

# **Domòtica i Internet de les Coses**

**Marc Marcos Madruga**

**Merche Martín**

**17/12/2021**

**Curs 2021-2022**

# Abstrac

## Castellano

La domótica es un campo que pretende solucionar los problemas domésticos a través de soluciones tecnológicas. Estas soluciones pasan por gestionar el sistema de vigilancia y seguridad, gestionar las luces, medir la temperatura i la luz o subir y bajar las persianas en función de la hora y de la luz exterior.

Estos cambios ya están siendo instalados en cada año más hogares, pero es una transición lenta y que aún necesitará bastantes años para convertirse en algo adoptado por la mayoría de la población. La mayoría de esta transición está empezando por el uso creciente de asistentes virtuales que pueden desempeñar simples tareas tales como apagar las luces o cambiar de canción, estos parten con ventaja ya que son simples de usar y baratos.

Al intentar construir y programar una solución domótica por nuestra cuenta nos encontramos con que es un trabajo muy complicado y que requiere conocimientos en varias áreas. Necesitaríamos desde conocimientos de soldadura hasta programación de servidores. Esto nos demuestra que, aunque la domótica es un campo prometedor i con mucho futuro, es una mucha mejor idea recurrir a una de las muchas empresas privadas que se dedican a la instalación de sistemas domóticos antes que intentar montarlo uno mismo.

Estas empresas se encargan de que quede un sistema más seguro y profesional. Esto es primordial ya que exponer al sistema domótico a una brecha de seguridad puede ser devastador debido al gran daño que puede hacer un atacante a través de, por ejemplo, abrir la puerta del garaje o almacenar datos de cámaras o micrófonos.

## English

Domotics is a field that aims to solve domestic problems through technological solutions. These solutions involve managing the surveillance and security system, managing the lights, measuring the temperature and the light or raising and lowering the blinds depending on the time and the external light.

These changes are already being installed in more homes each year, but it is a slow transition and it will still take many years to become something adopted by the majority of the population. Most of this transition is beginning with the increasing use of virtual assistants that can perform simple tasks such as turning off the lights or changing songs, these start with an advantage since they are simple to use and cheap.

When trying to build and program a home automation solution on our own, we find that it is a very complicated job and that it requires knowledge in several areas. We would need everything from welding knowledge to server programming. This shows us that, although home automation is a promising field with a great future, it is a much better idea to turn to one of the many private companies that are dedicated to the installation of home automation systems rather than trying to assemble it yourself.

These companies ensure that a more secure and professional system remains. This is essential since exposing the home automation system to a security breach can be devastating due to the great damage that an attacker can do through, for example, opening the garage door or storing data from cameras or microphones.

# Índex

Abstrac .....	2
Índex.....	4
Introducció .....	5
Motivació .....	5
Objectius .....	5
Metodologia.....	6
Estructura.....	6
La Domòtica.....	7
Què és la domòtica?.....	7
Historia de la domòtica .....	8
El present de la domòtica .....	8
Quin és el futur de la domòtica? .....	9
La seguretat en el camp de la domòtica .....	10
La maqueta.....	12
El Cervell del sistema: l'ordinador.....	12
La interfície d'usuari: la aplicació .....	13
El microcontrolador: la placa Arduino .....	15
Taula de tots el components electrònics.....	19
Esquema elèctric.....	20
Els problemes que m'he trobat .....	20
Que fa la maqueta? .....	21
Conclusió .....	25
Webgrafia .....	27
Annex .....	28

# Introducció

## Motivació

He triat el tema de domòtica i Internet de les Coses perquè és un tema lligat als estudis posteriors que vull fer (que en aquest cas és Enginyeria informàtica) i perquè sempre m'ha interessat el tema d'automatització i programar ordinadors i màquines per tal d'ajudar als humans tant com sigui possible. Llavors aquesta oportunitat de treballar en un projecte de programació durant cert temps, encara que sigui a nivell molt bàsic em sembla una molt bona oportunitat per copsar si vaig ben encaminat als estudis posteriors que vull fer.

També l'he escollit perquè em sembla que una de les pròximes revolucions que podria haver-hi dins del camp de la informàtica que més afectarien els humans en la seva vida quotidiana podria ser la domòtica.

## Objectius

En aquest treball m'he marcat l'objectiu de desenvolupar i aplicar un sistema de domòtica en una maqueta i provar que desplegar-lo no és tan difícil i està a l'abast de tothom qui tingui unes mínimes ganes. La maqueta hauria de ser capaç de controlar les llums de diverses habitacions, llegir la temperatura, controlar amb les lectures de la temperatura una simulació d'aire condicionat i tot això des d'una aplicació programada. Part d'aquest objectiu radica en esbrinar que tan viable és per una persona amb coneixements comuns poder desplegar una solució domòtica a la seva llar.

El segon objectiu serà millorar el meu coneixement sobre la domòtica i totes les parts que estan implicades en aquesta branca de la informàtica.

El tercer objectiu és aprendre diverses tecnologies utilitzades en el camp de l'enginyeria de Software (a nivell molt bàsic, està clar) però per tenir una idea del que fa un Enginyer Informàtic. Aquestes tecnologies podrien ser per exemple Python (pel disseny del Servidor Web), Java (pel desenvolupament de l'Aplicació per Mòbils) i Arduino per controlar els LED i les parts físiques de la maqueta.

## Metodologia

Per tal de dur a terme aquest treball, portaré una metodologia dual, tant pràctica com teòrica. Al començament faré una recerca per tenir uns coneixements bàsics sobre la domòtica. Llavors començaré a treballar la maqueta aplicant el que he après en la part teòrica, sense deixar de seguir fent recerca teòrica.

Ja que continuaré fent recerca durant tota la duració del treball, la maqueta està subjecta a canvi constant, per tal de poder aplicar els coneixements adquirits.

## Estructura

El meu treball de recerca es dividirà en dues grans seccions. La primera serà el treball teòric on explicaré les bases de la domòtica i de l'Internet de les coses. Més endavant sobre les bases que hauré explicat explicaré tot el procés i el resultat final d'intentar construir i desenvolupar un sistema de domòtica i el procés de construir la maqueta des de zero. També adjuntaré tot el codi que hauré programat i intentaré comentar-ho per parts per tal d'aclarir com funciona.

# La Domòtica

## Què és la domòtica?

La domòtica és la disciplina de la informàtica que ens permet automatitzar processos en un habitatge per tal de fer la vida més fàcil per als seus habitants. Aquests processos poden ser molt variats, des de la intel·ligent gestió de la il·luminació, passant per la climatització, fins a l'augment de la seguretat, gràcies a poder connectar l'alarma domestica al sistema domòtic.

Totes aquestes solucions es despleguen mitjançant una sèrie de sensors i actuadors, tot governat pel sistema central de control. Els sensors són totes aquelles eines que utilitzem per recaptar informació de l'habitatge (sensor de temperatura, de llum, de gas, d'humitat, de l'obertura de portes, etc.). La informació recaptada per aquests sensors es dirigeix al sistema central. Aquest sistema aporta la lògica necessària per fer funcionar el sistema domòtic. Per exemple, el sensor de temperatura detecta que fan 28 °C al menjador, i el sistema central de control el que fa és ordenar-li a l'aire condicionat que s'encengui. Aquesta comparació entre la temperatura que fa i la temperatura a la qual s'ha d'engegar l'aire condicionat és la peça fonamental de la qual s'encarrega el sistema de control.

L'última part del sistema domòtic són els actuadors, aquests són els llums, l'aire condicionat, el motor per obrir la porta, una pantalla, altaveus, etc. Aquests són els encarregats d'actuar al medi i de solucionar pròpiament els problemes als que respon la domòtica.

La domòtica permet que mitjançant la unió d'aquests tres elements podem dotar "d'intel·ligència" a bombetes tradicionals, perquè no cal que les bombetes siguin intel·ligents, només cal que hi hagi un sistema de control ben programat al darrere.

La domòtica, per tant, té diversos avantatges. L'avantatge més clar seria la comoditat de, per exemple, no haver-se de preocupar d'encendre les llums o de apujar les persianes o bé de posar el sistema de reg perquè fa dies que no plou. Són aquestes excepcions les que doten a la domòtica d'unes possibilitats quasi infinites.

## Historia de la domòtica

La domòtica és una disciplina de la informàtica relativament jove i, per tant, no té una història molt clara, així i tot, és interessant conèixer els seus orígens per tal de descobrir cap a on va. El primer gran exemple de domòtica al món va córrer al càrrec de Nikola Tesla que l'any 1898 va crear el primer comandament a distància. Ell va utilitzar aquest comandament per dirigir un vaixell, això va suposar un gran canvi com que va ser la primera innovació dirigida a facilitar la vida domèstica. Avui en dia ja no consideràriem un comandament a distància com a un invent domòtic, però en aquella època sí que es considerava domòtic, perquè no hi havia hagut cap intent així per facilitar una tasca domestica.

