# Introducció

## Descripció general del projecte.

La principal idea del projecte és construir una estació atmosfèrica amb components d’*Arduino* i gestionar les dades per visualitzar en un posterior moment.

*Aduino* és un mòdul semblant a un ordinador de baix cost i codi obert, perfecte per modificar i controlar a gust de l’usuari aparells i a la vegada sense consumir molt, pot funcionar amb una pila que trobem per casa. En ser un microordinador que consumeix poc, no pot fer molts càlculs importants com els processadors grans.

Per complir amb la funcionalitat del projecte, el microordinador incorpora sensors per cada dada que es vol controlar, tots els sensors els poden trobar per internet amb cost mínim i desenvolupats per *Arduino*.

Per tal de veure les dades hi haurà una manipulació, programada amb *Java* des d’un projecte *Maven* que està explicat en el següent l’apartat del projecte.

La manipulació també integra l’emmagatzematge de les dades si es vol i la visualització d’elles en un futur, en aquest projecte té dues opcions explicats en l’apartat següent del projecte.

## Objectius personal

Al principi del projecte estava la idea de treballar en l’àmbit informàtic relacionat amb l’electrònica, perquè l’origen del projecte és d’un curs de desenvolupament d’aplicacions multimèdia.

La raó de relacionar-la amb l’electrònica és perquè els components del grup els agrada, per un futur estendre les idees apreses en un grau relacionat amb la informàtica o electrònica.

## Objectiu tècnics

Utilitzar totes les teories apreses en el curs de desenvolupament d’aplicacions multimèdia.

Comunicació entre dispositius amb servidors.

Manipulació de les dades per emmagatzemar.

Visualització de les dades a l’usuari final.

Autoaprenentatge dels components electrònics i funcionalitat de l’electrònica a partir d’*Arduino* i la seva programació d’alt nivell per manipular dades de baix nivell com ara el voltatge, binari, senyals digitals i analògics.

El microordinador amb els sensors comunica amb el servidor *java* des de la *wifi* que controla les mostres rebudes de l’*Arduino*, en connectar amb el servidor demana el temps d’espera per enviar les dades al mateix servidor que es connecta. Quan arriba el temps envia les mostres al programa que rep les dades.

El servidor rep les mostres per calcular una mitjana, però només ho fa si estan dintre d’un rang error. Per enviar les dades del programa *Firebase* el projecte ha de ser *Maven*, per poder afegir les llibreries necessàries de *Google* i així connectar amb la base de dades.

A més de més, el servei ho complementa amb un suport d’una segona base de dades per emmagatzemar les dades permanentment cada cert temps, aquest temps normalment és d’un dia. La segona base de dades és SQL i local, el motiu d’utilitzar ho és perquè *Firebase* té un límit de capacitat de forma gratuïta.

## Eines utilitzades

Un sistema operatiu amb programes per desenvolupar aplicacions, com *Netbeans*, *Android Studio, Arduino* per crear el servidor i l’app que utilitzarà l’usuari final.

Un sistema operatiu amb els programes per desenvolupar aplicacions, com *Arduino, Netbeans, Android Studio*.

* *Arduino*, base per programar el microordinador.
* *NetBeans*, plataforma per construir aplicacions com servidor.
* *Android Studio*, pilar per crear una aplicació per dispositius mòbils.

Microordinador ESP8266, Centraleta que rebrà les dades dels sensors.

* DHT22, mòdul per captar dades de temperatura i humitat.
* BME80, aparell per captar la informació de la pressió.
* Sensor hall, destaca perquè la seva funcionalitat és activar-se quan hi passa un imant per sobre.
  + Mòdul que incorpora l’anemòmetre per calcular les dades de velocitat del vent.
  + Mòdul que incorpora el pluviòmetre per captar les mostres d’aigua caiguda per metres quadrats .

Servidor controlador *Raspberry Pi*, un microordinador una mica més poderós que l’ESP8366 i incorporarà el programa de servidor per controlar els possibles errors de les mostres i pujar-les al *Firebase*.

# Desenvolupament

## Anàlisis de requisits

Control de errors sobre les dades adquirides dels dispositius.

Control d’emmagatzemar les dades en un servidor.

Visualització d’una interfície pel usuari amb dispositius mòbils i televisors.

## Introducció de dades

Les dades són introduïdes per dispositius amb sensors, al principi del projecte només tindrà un dispositiu. Per complementar existeix una introducció de dades amb format de qüestionari per veure que medi utilitza el usuari per arribar al destí.

## Visualització de dades

Els usuaris poden veure les dades guardades en el servidor amb els dispositius.

## Altres

## Requisits funcionals

|  |  |
| --- | --- |
| Autor | Dispositiu |
| Descripció | Un dispositiu té sensors que envien dades a un servidor fen de controlador. |
| Guió | * Rebre el temps d’espera per part del servidor. * Gestionar els errors de connexió com Wifi i servidor. * Guardar les dades fins el moment determinat. * Enviar les dades al servidor/controlador. |

|  |  |
| --- | --- |
| Autor | Controlador |
| Descripció | Servidor que controla els dispositius i els possible errors que pot tenir al rebre les dades. |
| Guió | * Enviar el temps d’espera al dispositiu. * Espera rebre les dades dels sensors. * Calcula una mitjana si les dades estan dintre del rang d’error. * Enviar les dades a un servidor *Firebase* per emmagatzemar. * Guardar les dades de forma permanent en una base de dades local. |

|  |  |
| --- | --- |
| Autor | Usuari |
| Descripció | Usuari final que utilitza l’aplicació. |
| Guió | * Registrar-se com anònim en *Firebase* per poder utilitzar l’aplicació. * Gestionar un perfil senzill de nom, edat i sexe. * Visualització de les dades guardades en el servidor *Firebase*. * Respondré una enquesta de moviment. * Visualitzar l’enquesta en l’aplicació. |

## Diagrama de casos d’us

## Planificació especificació del casos d’us

|  |  |
| --- | --- |
| Nom: | Introducció de les dades al controlador. |
| Descripció: | És un dispositiu ESP8266 molt similar a *Arduino* amb mòduls de humitat, temperatura, pressió ... |
| Actors: | Dispositiu amb sensors incorporats. |
| Pre-condició: | Connectar l’aparell al corrent. |
| Post-condició: | Enviar les dades al controlador. |
| |  |  |  | | --- | --- | --- | | Flux Normal | Flux Alternatiu |  | | 1. Connecta amb la *Wifi*. 2. Connecta amb el controlador. 3. Envia l’opció de rebre el temps 4. Rep el valor d’espera per enviar les dades al controlador. 5. Bucle que espera el temps rebut del controlador. 6. Mira si encara esta connectat a la *Wifi*. 7. Mira si encara esta connectat al controlador. 8. Mira si el temps esperat es superior a la demanada. 9. Envia l’opció d’enviar dades. 10. Envia les dades al controlador. | * 1. Si no esta connectat, re intenta connectar amb la *Wifi.*   2. Entra en un bucle per veure si rep la acceptació.   3. Torna al punt cinc si es correcte.      1. Torna al punt 5.1 si encara no es connecta. | * 1. Si no esta connectat re intenta connectar amb el controlador.   2. Entra en un *loop* per veure si connecta.   3. Torna al punt sis si es connecta.      1. Torna al punt 6.1 si encara no es connecta. | | |

|  |  |
| --- | --- |
| Nom: | Introducció de les dades al servidor *Firebase* |
| Descripció: | És un servidor que fa de controlador pels controladors, escrit en un projecte *maven* amb *java*. |
| Actors: | Controlador |
| Pre-condició: | Execució de l’arxiu .java |
| Post-condició: | Enviar les dades al servidor *Firebase*. |
| |  |  | | --- | --- | | Flux Normal | Flux Alternatiu | | 1. Configura el servidor amb el número de port indicat. 2. Servidor rep l’odre que vol fer el dispositiu. 3. Rep l’opció d’enviar dades. 4. Emmagatzemar tres mostres en la memòria. 5. