB hem afegit i fet servir el paquet de Mongoose.
- Actualment es poden guardar a MongoDB les configuracions de l'estructura del moble que s'ha editat en el client.



## Próxima versió

A continuació es llisten els punts que tenim previstos implementar cap a les futures versions:

- Habilitar la gestió de sessions (login) per controlar també els models guardats de cada usuari.
- Implementar la càrrega de models des de la base de dades.
- Actualment el servei s'ha d'aixecar de forma manual i de forma local. En el futur, volem que aquest servidor sigui públic.
- Millorar la robustesa de l'aplicació per la part del client.
- Millorar la gestió de models 3D externs (com la televisió de la figura anterior).

## Annexos:

Dades extretes del client, visualitzades a MongoDB Compass (interfície gràfica de MongoDB):

MongoDB Compass Community - localhost:27017/furniture.structures

Connect View Collection Help

My Cluster

localhost:27017 STANDALONE

5 DBS 3 COLLECTIONS

filter

> admin

> config

▼ furniture

structures

> local

> nodeTest

furniture.structures DOCUMENTS 2

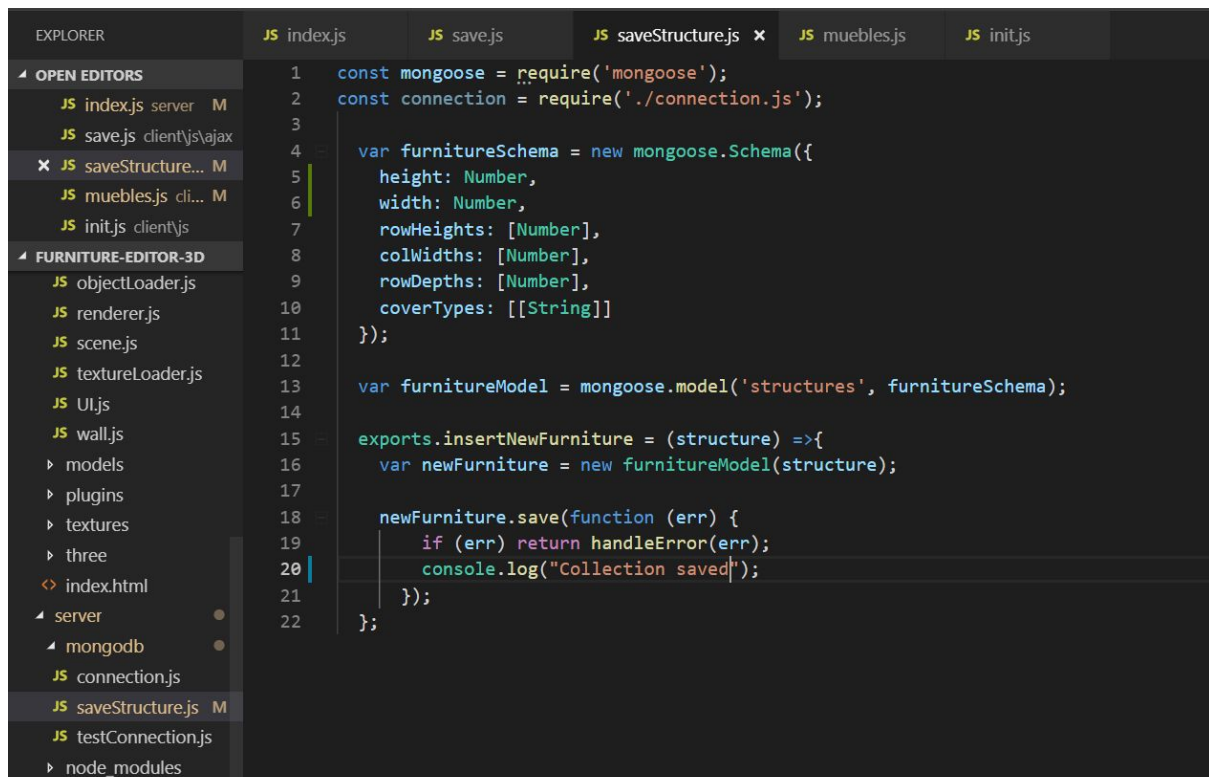
Documents Aggregations Explain Plan

FILTER

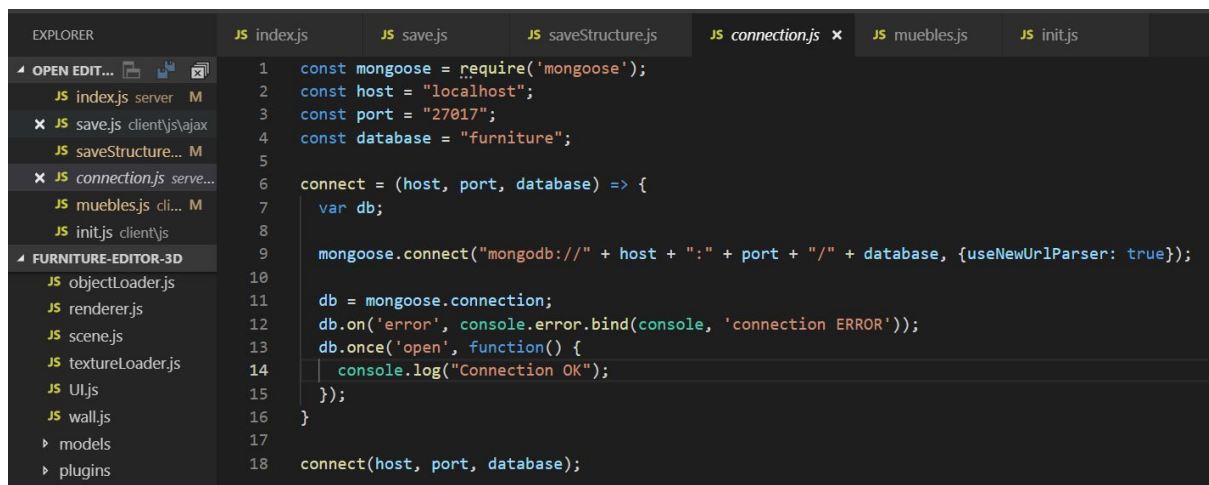
INSERT DOCUMENT VIEW LIST TABLE

```
{
  "_id": ObjectId("5cd5cc87175c9634ec43cdc3"),
  "rowHeights": Array
    0: 0.38
    1: 0.19
    2: 0.26
  "colWidths": Array
    0: 0.28
    1: 0.28
  "rowDepths": Array
    0: 0.48
  "coverTypes": Array
    0: Array
      0: "Drawers"
      1: "Drawers"
      2: "None"
      3: "Drawers"
      4: "None"
      5: "Drawers"
      6: "None"
      7: "Drawers"
      8: "None"
      9: "Drawers"
      10: "None"
    1: Array
      0: "None"
      1: "Drawers"
      2: "None"
      3: "Drawers"
      4: "None"
      5: "Drawers"
```

## Inserció de dades emprant Mongoose:

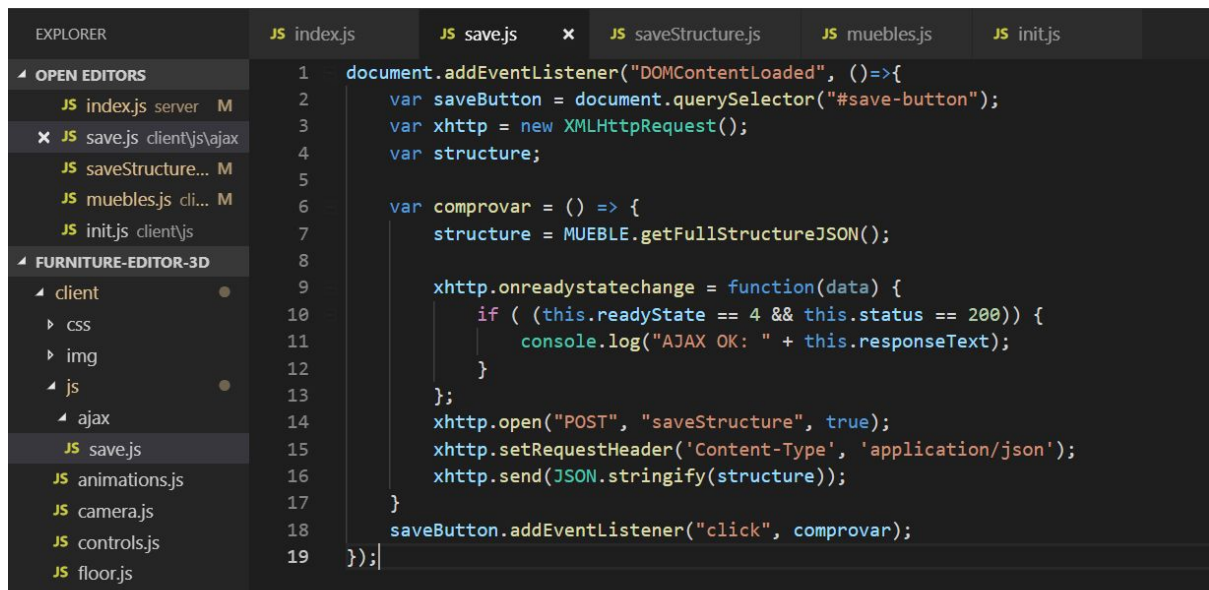


```
1 const mongoose = require('mongoose');
2 const connection = require('./connection.js');
3
4 var furnitureSchema = new mongoose.Schema({
5   height: Number,
6   width: Number,
7   rowHeights: [Number],
8   colWidths: [Number],
9   rowDepths: [Number],
10  coverTypes: [[String]]
11 });
12
13 var furnitureModel = mongoose.model('structures', furnitureSchema);
14
15 exports.insertNewFurniture = (structure) =>{
16   var newFurniture = new furnitureModel(structure);
17
18   newFurniture.save(function (err) {
19     if (err) return handleError(err);
20     console.log("Collection saved");
21   });
22 }
```



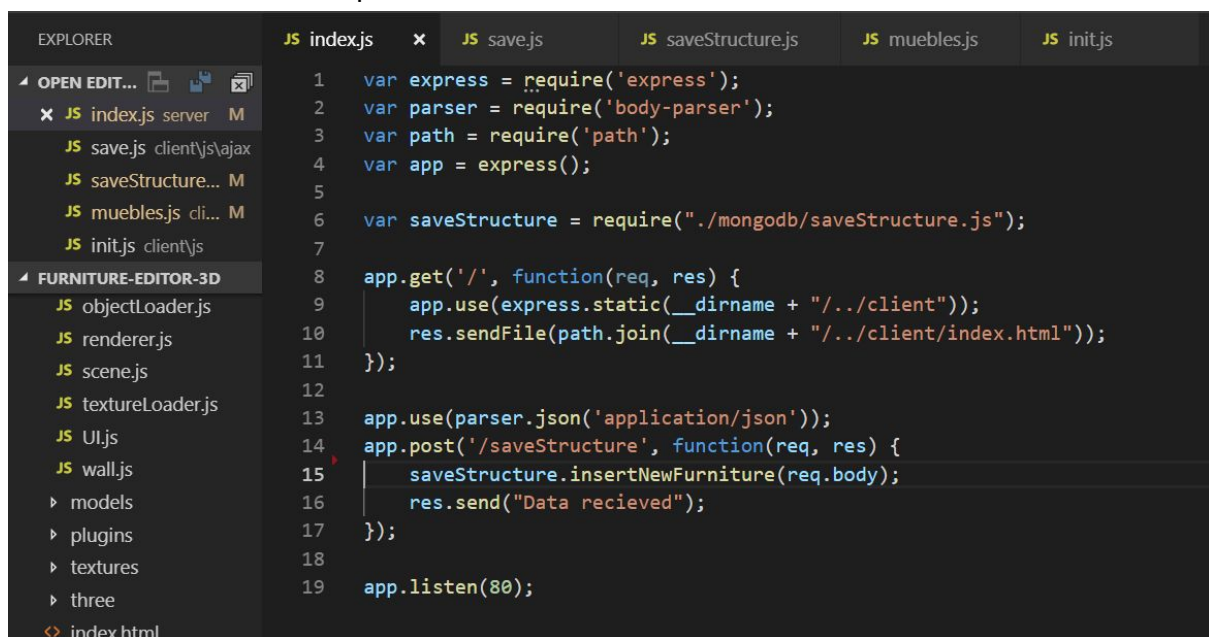
```
1 const mongoose = require('mongoose');
2 const host = "localhost";
3 const port = "27017";
4 const database = "furniture";
5
6 connect = (host, port, database) => {
7   var db;
8
9   mongoose.connect("mongodb://" + host + ":" + port + "/" + database, {useNewUrlParser: true});
10
11   db = mongoose.connection;
12   db.on('error', console.error.bind(console, 'connection ERROR'));
13   db.once('open', function() {
14     console.log("Connection OK");
15   });
16 }
17
18 connect(host, port, database);
```

Enviament de dades al servidor per part del client emprant AJAX:



```
1 document.addEventListener("DOMContentLoaded", ()=>{
2     var saveButton = document.querySelector("#save-button");
3     var xhttp = new XMLHttpRequest();
4     var structure;
5
6     var comprobar = () => {
7         structure = MUEBLE.getFullStructureJSON();
8
9         xhttp.onreadystatechange = function(data) {
10             if ( (this.readyState == 4 && this.status == 200)) {
11                 console.log("AJAX OK: " + this.responseText);
12             }
13         };
14         xhttp.open("POST", "saveStructure", true);
15         xhttp.setRequestHeader('Content-Type', 'application/json');
16         xhttp.send(JSON.stringify(structure));
17     }
18     saveButton.addEventListener("click", comprobar);
19 });
```

Gestió del servidor amb Express:



```
1 var express = require('express');
2 var parser = require('body-parser');
3 var path = require('path');
4 var app = express();
5
6 var saveStructure = require("./mongodb/saveStructure.js");
7
8 app.get('/', function(req, res) {
9     app.use(express.static(__dirname + "../client"));
10    res.sendFile(path.join(__dirname + "../client/index.html"));
11 });
12
13 app.use(parser.json('application/json'));
14 app.post('/saveStructure', function(req, res) {
15     saveStructure.insertNewFurniture(req.body);
16     res.send("Data recieved");
17 });
18
19 app.listen(80);
```

# Webgrafia

## **Eines:**

*Repositori GIT:*

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

*Trello:*

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

## **Documentació:**

*MongoDB:*

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

*NodeJS:*

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

*Express:*

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

*Mongoose:*

<https://mongoosejs.com/docs/guide.html>