Tot i aquest primer intent de domòtica, la primera vegada que veiem referència a la domòtica pròpiament és a la Fira Mundial de 1932. En aquesta fira es van presentar diversos conceptes d'electrodomèstics automatitzats e intel·ligents.

No obstant això, no va ser fins al 1966 que una empresa va intentar portar a un entorn comercial d'un producte domòtic. Va ser Westinghouse Electric amb el seu Echo IV. Aquest aparell pretenia emmagatzemar receptes, reenviar missatges, controlar la temperatura d'una casa, fer una llista de la compra i fins i tot apagar i encendre electrodomèstics. Potser el fet de ser tan avançat per a la seva època és el que va portar al desastre financer d'aquest producte, a causa del fet que era molt difícil i molt lent d'utilitzar.

I després d'això i fins a l'actualitat, veiem el gran boom dels telèfons intel·ligents. Des de que els humans ens vam acostumar a carregar el mòbil a tot arreu la domòtica ha pres un ritme vertiginós. Des de assistents amb intel·ligència artificial a la nostra butxaca, passant per detectors que ens avisen si detecten una alta presència de gas a l'ambient fins a persianes que s'obren detectant la llum de l'exterior.

## El present de la domòtica

La domòtica és un camp en constant canvi, per tant, és difícil parlar d'un "present" com a tal. El present de la domòtica passa inevitablement pels assistents de veu.



Aquests sistemes deuen el seu èxit a un hardware molt senzill (per tant, molt barat) però un software molt complex que els permet fer una gran varietat de tasques.

Aquests assistents poden fer tasques senzilles com posar música, posar recordatoris o alarmes o consultar la predicció meteorològica. Tot i poder fer tasques senzilles també poden fer tasques força complexes, tals com comprar certs productes en botigues en línia, consultar les últimes notícies basades en els interessos dels usuaris o fins i tot consultar les dades del compte bancari dels residents.

Aquests assistents artificials també deuen el seu èxit a la comoditat que té dir el que vols en veu alta respecte a, haver-te d'aixecar, obrir el mòbil o l'ordinador, i buscar-ho tu. Són els encarregats de gestionar tota la domòtica d'una casa, fan de nucli central. S'encarreguen de controlar totes les bombetes, altaveus, electrodomèstics, sensors de temperatura, televisors i climatitzadors mencionats anteriorment.

En l'actualitat, hi ha a Espanya 4,3 milions de cases que utilitzen assistents virtuals, això suposa un 22% de les llars. Això pot semblar una xifra força baixa, però tenint en compte que fins a l'octubre de 2018 els assistents no van arribar a Espanya és un creixement molt ràpid en un espai molt curt de temps.

## Quin és el futur de la domòtica?

Tots els camps de la informàtica pateixen un ràpid creixement, tant de les tecnologies utilitzades com dels objectius que es van assolint, però potser el camp de la domòtica és un dels camps on podem veure expectatives més ambiciosos i amb més possibilitats d'ajudar-nos en el nostre dia a dia.

El punt més interessant de la domòtica és la capacitat que té per agafar tecnologies que ja existeixen i ajuntar-les totes per tal que siguin accessibles de la manera més còmoda possible, i les expectatives de futur de la domòtica van totes encaminades en aquest camp.

Per exemple podem trobar voreres exteriors que s'escalfin quan detectin una temperatura de menys de zero graus per tal d'evitar que els usuaris rellisquin a causa

del gel; a la cuina podríem trobar frigorífics que detectin quan un producte estigui a punt de caducar-se o bé un sistema de compres recurrents, si hem de comprar cert producte regularment el sistema domòtic podrà comprar-ho sense la necessitat que un humà hi intervingui per res. Si ens abstraïem de l'àmbit domèstic i pensem en una comunitat de veïns, es podran instal·lar bústies intel·ligents que detectin quan estan plenes o bé punts de recollida de paquets amb el mateix fi.

Potser una de les aplicacions més futuristes que s'han presentat de la domòtica i la Internet de les Coses es el *metaverse* de Facebook. El passat 21 d'Octubre el CEO de Facebook va anunciar les seves intencions de crear un metavers en el qual es pugues interactuar amb diversos elements del món real a la vegada que això tenia repercussions al món virtual. A la imatge de l'esquerra podem veure a Mark Zuckerberg (CEO de Facebook) interactuant amb el seu avatar virtual a través d'un mirall intel·ligent. Moltes empreses ja s'han sumat a aquesta iniciativa. Altres escèptics, i amb arguments bastant sòlids, defensen que no s'hauria de donar tot aquest poder a Facebook, el mateix Facebook que ha tingut diversos sonats escàndols d'emmagatzematge massiu de dades de part dels usuaris. Facebook ha destinat una gran part dels seus recursos a desenvolupar tecnologies envers aquest futur metavers. Deixant de banda els pros i contres que pugui tenir, es sens dubte emocionant tecnològicament parlant, ja que integrar el món virtual al nivell en el que han ensenyat es difícil tècnicament parlant.

## La seguretat en el camp de la domòtica

Quan més es desenvolupa el camp de la domòtica més preocupacions aixeca la seguretat que la envolta. Ja que, si posem el nostre compte bancari, els nostres hàbits i diversos micròfons a casa nostra, tenir una escletxa de seguretat podria ser devastador. De fet, en els darrers anys la consciència col·lectiva sobre la seguretat informàtica ha crescut molt. Ja és una opinió

Aquestes preocupacions no son infundades ja que a l'Abril de 2017, conjuntament amb el llançament del Home Mini (un dels assistents més populars de Google) el periodista tècnic Artem Russakovskii va descobrir que l'assistent estava enregistrant i pujant tot el que es parlava a la seva casa, independentment de que el periodista

estigués o no parlant amb l'assistent. Russakovskii va reportar que en dos dies que havia tingut a l'assistent endollat aquest havia enregistrat una gran quantitat de dades molt gran sense que el periodista hagués activat l'assistent. Va reportar el problema a Google però només van dir que havia estat una errada de funcionament sense més, sense donar-li el pes que realment tenia.

Que les empreses enregistrin dades sense el nostre consentiment es un problema greu, pero encara pot ser pitjor que alguna persona aliena a la nostra casa aconsegueixi intervenir el nostre sistema domòtic. Al 2019 va ser molt sonat el cas de Ken Munro i Andre Tirney, dos investigadors de seguretat que a la DEFCON de les Vegas d'aquell mateix any van mostrar un *ransomware*<sup>1</sup> que “segrestava” un sistema domòtic, enregistrava tots els micròfons i descontrolava els termòstats i les llums fins que es pagava el “rescat” demanat, que tal com es veu a la fotografia, es un Bitcoin (56.920€ al canvi actual). Això es especialment preocupant perquè un virus d'ordinador es relativament fàcil de eliminar però un *ransomware* d'aquestes característiques pot ser molt més difícil d'erradicar. El problema afegit d'aquest tipus d'atacs es que es poden propagar amb molta més facilitat avui en dia. Fa deu anys les úniques coses que es podien veure afectades per un virus, si s'estenia per la xarxa local, eren un parell de mòbils i un ordinador, però si connectem a la xarxa local, bombetes, sensors de temperatura, altaveus, televisors, ordinadors i sensors de llum, les conseqüències poden ser devastadores.



Un altre problema que no es nou però es veu accentuat per l'aparició dels assistents virtuals i per l'augment de dispositius són les estafes a Internet, la domòtica pot suposar un augment d'aquestes ja que es molt més fàcil fer compres per Internet si l'únic que hem de fer es dir-ho en veu alta i l'assistent que està connectat amb el nostre compte bancari pot comprar-ho directament en lloc d'haver de comprar-ho nosaltres manualment.

---

<sup>1</sup> Ransomware: es un tipus de virus d'ordinador que pel general pren el control del dispositiu fins que l'usuari paga la quantitat demanada per l'atacant. En cas de no pagar-ho o s'esborren totes les dades del dispositiu o es deixa inservible.

# La maqueta

## El Cervell del sistema: l'ordinador

El cervell de tota la maqueta es el ordinador al que està connectat la maqueta. Aquest ordinador s'ocupa de rebre les peticions que mana el mòbil mitjançant l'aplicació allotjant un servidor. Llavors processa les peticions i les "tradueix" a ordres que llavors mana a la placa Arduino per tal d'encendre els diferents LED's o actualitzar la pantalla. La connexió entre Arduino i l'ordinador també serveix perquè Arduino li proporcioni a l'ordinador les diferents dades com per exemple: temperatura, humitat, llum, presència, etc.

El programa que he desenvolupat per ordinador està escrit en el llenguatge Python. Aquest es avui en dia el llenguatge de programació més utilitzat i el segon llenguatge amb més creixement a nivell global.

El programa allotja un servidor local que connecta el mòbil amb l'ordinador. El ordinador rep cadenes de un màxim de 18 caràcters que representen una acció que farà la maqueta. Algunes d'aquestes cadenes son *encendreLlum1*, *encendreAlarma*, *encendreAutoMode* o *apagarFogo*. Després de rebre aquestes cadenes de caràcters el programa busca en una llista les accions preprogramades. És a dir, si rep *encendreLlum1* busca en una llista la sèrie d'accions que ha de dur a terme si rep aquest missatge, que en aquest cas seria encendre el grup de LED's 1, que en aquest cas es el menjador.

Un altre funció que exerceix el programa del ordinador es registrar en una base de dades totes les accions que fa la maqueta, per tal de poder consultar-les. Utilitzant aquest mateix sistema el ordinador registra totes les temperatures i humitats per després poder graficar-les.

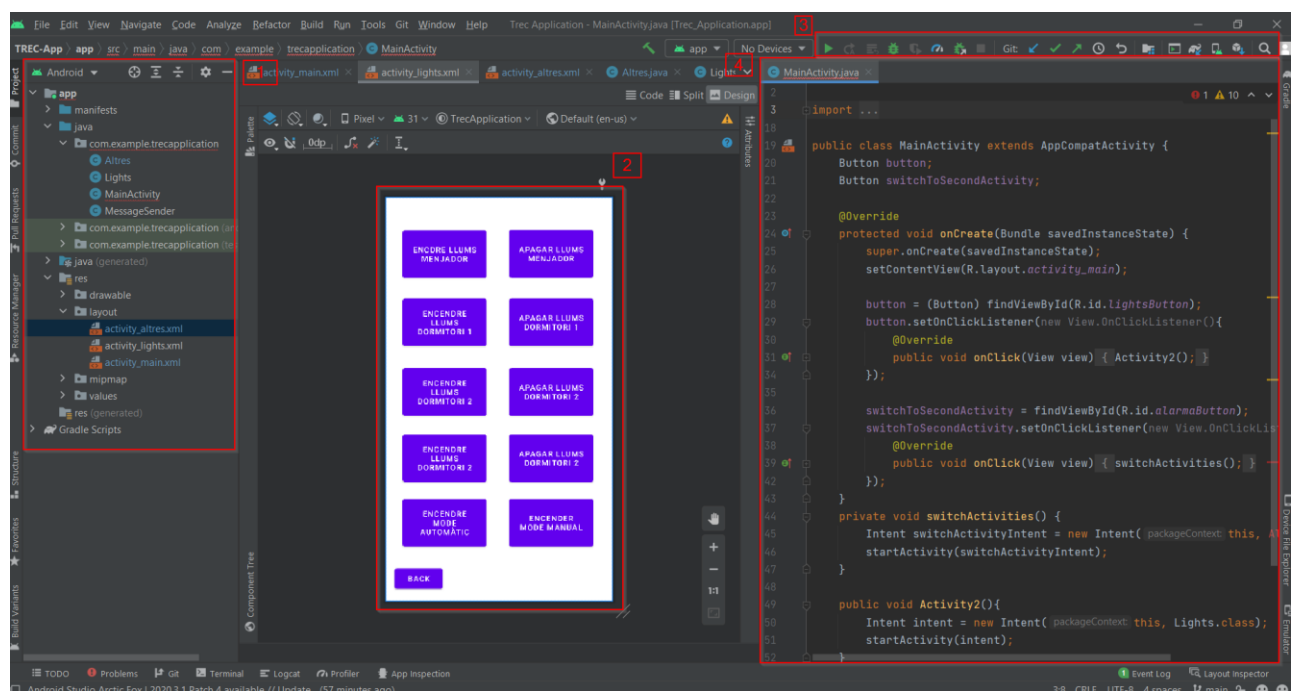
Aquesta arquitectura pot semblar un tant enrebesada, perquè ho es, pero després de provar altres mètodes com per exemple posar-li una placa inalàmbrica directament a la placa Arduino he optat per aquesta solució ja que es la més robusta. Amb les altres

solucions, quasi bé una de cada quatre accions es perdia, no arribava a la placa. Així que tot i tot el treball que ha suposat utilitzar una arquitectura molt més complexa crec que val la pena.

## La interfície d'usuari: la aplicació

La manera que tenen els usuaris d'interactuar amb la maqueta es a través d'una aplicació també desenvolupada des de zero. Aquesta aplicació està desenvolupada en Android Studio, que es la manera més estandarditzada de la indústria de desenvolupament d'aplicacions. Aquest *framework*<sup>2</sup> utilitza Java, un dels llenguatges històricament més important, per tal de gestionar totes les peticions i XML per organitzar tots els elements de la aplicació.

Aquest es Android Studio, el programa que he utilitzat. Android Studio organitza les aplicacions en “vistes” en el meu cas la aplicació tenia tres “vistes”: el menú principal, la que controla els llums i la que controla els altres (que en aquest cas es l'alarma i els fogons). Aquesta es la interfície d'usuari que Android Studio ofereix per desenvolupar les aplicacions.



<sup>2</sup> Conjunt d'eines de programació que s'utilitzen en molts projectes per tal de facilitar el treball als programadors.

Tal com es poden veure enumerades a la imatge la interfície té diverses parts:

1. El sistema de fitxers del projecte, aquí s'integren tots el paquets que utilitza la aplicació per funcionar, tots els arxius .XML i tots el arxius Java. Aquesta vista és molt útil perquè d'un cop d'ull es pot veure l'estructura de tot el projecte i totes les parts que l'integren.
2. Aquesta es la interfície d'usuari de la aplicació que volem desenvolupar. Mitjançant un sistema relativament senzill de arrossegat i deixar anar podem anar donant forma a la part estètica de la nostra aplicació. Aquesta part també es podria portar a terme picant codi .XML directament, però es realment ineficient i tenint en compte el còmode que es el sistema d'arrossegat no vaig veure-l'hi cap avantatge.
3. Aquesta es la barra de eines. Des d'aquesta barra podem compilar l'aplicació, instal·lar-la tant a un dispositiu físic com a un dispositiu virtual que tinguem configurat. Des d'aquí també podem pujar el nostre codi a un repositori de *GitHub*, a més a més de descarregar la copia del repositori per tenir la ultima versió, cosa que he trobat extremadament útil per tenir organitzat tots els canvis que anava fent.
4. Aquesta es la part on es programa tota la lògica de l'aplicació. El llenguatge que utilitza Android Studio es Java. És un llenguatge realment tosc quan el comparem amb Python per exemple. Té una quantitat desmesurada de claus (`{}`) i de punts i comes i estructures que necessiten seguir un orde i una forma molt concreta, mentre que Python no té gaires normes a seguir i es molt més fàcil de depurar un error.

La aplicació s'encarrega de disposar una sèrie de botons que són els que manen les cadenes de caràcters que després l'ordinador interpretarà. Aquestes senyals, com en aquest cas es una maqueta, no caldria encriptar-les no hi ha risc de que algú les intervingui i pugui actuar maliciosament. Tot i això he encriptat totes les ordres manades a través de la connexió inalàmbrica mitjançant una sistema anomenat *clau pública i clau privada*.

El sistema de clau pública i clau privada es un mètode molt útil i molt utilitzat al llarg i ample de la informàtica. Es especialment útil quan enviem recursos a través de

connexions que poden ser interceptades per possibles atacants. Aquest mètode de xifratge pertany a un tipus de sistemes anomenat sistemes de xifratge asimètric. Aquest sistema es basa en que el receptor del missatge té una clau pública i una clau privada. La clau pública pot ser coneguda per tothom, incloent als possibles atacants, aquesta clau només serveix per encriptar el missatge, una vegada tenim el missatge encriptat el transmetem com faríem amb qualsevol missatge desencriptat. Una vegada el receptor rep el missatge utilitza la clau privada (que només ell coneix) per tal de desencriptar aquest missatge. Els sistemes asimètrics son especialment útils en aquests casos ja que tota la seguretat recau en que la clau privada no sigui compromesa, i com el receptor en aquest cas es estàtic (no ha de transmetre dades encriptades, només rebre-les) hi ha molt poques possibilitats de que aquesta clau sigui compromesa. En aquest cas l'aplicació utilitza un sistema molt similar al que utilitza WhatsApp per xifrar els missatges entre els seus usuaris.

La aplicació és probablement la una de les parts més senzilles del projecte ja que realment no fa cap tasca complicada. Si ho mirem des de la base son només botons que transmeten un missatge a un servidor web (que es la part realment complicada). Tot i ser una part senzilla tècnicament ha de funcionar bé ja que serà la part amb la que l'usuari interactuarà per controlar la maqueta. Un mal funcionament d'aquesta aplicació significarà una mala experiència amb la maqueta.

## El microcontrolador: la placa Arduino

El últim pas per tal de utilitzar la maqueta es representar les ordres que enviem a la maqueta i d'això s'encarrega la placa Arduino. Una placa Arduino es un microcontrolador que té una sèrie de pins pels quals pot rebre i enviar corrent elèctrica. Això en la pràctica li permet exercir tasques com encendre LED's manant 5V a un dels seus pins digitals o llegir el voltatge que li arriba després de passar per una fotoresistència i detectar el nivell de llum. Hi ha diverses plaques Arduino, tot i la més comú ser l'Arduino Uno (veure a l'annex \$\$) he decidit utilitzar l'Arduino Mega 2560 ja que es superior en potencia i capacitat.



La placa Arduino rep des de l'ordinador a través de Python ordres senzilles, sense més context. Aquestes ordres poden ser: *encén el pin12*, *encén el pin34*, *espera 10 segons* o *llegeix el valor del pin12*. La placa només s'encarrega de transmetre aquestes dades, ja que de tota la lògica se'n ocupa Python.

Aquesta connexió per la qual es transmeten les dades és a través de cable (ja que es el més còmode i estable) però gracies a que es fa amb cable podem disposar d'una molta major seguretat ja que tenim la gaire bé certesa de que ningú pot intervenir la connexió abans de que arribi. Tot i tenir aquesta gaire bé certesa he decidit encriptar totes les ordres que rep l'Arduino i restringir la connexió que pot establir. És a dir, que si en un hipotètic cas algú intervingués la connexió, tot el que veuria serien lletres i números sense sentit i si tot i així aconseguís desxifrar-les, no podria manar les ordres que volgués ja que la connexió està restringida perquè només pugui ser establida per l'ordinador.

Hi ha molts tipus de plaques que podrien exercir la mateixa funció que l'Arduino tal com poden ser les de la gama MicroPython, o les Raspberry Pi, però m'he decantat per Arduino perquè ja tenia coneixements previs del llenguatge de programació que utilitza i perquè es una plataforma força senzilla i còmoda per treballar.

## Led's

Els Led's tenen dos terminals i estan connectats amb un dels terminals a un pin digital del que reben corrent i amb l'altre a terra. Depenent de la intensitat nominal del LED necessiten una resistència específica però en aquest cas per tots he utilitzat una resistència de 220 ohms, ja que la diferencia d'intensitat requerida no es la suficient com pel fet de no utilitzar la resistència perfecta sigui rellevant. Tot i no generar massa llum els led's em permeten simular tant les llums d'una casa com per exemple els fogons de la cuina.

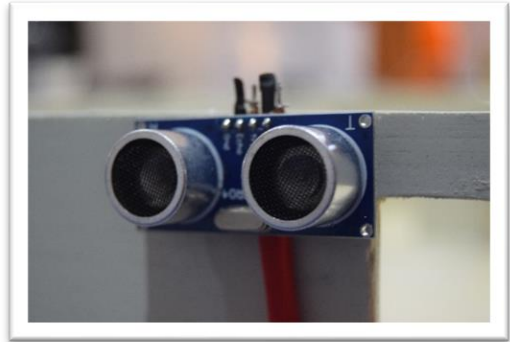


Aquests LED's consumeixen una quantitat ínfima de corrent (tant ínfima com 100 vegades menys que el que consumeix un a carregador de mòbil.) i això em permet fer la maqueta sense por de sobrepassar la intensitat màxima de la placa Arduino.



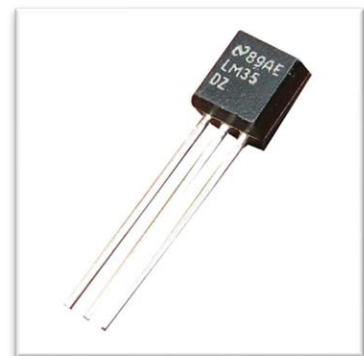
## Sensor d' Ultrasons

Els sensor d'ultrasons tenen dos terminals que són el disparador i el receptor d'ones ultrasòniques. Si disparem una ona des de el disparador i mesurem el temps que triga en tornar a detectar-la el receptor, i llavors dividim la velocitat de la ona entre el espai recorregut podem mesurar amb exactitud la distància a la que es troba el objecte. La exactitud amb la que puguem mesurar aquest espai dependrà exclusivament de la precisió amb la que puguem mesurar el temps entre les dues ones, que en aquest cas amb Arduino té un error  $\approx$  microsegons.



## Sensor de temperatura

El sensor de temperatura que he utilitzat pel projecte no es pròpiament un sensor de temperatura si no que es un transistor molt específic amb unes característiques que podem utilitzar per mesurar la temperatura. Els díodes realment son uns components electrònics amb tres terminals que al conduir-li una intensitat molt baixa per la base deixa passar més o menys corrent pels altres dos terminals. En aquest díode específic, el seu funcionament varia depenent la temperatura que hi hagi, mesurant aquest canvis podem detectar la temperatura amb bastanta facilitat. Utilitzant aquest mètode podem arribar a tenir una precisió de  $\approx 0.25$  °C sempre que les temperatures siguin majors a -25°C i menors a 120 °C, al sortir d'aquest rang de temperatures la precisió va disminuint exponencialment fins que les mesures son aleatòries.



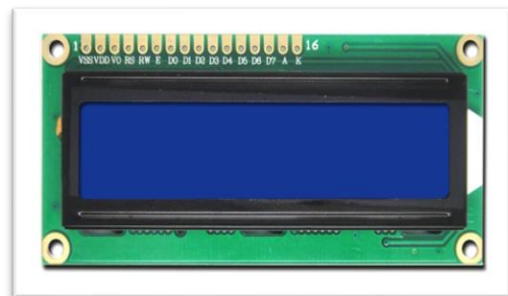
## Sensor de llum

El sensor de llum es basa en un principi molt senzill, te dos terminals separats per un material fotoconductor. Quan tapem la llum la resistència d'aquest material fotoconductor augmenta i deixa passar molta menys corrent i mesurant això podem veure quanta llum (tot i que realment mesurem quanta resistència hi ha, és a dir quanta foscor hi ha) tenim a l'ambient. Per la manera en la que funciona Arduino això es tradueix a que obtenim un valor entre 0 i 1023, sent 0 una quantitat exagerada de llum i 1023 completa foscor.



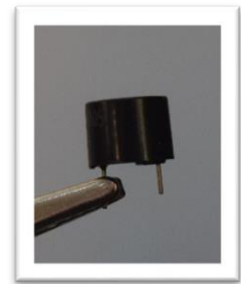
## Pantalla LCD

La pantalla es el component més complex electrònicament que té la maqueta. Té un total de 13 connexions. 4 estan dedicades a alimentació i preses de terra, 8 estan dedicades a la transmissió d'informació i l'últim està dedicat a la regulació del contrast de la pantalla mitjançant un potenciòmetre. La pantalla funciona bàsicament projectant llum blanca des de darrere de la pantalla i aplicant corrent elèctric a cada una de les cel·les de la pantalla aconseguim que s'encengui o no. Mitjançant els 8 pins digitals podem transmetre els caràcters que volem mostrar a la pantalla. La pantalla utilitza la codificació ASCII (*American Standard Code For Information Interchange*) que es una llista de 255 caràcters permesos. Aquesta codificació va ser desenvolupada a Estats Units i per tant no suporta ni accents ni símbols massa complicats, degut als pocs caràcters disponibles.



## Brunzidor Piezoelèctric

El bronzidor piezoelèctric té un funcionament molt senzill, al subministrar-li corrent elèctric entre els seus dos terminals la membrana que té dintre vibra i genera soroll. No pot generar notes ni canviar de freqüència però pels efectes pràctics de la maqueta es més que suficient i fa tant la funció de timbre com d'alarma.



## Botons

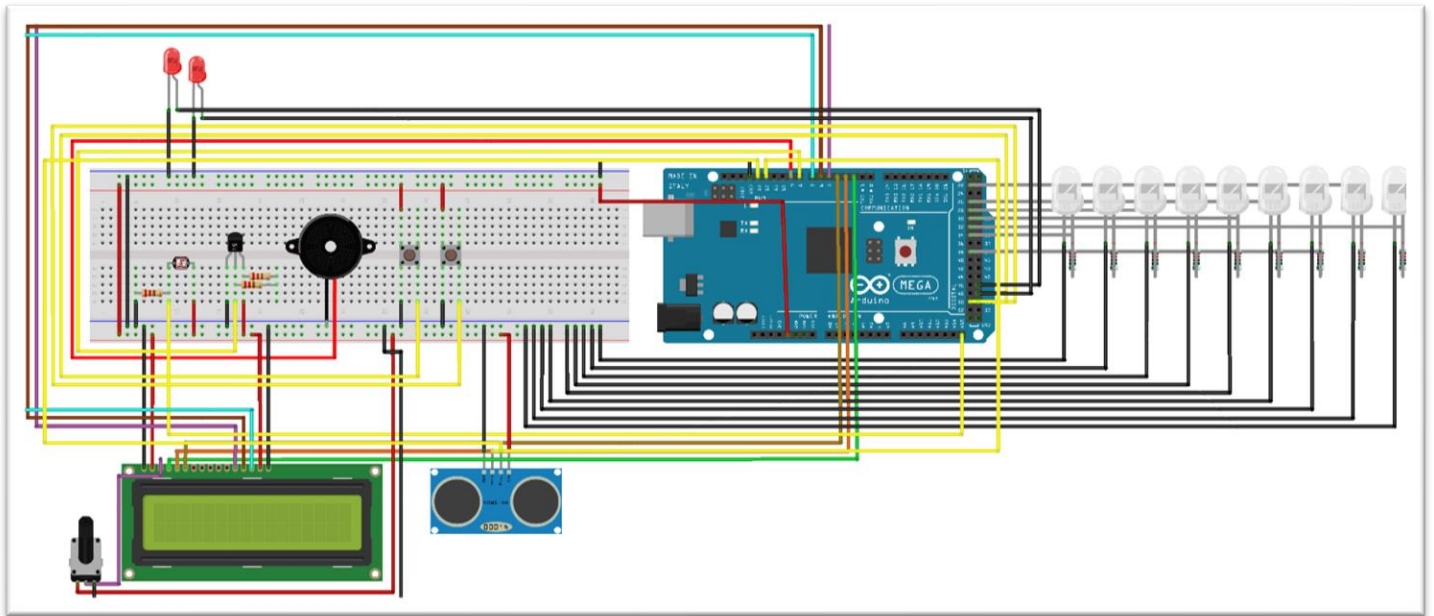
Els botons també tenen un funcionament molt senzill, quan pressionem el botó tenca el circuit entre els seus dos terminals i deixa passar l'electricitat, al detectar aquest pas de corrent podem detectar quan el botó està premut i quan no ho està. Hi han dos tipus de botons majoritàriament, de 2 pots i de 4. A la maqueta he escollit utilitzar els de 4 perquè son més senzills de utilitzar perquè porten per separat la connexió al pol negatiu i la connexió a l'entrada digital.



## Taula de tots el components electrònics

Nom	Quantitat	Ús
LED blancs	9	Donar llum.
LED vermells	2	Simular els fogons.
LDR (Fotoresistència)	1	Detectar el nivell de llum.
Pantalla LCD	1	Mostrar diverses dades.
Díode LM35	1	Detectar la temperatura.
Potenciòmetre	1	Regular el contrast de la pantalla.
Resistències (de diversos valors)	15	Evitar que es cremin els components.
Brunzidor Piezoelèctric	1	Timbre i alarma.
Sensor d'Ultrasons	1	Detectar presència.
Arduino Mega 2560	1	Microcontrolador.

## Esquema elèctric



## La maqueta

La maqueta està feta de planxes de fusta conglomerada tallada amb una serra de calar. He escollit aquest material perquè es relativament resistent, lleuger i perquè es un material fàcil de treballar i sobretot barat. El fet de triar aquesta material fa que fer forats per passar cables cap a sota de la maqueta sigui molt senzill ja que un trepant qualsevol té potencia més que de sobres per perforar-lo. El conglomerat al ser un material tant porós fa que sigui molt fàcil enganxar-li els components i les plaques. les Tot i això el seu principal desavantatge es que amb la mínima humitat planxes s'esbomben i que són un material no gaire resistent, però pel que fa a la maqueta son el millor material.

## Els problemes que m' he trobat

El primer problema que em vaig trobar fent la maqueta va ser a l'hora de tallar les fustes, que no acabaven de coincidir del tot. Per tal de solucionar-ho vaig haver de tornar a tallar dues parets tenint més en compte la precisió. L'altre problema que hem vaig trobar va ser al començar a connectar tots els components a la placa. Tenint en compte que cada component com a mínim te una connexió a la placa i una al pol negatiu, sense comptar per exemple la pantalla, que té 8 connexions, arribem a tenir un total aproximat de 60 connexions soldades. Tenint en compte que no tenia gens

d'idea de soldar al començar el projecte ha sigut un procés bastant complicat i de prova i error.

A l'hora de programar el servidor si que em vaig trobar amb altres problemes, sobretot amb la comunicació entre l'ordinador i Arduino. Per tal d'establir aquesta comunicació vaig utilitzar una llibreria anomenada *pyfirmata* que va estar desenvolupada fa 5 anys i ja no rep suport ni actualitzacions. Tot funcionava correctament, tot i que amb els seus problemes fins que vaig arribar a programar el sensor d'Ultrasons i la pantalla. La llibreria no tenia suport per cap d'aquests dos components. Tot i que vaig intentar fer les modificacions que calien a la llibreria per que acceptés aquests dos components no hi va haver manera. Al final vaig optar per reprogramar, seguint la mateixa estructura, el programa des de zero. Ha sigut definitivament difícil però el fet d'haver-ho reprogramat em permet no haver de dependre de la llibreria escrita per algú altre per haver de programar la maqueta.

## Que fa la maqueta?

La aplicació te dos menús, dins del primer menú anomenat *Llums* trobem totes les opcions per encendre i apagar les diferents llums de la casa. Les llums estan organitzades per habitacions (menjador, primer dormitori, segon dormitori i el lavabo) i tenim un últim botó que permet activar un mode automàtic de les llums. Aquest mode automàtic el que fa es detectar quan la llum es més baixa que un cert llindar encén les llums, això simula el que passaria al activar el mode automàtic en una casa real, que les llums només s'encendrien quan fos necessari, per tal d'estalviar energia.

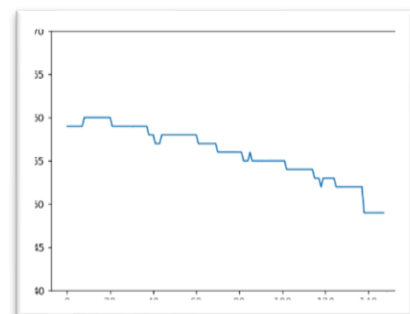
Al segon menú de l'aplicació anomenat *Altres* tenim quatre botons. Els dos primers ens permeten encendre i apagar els fogons mentre que els dos últims ens permeten encendre i apagar l'alarma de la casa. Aquesta alarma consisteix en que quan l'activem des de l'aplicació posa a funcionar el sensor d'ultrasons i quan detecti un objecte més a prop del normal (és a dir, moviment) salta la alarma. Aquesta alarma consisteix en el bronzidor piezoelèctric sonant i les llums encenent-se i apagant-se per cridar l'atenció. Fins que no es pressiona un botó amagat l'alarma no deixa de sonar i les llums no tornen a la normalitat. Aquest botó simula el que seria el teclat o

la combinació per desactivar la alarma, que no he posat per tal de no sobre complicar-ho innecessàriament.

Un altre sistema del que disposa la casa és l'enregistrament i gràfics de temperatura. A sota de la maqueta hi ha un sensor de temperatura que cada 1 segon enregistra un nou valor de temperatura i l'afegeix a una llista, cada 30 segons l'ordinador guarda aquesta llista en un fitxer d'Excel per tal de que sigui molt senzill fer un gràfic de la temperatura respecte al temps. Guarda el fitxer cada 30 segons perquè no es necessari actualitzar el fitxer cada segon, no ens aporta cap benefici i l'únic que provoca es que el programa de la maqueta tingui un funcionament molt més lent i tosc.

Els mateixos enregistraments de llum que pren la maqueta per decidir si encendre o no les llums quan té el mode automàtic encès també les guarda en una llista per poder graficar-les juntament amb les mesures de temperatura. Les actualitza amb el fitxer cada 30 segons, al mateix fitxer que les mesures de temperatura.

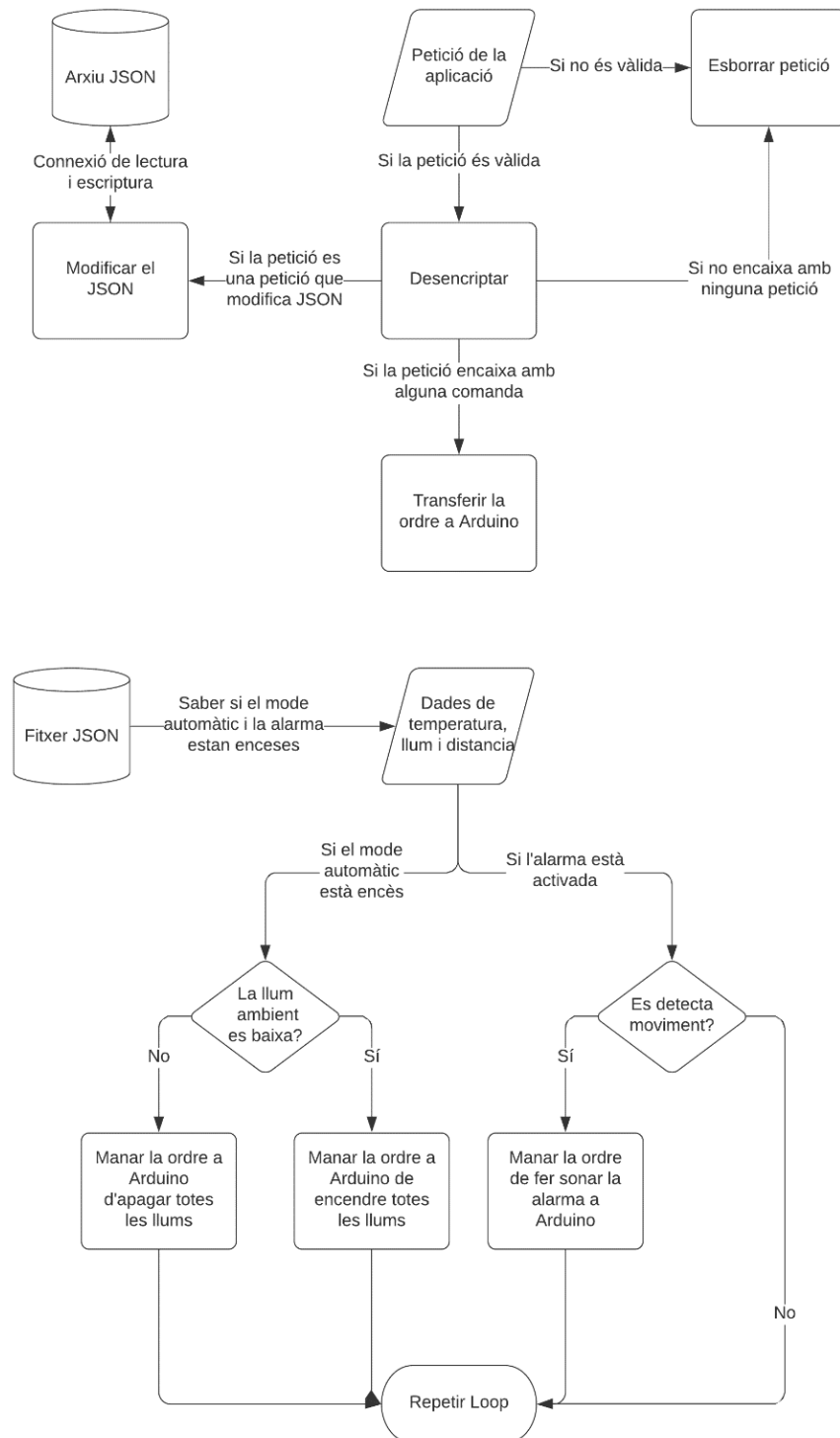
En aquesta figura podem veure la representació de les mesures de temperatura al llarg de mitja hora. Va ser registrada justament quan es va fent més fosc, i per tant podem veure com el nivell de llum va caient (des de el 59% fins al 49%). Les mesures s'estanquen en un mateix valor durant cert temps perquè el sensor de llum no té la precisió suficient com per detectar canvis tan petits al nivell de llum.



La maqueta també té una pantalla LCD on mostra diferent informació rellevant sobre la maqueta. Mostra en tot moment la hora i va alternant cada deu segons mostrant la temperatura i el nivell de llum en tant per cent.

L'últim sistema domòtic de la casa es el timbre. A la part davantera de la casa hi ha un botó que actua com a timbre que fa que el bronzidor piezoelèctric (el mateix que s'utilitza per a la alarma) soni fent un to de timbre.

## Diagrames de Flux de la maqueta







## Conclusió

La domòtica es un camp de la informàtica realment interessant, sent els seus principals avantatges l'estalvi energètic, la major comoditat i el poder integrar millor sistemes de la casa tot i que siguin d'àmbits diferents.

El canvi que la domòtica ha patit en els últims anys es aclaparador, el fet de poder demanar amb la nostra veu directament que les llums s'apaguin o bé que es faci certa compra a distància semblaria una idea completament de ciència ficció i ara gràcies a la domòtica es pot convertir en un eina quotidiana i que faciliti la vida a aquells qui la utilitzin.

Tot i que la domòtica es un camp en creixement i promet moltes coses, no tot es tan fàcil com sembla. Si els habitants d'una casa volen, des de zero, i sense tenir cap tipus de coneixement muntar ells mateixos una solució domòtica per a la seva casa, i si opten per muntar-ho ells tot des de zero, com es el que es pretenia demostrar amb la maqueta, es completament impràctic. Per començar, per tal d'aplicar el que s'ha fet a la maqueta al mon real cal saber d'electrònica, soldar, programació i tenir molta paciència. Relacionant-ho amb el primer objectiu que em vaig marcar abans de començar el treball i després d'haver passat mesos treballant-ho, sí, es possible desplegar una solució domòtica des de zero, però que sigui possible no implica que sigui senzill, perquè no ho es per res. Si abans del treball no hagués comptat amb els coneixements d'electrònica i de programació bàsics, hagués sigut completament impossible dur-lo a terme, o almenys extremadament difícil. Dit això, una instal·lació domòtica és força convenient (tot depenent de la situació de la llar, és clar) i després d'haver-me donat compte de que muntar-ho un mateix no es la idea més convenient vaig estar investigant i vaig trobar força empreses que es dediquen al muntatge de sistemes domòtics. Això ens treu l'inconvenient que suposa haver de muntar-ho un mateix, però també comporta perdre el control sobre que el que estàs fent es completament segur, així que té les seves avantatges i els seus inconvenients.

Respecte al segon objectiu, crec que després d'haver passat aquests mesos tant rodejat de programació i prenent les dinàmiques que du a terme un Enginyer de

Software crec que l'objectiu s'ha complert amb creixos. He pres pràctiques molt comuns en l'enginyeria tals com la fabricació de diversos prototips a més a més d'haver tingut que submergir-me de ple en diverses plataformes utilitzades a l'indústria de la programació com poden ser *GitHub* i *StackOverflow*.

