



# SynapTree

**Llorenç Coll Hernández**  
Grau en Multimedia  
Arduino

**Antoni Morell Pérez**  
**Pere Tuset Peiró**

6 de Juny de 2021



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

<b>Títol del treball:</b>	<i>SynapTree</i>
<b>Nom de l'autor:</b>	<i>Llorenç Coll Hernández</i>
<b>Nom del consultor/a:</b>	<i>Antoni Morell Pérez</i>
<b>Nom del PRA:</b>	<i>Pere Tuset Peiró</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>06/2021</i>
<b>Titulació o programa:</b>	<i>Grau en Multimedia</i>
<b>Àrea del Treball Final:</b>	<i>Arduino</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Arduino, sensors, web</i>

**Resum del Treball (màxim 250 paraules):** *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

Amb el projecte duc a terme el disseny i implementació d'un lector de chips RFID capaç de geoposicionar-se, connectar-se a Internet i gestionar una base de dades externa a través d'una aplicació web que permet dur a terme una gestió arbòria.

Aquesta gestió arbòria, o catalogació, es farà a partir de la col·locació d'etiquetes RFID en els arbres a catalogar, de manera que quedaran identificats unívocament i també posicionats amb les seves coordenades. Es guardarà informació seva i també de les intervencions que un tècnic especialista dugui a terme per aquest arbre.

He emprat una placa NodeMCU, la que proveeix de connexió a Internet al lector a través d'una xarxa WiFi i un node 3G/4G, un mòdul GPS que brinda les coordenades d'un donat arbre, una pantalla LCD per treure resultats i informacions diverses, i una aplicació web feta amb HTML, CSS, PHP i JavaScript, la qual ataca una base de dades allotjada a un servidor extern.

Per tant, SynapTree és una eina molt útil que permet una gestió força concreta, efectiva i intuïtiva que sense cap cost elevat es pot convertir amb la principal eina d'una empresa o òrgan públic que a això es dediqui.

**Abstract (in English, 250 words or less):**

With the project I carry out the design and implementation of an RFID reader able to geoposition, connect to the Internet and manage an external database through a web application that allows to carry out a tree management.

The procedure of tree management, or cataloging, will be done by placing RFID tags on the trees to be cataloged, so that they will be exclusively identified and also positioned with their coordinates. Tree's information will be saved and also the interventions that a technician performs for that tree.

I used a NodeMCU board, which provides an Internet connection to the reader via WiFi network and a 3G/4G node, a GPS module that provides the coordinates of a given tree, an LCD screen to output results and some information, and a web application made with HTML, CSS, PHP, and JavaScript, which queries a database hosted on an external server.

Therefore, SynapTree is a very useful tool that allows a very effective and intuitive tree management at low cost. It can become the main tool of a company or public administration dedicated to this.

## Índex.

0. Índex de taules i il·lustracions .....	Pàg. 6
1. Introducció .....	Pàg. 9
1.1. Explicació i contextualització del projecte .....	Pàg. 9
1.2. Planificació .....	Pàg. 10
1.3. Logotip .....	Pàg. 12
1.4. Viabilitat econòmica .....	Pàg. 12
1.5. Objectius .....	Pàg. 17
2. Arduino .....	Pàg. 18
2.1. Placa NodeMCU .....	Pàg. 18
2.2. Lector RFID RC522 .....	Pàg. 19
2.3. Mòdul GPS NEO-6M .....	Pàg. 21
2.4. Pantalla LCD I2C 16x2 .....	Pàg. 22
2.5. Prototipat definitiu del lector SynapTree .....	Pàg. 23
3. Aplicació web .....	Pàg. 29
3.1. Domini i hosting .....	Pàg. 29
3.2. La navegació .....	Pàg. 29
3.3. Menú horitzontal .....	Pàg. 39
3.4. La base de dades .....	Pàg. 42
3.5. Creació del lloc .....	Pàg. 44
3.6. API de GMaps .....	Pàg. 53
4. Desenvolupament del projecte .....	Pàg. 59
5. Usabilitat .....	Pàg. 62
6. Conclusió.....	Pàg. 66
6.1. Problemes .....	Pàg. 66
6.2. Projectió a futur del projecte .....	Pàg. 67
7. Annexes .....	Pàg. 68
7.0. Índex d'annexes .....	Pàg. 68
7.1. Documents adjunts a la memòria .....	Pàg. 69
7.2. Lògica de SynapTree .....	Pàg. 70
7.3. Llibreries Arduino .....	Pàg. 72
8. Bibliografia .....	Pàg. 74

## 0. Índex de taules i il·lustracions.

Il·lustració 1. Diagrama de Gantt .....	Pàg. 5
Il·lustració 2. Logotip SynapTree .....	Pàg. 6
Taula 1. Components hardware .....	Pàg. 7
Taula 2. Components hardware opcionals .....	Pàg. 7
Taula 3. Software emprat .....	Pàg. 8
Taula 4. Serveis .....	Pàg. 8
Taula 5. RRHH .....	Pàg. 9
Taula 6. Resum de despeses .....	Pàg. 10
Il·lustració 3. Circuit placa NodeMCU v3 .....	Pàg. 12
Il·lustració 4. Circuit lector RFID RC522 .....	Pàg. 14
Il·lustració 5. Circuit Mòdul GPS NEO-6M .....	Pàg. 16
Il·lustració 7. Circuit LCD I2C .....	Pàg. 17
Il·lustració 8. Prova WiFi .....	Pàg. 18
Il·lustració 9. LED on .....	Pàg. 18
Il·lustració 10. LED off .....	Pàg. 18
Il·lustració 11. Serial lectura RFID .....	Pàg. 19
Il·lustració 12. Serial lectura de coordenades .....	Pàg. 20
Il·lustració 13. Serial lectura de coordenades .....	Pàg. 20
Taula 6. Paràmetres de coordenades NMEA .....	Pàg. 20
Il·lustració 14. Lector SynapTree amb Fritzing .....	Pàg. 22
Il·lustració 15. Detall factura domini + hosting .....	Pàg. 23
Il·lustració 16. Navegació per SyanpTree .....	Pàg. 24
Il·lustració 17. Index.php .....	Pàg. 24
Il·lustració 18. Login.php .....	Pàg. 25
Il·lustració 19. Alert d'accés invàlid .....	Pàg. 25
Il·lustració 20. Home.php .....	Pàg. 25
Il·lustració 21. Globus .....	Pàg. 26
Il·lustració 22. Client.php .....	Pàg. 26
Il·lustració 23. Edita_client.php .....	Pàg. 27
Il·lustració 24. Crea_client.php .....	Pàg. 27
Il·lustració 25. Catalogats.php .....	Pàg. 28

Il·lustració 26. Filtres per clients .....	Pàg. 28
Il·lustració 27. Edita_arbre.php .....	Pàg. 29
Il·lustració 28. Intervencio.php .....	Pàg. 29
Il·lustració 29. Crea_arbre.php .....	Pàg. 30
Il·lustració 30. Alert RFID no detectat .....	Pàg. 30
Il·lustració 31. Cerca_intervencio.php .....	Pàg. 31
Il·lustració 32. Filtres per intervencions .....	Pàg. 31
Il·lustració 33. Crea_intervencio.php .....	Pàg. 32
Il·lustració 34. Escaneja.php .....	Pàg. 32
Il·lustració 35. Alert RFID no detectat .....	Pàg. 32
Il·lustració 36. Alert RFID existent .....	Pàg. 33
Il·lustració 37. Alert RFID no existent .....	Pàg. 33
Il·lustració 38. Menú horitzontal .....	Pàg. 33
Il·lustració 39. Menú horitzontal desplegable 1 .....	Pàg. 34
Il·lustració 40. Menú horitzontal desplegable 2 .....	Pàg. 34
Il·lustració 41. Diagrama base de dades SynapTree .....	Pàg. 36
Il·lustració 42. Creació API GMaps 1 .....	Pàg. 48
Il·lustració 43. Creació API GMaps 2 .....	Pàg. 48
Il·lustració 44. Creació API GMaps 3 .....	Pàg. 49
Il·lustració 45. Creació API GMaps 4 .....	Pàg. 49
Il·lustració 46. Creació API GMaps 5 .....	Pàg. 50
Il·lustració 47. Creació API GMaps 6 .....	Pàg. 50
Il·lustració 48. Creació API GMaps 7 .....	Pàg. 50
Il·lustració 49. Creació API GMaps 8 .....	Pàg. 51
Il·lustració 50. Creació API GMaps 9 .....	Pàg. 51
Il·lustració 51. Prova amb GMaps a la web .....	Pàg. 52
Il·lustració 52. Intent de connexió per LCD .....	Pàg. 56
Il·lustració 53. Connexió a WiFi feta .....	Pàg. 56
Il·lustració 54. Benvinguda per LCD .....	Pàg. 57
Il·lustració 55. Cercant GPS .....	Pàg. 57
Il·lustració 56. Lector preparat .....	Pàg. 57
Il·lustració 57. Lectura d'etiqueta .....	Pàg. 58

Il·lustració 58. Disseny responsive del menú ..... Pàg. 58



## 1. Introducció.

### 1.1. Explicació i contextualització del projecte.

SynapTree és un nom que neix de la unió de dues paraules d'origen anglès que crec que descriuen a la perfecció aquest projecte. La primera és “synapse”, sinapsi en català, i la segona és “tree”, arbre. Sinapsi m'ha vingut a la ment perquè crec que la forma en que un xip es comunica amb el lector es per proximitat, igual que la comunicació entre neurones; i arbre, òbviament perquè el projecte s'enfoca cap els arbres i la combinació queda força bé i penso que és molt descriptiva del caràcter que té aquest projecte i de la seva missió.

Amb aquest projecte vull desenvolupar una eina de feina que permeti portar un control de cada arbre sobre la/les plagues que té o ha tingut, els tractaments que ha rebut, les podes, l'edat, el tipus d'arbre que és (palmera, pi, llimoner, etc.), el sexe (si esdevé de rellevància com en el cas de les palmeres) i també un historial detallat de les feines que s'hi ha desenvolupat (intervencions), un historial des del que és podrà accedir a les fitxes de feina que han generat els treballadors. Això implica que l'arbre tindrà una espècie d'historial mèdic i a més, cada arbre estarà identificat dins el sistema per un codi únic proveït per l'etiqueta RFID que passarà a formar part de l'arbre, on cada un en disposarà d'una, i posicionat geogràficament, cosa que permetrà dur una estadística visual de l'estat dels arbres que es tinguin censats, per zones o bé per clients. Això és un mapa que es podrà fer servir per veure l'avanç d'una determinada plaga, com ara la processionària en els pins, el morrut vermell en les palmeres, etc. i també es podrà determinar el nivell de control que es té sobre una plaga i els arbres que no han superat la malaltia, i els que sí. Aquesta distinció es farà amb un mapa integrat en l'espai de feina de cada empresa dins la plataforma web i es tindrà un joc de marcadors en el mapa, distingits per colors, on cada un indicarà l'estat d'un determinat arbre.

El projecte es desenvoluparà amb els següents materials:

- Placa NodeMCU v3 (LoLin).
- Lector RFID a 13,56MHz RFID RC522 i etiquetes RFID a 13,56MHz.
- Pantalla LCD I2C.
- Mòdul GPS NEO-6M
- Bateria, font d'alimentació MB102, protoboard i cables per les connexions.
- Dispositiu mòbil amb connexió 4G.
- Lloc web per administrar la base de dades

A cada arbre es col·locarà un xip RFID a un lloc accessible, de mida reduïda, que servirà d'identificador per l'arbre, ja que cada codi de xip serà únic, com he dit més amunt. Un cop l'arbre disposa de xip, es passarà el lector i automàticament es podrà donar d'alta amb les seves dades a la base de dades. Això es farà a través d'una plataforma web, on cada empresa o “client” de SynapTree, tindrà els seus usuaris i cada usuari o treballador serà l'encarregat de portar al dia l'estat dels arbres que necessitin un tractament o revisió, per així portar al dia la base de dades.

D'aquesta manera, un treballador per fer ús del lector i poder-hi treballar, només necessitarà:

- El lector SynapTree.
- Un dispositiu mòbil 3G/4G que compartirà l'accés a Internet, preferiblement una tauleta, perquè també servirà per accedir a l'aplicació web.

L'objectiu d'això és facilitar als controladors de plagues un historial del que han fet, fer recompte de víctimes, curats, en tractament, amb plaga, pendents de valorar o sans (no han tingut plaga o malaltia) i així veure una evolució de forma gràfica i tan actualitzat com l'empresa desitgi, d'aquesta depèn la taxa d'actualització. Per les empreses SynapTree serà una eina de feina, ja que al revisar o tractar un arbre, les fitxes que tindran penjades a la base de dades, serviran com a fitxes de feina i no caldrà que omplin la fitxa a posteriori i a ma, ho podran fer in situ gràcies a la tauleta que portaran. A més, el fet de poder veure la ubicació de cada arbre en un mapa, pot facilitar molt la tasca de planificar cada intervenció, estalviant molt temps i creant uns itineraris molt més eficients. De cara a administracions públiques, al viure a Menorca, la vegetació és un tema molt interessant, així com el seu manteniment i cura. Aquest projecte també es podria orientar una mica cap als arbres centenaris o d'alt valor, ja que es pot tenir una fitxa de cada un i per tant, un cens molt més acurat i actual.

Segui quina sigui el propòsit del projecte, serà una eina molt gràfica, ja que permetrà mostrar zones afectades per plagues i la seva evolució, o permet delimitar les zones per cada tipus d'arbre, com ara pinars, alzinars, i demés espècies autòctones i controlar que les zones no minvin.

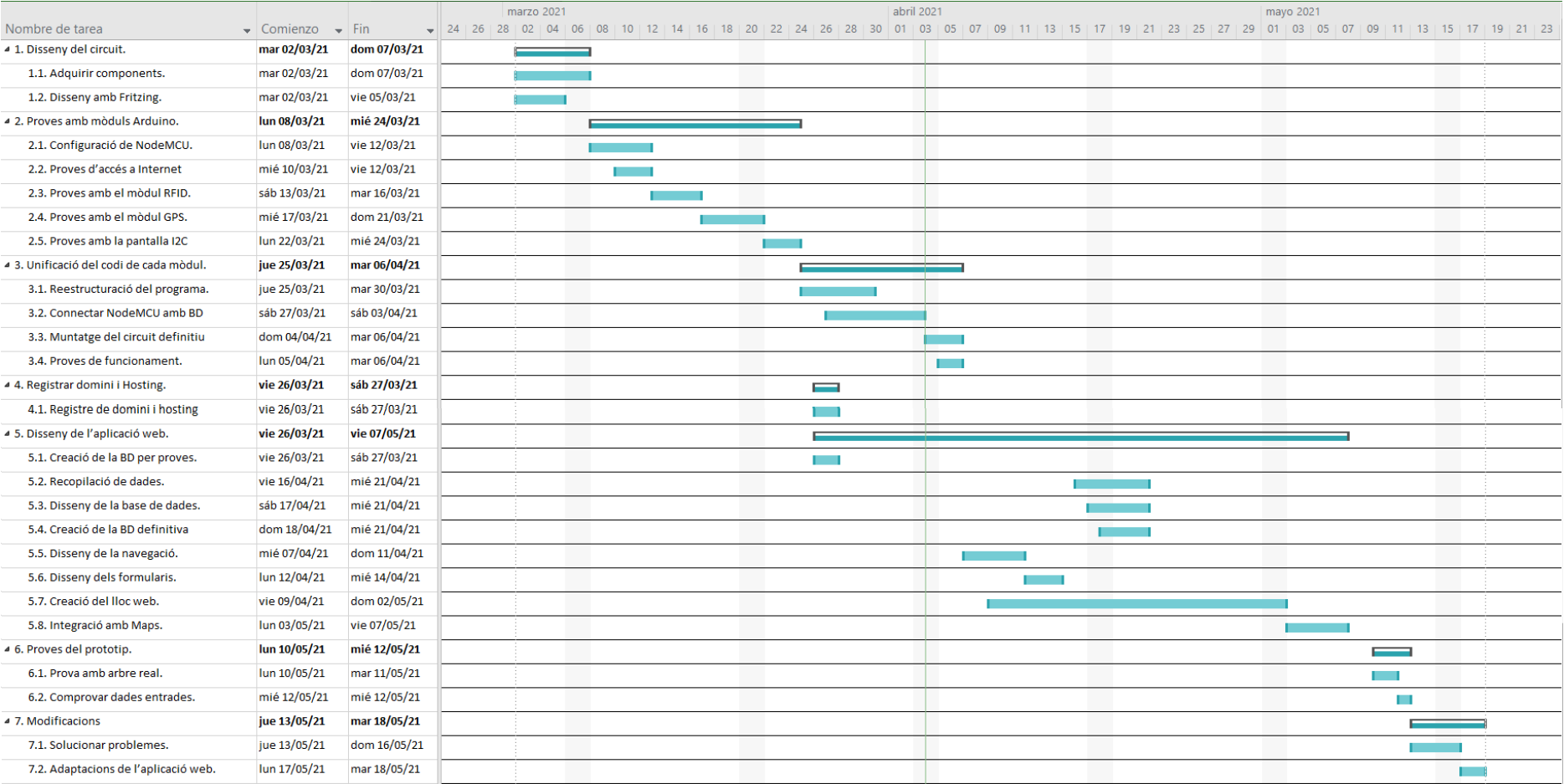
## 1.2. Planificació.

En la planificació del projecte, consten 77 dies des de l'inici del projecte pel primer pas, el 2 de març del 2021, fins la seva posada en producció, el dia 18 de maig del 2021. Al llarg dels dies he anat desenvolupant diferents tasques i des de la planificació inicial passant per totes les tasques i subtasques, he hagut de fer ajustos, perquè el càlcul inicial d'una feina, en realitat m'ha portat més temps del que pensava. També m'ha passat al revés, que algunes tasques m'han sigut més fàcils del que esperava i he trigat menys. Segui com sigui, la planificació inicial era més bé especulativa, i l'he haguda de readaptar i canviar d'ordre o prioritats d'algunes coses, perquè un cop m'hi trobava, necessitava abans desenvolupar alguna tasca per poder prosseguir.

Aquesta situació s'ha donat en la tasca 3.2 de la planificació definitiva, en la que connecto la NodeMCU amb la base de dades, i és que en un principi, aquesta tasca s'havia de fer en el moment de la creació del lloc web i la base de dades, però vaig notar que això s'havia de fer molt abans. També ha passat amb la tasca de la creació del lloc web, en un principi vaig assignar a aquesta tasca un temps de 15 dies, quan realment vaig veure que en requeria uns quants més, i es va passar a 23 dies. És veritat que també he fusionat tasques, perquè era absurd tenir-les per separat, com ara la contractació del domini i el hosting, que en un principi anaven separades i realment formen part de la mateixa.

A més, també he afegit funcionalitats des del plantejament del projecte, com la pantalla LCD i òbviament, comporta la seva feina programar la pantalla, que tot i que no és complicat, pota temps i vol molt assaig i error.

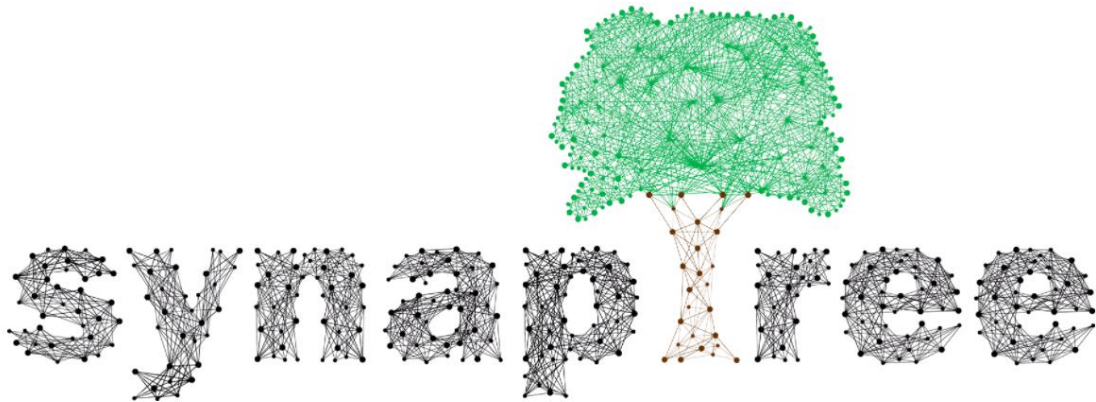
Com s'aprecia amb aquests exemples, la reestructuració del temps s'ha produït en diversos casos i també la reassignació de prioritats d'una tasca, ja que al principi, l'ordre de les feines no estava del tot ben plantejat. Per tant, la planificació de les tasques, la prioritats i el temps requerit en cada una, és el següent:



Il·lustració 1. Diagrama de Gantt

### 1.3. Logotip.

Per el logotip vaig quedar-me amb la primera idea que va venir-me al cap. Es cert que n'he anat tenint més, però la primera m'ha semblat la millor des del principi. És el següent:



Il·lustració 2. Logotip SynapTree

La seva justificació es molt senzilla: comunicació entre nodes i una referència molt clara als arbres. Un node ve a ser cada circumferència que forma part de cada lletra i de l'arbre, i les línies que comuniquen els nodes o circumferències són la metàfora d'una comunicació entre aquests. M'ha semblat molt adient des del començament del projecte perquè igualment que amb el nom i les sinapsis neuronals, cada arbre (node) es capaç de comunicar-se amb el lector (un altre node). Per tant, de nou faig referència a un acte comunicatiu. Quant a l'arbre, per la seva forma, substitueix a la lletra "T" del nom del projecte, i al ser un arbre, crec que hi queda bastant ben cohesionat perquè el projecte es basa al voltant d'aquests.

### 1.4. Viabilitat econòmica.

Per desenvolupar el projecte, he calculat que puc tenir quatre grans grups de despeses en funció de la seva naturalesa:

- **Costos derivats del hardware:** són els costos derivats de la compra de mòduls Arduino, components, cables i bateries i la tauleta que farà de punt d'accés i servirà per consultar l'aplicació web.
- **Costos derivats del software:** són les despeses que provenen del lloguer del domini i el hosting i el software emprat per desenvolupar el projecte sencer.
- **Costos per serveis:** inclou les despeses per quotes fixes relacionades amb serveis externs, com ara el domini.
- **Costos de recursos humans:** ve a ser el cost de les hores de feina que implica desenvolupar el projecte. En funció del perfil requerit per fer cada tasca, el preu per hora variarà. Els perfils possibles serien:
  - **Programador:** desenvolupa tant la part Arduino com la part web del projecte. Tot el que es relaciona amb la programació. En funció del tipus de programació, també variarà el preu d'una hora de feina.

- Cap de projecte: és qui coordina el projecte des de principi a final, establint el temps dedicat a cada feina i analitza la viabilitat i el disseny del projecte en general.
- Dissenyador gràfic: és la persona que elabora el disseny del logotip, la part de maquetació i disseny de la web i també la seva navegació.

#### 1.4.1. Despeses derivades del hardware.

Les despeses derivades del hardware impliquen tots els costos que suposa la compra de materials electrònics que integren el lector. Els costos per lector es desglossen en la següent taula:

Component	Preu
NodeMCU	4,50€
Lector RFID RC522	2,80€
Mòdul GPS NEO-6M	3,95€
LCD I2C	3,50€
Font d'alimentació MB102	2,00€
Etiqueta a 13,56MHZ (unitat)	0,65€
Protoboard, cables i bateria	16,40€
<b>Total</b>	<b>33,80€</b>

Taula 1. Components hardware

S'ha de tenir en compta una despesa opcional per la tauleta perquè el client a vegades també necessitarà comprar la tauleta. És veritat que a aquesta la pot proporcionar SynapTree o bé el propi client. El preu pot variar ja que pot estar subjecte a ofertes del majorista o bé a problemes de disponibilitat. També podria passar que el client desitgi un model diferent o d'una altra marca. Jo m'he decidit pel següent model de tauleta perquè ja compleix els requisits de resolució per suportar bé la web i la seva programació d'estils responsive. Per tant, una tauleta amb les especificacions mínimes valdria això:

Component	Preu
Galaxy Tab S6 (10.4" + 4G)	295,00€
<b>Total</b>	<b>295,00€</b>

Taula 2. Components hardware opcionals

\* Per desenvolupar el projecte no he necessitat cap tauleta, ja que ja disposava d'una, es per això que el preu que agafaré com a referència pel còmput total de despeses, serà excloent la tauleta. A més, aquesta no forma part del projecte i serveix qualsevol amb connexió 4G o 3G.

Per tant, les despeses de hardware per una unitat del lector, suposant que s'inclou la tauleta, són de **328,80€\*** i si només es té en compta el lector, **33,80€**.

#### 1.4.2. Despeses derivades del software.

En aquest apartat tindrè en compte els programes o webs utilitzats al llarg del projecte, tant els emprats per la part de programació, en la part de disseny com en la de planificació i redacció. Faré una discriminació en funció de les

funcions que acabo d'enumerar per així tenir algun criteri de classificació de les despeses de software. Així, les despeses seran:

<b>Programa</b>	<b>Preu</b>
<b>Redacció i planificació</b>	
LibreOffice Writer	0,00€
LucidChart	0,00€*
<b>Disseny</b>	
Verbatelo	0,00€*
Lucid	0,00€*
LibreOffice Draw	0,00€
<b>Programació</b>	
Arduino IDE	0,00€
Fritzing	0,00€
Visual Studio Code	0,00€
FileZilla	0,00€
<b>Total</b>	<b>0,00€</b>

Taula 3. Software emprat

\* S'han emprat les versions bàsiques i gratuïtes de serveis de pagament.

Les despeses derivades del software són completament inexistents, ja que he fet servir programari lliure o les versions gratuïtes d'algunes plataformes web.

Per elaborar la memòria, les entregues i els diagrames de Gantt, he emprat LibreOffice Writer i LucidChart, respectivament. Pel disseny de formularis i de la web he fet ús de la plataforma web Lucid en la seva versió gratuïta; pel diagrama de la base de dades he emprat la versió gratuïta de la plataforma web Verbatelo; i pel disseny de la navegació, he emprat LibreOffice Draw. I per la banda de programació, l'IDE d'Arduino per programar el firmware del lector i els mòduls, carregat en memòria de la NodeMCU; Fritzing pel disseny gràfic del prototip del lector; Visual Studio Code per tota la part de programació web amb HTML, CSS, PHP i JavaScript.

En conclusió, la despesa per software seria de **0,00€**.

#### 1.4.3. Despeses derivades de serveis externs.

Els serveis externs poden incloure serveis derivats de l'ús de la infraestructura necessària per treballar. Per exemple una tarifa de dades (Internet mòbil) per connectar amb la base de dades allotjada al servidor. També hi haurà una quota fixa que serà el lloguer del domini i el hosting de la base de dades i la web. Per tant:

<b>Servei</b>	<b>Preu</b>
Internet mòbil (7GB)	4,00€/mes*
Domini i hosting	3,63€/mes
<b>Total</b>	<b>7,63€</b>

Taula 4. Serveis

\* La quota de dades mòbils és opcional. En el meu cas ha sigut el que he emprat, de manera que així figurarà en el càlcul de despeses.

La quota fixa segura serà la del domini i el hosting que són **3,63€** segurs. Opcionalment hi ha la tarifa de dades mòbils, que pot variar en funció de cada cas degut a les necessitats, però amb una tarifa de 7GB hi ha més dades de les necessàries i suposa una despesa de **4,00€\***.

#### 1.4.4. Despeses derivades dels recursos humans.

En aquest tipus de despeses, hi vindrien les referents al cost del personal. Pel cas del projecte, que ha sigut desenvolupat per una sola persona, depenent de la tasca desenvolupada i el temps dedicat, el preu d'una hora de feina dependrà del perfil requerit per desenvolupar-la. Per tant:

Tasca	Perfil	Hores	Preu/hora	Import
<b>Anàlisi del projecte</b>	Cap de projecte	23	20,00 €	460,00 €
Anàlisi de viabilitat		10	20,00 €	200,00 €
Objectius		5	20,00 €	100,00 €
Anàlisi de costos		8	20,00 €	160,00 €
<b>Firmware del lector</b>	Programador	115	15,00 €	1.725,00 €
NodeMCU		10	15,00 €	150,00 €
RC522		10	15,00 €	150,00 €
GPS NEO-6M		20	15,00 €	300,00 €
LCD I2C		5	15,00 €	75,00 €
Unificar firmware		15	15,00 €	225,00 €
Disseny del circuit		10	15,00 €	150,00 €
Muntatge		10	15,00 €	150,00 €
Connexió a BD		20	15,00 €	300,00 €
Test funcionament		15	15,00 €	225,00 €
<b>Disseny</b>	Dissenyador gràfic	28	10,00 €	280,00 €
Logotip		5	10,00 €	50,00 €
Navegació web		10	10,00 €	100,00 €
Formularis		5	10,00 €	50,00 €
Maquetació web		8	10,00 €	80,00 €
<b>Programació web</b>	Programador web	104	10,00 €	1.040,00 €
Domini i hosting		1	10,00 €	10,00 €
Creació de la BD		5	10,00 €	50,00 €
Creació del lloc		90	10,00 €	900,00 €
Integració GMaps		8	10,00 €	80,00 €
<b>Redacció</b>	Cap de projecte	30	15,00 €	450,00 €
Memòria		20	15,00 €	300,00 €
Presentació		10	15,00 €	150,00 €
<b>Total</b>		<b>600</b>		<b>7.910,00€</b>

Taula 5. RRHH

Com es veu, la llista de despeses és la més important, ja que desenvolupar cada tasca comporta moltes hores i el preu de cada hora és molt elevat. He de dir que els costos són una aproximació i provenen de càlculs a partir d'informació que tinc al meu abast, i d'altra que he cercat per Internet. En total, surt una inversió temporal de **600 hores** repartits entre 77 dies, segons el que s'estipula en la planificació. I un total de despesa econòmica de

**7.910,00€.** Surt aproximadament a unes 8 hores cada dia feiner si faig la divisió entre el total d'hores i els 77 dies.

Com es pot veure en la taula de costos de recursos humans, cada part ha sigut elaborada per un perfil en concret, i en funció d'aquest, el preu d'una hora de feina es diferent. He tingut en compte la complexitat i el nivell d'exigència de cada tasca i pel cas del cap del projecte, el preu de l'hora per les feines d'anàlisi no és el mateix que per les de redacció perquè les primeres són més costoses a nivell de complexitat.

Per tancar aquest apartat de despeses, resumiré en una taula el cost de cada tipus de despesa perquè quedi la informació més concentrada:

Tipus de despesa	Cost fixe	Quota mensual
Hardware	33,80€	-
Software	0,00€	-
Serveis	-	7,63€/mes
Recursos humans	7910,00€	-
<b>Total</b>	<b>7943,80€</b>	<b>7,63€/mes</b>

Taula 6. Resum de despeses

\* Consten totes les despeses que en el meu cas he tingut. No està inclosa la tauleta perquè pel cas del desenvolupament del projecte he emprat un portàtil i no ha implicat una despesa per cap tauleta.

Amb tot això, la despesa total inicial del projecte pujaria a **7943,80€**, més **7,63€** mensuals un cop aquest ja estigui implementat i en producció. La despesa inicial no seria aplicable per un client en cas que estigués interessat amb SynapTree, ja que el desenvolupament del projecte s'hauria d'amortitzar amb el total dels clients que es tinguessin, no amb un. És com una inversió a llarg termini (més d'un any) que es recupera amb el temps. Fins aquí serien les despeses meves com a cap de projecte, on disposaria d'un lector (mòdul Arduino i placa), el firmware desenvolupat i les tasques dels recursos humans també acabades.

El mercat objectiu, o "target", poden ser empreses amb grans propietats de vinyes o terrenys arboris de qualsevol tipus explotats per produir fruits, administracions públiques, agricultors, jardineros i també particulars que vulguin tenir informació sobre els seus arbres del jardí. Amb les despeses per hardware que he esmentat, significa que cada lector SynapTree, a preu de cost surt per 33,80€. Això vol dir que es pot vendre molt bé per uns **69,00€** al client i una quota mensual de **19,90€** per lector, la qual inclouria servei de suport en cas de dubtes de funcionament, problemes tècnics i manteniment de serveis com el domini i el hosting. Amb aquesta quota s'inclouria la configuració inicial de la tauleta, ja sigui aportada pel client o facilitada des de SynapTree, ja que realment només s'ha de configurar com a punt d'accés i en el firmware del lector indicar el nom de la xarxa (SSID) i la seva contrasenya. Cal afegir també la despesa de dades mòbils, que seria de **4,00€** mensuals també.



## 1.5. Objectius.

Els objectius que vull assolir amb el projecte son:

1. Dissenyar i construir un lector que pugui llegir etiquetes RFID a 13,56MHz i comunicar-se amb una base de dades externa i sigui capaç de geoposicionar-se, emprant Arduino
2. Dissenyar i programar una aplicació web que ataqüi una base de dades i en mostri la informació, permeti introduir-ne de nova i modificar la ja existent.
3. Fer que el procés d'ús sigui senzill i amigable per l'usuari.

També hi ha una sèrie de petites metes, punts de control realment, que s'han d'anar aconseguint per assolir el projecte en la seva totalitat:

1. Aconseguir que tots els mòduls funcionin junts.
2. Dissenyar correctament la base de dades.

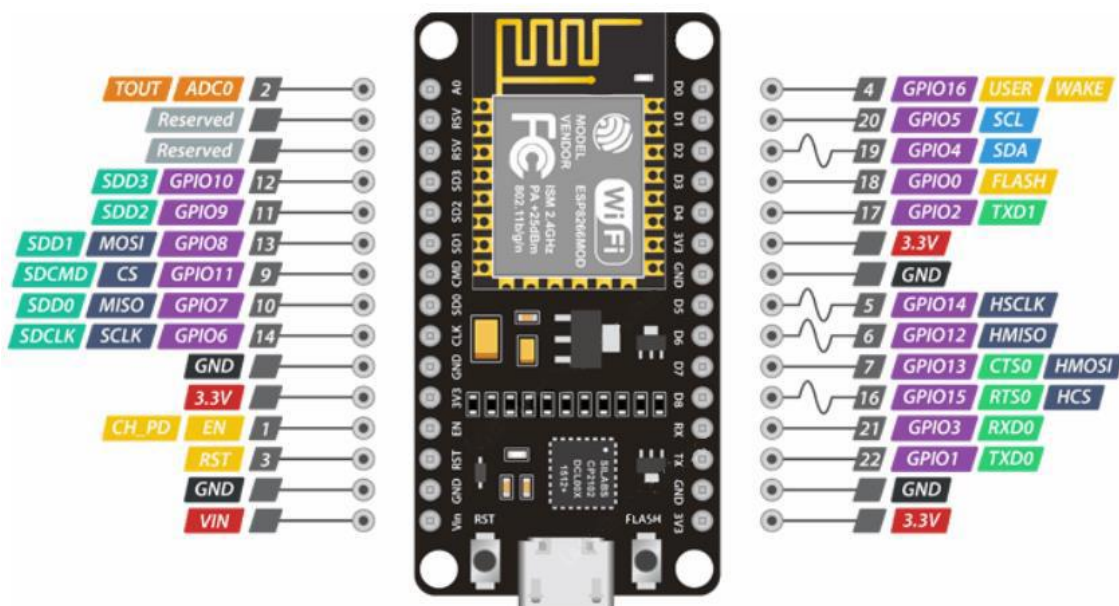
## 2. Arduino.

### 2.1. Placa NodeMCU.

NodeMCU es un kit de desenvolupament per IoT (Internet of Things, o Internet de les coses en anglès) que permet connectar objectes emprant l'estàndard WiFi 802.11 b/g/n a 2.4GHz. El seu firmware està basat en LUA i està especialment desenvolupat pel SoC (System on Chip, o sistema en xip) ESP8266EX, un xip WiFi basat en el model TCP/IP, que a més, té un microcontrolador que també pot ser programat amb l'Arduino IDE, encara que hi haurà diferències en quan als pins de la placa, ja que no es corresponen amb els de les plaques Arduino UNO per exemple, o Arduino MEGA. El disseny del seu hardware, està basat en el mòdul ESP-12, concretament la versió ESP-12E, que es la versió millorada. Les seves especificacions d'interès pel projecte, són:

- Processador de 32 bits Tensilica Xtensa® LX106 RISC a 80 - 160MHz.
- RAM de 128kB
- Memòria interna Flash de 4MB
- Tensió d'alimentació: 4,5V a 10V (pel prototip empraré 5V).
- Tensió en els pins e/s: 3,3V a 600mA.
- Estàndard WiFi 802.11b/g/n a 2.4GHz.
- Suporta encriptats WPA/WPA2.
- Permet comunicacions per interfície Serial, UART o SPI.
- 13 pins digitals (D0 a D12), 1 pin analògic de 10 bits de resolució (A0).
- 3 pins de 3,3V, 2 pin de 5V (VIN i RSV) i 4 pins de terra (GND).
- Convertidor de senyal USB a sèrie (USB to TTL).
- LEDs d'estat.
- Botons de Flash i Reset.
- Dimensions: 57 x 30,4 x 10mm.

Tècnicament, la versió 3, que és la que estic emprant pel projecte, es una versió 2, però feta per LoLin. Per tant, la v3 no és en si un estàndard nou per part de NodeMCU, ja que és v2, la darrera versió oficial.



Il·lustració 3. Circuit placa NodeMCU v3

Amb aquest esquema, la distribució de pins quedaria així, segons la funció de cada tipus de pin:

- Pins 5V i 3V3** En total, segons l'estàndard hi ha 4 pins, un d'entrada de 5V i 3 de sortida a 3,3V pels perifèrics que es desitgin connectar.
- Pins GND** També son 4 i són els pins de terra.
- Pins GPIO** Són 17 i es poden assignar a diverses funcions com I2C, UART, I2S, PWN, IR, LED, etc.
- Pins I2C** Ho són el pin D1 i D2. I2C segueix l'arquitectura Màster – Slave i permet establir una comunicació síncrona entre el Màster i els esclaus.
- Pin Analògic** Ho es A0 i com he dit unes línies més amunt, té una resolució de 10 bits.
- Pins UART** Són els pins D4 (TXD1), D7 (RXD2), D8 (TXD2), D9 (RXD0) i D10 (TXD0). Aquests pins proporcionen una comunicació asíncrona, generen i gestionen interrupcions i envien dades.
- Pins SPI** Són els pins D5, D6, D7, D8, CLK, SD0, CMD i SD1. Segueixen l'arquitectura Màster – Slave on un Màster, pot iniciar una comunicació amb un o varis esclaus y rebre les seves dades, sense que aquests puguin comunicar-se amb la resta d'esclaus. En aquest cas la comunicació es síncrona.
- Pins SDIO** Ho són els CLK, SD0, CMD, SD1, SD2 i SD3 i són pins d'e/s digital segura.
- Pins PWN** Disposa de quatre canals de modulació per amplada de banda que són D2, D5, D6 i D8.
- Pins control** Permeten controlar l'ESP8266 i ho són el RST, que s'empra per restablir ESP8266, EN permet habilitar ESP8266 o deixar-lo funcionant a potència mínima, WAKE que serveix per reactivar ESP8266, i FLASH que permet comunicar la placa amb un PC per Serial i així pujar un firmware.

## 2.2. Lector RFID RC522.

La tecnologia RFID (identificació per radiofreqüència) es fonamenta en dos elements indispensables:

- Un lector, format per una antena que genera un camp electromagnètic, i un mòdul de radiofreqüència;
- Una etiqueta, (o moltes) que conté un xip amb informació numèrica, o alfanumèrica, que va a l'objecte que es vol identificar. Aquesta no necessita cap mena de bateria, ja que és excitada pel camp

electromagnètic que genera el lector a través de l'antena, gràcies a la inducció.

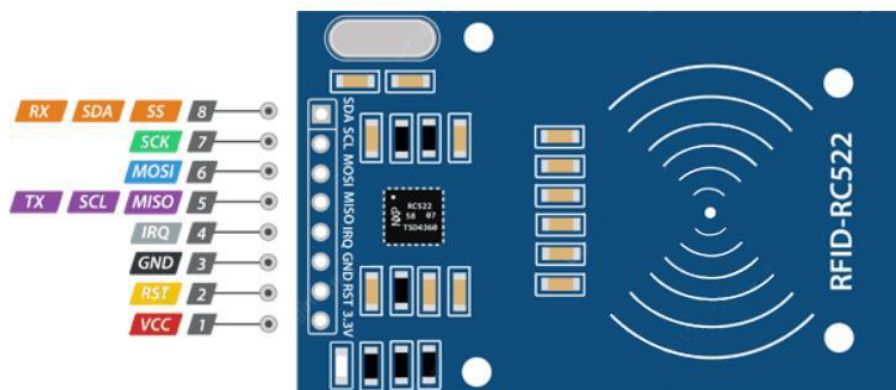
Aquesta tecnologia, treballa amb diverses freqüències:

- **Baixa freqüència o Low Frequency (LF):** treballa amb una freqüència de 125 - 134 KHz.
- **Alta freqüència o High Frequency (HF):** treballa a 13,56 MHz i per tant, la que s'empra en aquest projecte.
- **Freqüència ultra alta o Ultra High Frequency (UHF):** empra les freqüències 433, 860 i 960 MHz

Per fer una lectura, bastarà en apropar els dos elements fins que l'etiqueta quedi dins el camp electromagnètic generat pel lector, de manera que aquest sigui capaç de recórrer la memòria del xip i descodificar la informació que conté, i tot a través d'ones electromagnètiques, procés denominat retrodispersió. Una vegada el senyal de ràdio arriba al lector, aquest la interpreta i envia les dades a un ordinador o a un microcontrolador. Pel cas del lector RC522, aquest genera un camp electromagnètic de 13,56MHz capaç de comunicar etiquetes que compleixen l'estàndard ISO 14443A. A més de llegir-les, també permet escriure la informació que es desitgi dins d'una etiqueta, cosa que queda fora de l'àmbit del projecte per motius pràctics, ja que no és necessari i a més, amb la quantitat de memòria que ofereixen, no podria guardar la informació que vull manejar, i les de més memòria tenen un preu més elevat.

Les especificacions tècniques del lector RFID RC522 són:

- Freqüència acceptada: 13,56MHz.
- Interfícies de comunicació: SPI, UART i I2C.
- Voltatge de funcionament: de 2,5V a 3,3V a 13 – 26mA.
- Rang de lectura: fins a 5cm.



Il·lustració 4. Circuit lector RFID RC522

Com es veu en l'esquema anterior, consta de 8 pins:

#### **Pin 1 - VCC**

S'empra per l'alimentació del mòdul i admet entre 2,5 i 3,3V.

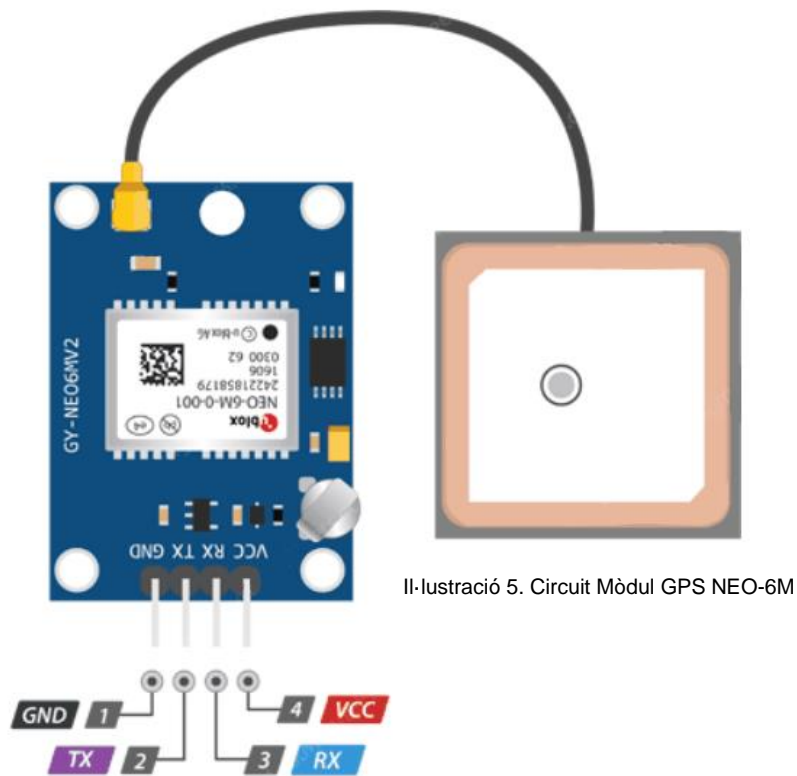
<b>Pin 2 - RST</b>	Té dos estats: HIGH i LOW. HIGH s'empra per mantenir el mòdul en funcionament, mentre que LOW, per apagar-lo. Per tant, la seva feina es encendre i apagar el mòdul.
<b>Pin 3 - GND</b>	És el pin de terra.
<b>Pin 4 - IRQ</b>	Serveix per "fer saber" al lector que hi ha una etiqueta dins el rang de lectura.
<b>Pin 5 - TX/SCL/MISO</b>	Actua com a sortida per Serial si UART esta habilitat (TX); si la interfície I2C està en funcionament, actua com a rellotge (SCL – Serial Clock); o si la interfície SPI esta en funcionament, el pin serveix com a entrada Màster i sortida esclava (MISO – Master In Slave Out).
<b>Pin 6 - MOSI</b>	És l'entrada per la interfície SPI al mòdul, sent una sortida per el Màster i una sortida per l'esclau (MOSI – Master Out Slave In).
<b>Pin 7 - SCK</b>	S'empra per entrar la senyal de rellotge d'arduino a la interfície SPI.
<b>Pin 8 - RX/SDA/SS</b>	Si la interfície UART esta activada, funciona com entrada de dades Serial (RX); si la interfície I2C és activa, funciona com entrada de dades; o si la interfície SPI està habilitada, funciona com una entrada de senyal (SS).

### 2.3. Mòdul GPS NEO-6M.

El mòdul GPS NEO6 és un receptor GPS que es capaç de determinar la seva ubicació arreu del mon, esbrinant quant de lluny es troben diversos satèl·lits GPS. Aquests satèl·lits GPS envien ones de radio amb informació com la seva posició i la hora i la data cap a la terra, permetent que el mòdul GPS sigui capaç de calcular diverses distàncies de diversos satèl·lits i així, determinant de forma ben precisa on es troba el mòdul, procés denominat trilateració. NEO6M v2 es capaç de rastrejar fins a 22 satèl·lits GPS amb una sensibilitat de -161dB i actualitzar la seva ubicació 5 vegades cada segon amb una precisió de 2,5 metres, cosa que pel meu projecte, es una mica massa, però de cara a la feina de l'usuari, té molta utilitat i permet estalviar molt de temps. Resumidament, les seves característiques de funcionament són:

- Tipus de receptor: GPS L1 amb 50 canals.
- Voltatge: 2,7V a 3,6V a 45mA
- Precisió: 2,5m.
- Taxa d'actualització: fins a 5 per segon.
- Sensibilitat: -161dB gràcies a l'antena.

- Velocitat de comunicació per Serial: 4800 – 230400 baud.



Il·lustració 5. Circuit Mòdul GPS NEO-6M

Aquest mòdul GPS, consta de 4 pins:

**Pin 1 - GND** És el pin de terra.

**Pin 2 - TX** És el pin transmissor i s'empra per la comunicació en sèrie.

**Pin 3 - RX** És el pin receptor i també s'empra per la comunicació en sèrie.

**Pin 4 - VCC** s'empra per fer arribar el corrent elèctric al mòdul.

#### 2.4. Pantalla LCD I2C 16x2.

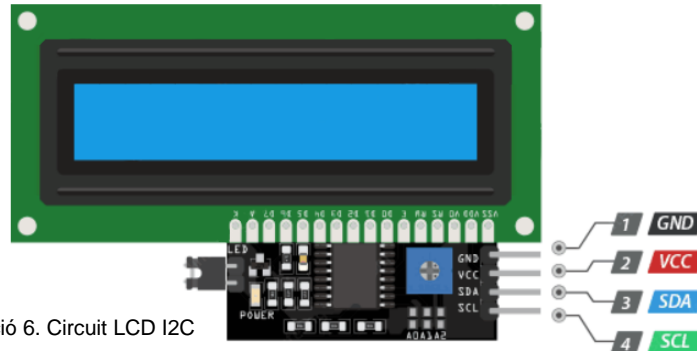
Aquestes pantalles consten de dues parts:

- **Pantalla LCD:** consta d'una matriu de 16x2, això es, dues files que admeten fins a 16 caràcters ASCII cada una, i de fons, disposa d'un LED. Cada espai on surt un caràcter, esta format per una petita matriu de píxels de 5x8.
- **Adaptador I2C:** és una placa que converteix dades I2C en les dades en paral·lel que requereix la pantalla. A més, disposa d'un potenciòmetre per regular la brillantor del LED de fons que il·lumina la pantalla.

Les seves característiques de funcionament són:

- Número de caràcters: 32 (16x2) ASCII.
- Files: 2.

- Columnes: 16.
- Voltatge: 5V.
- Resolució per caràcter: 5x8 píxels.
- Retroil·lumincació: per LED.



Il·lustració 6. Circuit LCD I2C

La pantalla amb el mòdul I2C, consta de 4 pins:

**Pin 1 - GND** És el pin de terra.

**Pin 2 - VCC** Permet l'arribada de corrent elèctric al mòdul I2C. Accepta 5V.

**Pin 3 - SDA** És el pin encarregat d'enviar i rebre dades en sèrie.

**Pin 4 - SCL** Pin pel que es rep la senyal del rellotge de la placa. Es un pin de temporització.

## 2.5. Prototipat definitiu del lector SynapTree.

El muntatge del lector consta de tots aquests mòduls que he explicat fins ara i cada un desenvolupa la seva funció dins el lector SynapTree. Abans d'ajuntar-los i fer que tots funcionin i treballin en equip, ha hagut de fer una sèrie de proves individualment per conèixer-los i saber com funcionen i com puc desenvolupar el firmware de forma òptima. Un cop he tingut aquestes proves individuals i tot funcionant de la forma esperada, ve el moment de la unificació del codi i generar un únic firmware. Unificar el codi, significa que les proves que vaig fer amb cada mòdul Arduino, han de funcionar de forma conjunta i han de tenir una funció mes definida dins el programa general:

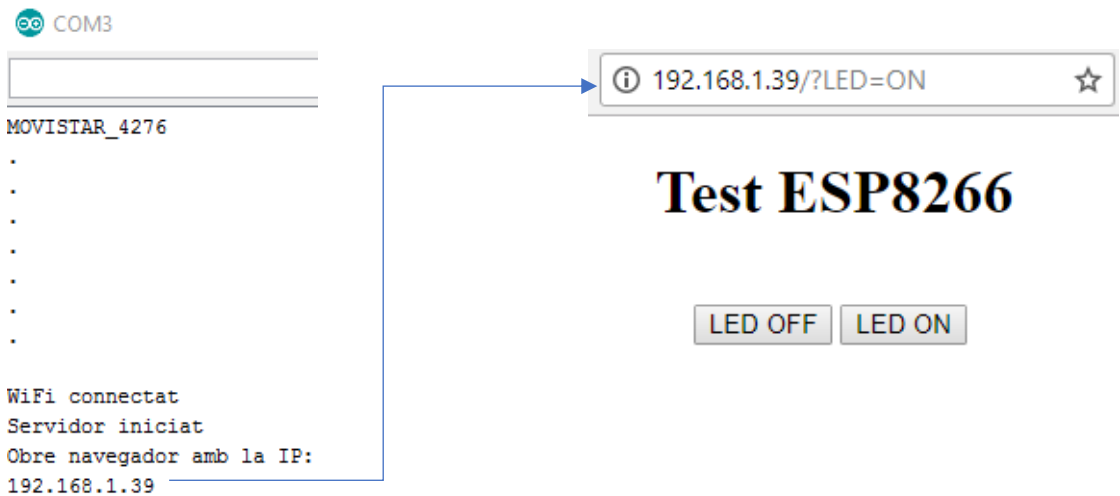
- Placa NodeMCU:** ha de ser el centre del projecte ja que és el nexe d'unió de tots els altres mòduls. Cada mòdul emprarà els pins de la placa i connectarà els mòduls entre si. A més, servirà per fer tenir connexió a Internet i per tant, al domini i a la base de dades. Com a parts importants destacables del codi per la placa, hi ha la que inicia la connexió WiFi a partir de les variables "SSID" i "contrasenya" proporcionades en la part de declaració de variables. Aquesta part està dins la funció `setup()` del codi i realment és una funció, però es força important pel projecte i queda com es veu a continuació:

```

void setup() {
  ...      OMISSIÓ DE PARTS
  WiFi.begin(wifi, contrasenya); //COMENÇA EL PROCÉS DE CONNEXIÓ
  ...      OMISSIÓ DE PARTS
}

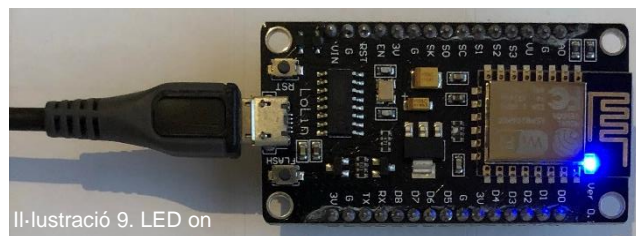
```

Per veure de forma exemplificada com funciona la connexió a WiFi de la NodeMCU, el que he fet ha estat crear un formulari web que permet la comunicació entre aquest formulari i la placa, permetent encendre i apagar un LED amb les opcions que ofereix aquest formulari HTML. El funcionament es pot resumir amb aquestes imatges:

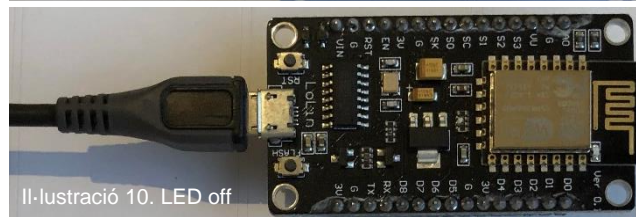


Il·lustració 8. Prova WiFi

En les imatge de dalt, es veu que per Serial es notifica la connexió a la xarxa amb la IP, i després, en al de la dreta, es veu els botons web que permeten interactuar amb el LED de la placa. Per tant, na vegada el codi esta pujat a la placa, automàticament s'estableix connexió amb el router que s'ha configurat, s'assigna una IP a NodeMCU i és accessible a través del navegador. Un cop en el navegador, es veuen dos botons com els de la imatge superior dreta, des d'on es pot controlar el LED integrat, encès per defecte i es veu que està encès en la imatge inferior esquerra i apagat en la inferior dreta:



Il·lustració 9. LED on



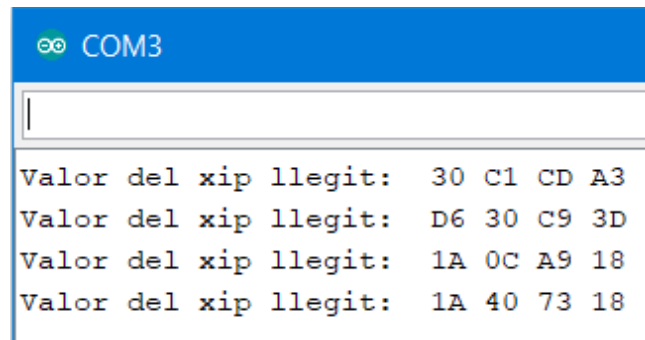
Il·lustració 10. LED off



- b. **Mòdul RFID RC-522:** ha de fer la funció per la que ha estat dissenyat: llegir etiquetes RFID per inducció a 13,56MHz. Aquest bucle el que fa és recórrer la memòria de l'etiqueta que s'excita per proximitat, en blocs. Aquesta part va dins la funció `loop()` i queda així:

```
... OMISSIÓ DE PARTS
if (lecturaRFID.PICC_ReadCardSerial()){ //CONDICIÓ QUE PROCESA LA LECTURA DEL XIP
    for (byte i = 0; i < 4; i++) { //ES VA LLEGINT LA MEMÒRIA DEL XIP,
        valorRFID += lecturaRFID.uid.uidByte[i]; //AFEGEIX A valorRFID LA PART LLEGIDA
    }
}
```

Per aquesta part, ja he necessitat, el lector i les etiquetes RFID, a part, com és natural, la NodeMCU. El funcionament d'aquest lector es força senzill, basta amb atracar l'etiqueta perquè el lector faixi la seva feina i envii el nombre que porta en memòria al port sèrie. Com s'aprecia en la imatge, una vegada el codi ha estat pujat a la placa, si s'aproxima una etiqueta al lector, aquest la llegeix i es mostra per port sèrie.



```
COM3
Valor del xip llegit: 30 C1 CD A3
Valor del xip llegit: D6 30 C9 3D
Valor del xip llegit: 1A 0C A9 18
Valor del xip llegit: 1A 40 73 18
```

Il·lustració 11. Serial lectura RFID

- c. **Mòdul GPS NEO-6M:** ha de permetre obtenir les coordenades del lloc de lectura del xip RFID, de forma aproximada, ja que servirà d'aproximació per carregar un mapa en l'aplicació web. No es necessita precisió mil·limètrica, ni molt menys. Per obtenir una posició s'empra aquets codi que va dins la funció `loop()`:

```
... OMISSIÓ DE PARTS
if (ubicacio.encode(comSerieGPS.read())){ //SI ES REP LA TRAMA NMEA, ES DECODIFIQUEN LES FRASES
    if(ubicacio.location.isValid()){ //I SI LES COORDENADES REBUDES SÓN BONES
        lati = ubicacio.location.lat(); //S'ASSIGNA LA LATITUD A LA VARIABLE lati
        longi = ubicacio.location.lng(); //I TAMBÉ LA LONGITUD A LA VARIABLE longi
    }
}
```

Ben bé tots els receptors de GPS reben la informació en un format especial, aquest format es diu NMEA. Una sentència NMEA (National Marine Electronics Association) està composta per oracions. Per comprovar el funcionament del mòdul, empraré la placa NodeMCU com a conversor USB a TTL, deixant a la vista les sentències NMEA i les seves frases o oracions sense descodificar i interpretables, però amb molta

informació no útil. A continuació, es veu el resultat d'aquest sketch que ve d'exemple amb la llibreria que empra el mòdul.

```

COM4 (Arduino/Genuino Uno)
$GPVTG,,T,,M,0.053,N,0.098,E,A*24
$GPGGA,174819.00,3959.74092,N,00350.00700,E,1,05,2.24,21.8,M,49.1,M,,*67
$GPGSA,A,3,20,23,21,27,08,,,,,,,,,3.19,2.24,2.27*07
$GPGSV,3,1,11,01,,,33,07,10,301,29,08,36,305,45,10,,,26*7A
$GPGSV,3,2,11,11,06,269,35,18,,,29,20,24,047,43,21,22,252,46*43
$GPGSV,3,3,11,23,41,060,37,26,,,29,27,70,343,44*76
$GPGLL,3959.74092,N,00350.00700,E,174819.00,A,A*64
$GPRMC,174820.00,A,3959.74084,N,00350.00701,E,0.077,,010421,,,A*77
$GPVTG,,T,,M,0.077,N,0.142,E,A*24
$GPGGA,174820.00,3959.74084,N,00350.00701,E,1,05,2.24,21.8,M,49.1,M,,*6B
$GPGSA,A,3,20,23,21,27,08,,,,,,,,,3.19,2.24,2.27*07
$GPGSV,3,1,11,01,,,32,07,10,301,29,08,36,305,45,10,,,26*7B
$GPGSV,3,2,11,11,06,269,35,18,,,29,20,24,047,43,21,22,252,46*43
$GPGSV,3,3,11,23,41,060,37,26,,,29,27,70,343,44*76
$GPGLL,3959.74084,N,00350.00701,E,174820.00,A,A*68
$GPRMC,174821.00,A,3959.74096,N,00350.00692,E,0.018,,010421,,,A*77
$GPVTG,,T,,M,0.018,N,0.033,E,A*2A
$GPGGA,174821.00,3959.74096,N,00350.00692,E,1,06,1.12,21.8,M,49.1,M,,*67
Autoscroll Ambos NL & CR 9600 baudio Clear output
  
```

Il·lustració 12. Serial lectura de coordenades

Com es veu, hi ha molta informació dins cada frase NMEA, i la més útil i més emprada és la frase GPRMC (Global Positioning Recommended Minimum Coordinates). Té la següent gramàtica:

Paràmetre	Valor
Format de la trama	\$GPRMC
Hora UTC (hhmmss)	174820.00 (17:48:20)
Estat de la trama	A (vàlida) – 9 (trama de caché) – V (invàlida). En aquest cas és vàlida (A)
Latitud	3959.74084
Hemisferi	N (Nord) – S (Sud). En aquest cas, val N
Longitud	00350.00701
Hemisferi	E (Est) – W (Oest)
Velocitat (nusos)	0.077
Rumb	En aquest cas, no es recullen dades
Data (ddmmaa)	010421 (01-04-2021)
Variació magnètica	En aquest cas, tampoc es recullen dades
Checksum	A*77

Taula 6. Paràmetres de coordenades NMEA

Per obtenir les coordenades a partir de les sentències o frases NMEA contingudes dins les trames NMEA, aquestes han de ser descodificades. Per descodificar les trames, es pot emprar la llibreria TinyGPS++ i la funció `encode()`. Si s'aplica aquesta funció, ja podem seleccionar i guardar dins variables la informació que més interressi, com ara la longitud i la latitud, entre moltes altres (com la velocitat, rumb, data i hora, etc.):

```

COM4 (Arduino/Genuino Uno)
Latitud: 39.995971,Longitud: 39.995971
Latitud: 39.995971,Longitud: 39.995971
Latitud: 39.995971,Longitud: 39.995971
Latitud: 39.995971,Longitud: 39.995971
  
```

Il·lustració 13. Serial lectura de coordenades

- d. **Pantalla LCD I2C:** ha de treure informació per pantalla, com ara l'estat de la connexió WiFi, l'estat del GPS, el valor de lectura RFID, etc. Aquest mòdul si que funciona de forma força senzilla i no es gens complicat fer que un missatge surti per pantalla. De fet, només s'ha d'emprar la funció `pantalla.print()` de la seva llibreria i indicar a quina part de la matriu de la pantalla ha de començar el missatge amb la funció `pantalla.setCursor()`:

```
pantalla.setCursor(0,0); //EN LA PRIMERA LÍNIA S'ESCRIU
pantalla.print("Hola mon"); //EL NOM DE LA XARXA A LA QUE S'ESTÀ CONNECTAT
```

- e. **Connexió amb la base de dades:** aquesta també és una part clau i fonamental del projecte. La comunicació entre el lector i la base de dades que està sobre el servidor. Sense aquesta comunicació el projecte plantejat tal i com ho he fet no tindria cap sentit. És veritat que tot i ser una part relativament petita del codi, no és per això menys important. Les línies que entaulen la comunicació són aquestes:

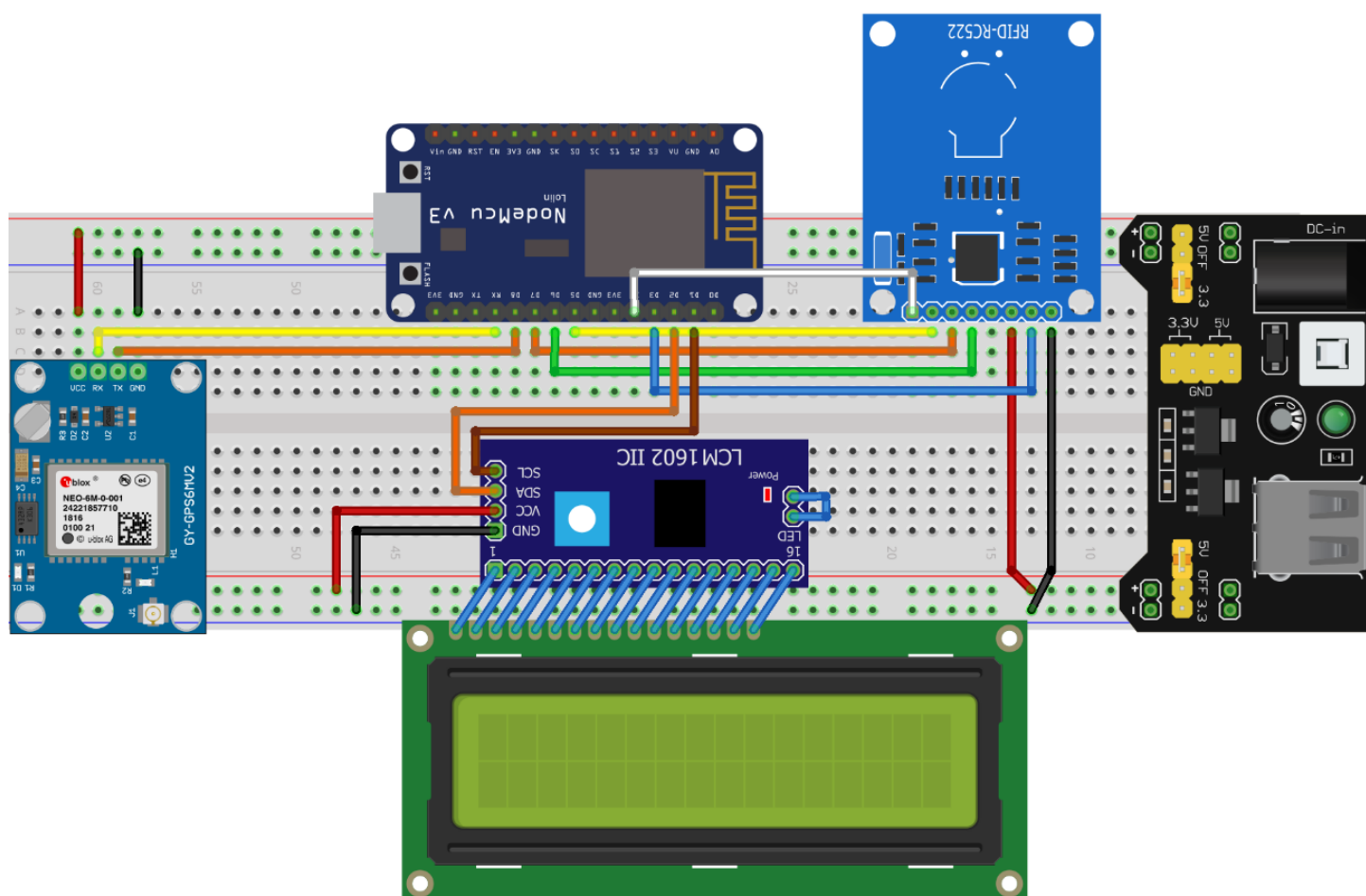
```
//ES CONSTRUEIX LA URL PER GET, S'ENVIA TOT EL QUE CONFORMA LA URL DEL LLOC
client.print(String("GET ") + url + rfid + valorRFID + latitud + latiCadena + longitud + longiCadena + "
HTTP/1.1\r\n" + "Host: " + host + "\r\n" + //S'AFEGEIX EL DOMINI A LA URL
"Connection: close\r\n\r\n"); //DESPRÉS, ES TANCA LA CONNEXIÓ
```

En aquestes dues línies hi ha variables que com és normal, són prèviament declarades o sorgeixen resultat d'alguna funció programada per tal fi. Un cop es té la banda d'arduino solucionada, toca elaborar un document PHP per rebre la informació d'Arduino i inserir la informació a la base de dades. El document queda així:

```
<?php
//ES RECALL EL VALOR DE LA URL ENVIAT DESDE NODEMCU I ES GUARDA DINS UNA VARIABLE
$rfid = $_GET ['rfid'];
//CONNEXIÓ AMB LA BD
$host="db5000302980.hosting-data.io:3306";
$user="dbu160941";
$password="2020.CJm";
$db="dbs295981";
$con = new mysqli($host,$user,$password,$db) or die("Error en la conexión.");
//CONSULTA QUE INSEREIX EL VALOR DEL XIP RFID ESCANEJAT AMB EL LECTOR
$query = "INSERT INTO chip `chip` VALUES ('$rfid')";
$resultado = mysqli_query($con,$query);
?>
```

Fins aquí la part de la lògica més rellevant, ja que el codi sencer ve adjunt a aquesta memòria i està degudament comentat per explicar que fa cada cosa. És el moment de presentar el disseny del prototip de forma visual. Durant el muntatge i la programació, és completament indispensable tenir molta cura de l'ordre dels pins de cada mòdul i sempre s'ha de respectar, sinó el funcionament

presentarà comportaments inesperats. En el següent gràfic, es veu el mòdul compost per tots els mòduls i l'ordre de pins que he seguit:



fritzing

Il·lustració 14. Lector SynapTree amb Fritzing

### 3. Aplicació web.

#### 3.1. Domini i hosting.

El domini que he escollit s'anomena synaptree.es i està comprat i allotjat a 1&1. He decidit emprar aquest proveïdor degut a la configuració dels seus servidors, són molt més flexibles i és que també ja he treballat amb la companyia per diversos motius, i no he tingut queixes. És per això que m'ha paregut una bona idea, ja que tampoc es que tingui molt temps per dedicar a esbrinar com fer algunes coses i perdre temps amb processos que no m'haurien de suposar una inversió de temps més enllà del que tinc programat. Amb 1&1, al comprar un domini, ja ve un pla de hosting, de manera que es factura junt i de forma mensual. L'import que se me facturarà de forma mensual es de 3,63€. Amb aquest pla "Hosting Essential" tinc dret a un lloc web de fins a 50GB de mida o 262.144 arxius que conformin el lloc, 10 usuaris FTP i 5 bases de dades. Suficient pel projecte i per 3,63€ mensuals. Amb la següent factura es veu el que estic explicant:

#### Su factura (Hosting Essential)

Fecha de facturación: 19/05/21

Con.	Servicios facturados	Tarifa	Cantidad	Importe (EUR)	IVA (%)
1	Cuota mensual 19.05.2021-19.06.2021 synaptree.es	3.00 EUR al mes	1 m.	3,00	21,00
Total (base imponible)				3,00 EUR	
+ IVA (21,00%)				0,63 EUR	
<b>Total a pagar</b>				<b>3,63 EUR</b>	

El importe total a pagar será cargado en su cuenta de PayPal durante los próximos días. Gracias.

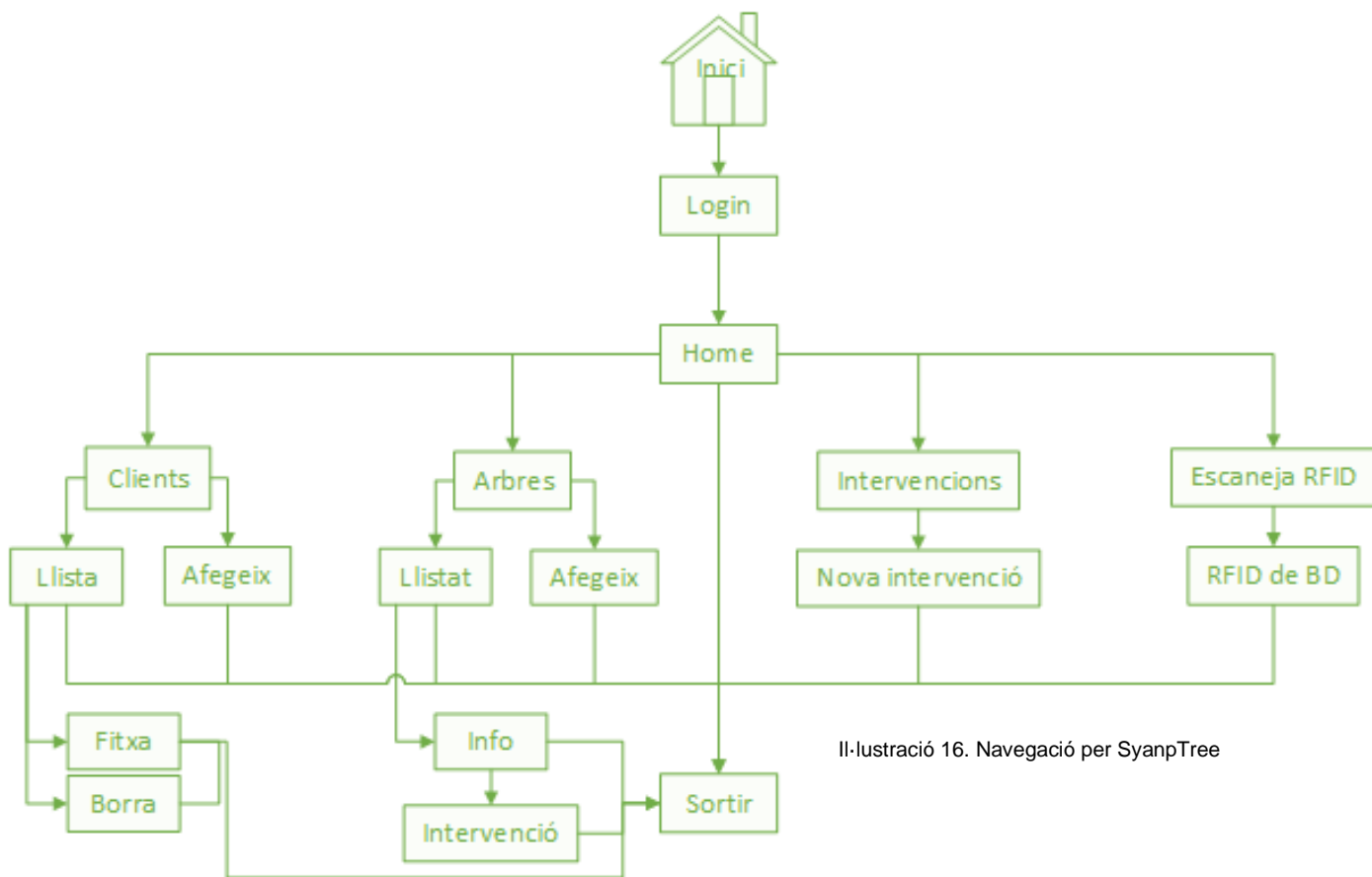
Il·lustració 15. Detall factura domini + hosting

Aquest pla és el més bàsic de tots, ja que els serveis que oferta són ben prou per la infraestructura del projecte. Com es normal, si es mira cap al futur, aquest pla es quedarà obsolet de seguida perquè necessitaré més bases de dades i més espai en els servidors de 1&1, per exemple.

Per pujar els documents que conformen el lloc web als servidors, he emprat FileZilla i a través d'FTP els he anat pujant cada cop que els modificava o actualitzava.

#### 3.2. La navegació.

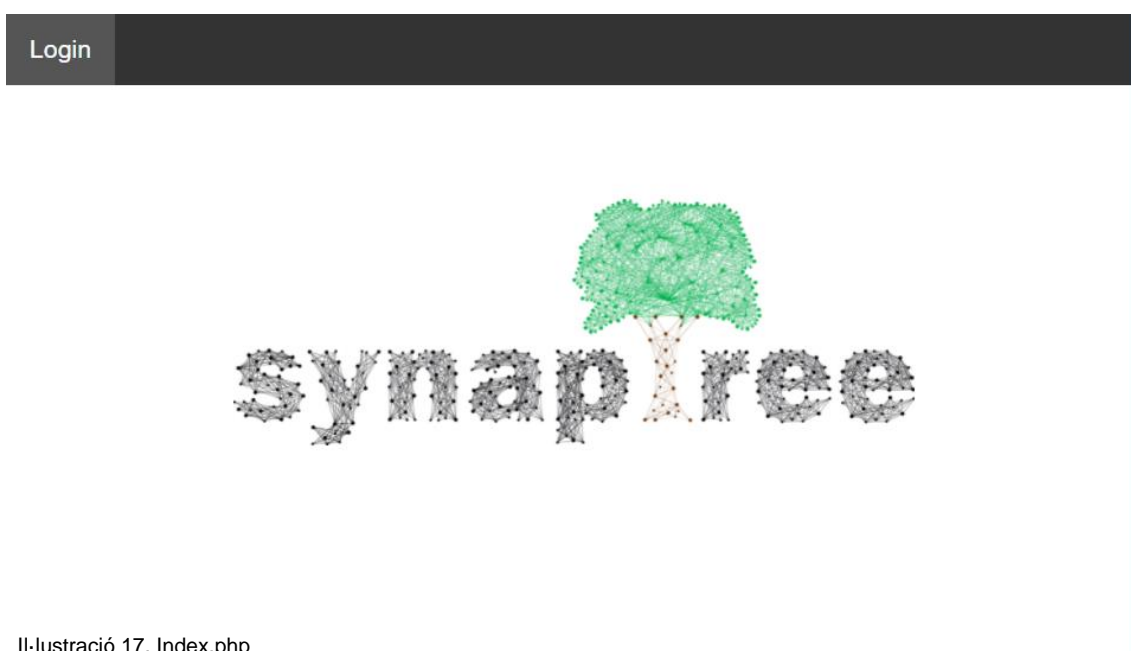
A continuació segueix un diagrama del lloc web, un mapa del lloc. Això és per il·lustrar com és la navegació dins del web i veure la jerarquia de les pàgines que el conformen. Amb la imatge de baix, es pot apreciar de forma esquemàtica la navegació pel lloc:



II-lustració 16. Navegació per SyanpTree

A continuació, explicaré com és cada pàgina:

- a. **La pàgina principal:** en el diagrama l'anomeno "Inici" perquè realment és el que és, tot i que el nom del document és "index.php". Només consta del logotip i d'un menú horitzontal en la part superior que només mostra un enllaç per iniciar sessió:



II-lustració 17. Index.php

- b. Login:** pàgina on l'usuari es trobarà el formulari d'inici de sessió, on haurà d'introduir les seves credencials per poder accedir al contingut del lloc web. Aquesta pàgina de login i el seu formulari d'inici de sessió, quedaran així:



**Bienvenido. Introduce tus datos de acceso.**

Usuario

\*\*\*\*\*

Il·lustració 18. login.php

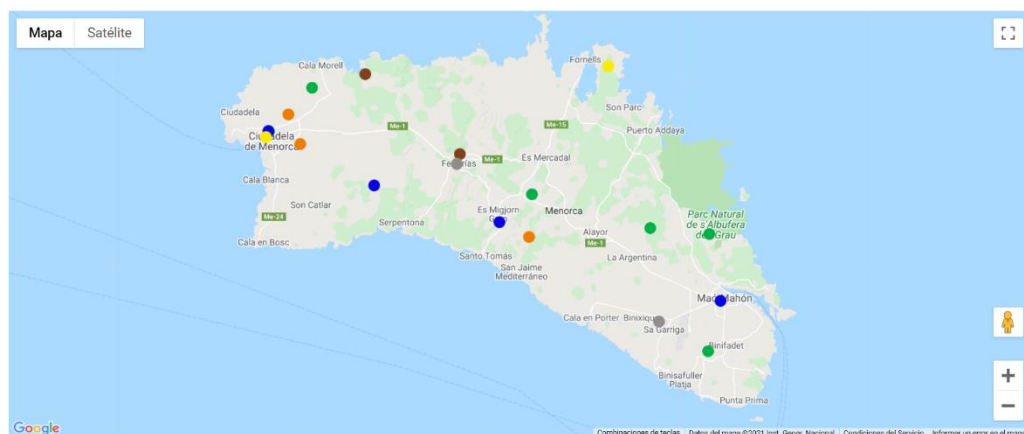
En aquest document, el que es fa és agafar les dades que l'usuari posa com a credencials i les valida sobre la base de dades. En cas que siguin correctes, s'inicia una sessió i es permet entrar a l'aplicació. En cas contrari es notifica a l'usuari que les credencials no són vàlides:

synaptree.es dice

Acceso invalido.

Il·lustració 19. Alert d'accés onvàlid

- c. Home:** en aquesta primera pàgina, bàsicament es veuran totes les opcions del menú i un mapa amb tots els arbres donats d'alta al sistema en forma de punts sobre un mapa. Aquests punts seguiran un codi de color en funció de l'estat de l'arbre i anirà canviant a mesura que canviï l'estat d'aquest. Per tant, aquesta pàgina llueix així:



Il·lustració 20. Home.php

Com es pot apreciar, en el mapa hi ha una sèrie de punts, els quals simbolitzen arbres. Cada arbre que està donat d'alta en el sistema conté una sèrie de dades i si es clica sobre un d'aquests punts, apareix la informació més rellevant sobre aquest:



Il·lustració 21. Globus

d. **Clients:** aquest apartat no es realment una pàgina de l'aplicació web, es més bé un menú desplegable que porta, ara si, a dues pàgines:

- **Llista:** el seu nom real (el del document PHP) és client.php. En aquesta pàgina apareix un llistat de tots els clients dels que l'empresa té un arbre censat. Apareixen les dades d'aquest client i també un formulari de cerca per poder trobar clients més fàcilment. També hi ha un botó que porta a la pàgina d'afegir un client. El seu disseny de client.php és aquest:

Inici Mapa **Clients** ▾ Arbres ▾ Intervencions RFID Sortir

### Llistat de clients

Cerca Afegeix

DNiC/IF	Nom	Adreça	Municipi	País	Telèfon	Mail		
41744330C	Júlia Torres Florit	C/ Mercadal, 39, F12	Ciutadella	Espanya	672265444	juliatorresflorit@hotmail.com		
45223453V	Llorenç Coll de Arredondo	C/ Forners i pastissers, 44, Parc 1, Nave 1	Ciutadella	Espanya	607334486	lore_coll@quimicoll.net		
24567659Y	Toni Moll Moll	C/ Ferreries, 54	Ferreries	Espanya	677877545	tonimoll@hotmail.com		
A07654009	Consell Insular de Menorca	Plaça Biosfera, 1	Maó	Espanya	971150000	consell@insular.org		
45631987Y	Josep Marques Satorc	C/ Canonge Moll, s/n	Ciutadella	Espanya	971381502	josep@gmail.com		
76589000Y	Pedro Florit Altes	C/ Constel·lació, 3	Ciutadella	Espanya	644378600	pedroflorit@gmail.com		
21234551U	Tonia Hernández Serra	C/ Muradefa, 6	Ciutadella	Espanya	607334417	tonia.he.se.13@gmail.com		
5476928T	Tomeu Riera Rüdavets	C/ Nou, 40	Ferreries	Espanya	657977879	tomeu.riera@juaneda.es		
65900127E	Efren Gomila Garriga	Avda. Ramon Llull, 6	Maó	Espanya	971480505	efren.gomila@juaneda.es		

Il·lustració 22. Clnet.php



D'aquesta pàgina, segons es veu al diagrama de navegació, surten les pàgines fitxa i borra:

- **Fitxa:** permet consultar la informació del client, les seves dades, i actualitzar-les si escau, guardant els canvis amb un botó "Modifica". S'hi accedeix des de la icona d'un llapis de la llista de clients i la pàgina realment es diu edita\_client.php i és així:

Inici Mapa **Clients** Arbres Intervencions Sortir RFID

### Actualització del client

<b>Informació personal</b>	<b>Direcció</b>	<b>Contacte</b>	
DNI 41744330C Nom Júlia Llinatges Torres Florit	Direcció C/ Mercadal Nombre 39, F.12 Municipi Ciutadella C. Postal 07760 C. Autònoma Balears País Espanya	Telèfon 672265444 Mail juliatorresflorit@hotmail.co	

[Modifica](#)


Il·lustració 23. Edita\_client.php

Consta de camps de text editables per modificar la informació que sigui necessari i de pas, la treu directament per veure-la. També surt un mapa per localitzar el client.

- **Borra:** realment es un document PHP que elimina el client de la BD. Aquesta acció d'eliminar el client, es du a terme a través de la icona de paperera de la llista de clients.
- **Afegeix:** el nom del PHP per aquesta pàgina és crea\_client.php. Aquesta, permetrà afegir un client nou aportant les seves dades. S'hi pot accedir des del menú client o bé clicant al botó d'afegir de la pàgina llista. El seu disseny es força similar al de fitxa, ja que les informacions que maneja són les mateixes:

Inici Mapa **Clients** Arbres Intervencions RFID Sortir

### Client nou

<b>Informació personal</b>	<b>Direcció</b>	<b>Contacte</b>	<b>Ubicació</b>
DNI Nom Llinatges	Direcció Nombre Municipi C. Postal C. Autònoma País	Telèfon Mail	

[Crea](#)

Latitud  
Longitud

Il·lustració 24. Crea\_client.php

e. **Arbres:** igual que en el cas de clients, aquest apartat també és un menú que porta a dues pàgines. El seu comportament es gairebé idèntic al d'afegir client o llistar-los tots, però extrapolat als arbres i unes funcions una mica més concretes. Les pàgines de les que estic xerrant són:

- **Llistat:** el seu nom pel que fa al document en si, és catalogats.php i el que fa la pàgina és mostrar una llista de tots els arbres censats en la base de dades, amb informació del propietari, el seu estat i el tipus d'arbre. També consta d'un mapa que posiciona tots els arbres, i com en la pàgina home, cada arbre es representa amb un codi de colors que té a veure amb el seu estat. La pàgina queda així:

DNI/CIF	Client	Arbre	Estat	Plaga
41745159K	Llorenç Coll Hernández	Pi mediterrani	Sano	Proceccionària del pi
41745159K	Llorenç Coll Hernández	Pi pinyoner	Curado	Proceccionària del pi
45223453V	Llorenç Coll de Arredondo	Pi pinyoner	Muerto	Proceccionària del pi
A07654009	Consell Insular de Menorca	Palmera datllera	Sano	Morrut vermell
45223453V	Llorenç Coll de Arredondo	Palmera datllera	Sano	Morrut vermell
24567659Y	Toni Moll Moll	Palmera datllera	Curado	Morrut vermell
41744330C	Júlia Torres Florit	Palmera datllera	En tratamiento	Morrut vermell
24567659Y	Toni Moll Moll	Palmera canaria	Curado	Morrut vermell
45223453V	Llorenç Coll de Arredondo	Palmera canaria	Plaga	Morrut vermell
A07654009	Consell Insular de Menorca	Palmera canaria	Pendiente de valorar	Morrut vermell
A07654009	Consell Insular de Menorca	Alzina	Sano	Galeruca peluda
A07654009	Consell Insular de Menorca	Alzina	Sano	Galeruca peluda

II-lustració 25. Catalogats.php

Com es pot apreciar en el disseny, aquesta pàgina disposa d'un boto de filtres que bàsicament és un formulari que permet filtrar els arbres que es llisten. Aquest formulari serà flotant i apareixerà al clicar sobre el botó "filtres" de la pàgina del llistat d'arbres. Permetrà filtrar la llista d'arbres en funció del client, la plaga que l'afecta, l'arbre i el seu estat. D'aquesta manera es podrà fer minvar la llista segons uns criteris i així es facilitarà la cerca d'un arbre en concret, o un conjunt d'ells, per veure l'àrea afectada per una plaga, la seva evolució, etc. Aquest formulari queda així:

Client

Plaga

Arbre

Estat

Filtrar

A07654009 Consell Insu  
II-lustració 26. Filtres per clients

D'aquesta pàgina que llista els arbres de la base de dades, a cada element de la llista hi ha un botó que permet accedir a la fitxa d'informació de l'arbre:

- **Info:** la seva funció és la de mostrar la informació de l'arbre de forma més concisa i el nom del document PHP és edita\_arbre.php. Aquesta informació és la del seu propietari, el seu estat i característiques, la seva posició en un mapa, i a més les intervencions que ha rebut per part dels tècnics. El disseny d'aquesta pàgina serà aquest:

Il·lustració 27. Edita\_arbre.php

Destacar que en l'apartat on es llisten les intervencions rebudes, el que es fa és un llistat amb informació bàsica d'aquestes intervencions. Des de cada fila que representa una intervenció, es pot accedir als detalls d'aquesta amb un enllaç que porta a una pàgina diferent. Igualment es pot afegir una nova intervenció que afecti a l'arbre, especificant els detalls d'aquesta. Aquestes pàgines són:

- **Intervenció:** el seu nom és intervenció.php i s'hi accedeix des de l'enllaç "detalls" de la pàgina. Mostra per pantalla de forma detallada informació d'una intervenció donada i també una pinzellada d'informació sobre el propietari, què sempre és útil. El seu disseny es aquest:

Il·lustració 28. Intervencio.php

He de comentar que en la pàgina d'informació de l'arbre, també hi ha un botó "Nova intervenció" que permet introduir una nova intervenció per

l'arbre en qüestió. Aquesta pàgina serà explicada en l'apartat d'intervencions.

- **Afegeix:** s'anomena crea\_arbre.php i permet afegir un arbre nou aportant les seves dades i amb el previ escaneig d'una etiqueta RFID que es vincularà a l'arbre. Una vegada escanejat, podrà procedir-se amb la introducció de les dades del propietari i les dades de l'arbre, així com la seva ubicació. El disseny d'aquesta pàgina és aquest:

Inici Mapa Clients ▾ **Arbres ▾** Intervencions RFID Sortir

## Arbre nou

**Propietari**  
Selecciona client

**Característiques**  
Identificador 4819320514355333  
Arbre ▾  
Estat ▾  
Plaga ▾

**Ubicació**  
Mapa Satélite  
Google Imágenes ©2021 Condiciones del Servicio Informar un error en el mapa  
Latitud  
Longitud

Crea

Il·lustració 29. Crea\_arbre.php

Bàsicament el que s'ha de fer és emplenar les dades sobre l'arbre i el pacient es selecciona de la llista de clients. En cas que es vulgui accedir a aquesta pàgina sense haver fet prèviament un escaneig d'una etiqueta RFID, s'avisarà:

synaptree.es dice

Cap identificador RFID ha sigut detectat. Escaneja de nou i tanca avís.

Aceptar

Il·lustració 30. Alert RFID no detectat

- f. **Intervencions:** aquesta pàgina llista totes les intervencions realitzades per tots els arbres que es tenen a la base de dades. Com en el cas de la pàgina on es llistaven tots els arbres, hi ha uns filtres que permeten escollir o afinar una cerca per tal de filtrar tota la quantitat d'informació que mostra aquesta pàgina, però de forma inicial, ho mostrarà tot. Dir també que constarà d'un mapa on es mostrarà la posició de tots els arbres i en el cas que hi hagi filtres aplicats, només apareixeran els que calgui. Dir que el nom del PHP és cerca\_intervencio.php i el seu disseny, amb filtres aplicats o no, és així:

Inici Mapa Clients ▾ Arbres ▾ Intervencions RFID Sortir

## Cerca intervencions

DNI/CIF	Nom	Arbre	Estat	Plaga	Data d'intervenció	Tècnic
41745159K	Llorenç Coll Hernández	Pi mediterrani	Sano	Proccionària del pi	2021-02-01	Llorenç
41745159K	Llorenç Coll Hernández	Pi mediterrani	Sano	Proccionària del pi	2021-03-03	Miguel
41745159K	Llorenç Coll Hernández	Pi mediterrani	Sano	Proccionària del pi	2021-03-12	Pedro
41744330C	Julia Torres Florit	Palmera datllera	En tratamiento	Morrut vermell	2021-03-05	Llorenç
76245780P	Lluís Pons Camps	Alzina	En tratamiento	Galeruca peluda	2021-03-15	Miguel
45223453V	Llorenç Coll de Arredondo	Pi pinyoner	Muerto	Proccionària del pi	2021-03-18	Pedro

Il·lustració 31. Cerca\_intervencio.php

Com abans pels arbres, els filtres són molt similars però com és obvi, canvien els criteris:

✕

**Client**

**Tècnic**

**Desde**

**Fins**

**Filtrar**

Il·lustració 32. Filtres per intervencions

En el llistat d'intervencions, a la dreta, apareix una fletxa que permet accedir als detalls de la intervenció, al document que ja he explicat abans, anomenat `intervencio.php`.

- **Nova intervenció:** el documento s'anomena `crea_intervencio.php` i d'ell ja n'he començat a parlar abans. Permet afegir una intervenció per un arbre. S'accedeix a ell des de la fitxa de l'arbre i com es normal, la intervenció s'afegirà a l'arbre del que s'està consultant la seva fitxa. Per fer una intervenció nova per un arbre donat, s'ompliran les dades del tècnic, la plaga, la data i el motiu de la intervenció, i després, només s'indicaran els productes emprats i la concentració d'aquests. El seu disseny és aquest:



Il·lustració 33. Crea\_intervencio.php

- g. Escaneja RFID:** es una pàgina merament informativa que avisa de que s'està al pas previ a inserta un xip RFID a la base de dades. Disposa d'un enllaç que porta a la fitxa de l'arbre que s'identifica en el sistema amb aquell xip llegit si existeix, o bé porta a la pàgina on es donen d'alta els arbres si no existeix. El nom del PHP és escaneja.php i la pàgina és així:

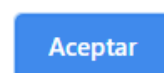


Il·lustració 34. escaneja.php

- **RFID a BD:** es un document PHP de nom lectura.php que fa d'interfície entre l'aplicació web i la base de dades. En ella, s'envia el nombre de xip a la base de dades i s'esbrina si existeix o és nou. Si no s'ha llegit cap etiqueta ho notificarà:

synaptree.es dice

Lectura no detectada. Escaneja de nou i tanca avís.



Il·lustració 35. Alert RFID no detectat

Si s'escaneja una etiqueta que ja existeix, també ho notificarà i portarà a la fitxa de l'arbre a la que correspon:

synaptree.es dice

El xip pertany a un arbre existent, seràs redirigit a la seva fitxa...

Aceptar

Il·lustració 36. Alert RFID existent

I si la lectura és d'una etiqueta que no pertany a cap arbre, també ho notificarà i portarà ala fitxa d'alta:

synaptree.es dice

El xip no està assignat a cap arbre, crearàs una fitxa nova...

Aceptar

Il·lustració 37. Alert RFID no existent

**h. Sortir:** és la pàgina que acaba amb la sessió de l'usuari. Realment és un document PHP que finalitza la sessió i prou, la destrueix i ja no torna a ser valida fins que s'iniciï de nou. El document es diu logout.php.

### 3.3. Menú horitzontal.

Un element important de la navegació, és el menú horitzontal superior, ja que és a partir del que un usuari es mourà dins l'aplicació web. Aquest menú, consta de 7 elements:

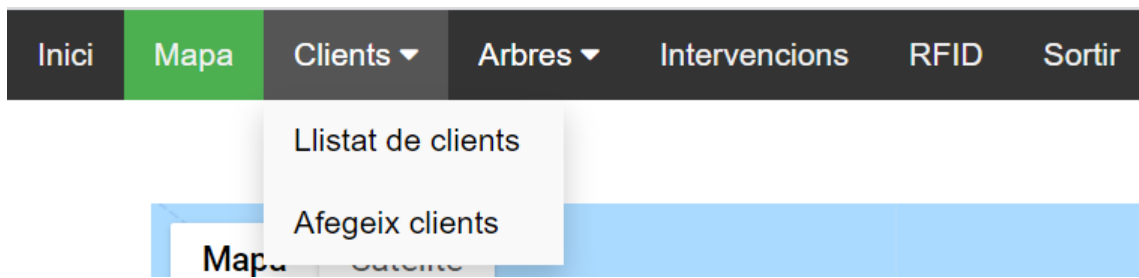
- **Inici:** porta a la pàgina d'inici de l'aplicació web.
- **Mapa:** porta a la pàgina home i és on es pot trobar un mapa gran amb tots els arbres que estan en el sistema. Els detalls es poden trobar unes pàgines més amunt, on dedico unes línies a explicar aquesta pàgina.
- **Clients:** consta de dos submenús que permeten alterar la informació de clients i les seves dades.
- **Arbres:** també consta de dos submenús que permeten el mateix que l'element clients, però de cara a la informació relativa als arbres del sistema.
- **Intervencions:** porta a la pàgina on es llisten les intervencions fetes per l'empresa, amb la possibilitat d'aplicar filtres i afinar uns resultats.
- **RFID:** porta a la pàgina prèvia del document que envia un valor llegit a la base de dades.
- **Sortir:** destrueix la sessió actual de l'usuari.

El disseny d'aquest menú és així:

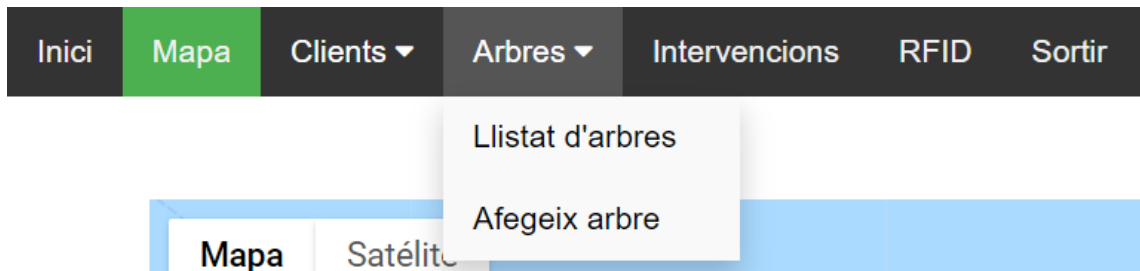


Il·lustració 38. Menú horitzontal

Els apartats “Clients” i “Arbres” són desplegable:



Il·lustració 39. Menú horitzontal desplegable 1



Il·lustració 40. Menú horitzontal desplegable 2

Per acabar, només dir que he fet que aquest menú sigui una mica interactiu mitjançant codi JavaScript i CSS. La opció activa, es la que està de color verd i si amb el mouse es passa sobre una part del menú, aquesta es torna grisa. Això es pot apreciar en les imatges superiors.

Per que l’element actiu sigui de color verd, he emprat el següent codi:

```
<!--MENÚ HORIZONTA-->
<div class="topnav" id="myTopnav">
  <a href="index.php">Inici</a>
  <a class="active" href="home.php">Mapa</a>
```

Com es pot observar, pel cas de la pàgina home (home.php), on surt el mapa amb tots els arbres, l’enllaç que porta a home té associada la classe “active” i això en l’apartat de CSS permet donar-li el color verd:

```
.active {
  background-color: #4CAF50;
  color: white;
}
```

Es veu que la classe “active” posseeix uns estils que fan que la lletra sigui blanca i el fons de l’element sigui verd (#4CAF50).

Pel tema dels submenús desplegable, ho he fet mitjançant CSS. El codi HTML del desplegable es aquest:

```
<div class="dropdown-content">
  <a href="clients.php">Llistat de clients</a>
```



```
<a href="crea_client.php">Afegix clients</a>
</div>
```

Els elements desplegable estan dins un div amb classe "dropdown-content" i mitjançant CSS, faig que es desplegui i obtinguin les característiques de color que s'han vist:

```
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  float: none;
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;
}

.dropdown-content a:hover {
  background-color: #ddd;
  color: black;
}

.dropdown-content {
  display: block;
}
```

En el codi de dalt, faig que al passar sobre un element que pertany a la coasse "dropdown-content" es desplegui un determinat contingut i obtingui uns estils en concret.

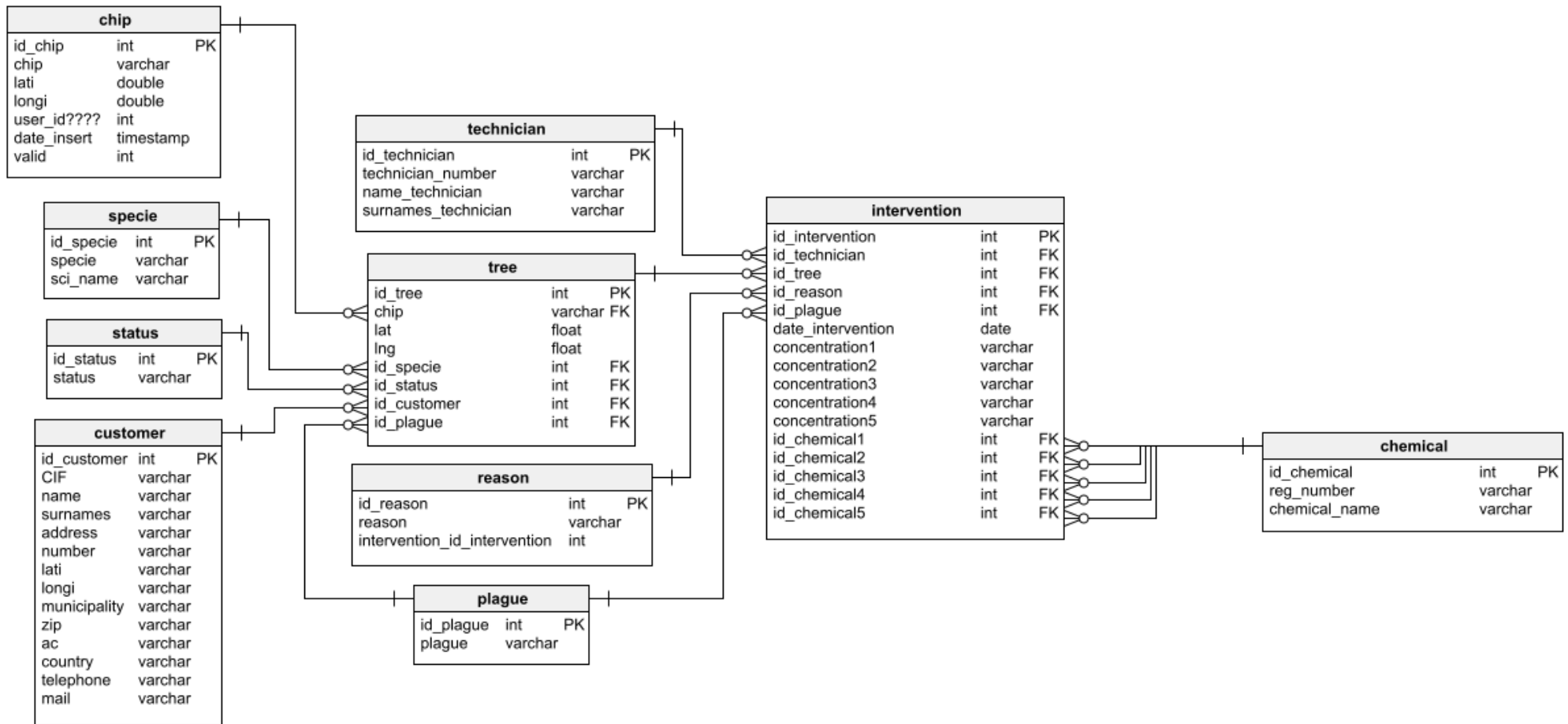
I per fer que quan es passi a sobre del menú, l'element pel que es passa es torni gris, també ho he fet amb CSS:

```
.topnav a:hover {
  background-color: #555;
  color: white;
}
```

Amb això, els elements continguts dins la classe "topnav" si són enllaços (elements <a> d'HTML) es posaran de color gris (#555).

### 3.4. La base de dades.

L'aplicació web permetrà interactuar amb la base de dades des de la mateixa web i també des del lector SynapTree, objecte d'aquest projecte. El disseny de la base de dades, és una mica complex, ja que ha de tenir en compte cada informació que s'ha de guardar, el seu tipus i quines informacions es relacionen amb altres per l'òptim funcionament de la comunicació lector SynapTree, base de dades i aplicació web. El disseny és aquest:



II-lustració 41. Diagrama base de dades SynapTree

Com es pot veure en el diagrama anterior, la base de dades consta de 10 taules:

- **customer (client):** és la taula que guardarà les dades dels clients, els propietaris dels arbres que es censaran i dels que es portarà control. Per això, la informació que guarda es relativa al client exclusivament: el nom, llinatges, DNI o CIF, adreça, nombre, la longitud i la latitud, el municipi, la comunitat autònoma (ac), el país, i les formes de contacte com ara el telèfon i l'adreça electrònica. Bàsicament aquesta taula s'empra per guardar aquesta informació serveix per determinar qui posseeix un determinat arbre, ja que en la taula arbre (tree) es farà referència a aquesta taula amb una clau forana.
- **tree (arbre):** aquesta taula guardarà la informació d'un arbre censat. S'identificarà un arbre pel seu ID, encara que també es podria identificar pel xip que es col·locarà a l'arbre per poder identificar-lo amb el lector, però no es així perquè el xip ja serà un identificador únic de la taula xip. També es guarda la longitud i la latitud de l'arbre, la seva posició en el globus terrestre. Igualment hi haurà informació relativa a la seva espècie, l'estat, el propietari i la plaga que l'afecte (si en té cap) en forma de claus foranes.
- **specie (espècie):** guardarà les espècies d'arbres disponibles a tractar o censar, per part d'una empresa o entitat.
- **stauts (estat):** guardarà els possibles estats d'un arbre.
- **plague (plaga):** guardarà les plagues que puguin afectar a un arbre.
- **chip (xip):** aquesta taula conté el valor dels xips emprats pel cens d'arbres. Realment, ve a ser la informació que es genera amb una lectura del lector SynapTree: el xip escanejat, la longitud i la latitud (posició) de la lectura, el temps i la hora d'aquesta i un valor que indica si la lectura es vàlida o no, en funció de si ja s'ha processat la informació d'aquesta o encara no. Quan es fa una lectura, aquesta pot ser d'un arbre que ja té xip (el xip ja va associat a un arbre) o bé un xip nou que identificarà un arbre en un futur pròxim, i aquesta comprovació es fa a després de la lectura, per això pot estar pendent de processar, on vàlid serà 1, o ja processada, on vàlid valdrà 0. Una peculiaritat d'aquesta taula, és que el seu camp "chip" té una restricció d'únic, ja que potser el lector realitzi diverses lectures per segon, de manera que no m'interessa tenir diverses lectures marcades com vàlides (pendents de processar) dins la taula, ja que es quedarien perpètuament en estat vàlid, degut a que el sistema agafarà sempre la darrera que s'ha fet, deixant les anteriors com a insercions errònies i sempre vàlides.
- **interention (intervenció):** una intervenció es portarà a terme sempre que un operari o tècnic valori un arbre o senzillament el doni d'alta al sistema, encara que estigui sa. És per això que aquesta taula guardarà informació relativa al que s'ha fet amb l'arbre en un moment donat. Una intervenció sempre estarà feta per un tècnic o operari i aquesta sempre quedarà vinculada a aquest tècnic, òbviament també es vincularà a un arbre, a un motiu i a una plaga (ja que potser una intervenció pugui tenir com a motiu una valoració prèvia d'una plaga per conèixer-ne el seu estat abans de

tractar-lo). També es guardarà la data d'aquesta intervenció i els químics emprats i la seva concentració, si es que s'han emprat químics.

- **technician (tècnic):** serà qui actua en una intervenció. Es guardarà el seu nombre de tècnic i el seu nom i llinatges.
- **reason (raó):** com he dit, guardarà el motiu d'una intervenció i podrà ser una raó molt diversa: tractament, valoració, control, etc.
- **chemical (químic):** és el producte emprat en una intervenció que calgui de productes per tractar una plaga. Aquests productes poden ser de qualsevol tipus i sempre hauran d'anar acompanyats pel seu nombre de registre i el seu nom, per seguretat.

Realment, hi ha una taula més en la base de dades, però només s'empra per fer el login en l'aplicació. No l'he posada en el diagrama perquè no es vincula amb cap altra taula, ja que la seva funció es proporcionar un ID de sessió, donar seguretat. Aquesta taula guarda la ID d'usuari, el seu nom, el correu electrònic, el nom d'usuari, la contrasenya i la data de creació.

### 3.5. Creació del lloc.

Pel disseny i la maquetació web, he emprat HTML i CSS. HTML (HyperText Markup Language) és un llenguatge fonamental per la programació web ja que és amb aquest que es defineix l'estructura d'una pàgina, i per tant d'un lloc sencer. El que fa HTML es dividir en seccions una pàgina, per donar a aquesta divisió un lloc dins la pàgina, i per fer-ho, empra les anomenades etiquetes. Aquestes etiquetes consten d'una paraula clau dintre de "<" i ">", per exemple <div>. Hi ha una gran quantitat d'etiquetes i totes elles serveixen per una cosa en concret, l'exemple de <div> permet crear un contenidor que contindrà més etiquetes en el seu interior i informació que serà impresa per pantalla quan es visiti la pàgina. De fet, qualsevol pàgina web, comença amb etiquetes HTML següents:

```
<html>
  <head>
  </head>

  <body>
  </body>
</html>
```

Per tant, aquest llenguatge permet dividir una pàgina en seccions d'informació, i donar una mica d'estil a la pàgina, que sempre serà millor complementar amb el llenguatge CSS (Cascading Style Sheets). El CSS el que permet es donar un aspecte visual molt elaborat, oferint una presentació elegant i al gust del dissenyador del web. Actualment està en la versió 3 (CSS3) i com deia, serà l'encarregat de descriure com es mostra per pantalla l'HTML ja que des de CSS es fa referència a un element HTML i s'estableix com ha de ser presentat:

```

.logo{
  max-width: 100%;
  height: auto;
  margin-top: 5%;
}

.titol{
  color: #457346;
  text-align: left;
  font-size: 37;
  letter-spacing: 0.2em;
  margin-left: 1.5%;
}

```

La referència a HTML es pot fer des de la pròpia etiqueta HTML, i per tant afectarà a totes, per la seva ID o per la seva classe (entre altres més complexes).

Per la comunicació entre l'aplicació i el servidor, he emprat PHP (Hypertext Processor), un llenguatge ideal per incrustar dins l'HTML, ja que està pensat pel desenvolupament web. Amb aquest llenguatge he fet possible coses com la connexió amb la base de dades:

```

<?php
  $host="nom_servidor";
  $user="usuari";
  $password="contrasenya";
  $db="nombd";
  $con = new mysqli($host,$user,$password,$db) or die("Error");
?>

```

També m'ha permès emprar sessions d'usuari:

```

<?php
  session_start();
  if(!isset($_SESSION["user_id"]) || $_SESSION["user_id"]==null){
  }
?>

```

I el més rellevant de tot, comunicar-me amb la base de dades i obtenir-ne dades o introduir-les:

```

<?php
  $query = "select * from plague";
  $resultado = mysqli_query($con,$query);
?>

```

També he emprat JavaScript pel tema dels mapes. JavaScript és un llenguatge de programació orientat a objectes, com PHP, i s'empra per definir

comportaments en un lloc web, per scripting web. El codi JavaScript que he emprat, no es que sigui molt abundant ni tampoc molt canviant al llarg del projecte, ja que més o menys es sempre el mateix, canviant petits matisos en funció del que vull obtenir del mapa o com vull aquest mapa. Un exemple de codi seria aquest:

```
var map;
var marker;
var myLatLng = new google.maps.LatLng(<?php echo $lati;?>,<?php echo $lon
gi;?>);
var geocoder = new google.maps.Geocoder();
    function initialize() {
        var mapOptions = {
            zoom: 20,
            center: myLatLng,
            mapTypeId: google.maps.MapTypeId.SATELLITE
        };
```

Finalment, pel llenguatge de consulta a la base de dades, he emprat l'SQL (Structured Query Language). És un llenguatge que permet administrar una base de dades y recuperar informació de sistemes de gestió de bases de dades relacionals. Com comentava unes línies amunt, l'he emprat juntament amb PHP perquè així ho permet PHP, i un exemple d'ús d'aquest llenguatge de consulta seria aquest:

```
delete from customer WHERE id_customer = '$id_customer'
```

En aquest cas el que es fa és eliminar un registre sempre que es compleixi la condició de l'apartat "WHERE".

En resum, he emprat 5 llenguatges:

- HTML
- CSS
- PHP
- SQL
- JavaScript

Cada un d'ells té la seva funció ben definida dins l projecte:

- L'HTML ha sigut emprat per la part de disseny web. Permet crear tots els elements presentats en el lloc web i que per tant el formaran.
- El CSS m'ha permès donar un toc més artístic als elements HTML i ha també ha fet possible seguir un estil comú entre els diferents documents que conformen l'aplicació, donant lloc a un disseny unificat al llarg de tota l'aplicació.
- PHP m'ha servit per presentar la informació que s'obté de la base de dades i presentar-la en elements HTML. També ha servit per comprovar

- certs valors provinents de formularis o de la pròpia URL, verificant si estaven buits o valien 0, entre altres operacions lògiques.
- El llenguatge SQL m'ha permès atacar la base de dades per recuperar-ne la informació que aquesta conté, i també, gràcies a PHP i HTML, per l'enviament de dades cap a la URL o bé per mètodes com POST o GET i la possibilitat de crear formularis web, respectivament, introduir-ne de nova o actualitzar la ja existent.
  - I JavaScript, que m'ha servit per pintar el mapa i els marcadors que aquest pugui contenir, o obtenir els valors de la longitud i la latitud d'un punt en concret del mapa.

Com es pot comprovar, tots els llenguatges que he emprat, fan feina junts i han permès fer tot el que estic explicant en el projecte. A continuació, analitzaré un document PHP que conté HTML, CSS (una referència al document que conté el codi CSS), PHP, SQL i JavaScript, part per part. També n'explicaré un altre que només conté lògica (PHP i SQL) per veure com funciona i quina funció tenen els documents que només serveixen per interactuar amb la base de dades, ja que no presenten informació. És important dir que ambdues castes de documents són vitals pel funcionament del projecte:

- Per una banda, els que presenten informació, són els que ja he explicat amb anterioritat i són els que apareixen en el diagrama de navegació que hi ha més amunt en aquest document.
- Per altra, tenim els documents que no en presenten, la tasca dels quals es extreu o introduir informació en la base de dades, fent unes comprovacions prèvies de la informació (variables) abans d'introduir-la, per així no deixar camps amb valors invàlids o que no compleixin els requisits demanats.

Per començar, explicaré el funcionament d'un document PHP que forma part de l'aplicació i que alhora que presenta informació, també conté una part de lògica. Aquets document serà el que llista els arbres en el mapa i els mostra com a marcadors en forma de punt damunt aquest. S'anomena "home.php":

- Per començar el document, es pot observar que el primer que es fa, es comprovar si la sessió d'usuari es bona, en cas que ho sigui, segueix executant la resta del document, i en cas contrari, amolla una alerta de sessió invàlida:

```
<?php
//SESSIÓ D'USUARI
session_start();
if(!isset($_SESSION["user_id"]) || $_SESSION["user_id"]==null){
    print "<script>alert(\"Accés invàlid!\");window.location='login.php';</script>";
}
?>
```

- Com que la sessió és bona, a partir d'ara començarà la part HTML, concretament la part <head> que és on s'estableix el títol de la pàgina, es vinculen els documents d'estils CSS i alguns scripts, com és el cas de l'API de Google i altres tipus d'scripts externs, si escau. A continuació, es veu com primerament s'estableix un títol, després s'inclou l'API de Google Maps i després es vincula el document d'estils "estils.css" i un full d'estils extern que conté un tipus de font concreta:

```
<html>
  <head>
    <title>Vista general</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--INCLOEIXO LA API DE GOOGLE MAPS AMB EL MEU PROPI NÚMERO DE CLAU PER L'API-->
    <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAYShtVEUqohEtfUuu_279
9TpmE_xrviqU"></script>
    <!--ESTILS-->
    <link rel="stylesheet" type="text/css" href="estils.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  </head>
```

- Just després del <head> comença la part <body> i és el cos de la pàgina, la part del document que conté la informació que serà presentada. En el cas d'aquest document, comença el <body> establint la connexió amb la base de dades, sempre dins una etiqueta PHP, ja que això permet incrustar codi PHP dins HTML:

```
<body>
  <!--CONNEXIÓ AMB LA BD-->
  <?php
    $host="servidor de BD";
    $user="usuari";
    $password="mot de pas";
    $db="nom BD";
    $con = new mysqli($host,$user,$password,$db) or die;
  ?>
```

- Segueixo amb el menú horitzontal. Aquest menú horitzontal es troba dins un element <div> i està compost per enllaços (elements <a>) i altres elements <div> i elements <button> pel cas dels submenús desplegable. Notar que alguns elements <div>, <a> o <button> consten d'un atribut "class", el qual serveix per aplicar uns estils a tota aquesta classe. Aquest és el codi HTML del menú horitzontal:

```
<!--MENÚ HORITZONTAL-->
<div class="topnav" id="myTopnav">
  <a href="index.php">Inici</a>
  <a class="active" href="home.php">Mapa</a>
  <div class="dropdown">
    <button class="dropbtn">Clients
      <i class="fa fa-caret-down"></i>
```



```

        </button>
        <div class="dropdown-content">
            <a href="clients.php">Llistat de clients</a>
            <a href="crea_client.php">Afegeix clients</a>
        </div>
    </div>
    <div class="dropdown">
        <button class="dropbtn">Arbres
            <i class="fa fa-caret-down"></i>
        </button>
        <div class="dropdown-content">
            <a href="catalogats.php">Llistat d'arbres</a>
            <a href="crea_arbre.php">Afegeix arbre</a>
        </div>
    </div>
    <a href="cerca_intervencio.php">Intervencions</a>
    <a href="escaneja.php">RFID</a>
    <a href="php/logout.php">Sortir</a>
    <a href="javascript:void(0);" style="font-size:15px;" class="icon" onclick="myFunction()">&#9776;</a>
</div>
<!-- JAVASCRIPT DE COMPORTAMENT DEL MENÚ HORIZONTAL -->
<script>
    function myFunction() {
        var x = document.getElementById("myTopnav");
        if (x.className === "topnav") {
            x.className += " responsive";
        }
        else {
            x.className = "topnav";
        }
    }
</script>

```

Al final de la part del menú, es veu com hi ha un element `<script>`. Aquest conté codi JavaScript i les etiquetes `<script>` permeten que JavaScript es pugui incorporar directament entre codi HTML. En aquest cas el que fa JavaScript és canviar el nom de la classe de l'element "myTopnav" per facilitar que aquest menú s'adapti a diferents resolucions de pantalla.

- Després del menú es procedeix a construir el `<div>` que contindrà el mapa amb tots els arbres senyalats:
  - o Primer s'obtenen tots els arbres del sistema amb una consulta SQL i es guarden dins la variable PHP `$resultado`:

```

<!-- DIV QUE CONTÉ EL MAPA -->
<div id="mapa" class="mapa"></div><br>
<?php

```

```

//CONSULTA QUE OBTÉ ELS ARBRES, LES UBICACIONS I LA SEVA INFORMACIÓ PEL GLOB
US
$query = "select * from tree, customer, status, plague WHERE status.id_statu
s=tree.id_status and plague.id_p
lague=tree.id_plague and customer.id_customer=tree.id_custome
r";
$resultat = mysqli_query($con,$query);
?>

```

- A continuació, s'invoca el codi JavaScript que construeix el mapa dins el <div>. S'indiquen el centre del mapa, el zoom que s'aplicarà per defecte, el tipus de mapa (satèl·lit o de carreteres) i també s'indica l'element que el contindrà:

```

<!--SCRIPT QUE ESTABLEIX EL CENTRE DEL MAPA, EL ZOOM I EL TIPUS DE MAPA-->
<script type="text/javascript">
var menorca = new google.maps.LatLng(39.940000,4.0689944);
var mapOptions = {
center: menorca,
zoom: 11,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
map = new google.maps.Map(document.getElementById('mapa'), mapOptions);
</script>

```

- Després, es construeixen els marcadors per cada arbre, recorrent la variable PHP \$resultat, la qual conté totes les posicions on hi ha d'haver un arbre posicionat

```

<?php
$i=1;
//ES COLOQUEN EN EL MAPA LES PUNTS QUE MARQUEN LA UBICACIÓ DE CADA ARBRE
while ($data = mysqli_fetch_assoc($resultat)) { ?>

```

Comença la part de l'script que crearà els marcadors en el mapa:

```

<script type="text/javascript">
var marker<?php echo $i;?> = new google.maps.Marker({
position: new google.maps.LatLng(<?php echo $data['lat']; ?>, <?php echo $data['lng']; ?>),
map: map,

```

Segons l'estat de l'arbre, poso el marcador d'un o altra color:

```

icon: 'imatges/<?php echo $data['id_status']; ?>.png',
//POSA UN MARCADOR O ALTRE SEGONS L'ESTAT
});

```

Estableixo la informació que contindrà el globus al clicar-hi a:

```

//INFORMACIÓ QUE GUARDA EL GLOBUS DEL MARCADOR AL CLICAR-HI
var contentString = "<h1><?php echo ".$data['tree']."; ?></h1><p><b>Client</b><br><?php echo ".$data['name']. " ".$data['surnames']."; ?><br><br><b>Direcció</b><br><?php echo ".$data['address']."; ?></p><p><b>P

```

```

laga</b><br><?php echo "".$data['plague']."";
?></p><p><b>Estat</b><br><?php echo "".$data['
status'].""; ?></p>"

var infowindow<?php echo $i;?> = new google.maps.InfoWindow({
    content: contentString
});

```

Finalment per mostrar el globus al clicar:

```

google.maps.event.addListener(marker<?php echo $i;?>, 'click', function() {
    infowindow<?php echo $i;?>.open(map,marker<?php echo $i;?>);
});
</script>

```

També s'incrementa la variable comptador \$i de marcadors:

```

<?php $i++;
} ?>

```

- Finalment, s'acaba la part <body> i es tanca el document <html>:

```

</body>
</html>

```

Com es veu, és el document "home.php" es bastant breu però conté tot el que he anat explicant fins ara quant a llenguatges emprats i part de lògica i de presentació. Al presentar informació, està present en el diagrama de navegació que he esmentat i explicat al principi. A continuació procediré a explicar l'altre tipus de document que conforma l'aplicació web, els que no contenen informació per presentar sinó que són codi PHP SQL purament.

Per aquest segon tipus de document, els lògics, encarregats exclusivament de comunicar-se amb la base de dades, només dir que la seva estructura és una mica diferent, ja que no contenen gairebé HTML, només PHP i SQL. És per això no tindran enllaços a fulls d'estils, però les etiquetes HTML d'estructura de documents (<html>, <head> i <body>) si que hi seran, pel fet que he d'incloure una línia amb l'etiqueta <script> per invocar una redirecció automàtica del document de comunicació amb la base de dades cap a un altre document. El document que he escollit es el que insereix un arbre nou al sistema i s'anomena "crea\_arbre.php":

- Comença de forma idèntica que l'altre, amb una comprovació de validesa de la sessió d'usuari:

```

<?php
//SESSIÓ D'USUARI
session_start();
if(!isset($_SESSION["user_id"]) || $_SESSION["user_id"]==null){
    print "<script>alert(\"Accés invàlid!\");window.location='login.php';</script>";
}
?>

```

- Després, segueix el procés de comprovació de les variables rebudes del document anterior a aquest. En aquest cas les dades provenen d'un

formulari on es permet la introducció d'informació sobre un arbre. Com es pot veure, es recullen dades enviades pel mètode POST, i en cada subcondicional, es comprova que la informació rebuda no sigui buida ( ' ') i si no ho és, es guarda dins una variable:

```
<?php
//ES RECULLEN LES DADES A INSERTAR DEL FORM QUE APUNTA A AQUEST DOCUMENT
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if ($_POST['id_customer'] != '') { //SI LA VARIABLE NO ES BUIDA,
        $id_customer = $_POST['id_customer']; //ES GUARDA DINS UNA VARIABLE
    }
    if ($_POST['identificador'] != '') {
        $identificador = $_POST['identificador'];
    }
    if ($_POST['tipus_arbre'] != '') {
        $tipus_arbre = $_POST['tipus_arbre'];
    }
    if ($_POST['plaga'] != '') {
        $plaga = $_POST['plaga'];
    }
    if ($_POST['estat'] != '') {
        $estat = $_POST['estat'];
    }
    if ($_POST['latitude'] != '') {
        $lati = $_POST['latitude'];
    }
    if ($_POST['longitude'] != '') {
        $longi = $_POST['longitude'];
    }
}
```

- Ara, es procedeix a connectar-se amb la base de dades, ben igual que he explicat abans:

```
<!-- CONNEXIÓ AMB LA BD -->
<?php
    $host="servidor de BD";
    $user="usuari";
    $password="mot de pas";
    $db="nom BD";
    $con = new mysqli($host,$user,$password,$db) or die;
?>
```

- Després d'establir la connexió, s'executa una consulta, en aquest cas per inserta informació i també una sentència per actualitzar altres valors, INSERT i UPDATE, i es tanca la connexió:

```
//CONSULTA QUE INSEREIX LES DADES ENTRADES
```

```

$query = "INSERT INTO `tree` (`id_tree`, `chip`, `lat`, `lng`, `id_customer`, `id_specie`
      , `id_status`, `id_plague`) VALUES (NULL, '$identificador', '$lati', '$lon
      gi', '$id_customer', '$tipus_arbre', '$estat', '$plaga)";
$resultado = mysqli_query($con,$query);
//CONSULTA QUE INVALIDA UNA NOVA INSERCIÓ DEL XIP EMPRAT (EVITA DUPLICATS)
$query = "update chip set valid = '0' where chip = '$identificador'";
$resultado = mysqli_query($con,$query);
//eS TANCA CONNEXIÓ AMB LA BD
mysqli_close($con);
?>

```

- Finalment, continua la part HTML que conté l'script de redirecció cap a la pàgina "catalogats.php":

```

<html>
  <head>
  </head>
  <body>
    <!--SCRIPT QUE REDIRIGEIX UN COP FET L'INSERT-->
    <script type="text/javascript">
      setTimeout("window.location='catalogats.php'",0);
    </script>
  </body>
</html>

```

Aquest segon tipus de document es menys extens, però al tenir una part de lògica tan important, fan que sigui molt més complicat crear-los. És així perquè aquesta lògica no pot fallar i ha de funcionar perfectament i no deixar res a la sort, tot comportament ha d'estar previst.

Tota l'aplicació web es pot provar entrant al web <https://synaptree.es/login.php> i emprant les següents credencials:

- Usuari: llore
- Contrasenya: 661742124

Es pot veure com he anat introduint dades fictícies en base a la informació que he anat recollint.

### 3.6. API de GMaps.

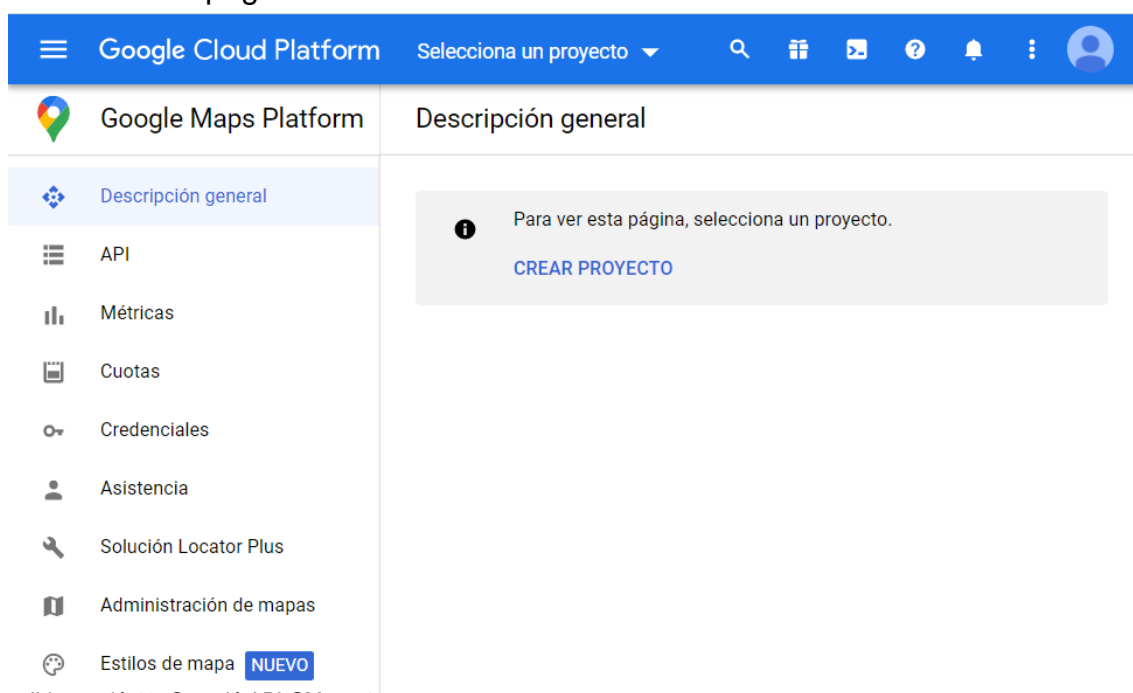
Per poder emprar els mapes i posicionar amb un marcador un arbre, o un client, que també els posiciono, s'ha de fer servir la API de Google Maps. Una API (Application Programming Interface, o Interfície de Programació d'aplicacions) és una clau que s'empra perquè una donada aplicació tingui permisos per emprar els mapes de Google, ja que està vinculada a un compte d'usuari i des d'aquesta compta, un programador genera la clau, o API, i a través del codi de l'aplicació la proporciona al sistema. Aquesta aplicació al fer la sol·licitud de càrrega d'un mapa, empra la clau per identificar-se i Google la verifica i permet carregar els

mapes. En cas que aquesta clau no sigui vàlida o no s'empri cap API, el mapa no es mostrarà.

Per obtenir una API caldrà seguir les passes següents:

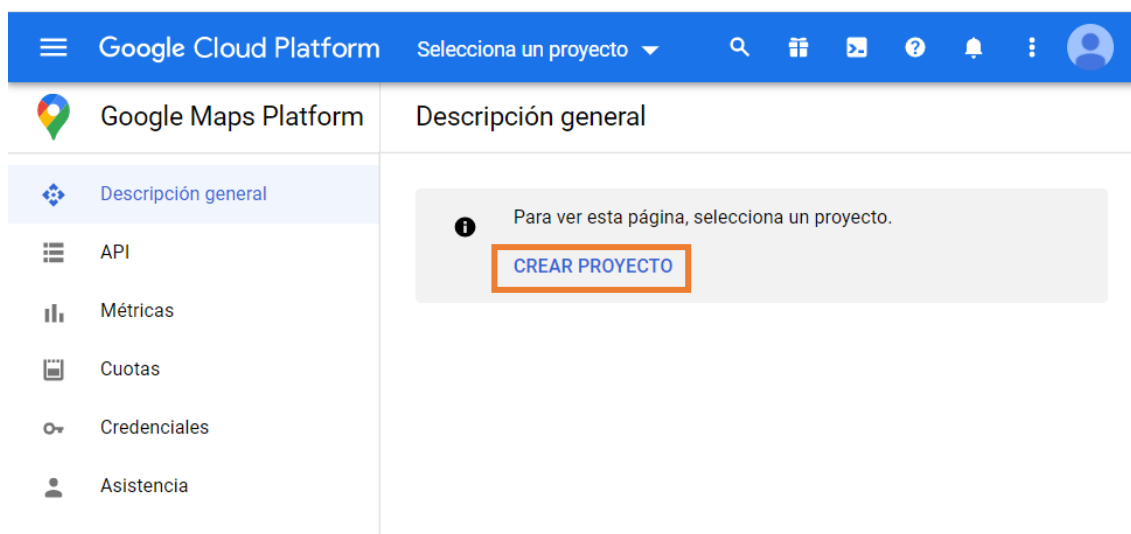
1. Crear un compte a Google, un compte de Gmail. Es pot obtenir des de <https://accounts.google.com/SignUp>
2. Acte seguit, amb la sessió iniciada a partir del compte creat anteriorment, o un ja existent, s'ha d'entrar al panell "Google API Console" <https://console.developers.google.com/apis/dashboard>

Es veurà una pàgina així:



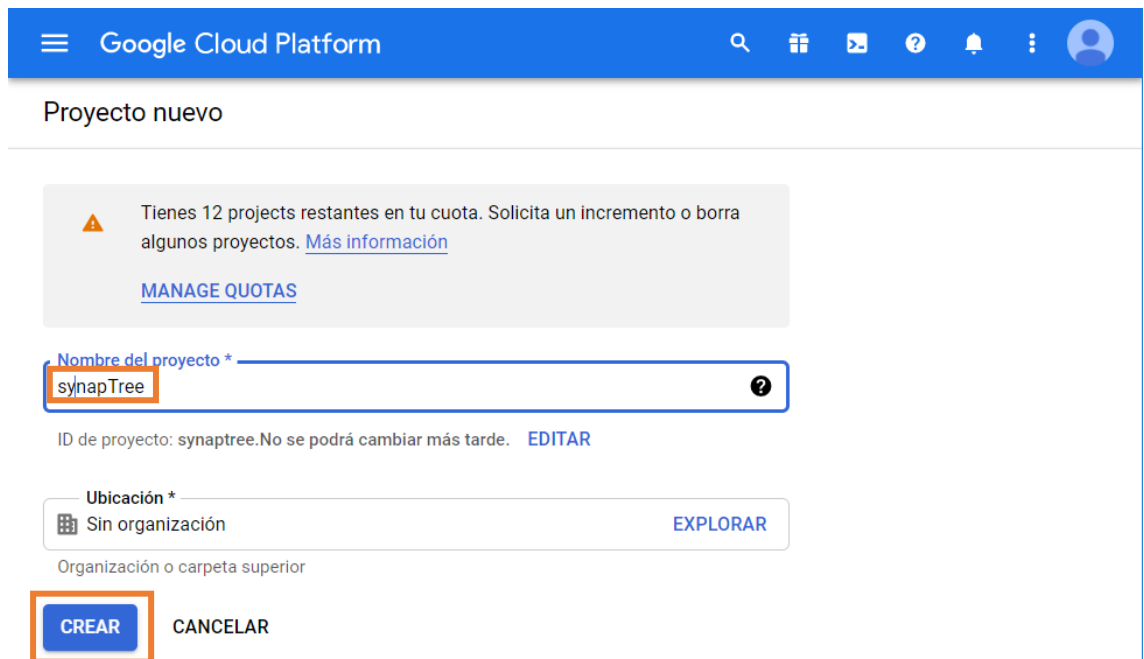
Il·lustració 42. Creació API GMaps 1

3. Per aconseguir una API, s'haurà de tenir un projecte en la plataforma, cosa que obliga a crear-ne un. Per fer-ho, s'ha de clicar a "Crear":



Il·lustració 43. Creació API GMaps 2

4. L'única dada que s'ha d'indicar és el nom del projecte, que pot ser qualsevol i clicar a "Crear":



Google Cloud Platform

Proyecto nuevo

Tienes 12 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)

Nombre del proyecto \*

ID de proyecto: synaptree.No se podrá cambiar más tarde. [EDITAR](#)

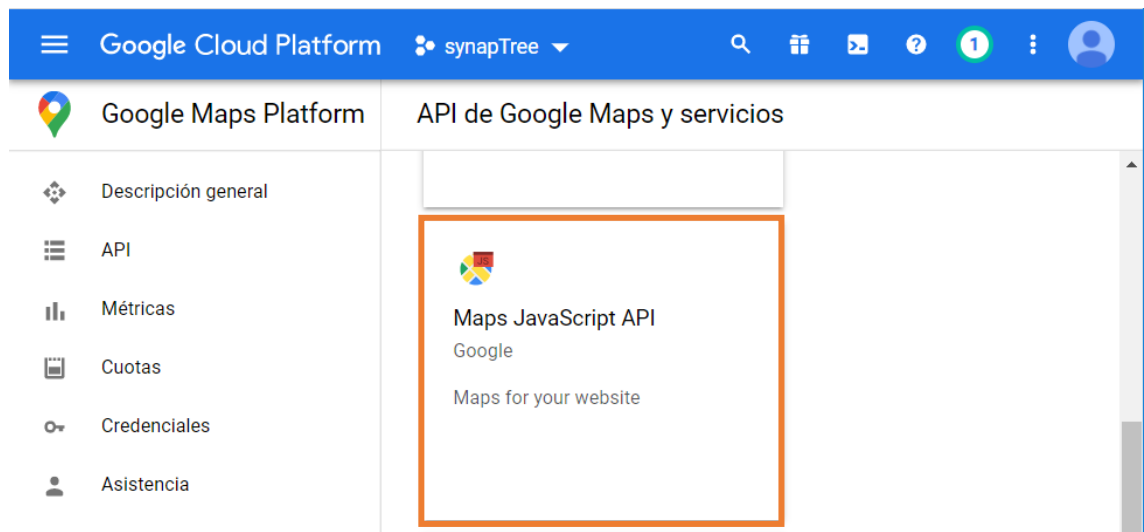
Ubicación \*  
Sin organización [EXPLORAR](#)

Organización o carpeta superior

[CREAR](#) [CANCELAR](#)

Il·lustració 44. Creació API GMaps 3

5. Acte següent, s'haurà d'indicar quin servei és el que es vol exactament. Es pot escollir entre diversos, com ara Geolocation API, Roads API, etc. El que interessa pel projecte, és "Maps JavaScript API". Aquest servei és el que permet incrustar mapes en un lloc web emprant JavaScript i emprant la clau API que es generarà en les passes pròximes. Per tant, s'escollirà "Maps JavaScript API":



Google Cloud Platform synapTree

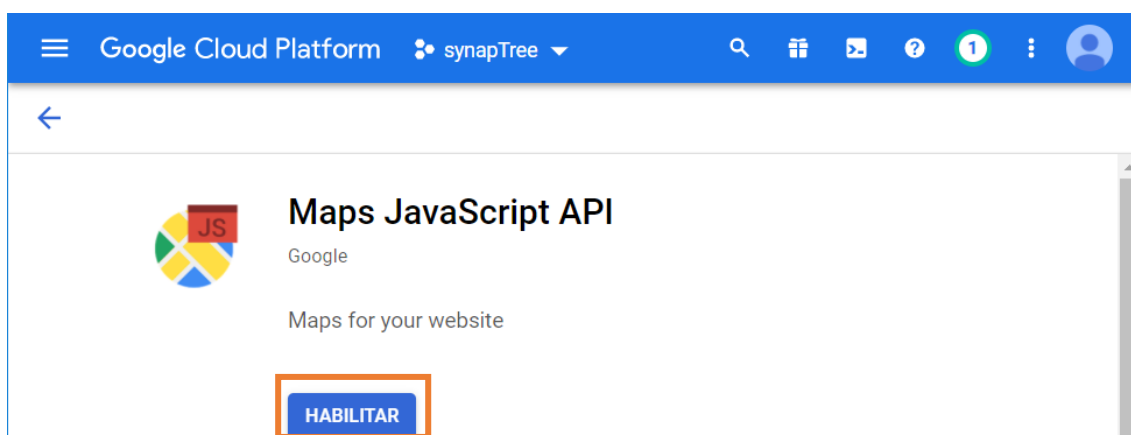
Google Maps Platform API de Google Maps y servicios

- Descripción general
- API
- Métricas
- Cuotas
- Credenciales
- Asistencia

Maps JavaScript API  
Google  
Maps for your website

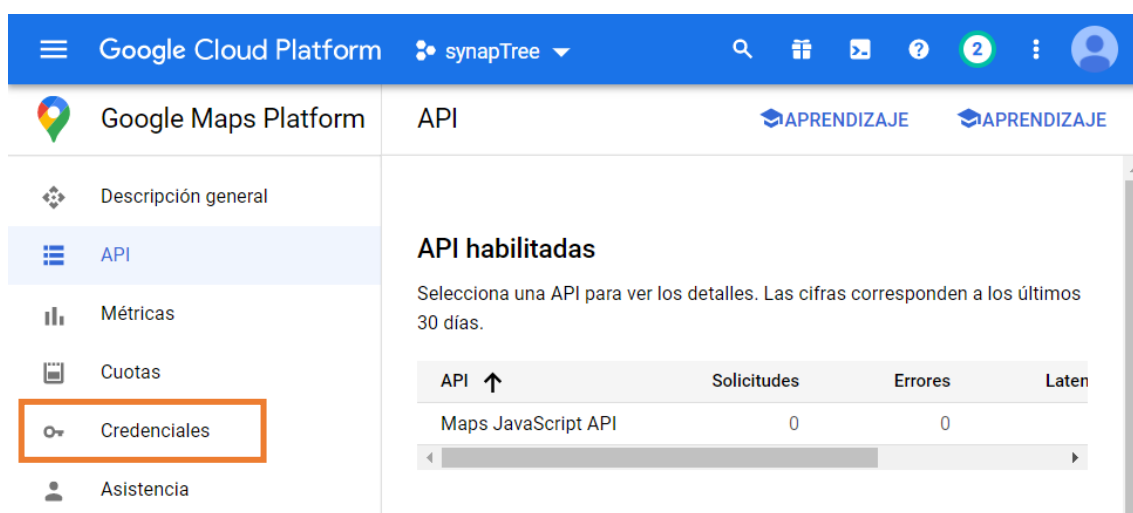
Il·lustració 45. Creació API GMaps 4

I un cop escollida, s'haurà d'habilitar en la pròxima pàgina:



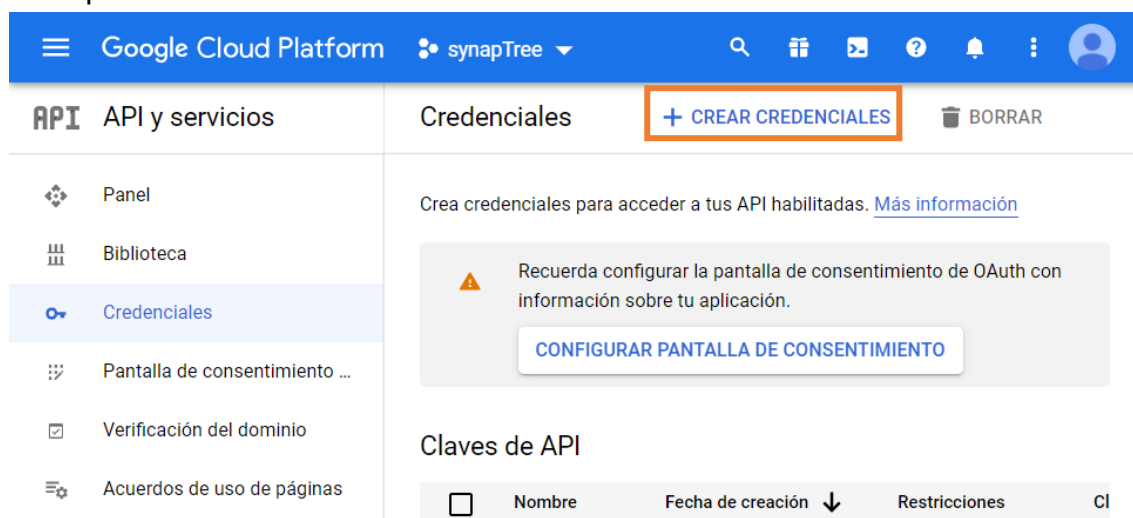
Il·lustració 46. Creació API GMaps 5

6. Després s'hauran de crear unes credencials, clicant sobre "Credenciales" del menú lateral:



Il·lustració 47. Creació API GMaps 6

I després a "crear credenciales"



Il·lustració 48. Creació API GMaps 7



Es mostraran diverses opcions, però cal triar “Clave de API”:

### Clave de API

Identifica tu proyecto con una clave de API simple para verificar la cuota y el acceso

### ID de cliente de OAuth

Solicita el consentimiento del usuario para que tu app pueda acceder a sus datos

### Cuenta de servicio

Habilita la autenticación de servidor a servidor en el nivel de la app mediante cuentas robot

### Ayúdame a elegir

Responde algunas preguntas para decidir qué tipo de credencial usar

Il·lustració 49. Creació API GMaps 8

- Finalment, apareixerà un requadre amb la clau generada i que és la que s'emprarà en el codi del lloc web quan s'hagi d'incloure un mapa:

## Se creó la clave de API

Para usar esta clave en tu aplicación, transfírela con el parámetro `key=API_KEY`.

### Tu clave de API

AIzaSyAhyUqe1W-6QgMuIAXEwfL3VjHHMR59HZ8



Restringe tu clave para impedir el uso no autorizado en producción.

Il·lustració 50. Creació API GMaps 9

- Un cop generada la API es podrà emprar. En el meu cas, l'he inclosa en el bloc `<head>` de cada document on l'he necessitada i dins una etiqueta `<script>`:

```
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAhyUqe1W-6QgMuIAXEwfL3VjHHMR59HZ8"></script>
```

- Una vegada obtinguda l'API i mencionada en el document HTML o PHP, ja es pot emprar JavaScript per pintar el mapa i que aquest sigui visible des del web:

## Informació de l'arbre

### Client

DNI 41745159K  
Nom Llorenç  
Llinatges Coll Hernández

### Característiques

Identificador 48193205163  
Tipus d'arbre Pi mediterràni  
Estat Sano  
Plaga Processionària del pi

### Historial d'intervencions

Data 2021-02-01	Tècnic <i>Llorenç</i>	Plaga <i>Processionària del pi</i>	Motiu <i>Valoració</i>	<a href="#">Detalls...</a>
Data 2021-03-03	Tècnic <i>Miguel</i>	Plaga <i>Processionària del pi</i>	Motiu <i>Morrut verme</i>	<a href="#">Detalls...</a>
Data 2021-03-12	Tècnic <i>Pedro</i>	Plaga <i>Processionària del pi</i>	Motiu <i>Morrut verme</i>	<a href="#">Detalls...</a>

[Nova intervenció](#)



Il·lustració 51. Prova GMaps a la web

## 4. Desenvolupament del projecte.

El procés de desenvolupament del projecte ha sigut força seqüencial, excepte les feines de redacció marcades per l'assignatura que s'han desenvolupat de forma transversal. El projecte va començar amb una pluja d'idees de projectes Arduino que satisfessin necessitats actuals que no fossin cobertes per altres solucions. Almenys d'una forma molt específica i normalitzada. Durant la pluja d'idees vaig pensar amb una plaga que està afectant actualment a l'illa de Menorca i que m'afecta de ben a prop perquè el meu pare es dedica a combatre-la. Aquesta plaga és la de morrut vermell i afecta a les palmeres i si no s'intervé a temps, acaben morint de forma molt prematura a causa d'aquest insecte. Tenint aquesta problemàtica en ment, va venir-me al cap construir una eina que faciliti portar un control d'aquesta i altres plagues, com la processionària del pi. És per això que neix SynapTree, una eina que facilita el treball de camp dels tècnics en control de plagues, ja que no han de registrar les intervencions en paper, sinó que basta que ho registrin a l'aplicació web i ja queda en el sistema de l'empresa. A partir d'aquí, aquesta empresa pot fer anàlisi d'extensió de plagues, control, percentatge d'èxit o fracàs, etc. Un cop aquesta idea va anar agafant pes, vaig decidir comentar-la amb el meu pare i la seva valoració va resultar força satisfactòria. Però no només volia quedar-me amb la seva opinió. És per això que vaig parlar amb un enginyer agrònom que es dedica a la cura de vinyes. Vaig extrapolar la situació del morrut vermell en les palmeres, però amb la problemàtica actual d'un negoci que es dedica a la producció de vi. Vaig comentar-li que podria registrar els kilograms de raïm extrets d'una planta donada o els tractaments que havia rebut i a l'hora de treure els nombres, podia veure de forma molt gràfica el que s'havia fet, quan, com i on. Amb això, va veure que podia controlar el temps entre una intervenció i altra i el que es tardava en cobrir tota l'àrea de vinyes per una mateixa intervenció però a diferents plantes. Això permet un control de temps força important per implementar uns procediments de cura i també per controlar el que s'extreu de les plantes en anys diferents. La seva valoració també va ser sorprenentment bona, ara bé, jo m'he centrat en el control de plagues en general, perquè el puc implementar just acabar el projecte.

Un cop la idea està agafada, segueixo amb la planificació del projecte. En aquesta part he de pensar quines tasques he de seguir i desenvolupar per arribar al meu objectiu. És important tenir clars els objectius del projecte y la durada que ha de tenir cada tasca. En aquesta part, he hagut de fer modificacions així com anava avançant, ja que segons una planificació inicial, una tasca no ha durat tant com estava planejat, i al revés, algunes han portat més temps del pensat en un principi. Per fer la planificació de les tasques i el temps que s'ha de dedicar a cada una, vaig fer un diagrama de Gantt que està al principi d'aquesta memòria, i va ser desenvolupat amb l'eina online LucidChart. Com he dit, aquest diagrama ha sigut modificat en dues ocasions per motius de planificació i també de durada de les tasques.

És hora de pensar com fer un sistema que permeti fer això. Vaig decidir que es podien etiquetar els arbres i amb un lector que es connecta amb una base de dades es pot obtenir informació, modificar-la o crear-ne de nova. Aquí és on el lector conformat pel lector RFID i la NodeMCU agafa forma. Posteriorment, afegeixo el mòdul GPS i més tard, la pantalla. Una vegada tots els elements estan definits, passa a ser el moment d'unificar-los per crear el lector. Començo amb els mòduls per individual fent petites proves de funcionament per conèixer-los una mica. Després creo el firmware definitiu que fusiona el funcionament de cada mòdul i dota de comunicació amb una base de dades al lector. Pel disseny del firmware i les proves, empro l'IDE d'Arduino i el muntatge queda ben il·lustrat amb el software Fritzing. Aquesta part ha tingut la problemàtica de perdre molt temps per provar coses i mirar què funciona i com funciona. Per obtenir fruits d'aquesta part, seguir una metodologia de prova i error ha sigut fonamental. Ja que he hagut de fer ajustos contínuament a mesura que incorporant els diferents mòduls al lector. Sense dubte ha sigut una part complicada i la part que m'ha fet perdre més temps i que he estat a punt de descartar, ha sigut la del mòdul GPS. HA sigut així perquè on feia les proves amb el mòdul, no tenia cobertura GPS, de manera que pensava que el que passava és que hi havia problemes amb el codi o el pinejat. En un darrer esforç, vaig decidir anar enmig del carrer amb un portàtil i tot el cablejat que això implicava i fer córrer un sketch de prova. Quan vaig veure que podia veure les coordenades per pantalla, va ser una alegria gegant. Me va permetre prosseguir el projecte tal i com havia planejat, amb el mòdul GPS inclòs.

Un cop es té el lector funcionant i atacat una base de dades de prova, és hora de prosseguir amb la part web del projecte. Aquesta part no es tan complicada com la primera, però sí que comporta una gran quantitat de temps. Decideixo programar aquesta part amb HTML, CSS, JavaScript, SQL i PHP amb un previ disseny de la navegació fet, que passava per dissenyar com funcionaria el lloc i com serien les pàgines que el conformarien. Aquesta fase de disseny, com dic, es va fer a partir de diagrames generats amb Lucid, una eina online que en la seva forma gratuïta ja va fer-me el servei. Per la navegació, vaig emprar LibreOffice Draw, ja que havia de dissenyar una forma intuïtiva i lògica, perquè qualsevol usuari pugues navegar pel lloc. Aquesta fase de disseny de la navegació va ser complicada i vaig emprar diverses persones per establir el lloc de cada cosa i en funció de l'opinió d'aquestes persones, vaig establir el camí òptim per la millor navegació.

El fruit d'aquesta combinació de llenguatges, disseny de navegació i els formularis, ja ha quedat ben explicada uns apartats més amunt. Aquesta part ha tingut una gran dedicació temporal degut al gran nombre de documents PHP, la creació de la base de dades, preparar l'intercanvi de dades entre l'aplicació web, la base de dades i el lector, perquè aquest fos introduït correctament dins la base de dades, amb el tipus de dada correcte. Aquí vaig trobar-me amb que quan passava dades GPS a base de dades, dins la base de dades només es guardaven un nombre molt petit de decimals, tot i enviar un decimal força llarg. Provant coses, vaig arribar a trobar-hi solució i aquesta passa per una solució

força fàcil: canviar el tipus de dada des del propi servidor de base de dades. Si no ho feia, obtenia unes coordenades molt imprecises i abarcaven una gran àrea. Per dissenyar la base de dades, vaig emprar una eina gratuïta d'Internet anomenada Verbatelo. Per programar tots i cada un dels documents que conformen el lloc web, vaig emprar Visual Studio Code, i per pujar els documents generats al servidor, FileZilla.

Un cop les dues parts estaven construïdes i es comunicaven, quedava la part de fer proves i posar-ho en marxa per veure els errors que sortien. Com he dit, un dels problemes amb els que vaig trobar-me, és que el GPS no acabava de ser prou precís com per posicionar un arbre. Per això, vaig decidir que enlloc de que el mòdul el posicionés, aquest passés a tenir una funció d'ajuda a l'usuari per posicionar l'arbre aquest últim. Per tant, el GPS servia per ubicar aproximadament l'arbre i l'usuari l'ubiqués pel mapa que té en pantalla. D'aquesta manera evitava errades de posició i aquesta queda molt precisa. Ho he decidit així perquè es dona el cas que un arbre a censar pot tenir arbres veïns, i amb un error de 3 o 4 metres, potser un arbre etiquetat amb l'etiqueta X, la posició dins el sistema passava a estar més propera a l'arbre veí que a la del arbre en qüestió.

De forma transversal a tot això i durant tot el projecte, d'inici a final, hi ha hagut una feina de redacció i documentació molt important. Aquestes tasques de redacció han sigut fonamentals per assolir les entregues que ha marcat l'assignatura, des de la PAC 1 fins a la presentació. Per elaborar aquestes entregues, he emprat la suite ofimàtica LibreOffice.