# Webgrafia

consultor@ramonmillan.com, R. M.--. (n.d.). *Domótica. Edificios Inteligentes*. Domotica. Edificios inteligentes. Retrieved December 13, 2021, from <http://www.ramonmillan.com/libros/libroDomoticaEdificiosInteligentes.php>

*Domótica: Un Enfoque Sociotécnico* - dit.upm.es. (n.d.). Retrieved December 13, 2021, from [https://www.dit.upm.es/~fsaez/intl/libro\\_domotica.pdf](https://www.dit.upm.es/~fsaez/intl/libro_domotica.pdf)

FRANCISCO MOYA and JUAN CARLOS LÓPEZ, Moya, F., Mancha, U. of C.-L., & López, J. C. (n.d.). *Senda: An alternative to OSGI for large scale domotics*. World Scientific. Retrieved December 13, 2021, from [https://www.worldscientific.com/doi/abs/10.1142/9789812776730\\_0014](https://www.worldscientific.com/doi/abs/10.1142/9789812776730_0014)

Google. (n.d.). *Dom*. Google Libros. Retrieved December 13, 2021, from <https://books.google.es/books?hl=es&lr=&id=8ERFqWcdHAEC&oi=fnd&pg=PR3&q=domotica&ots=WRPXwWxMQQ&sig=HwDlrRHF4PDGWf9FAf1fqw7osJM#v=onepage&q=domotica&f=false>

Group, I. T. D. M. (2019, April 17). *El 10,7% de la población española utiliza asistentes virtuales de voz*. Al día | IT Reseller. Retrieved December 13, 2021, from <https://www.itreseller.es/al-dia/2019/04/el-107-de-la-poblacion-espanola-utiliza-asistentes-virtuales-de-voz>

Simonet, C., & Noyce, A. J. (2021). *Domotics, smart homes, and parkinson's disease*. Journal of Parkinson's disease. Retrieved December 13, 2021, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8385512/>

*Welcome to ESP8266 Arduino Core's documentation!*¶. Welcome to ESP8266 Arduino Core's documentation! - Tài liệu ESP8266 Arduino Core 2.4.0. (n.d.). Retrieved December 13, 2021, from <https://arduino.readthedocs.io/vi/stable/>

*Welcome to pySerial's documentation!*¶. Welcome to pySerial's documentation - pySerial 3.4 documentation. (n.d.). Retrieved December 13, 2021, from <https://pyserial.readthedocs.io/en/latest/>

→ *tipos de Sistemas domóticos para viviendas – pentadom*. PENTADOM Edificios Inteligentes. (2021, October 11). Retrieved December 13, 2021, from <https://pentadom.com/sistemas-domoticos-para-viviendas/>.

Russakovskii, A., & Artem Russakovskii (1396 Articles Published) . (2017, October 10). *Google is permanently nerfing all home minis because mine spied on everything I said 24/7 [update x2]*. Android Police. Retrieved December 14, 2021, from <https://www.androidpolice.com/2017/10/10/google-nerfing-home-minis-mine-spied-everything-said-247/>

# Annex

## 1.1 Programa d' Arduino

```
#include <LiquidCrystal.h> // Incloem totes les llibreries necessaries
#include <OneWire.h>
#include <DallasTemperature.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // Inicialitzem la pantalla LCD

const int pin1_0 = 34; // Definim tots els pins que utilitzarem
const int pin1_1 = 28;
const int pin1_2 = 27;
const int pin1_3 = 31;

const int pin2_0 = 30;
const int pin2_1 = 26;

const int pin3_0 = 36;
const int pin3_1 = 22;

const int pin4_0 = 32;

const int pinLDR = A15;

const int pinTimbre = 51;
const int pinBuzz = 9;
const int pinAlarma = 50;

const int EchoPin = 12; // Ultrasons
const int TriggerPin = 13;

const int oneWirePin = 8; // Diode pel detector de temperatura
OneWire oneWireBus(oneWirePin);
DallasTemperature sensor(&oneWireBus);

int tim = 0; // Temporitzador

const int ledRojo1 = 47;
const int ledRojo2 = 49;

void setup(){
  lcd.begin(16, 2); // Iniciem la pantalla LCD
  // Escribimos el Mensaje en el LCD.
  lcd.print("Temperatura: ");

  Serial.begin(9600); // Iniciem la comunicació Serie

  pinMode(pin1_0, OUTPUT);
  pinMode(pin1_1, OUTPUT);
  pinMode(pin1_2, OUTPUT);
```

```
pinMode(pin2_0, OUTPUT);
pinMode(pin2_1, OUTPUT);

pinMode(pin3_0, OUTPUT);
pinMode(pin3_1, OUTPUT);

pinMode(pin4_0, OUTPUT);

pinMode(pinLDR, INPUT);

pinMode(pinTimbre, INPUT);
pinMode(pinBuzz, OUTPUT);

pinMode(TriggerPin, OUTPUT);
pinMode(EchoPin, INPUT); // Definim tots els ports digitals com
entrades y sortides segons convingui

sensor.begin(); // Inicialitzem el sensor de temperatura
}

void loop(){
  if (Serial.available() > 0)
  {
    char option = Serial.read(); // Llegim els missatges que li manem a
    Arduino desde Python i prenem les accions que convinguin.

    if (option == '1')
    {
      digitalWrite(pin1_0, HIGH); // Per exemple aquesta opció seria la
      d'encendre les llums del menjador, per tant posem a HIGH els 4 pins que
      hem posat que son els LED del menjador.
      digitalWrite(pin1_1, HIGH);
      digitalWrite(pin1_2, HIGH);
      digitalWrite(pin1_3, HIGH);
    }

    else if (option == '2')
    {
      digitalWrite(pin1_0, LOW);
      digitalWrite(pin1_1, LOW);
      digitalWrite(pin1_2, LOW);
      digitalWrite(pin1_3, LOW);
    }

    else if (option == '3')
    {
      digitalWrite(pin2_0, HIGH);
      digitalWrite(pin2_1, HIGH);
    }
  }
}
```

```
}

else if (option == '4')
{
    digitalWrite(pin2_0, LOW);
    digitalWrite(pin2_1, LOW);
}

else if (option == '5')
{
    digitalWrite(pin3_0, HIGH);
    digitalWrite(pin3_1, HIGH);
}

else if (option == '6')
{
    digitalWrite(pin3_0, LOW);
    digitalWrite(pin3_1, LOW);
}

else if (option == '7')
{
    digitalWrite(pin4_0, HIGH);
}

else if (option == '8')
{
    digitalWrite(pin4_0, LOW);
}

else if (option == '9')
{
    digitalWrite(pin1_0, HIGH);
    digitalWrite(pin1_1, HIGH);
    digitalWrite(pin1_2, HIGH);
    digitalWrite(pin1_3, HIGH);
    digitalWrite(pin2_0, HIGH);
    digitalWrite(pin2_1, HIGH);
    digitalWrite(pin3_0, HIGH);
    digitalWrite(pin3_1, HIGH);
    digitalWrite(pin4_0, HIGH);
}

else if (option == '0')
{
    digitalWrite(pin1_0, LOW);
    digitalWrite(pin1_1, LOW);
    digitalWrite(pin1_2, LOW);
    digitalWrite(pin1_3, LOW);
```

```

    digitalWrite(pin2_0, LOW);
    digitalWrite(pin2_1, LOW);
    digitalWrite(pin3_0, LOW);
    digitalWrite(pin3_1, LOW);
    digitalWrite(pin4_0, LOW);
}

else if (option == 'a') // Aquesta opció es l'activació de l'alarma
{
    int alarma = 0;
    while (alarma == 0){ // Si l'alarma es dispara no es parerà
fins que es pulsi el botó per desactivarla
        analogWrite(pinBuzz, 155);
        delay(500);
        analogWrite(pinBuzz, 0);
        delay(50);
        analogWrite(pinBuzz, 155);
        delay(500);
        analogWrite(pinBuzz, 0);
        delay(50);
        alarma = digitalRead(pinAlarma);
    }
}

else if (option == 'b')
{
    digitalWrite(ledRojo1, HIGH);
    digitalWrite(ledRojo2, HIGH);
}

else if (option == 'c')
{
    digitalWrite(ledRojo1, LOW);
    digitalWrite(ledRojo2, LOW);
}
}

// VOID LOOP
int LDR = analogRead(pinLDR);
String stringLDR = String(LDR);

int cm = ping(TriiggerPin, EchoPin);
String stringCM = String(cm);

float temp = sensor.getTempCByIndex(0);
String stringTemp = String(temp);

sensor.requestTemperatures();

```

`Serial.println(stringLDR + '/' + stringTemp + '/' + stringCM);` // Aquesta part s'encarrega també mitjançant comunicació serie de manar-li a Python les dades del sensor de distancia, de temperatura i de llum.

`if (digitalRead(pinTimbre) == HIGH)` // Aquí ens encarreguem de detectar si algú està fent sonar el timbre i fer sonar el timbre.

```
{
  analogWrite(pinBuzz, 155);
  delay(500);
  analogWrite(pinBuzz, 0);
  delay(50);
  analogWrite(pinBuzz, 155);
  delay(500);
  analogWrite(pinBuzz, 0);
  delay(50);
}
```

`tim = (millis() / 10000)%10;` // Aquí posem les dades que volem a la pantalla LCD i les canviem cada 10 segons.

```
fecha = now();
lcd.clear();
lcd.setCursor(0, 0);
```

```
String stringHora = String(hour(fecha));
String stringMinuto = String(minute(fecha));
```

`lcd.print("Hora: " + stringHora + ":" + stringMinuto);` // A la pantalla també ensenyem quina hora és en aquell moment

```
if (tim % 2 != 0){
  lcd.setCursor(0, 1);
  lcd.print("Temp: " + stringTemp + " C");
}
```

```
if (tim % 2 == 0) {
  lcd.setCursor(0, 1);
  lcd.print("Llum: " + stringLDR + " %");
}
}
```

// EXTRA FUNCTIONS

```
int ping(int TriggerPin, int EchoPin) {
  long duration, distanceCm;
```

```
  digitalWrite(TriggerPin, LOW); //para generar un pulso limpio ponemos
a LOW 4us
  delayMicroseconds(4);
```



```

digitalWrite(TriiggerPin, HIGH); //generamos Trigger (disparo) de 10us
delayMicroseconds(10);
digitalWrite(TriiggerPin, LOW);

duration = pulseIn(EchoPin, HIGH); //medimos el tiempo entre pulsos,
en microsegundos

distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en
cm
return distanceCm;
}

```

## 1.2 Programa de Python

```

import serial, time, socket, jsonHandling, threading, pandas # Importem
totes les lliberies
# serial -> per establir la comunicació serie entre Arduino i Python
# time -> per poder esperar certs intervals de temps
# socket -> pel servidor web
# jsonHandling -> per modificar els arxius JSON, aquesta es una llibreria
escrita integrament per mi
# threading -> per poder fer tasques en paralel
# pandas -> per poder exportar les dades de temperatura i llum com a for-
mat .xlsx (Excel)

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Iniciem el
servidor web
sock.bind(('0.0.0.0', 420))
sock.listen(1)
connections = []

f = open('filekey.key', 'r') # Obrim el fitxer on s'emmagatzema la clau
de xifratge
code = f.readline() # I la guardem en una variable amb la que la llibre-
ria socket automaticament la desxifrarà
f.close()

puerto = 'COM3'
arduino = serial.Serial(puerto, 9600)
time.sleep(2) # Definim a quin port està conecat l'Arduino i esperem 2
segons per tal de que s'efectui la connexió.

```

```
def handler(c, a): # Aquesta es la funció que rep les peticions de la
    aplicació, les desxifra i les passa a Arduino.
    global connections

    while True:
        data = c.recv(1024) # Li diem a Python que com a molt accepte
        1024 bytes de petició (més que suficient)
        decodedData = data.decode()

        if decodedData == 'turnOn1': # Tota la llista de comands que
            volem que Python sigui capaç d'enviar-li a Arduino
                arduino.write(b'1')
                print('testing')

        if decodedData == 'turnOff1':
            arduino.write(b'2')

        if decodedData == 'turnOn2':
            arduino.write(b'3')

        if decodedData == 'turnOff2':
            arduino.write(b'4')

        if decodedData == 'turnOn3':
            arduino.write(b'5')

        if decodedData == 'turnOff3':
            arduino.write(b'6')

        if decodedData == 'turnOn4':
            arduino.write(b'7')

        if decodedData == 'turnOff4':
            arduino.write(b'8')

        if decodedData == 'onFogon':
            arduino.write(b'b')

        if decodedData == 'offFogon':
            arduino.write(b'c')

        if decodedData == 'turnOnAuto':
            jsonHandling.writeData('ServerCalls.json', 'autoMode', True)
            print('true')

        if decodedData == 'turnOffAuto':
            jsonHandling.writeData('ServerCalls.json', 'autoMode', False)
            print('false')
```

`def handleServer():` # Aquesta funció serveix per gestionar les peticions a més baix nivell, es a dir rep les peticions i per cada petició que rep crea un nou "fil" per tal de que les peticions es processin el més ràpid possible.

```
while True:
    c, a = sock.accept()
    cThread = threading.Thread(target=handler, args=(c, a))
    cThread.daemon = True
    cThread.start()
    connections.append(c)
    #print(connections)
```

`def getIterators():` # Aquesta funció s'encarrega de totes les coses que tenen a veure amb els iteradors. Això és la alarma, el LDR i per tant el mode automàtic i la temperatura.

```
df = pandas.DataFrame(columns = ["Temperatura", "Light Level"])
i = 0
```

```
while True:
```

```
    cleanStr = arduino.readline().decode('utf-8') # Rebem totes les
dades per connexió serie i les pasem a format utf-8.
    # print(cleanStr)
```

`try:` # Aquí utilitzem el mètode try perquè hi ha vegades que les primeres connexions amb Arduino no retornen totes les variables i amb l'estructura try/except podem veure aquest errors sense que detinguin el programa.

```
    interiorLDR, temperature, distanceAlarm = cleanStr.split("/")
```

```
    if jsonHandling.readData('serverCalls.json', 'autoMode') ==
1: # Aquest if comproba al fitxer .json anterior si el mode automàtic
està encès i en cas de que ho estigui comproba si la llum es superior a
700 i en aquest cas encen totes les llums. Si som completament corosos
s'hauria de dir que el que mesurem es realment el nivell de fosc i no
el nivell de llum, per tant si el LDR detecta més de 700/1023 de fosc
encen totes les llums.
```

```
    if interiorLDR != None:
        print(interiorLDR)
        if int(interiorLDR) > 700:
            arduino.write(b'9')

    else:
        arduino.write(b'0')
```

```

        if jsonHandling.readData('serverCalls.json', 'alarm') == 1: #
Aquest if fa el mateix que el del LDR per amb l'alarma, llavors si
detecta presència a menys de 20cm fa que soni la alarma.

```

```

        print(distanceAlarm)
        if int(distanceAlarm) < 20:
            arduino.write(b'a')
            jsonHandling.writeData('serverCalls.json', 'alarm',
0)

```

```

        df = df.append({"Temperatura":float(temperature), "Light
Level":int(interiorLDR)}, ignore_index=True)
        # print(df)

```

```

        i += 1 # Cada vegada que aquest loop fa una volta li sumem 1
a i i quan i es multiple de 30 (més o menys cada 30 segons), agafem tots
els valors de temperatura i llum i els guardem a excel.xlsx .

```

```

        if i % 10 == 0:
            df.to_excel('excel.xlsx')

```

```

        except Exception as e: # En cas de detectar un error l'imprimim
per tal de que sigui més facil d'identificar-lo.
            print(e)

```

```

threading.Thread(target = handleServer).start() # Encenem la funció que
s'encarrega del servidor.
threading.Thread(target = getIterators).start() # Encenem la funció que
s'encarrega dels iteradors.

```

## 1.2 Programa de l'aplicació Java

### 1.2.1 Sistema de enviament al servidor

```

package com.example.trecapplication;
import android.os.AsyncTask;

import java.io.DataOutputStream;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.nio.charset.StandardCharsets;

```

```
public class MessageSender extends AsyncTask<String, Void, Void>{
    Socket s;
    DataOutputStream dos;
    PrintWriter pw;

    @Override
    protected Void doInBackground(String... voids){
        String message = voids[0];

        try{
            s = new Socket("192.168.43.49", 420);
            pw = new PrintWriter(s.getOutputStream());
            pw.write(message);
            pw.flush();
            pw.close();
            s.close();
        }

        catch(IOException e){
            e.printStackTrace();
        }

        return null;
    }
}
```

## 1.2.2 Sistema de botons

```
package com.example.trecapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Lights extends AppCompatActivity {
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lights);
    }
}
```

```
        button = (Button) findViewById(R.id.backButton);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Activity1();
            }
        });
    }

    public void Activity1(){
        Intent intent = new Intent(this, MainActivity.class);
        startActivity(intent);
    }

    public void turnOn1(View v){
        MessageSender messageSender = new MessageSender();
        messageSender.execute("turnOn1");
    }

    public void turnOff1(View v){
        MessageSender messageSender = new MessageSender();
        messageSender.execute("turnOff1");
    }

    public void turnOn2(View v){
        MessageSender messageSender = new MessageSender();
        messageSender.execute("turnOn2");
    }

    public void turnOff2(View v){
        MessageSender messageSender = new MessageSender();
        messageSender.execute("turnOff2");
    }

    public void turnOn3(View v){
        MessageSender messageSender = new MessageSender();
        messageSender.execute("turnOn3");
    }

    public void turnOff3(View v){
        MessageSender messageSender = new MessageSender();
        messageSender.execute("turnOff3");
    }
}
```

```
public void turnOn4(View v){
    MessageSender messageSender = new MessageSender();
    messageSender.execute("turnOn4");
}

public void turnOff4(View v){
    MessageSender messageSender = new MessageSender();
    messageSender.execute("turnOff4");
}

public void turnOnAuto(View v){
    MessageSender messageSender = new MessageSender();
    messageSender.execute("turnOnAuto");
}

public void turnOffAuto(View v){
    MessageSender messageSender = new MessageSender();
    messageSender.execute("turnOffAuto");
}

}
```