## 5. Usabilitat.

La usabilitat ha sigut un repte des del principi del projecte, perquè tant el lector com l'aplicació web, sobretot per aqueta darrera.

Pel lector, en principi ha de tenir un funcionament força senzill i intuïtiu. Basta amb apropar el lector a l'etiqueta RFID que estarà col·locada a un arbre. I ja n'hi ha prou. Ara bé, així com vaig anar avançant, vaig veure que a vegades hi havia errors de comunicació amb la base de dades o es feien lectures múltiples d'una etiqueta RFID. Davant aquestes dos problemes vaig optar per solucions contundents:

- Per saber l'estat del lector i evitar problemes de dubtes de si s'havia produït bé una lectura o no, si la xarxa estava ben configurada o si l'estat d'aquest era preparat per llegir, vaig decidir afegir una pantalla LCD per donar un feedback a l'usuari. Amb aquesta pantalla puc alertar a l'usuari dels diversos estats i errors que es poden produir. La pantalla ofereix un feedback en les següents situacions:
  - o Just a l'iniciar, avisa de que s'està intentat connectar a la xarxa que es té configurada al firmware del lector:



Il·lustració 52. Intent de connexió per LCD

- o Un cop s'ha establert connexió per WiFi a la xarxa, també es notifica i es dona la IP:



Il·lustració 53. Connexió a WiFi feta

- Quan la connexió està establerta, es dona un missatge de benvinguda:



Il·lustració 54. Benvinguda per LCD

- Al cercar senyal GPS abans de poder escanejar:



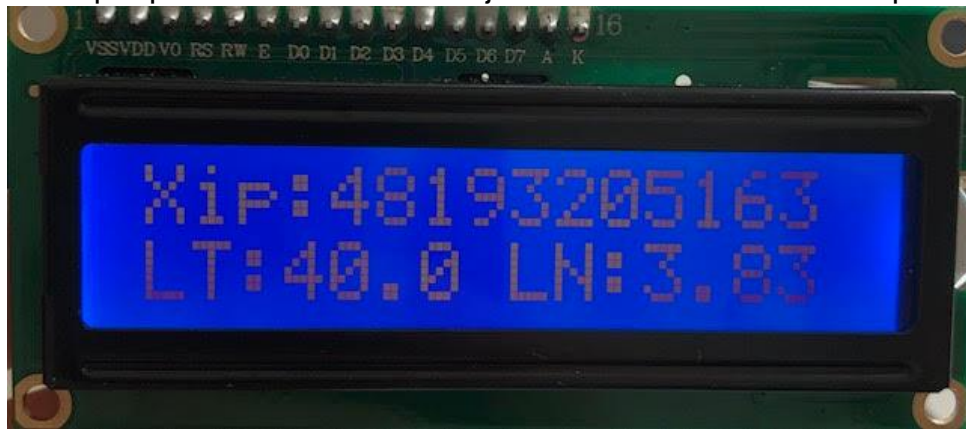
Il·lustració 55. Intent de connexió per LCD

- Quan s'està a punt per fer un escaneig d'una etiqueta:



Il·lustració 56. Lector preparat

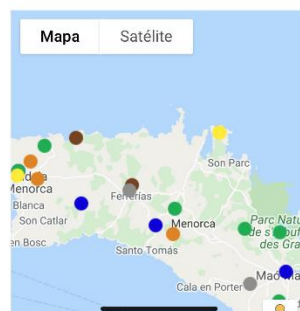
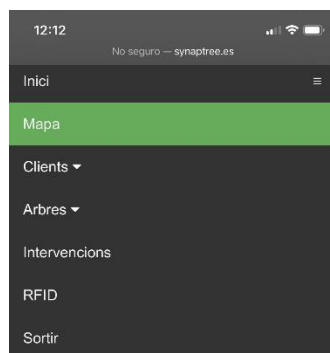
- I finalment, quan es produeix una lectura exitosa d'una etiqueta. Es treu per pantalla el valor escanejat i les coordenades de l'etiqueta:



Il·lustració 57. Lectura d'etiqueta

Quant a l'aplicació web, he intentat que l'experiència de l'usuari fos bona gràcies a una navegació intuïtiva i senzilla, on els apartats i la ubicació de cada pàgina per intentar satisfer una necessitat d'informació de l'usuari, està dintre de la categoria (o menú) que he cregut més lògic. És per això que he intentat que:

- La navegació sigui intuïtiva.
- No es requereixi un aprenentatge que impliqui perdre hores i hores per poder emprar correctament l'aplicació.
- Les coses siguin fàcilment llegibles i la web s'adapti a diferents dispositius, com ara els mòbils. Com es veu a continuació, en un mòbil, la informació en pantalla s'adapta a la resolució més petita i el menú enloc de ser sempre visible, es torna desplegable (clicant a les 3 barres de dalt a la dreta):



Il·lustració 58. Disseny responsive del menú



- El codi de colors emprat en el disseny es sempre el mateix, ja que emprava verds, negre i tons de gris.
- Si hi ha algun problema, sempre es notifica a l'usuari. Per exemple en el cas de no detectar una lectura d'un RFID des de l'aplicació es notifica que no s'ha detectat. També s'avisava si l'etiqueta és nova i ja existeix un arbre amb aquest identificador.

## 6. Conclusió

Estic força satisfet del projecte que he sigut capaç de desenvolupar de principi a fi. Gràcies a Arduino, una plataforma oberta que permet el desenvolupament d'eines com aquesta, ha sigut fàcil aprendre tot el necessari per poder-ho fer, i a un preu molt bo. El resultat ha sigut un lector que ja és funcional, amb una projecció a futur molt prometedora i una implementació que és força senzilla i no requereix més que un parell d'hores d'aprenentatge.

Si miro endarrere, crec que els objectius que proposava al principi d'aquesta memòria, han sigut assolits completament. Cosa que com autor me satisfà immensament.

### 6.1. Problemes.

Els problemes que he anat trobant han pogut ser solventats gràcies a la paciència i la dedicació de temps. El problema mes comú amb el que m'he trobat, han sigut els diversos canvis de planificació que he hagut de fer, ja que en un principi un es fa a la idea de la dedicació de X hores a una tasca, i a l'estar amb aquesta tasca, el temps planificat no és l'adequat. Aquest problemes bàsicament són 3:

- Al principi, quan estava provant el mòdul GPS, cap dels exemples de skecths de prova funcionava. Vaig arribar a demanar 4 mòduls GPS de diferents models per fer proves. Al veure que cap d'ells funcionava. Vaig arribar a la conclusió que el problema no eren els mòduls, sinó que era un altre. Vaig provar amb diferents plaques NodeMCU també, però veia que succeïa el mateix. Finalment, el problema era que dins casa, tot i estar aferrat a una vidriera ben gran, el senyal GPS no era prou bo, cosa que va fer-me sortir al carrer per fer les proves pertinents cada vegada que emprava el mòdul. Amb les comandes de nous mòduls i diferents proves amb plaques Arduino UNO, NodeMCU i fins i tot ordinadors diferents, vaig perdre no hores, sinó dies. Es clar que no vaig quedar aturat, seguia endavant amb altres tasques.
- Per no deixar els problemes amb el GPS, a darrera hora va sortir-ne un altre. I es que a la base de dades no arribaven prou decimals per les coordenades. Això si que va ser més fàcil de solucionar, ja que va bastar en canviar el tipus de dada sobre la base de dades. Des del principi, el tipus de dada que tenia assignada a la longitud i la latitud, era FLOAT. Però amb aquest problema vaig anar fent proves i ha acabat sent una dada de tipus DOUBLE. Amb això s'acaba amb aquest problema.
- I el tercer, però no menys important, és que el lector no tenia un feedback amb l'usuari. Un usuari no sabia si el GPS estava a punt, si la connexió amb el punt d'accés era exitosa o si la lectura havia sigut correcte. La solució d'això passa per afegir una pantalla al lector i avisar del que està passant a l'usuari. Ho he explicat en l'apartat d'usabilitat. Es clar que afegir una pantalla implica reestructurar el temps perquè s'arribi a la data límit amb el projecte acabat.

Fonamentalment tot ha estat qüestió de dedicar-hi més o menys temps, però aquest temps és un bé que al estar treballant, no sempre es disposa del temps que un voldria. Tot i així, he robat el temps d'on fes falta, sobretots dels vespres i els caps de setmana, tot i que cada dia hi he dedicat una mitja de 6 hores a desenvolupar tot això.

## 6.2. Projecció a futur del projecte.

La principal necessitat que satisfà aquest projecte, és la de proporcionar informació a una empresa, entitat o persona física, que disposi d'una sèrie d'arbres dels que ha de tenir cura en termes d'historials d'afeccions, podes, plagues, estats, etc. pel que passa un arbre donat. Això es permet gràcies a l'ús combinat del lector i l'aplicació web. Aquesta informació que s'obté permet treure estadístiques o crear mapes d'avanç o retrocés de plagues.

No obstant, el lector és un prototip, i encara que és petit i bo de manejar, hauria de dissenyar una placa amb un circuit imprès i que ja integri els components i mòduls que ara mateix té el lector però amb una versió més compacte. Igualment, es podria adaptar més l'aplicació web per portar un control d'estoc de químics.

Tot això, però, són millores que s'han de fer de cara al futur, ja que tant el lector com l'aplicació web són completament funcionals. Aquests canvis o millores que estic plantejant són purament estètics i pràctics, no funcionals.

## 7. Annexes.

### 7.0. Índex d'annexes

1. Documents adjunts a la memòria ..... Pàg. 64
2. Lògica de SynapTree ..... Pàg. 65
3. Llibreries Arduino ..... Pàg. 68

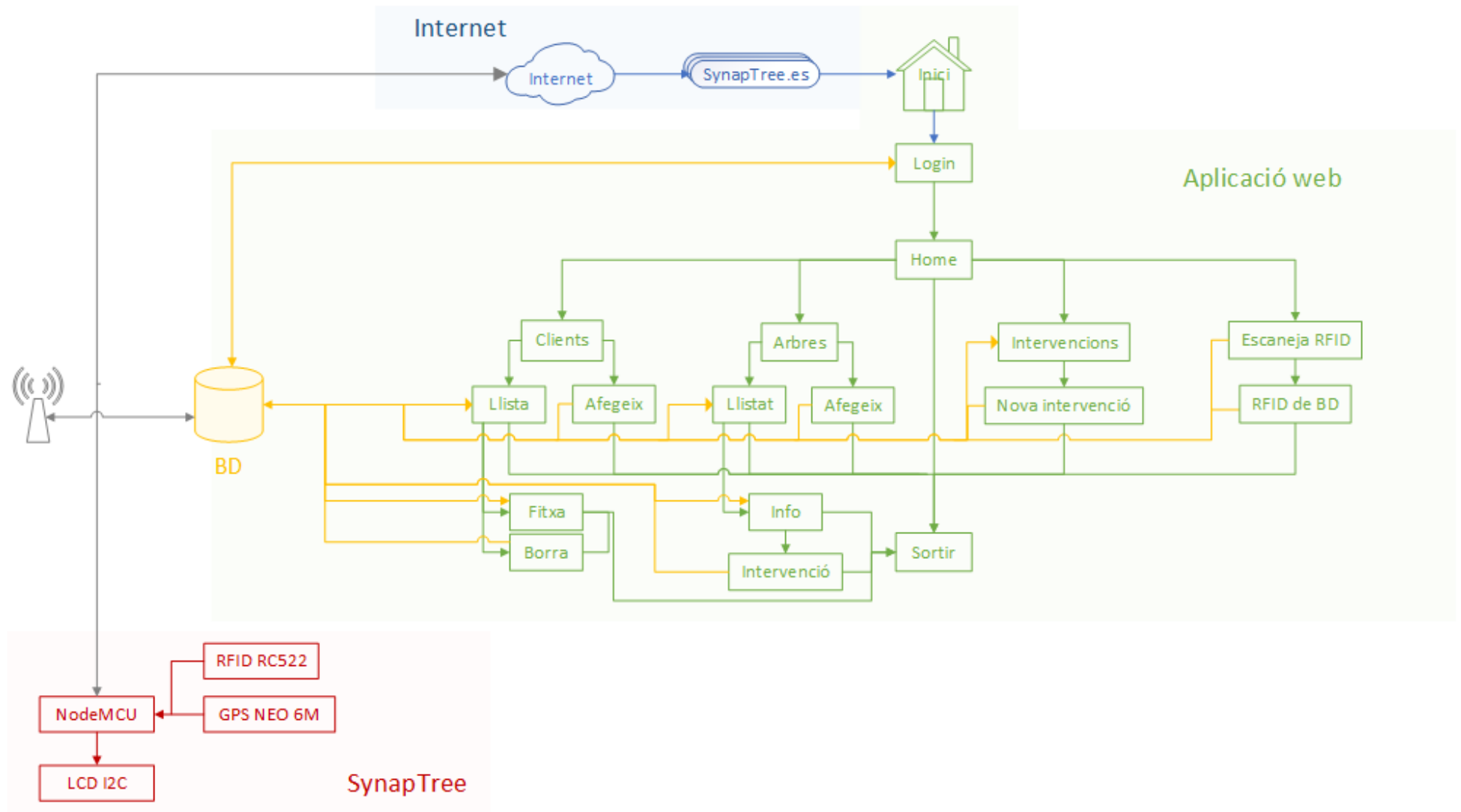
### 7.1. Documents adjunts a la memòria.

Dins l'arxiu comprimit de l'entrega anomenat "TFG.Memòria – Llorenç Coll Hernández.zip" hi ha una sèrie d'elements:

- **Diagrames:** dins aquest directori hi ha:
  - 1. El diagrama de Gantt del projecte amb nom "Gantt del projecte SynapTree.pdf": ve a ser el diagrama de Gantt del principi del document però en format PDF perquè sigui llegit de forma més còmode.
  - 2. El diagrama del projecte sencer del que parlo en l'annex 2, 2. Lògica de SynapTree. L'arxiu està en format PDF perquè sigui llegit i ampliat al gust. S'anomena "Diagrama del projecte SynapTree.pdf".
  - 3. El diagrama de navegació del que he parlat en l'apartat 3.2. La navegació. És una imatge en format PNG i amb nom "Diagrama web.png".
  - 4. El diagrama de la base de dades que s'ha vist en l'apartat 3.4. La base de dades. Aquest arxiu està en format PDF i s'anomena "Base de dades SynapTree.pdf".
  - 5. L'esquema de muntatge sencer del lector. Fet amb Fritzing. Amb el nom "Lector SynapTree.png"
  
- **Wireframes:** directori que inclou els dissenys inicials de com havien de ser les pàgines del lloc, així com alguns formularis.
  
- **Memòria:** no és un directori, és tracta d'aquest document que s'anomena "Memòria - SynapTree.pdf".



## 7.2. Lògica de SynapTree.



Aquest diagrama descriu la navegació, el flux de dades entre la base de dades i els documents que conformen el lloc, els elements que s'interconnecten gràcies a la xarxa 3G/4G i també els elements que conformen el lector SynapTree. Com s'aprecia en el diagrama, hi ha un codi de colors:

- **Blau:** és la part d'Internet, la part externa al servidor web i el domini a partir del qual s'accedirà al lloc web.
- **Vermell:** són els elements que conformen el lector SynapTree. Els mòduls Arduino i la placa NodeMCU.
- **Gris:** és la part de comunicacions. Són els elements tecnològics que permetran la comunicació a través d'Internet del lector, la base de dades i l'aplicació web. En aquest cas serà la tecnologia 4G/5G perquè el dispositiu mòbil que farà de porta d'enllaç pel lector, emprerà aquests estàndards.
- **Groc:** és la base de dades allotjada al servidor i les fletxes grogues, marquen la direcció del flux de les dades entre la BD i la pàgina en qüestió a través dels arxius PHP.
- **Verd:** és la part que té a veure amb aquest apartat ja que ve a ser el mapa del lloc web. En aquesta part hi ha totes les pàgines i subpàgines que conformen l'aplicació i les fletxes marquen com és la navegació.

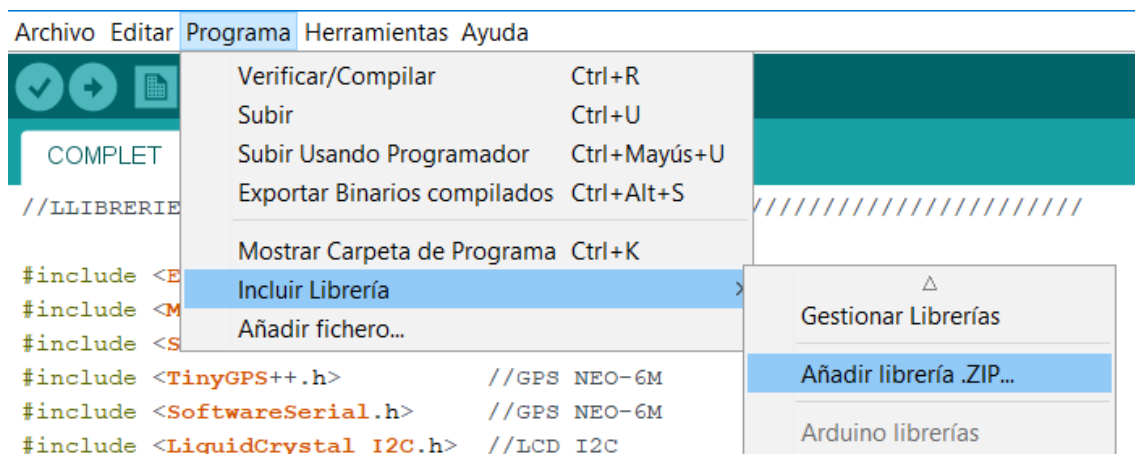


### 7.3. Lliberies Arduino.

Per programar el firmware el lector i emprar les funcions necessàries de cada mòdul, he fet ús d'una sèrie de lliberies Arduino. Les lliberies són un grapat de funcions desenvolupades per tercers i a més de les funcions que s'afegeixen, permet que el codi sigui més senzill i fàcil d'entendre. Normalment, les lliberies venen comprimides en ZIP i solen contenir:

- Un arxiu de codi escrit en C++ i amb l'extensió .cpp.
- Un arxiu en format .h que conté les propietats i funcions de la llibreria.
- Normalment també inclouen un arxiu TXT anomenat "keywords.txt" que ocnté les paraules reservades de la llibreria i fa que en l'IDE apareguin d'un altre color.
- Un readme amb informació addicional.
- Un directori d'exemples, on hi ha sketchs d'exemple per fer proves.

Per instal·lar una llibreria, basta en obrir l'IDE d'Arduino, anar al menú "Programa", "Inclur Llibrería" i "Añadir librería .ZIP..."



Les lliberies emprades en aquest projecte són:

- **ESP8266WiFi.h:** és la llibreria per la placa NodeMCU i facilita la connexió a una xarxa inalàmbrica. Empra la pila de protocols TCP/IP i conté funcions que faciliten aquest procés de connexió com "**wifi.begin(nom, contrasenya)**".
- **MFRC522.h:** és la llibreria que he emprat per llegir etiquetes RFID amb el lector RC522.
- **SPI.h:** també l'he emprada per el lector RFID RC522, concretament epr inicialitzar l'RC522 amb la funció "**SPI.begin()**".
- **TinyGPS++.h:** és la llibreria que permet descodificar les trames NMEA amb les que treballen els mòduls GPS y obtenir directament dades com la longitud i la latitud.
- **SoftwareSerial.h:** permet la comunicació per Serial des dels pins digitals de les plaques. Una funció molt emprada és "**print("text a treure per Serial");**"

- **LiquidCrystal\_I2C.h**: gràcies a aquesta llibreria es poden imprimir missatges per la pantalla LCD I2C. Una de les funcions que així ho permet és `LiquidCrystal_I2C.print("Missatge a treure per LCD")`.

## 8. Bibliografia

ESPRESSIF SYSTEMS. **ESP8266EX Datasheet** (2020). Consultat el març del 2021. Disponible a:

[https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)

Luis del Valle Hernández. **NodeMCU tutorial paso a paso desde cero** (2018). Consultat el març de 2021. Disponible a:

<https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

Luis del Valle Hernández. **Lector RFID RC522 control de acceso RFID con Arduino**. Consultat el març de 2021. Disponible a:

<https://programarfacil.com/blog/arduino-blog/lector-rfid-rc522-con-arduino/>

Tom PSH. **Parseo de una trama GPRMC** (2013). Consultat el març del 2021. Disponible a: <http://tohemu.blogspot.com/2013/11/parseo-de-una-trama-gprmc.html>

T. K. Hareendran. **NEO-6M GPS Module – An Introduction**. Consultat el març del 2021. Disponible a: <https://www.electroschematics.com/neo-6m-gps-module/>

Prometec. **Displays LCD**. Usando displays de texto LCD 16x2 con Arduino. Consultat el març del 2021. Disponible a: <https://www.prometec.net/displays-lcd/>

Apais. **Send Arduino Data to the web (PHP/MySQL/D3.js)** (2018). Consultat el març del 2021. Disponible a: <https://www.instructables.com/PART-1-Send-Arduino-data-to-the-Web-PHP-MySQL-D3js/>

Random Nerd Tutorials. **ESP32/ESP8266 Insert Data into MySQL Database using PHP and Arduino IDE** (2019). Consultat el març del 2021. Disponible a: <https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/>

MDN contributors. **HTML: Lenguaje de etiquetas de hipertexto** (2021). Consultat l'abril del 2021. Disponible a: <https://developer.mozilla.org/es/docs/Web/HTML>

W3Schools. **CSS Tutorial**. Consultat l'abril de 2021. Disponible a: <https://www.w3schools.com/css/>

MDN contributors. **JavaScript**. Consultat el Maig de 2021. Disponible a: <https://developer.mozilla.org/es/docs/Web/JavaScript>

PHP.net. **What is PHP?** Consultat el Maig de 2021. Disponible a: <https://www.php.net/manual/es/intro-what-is.php>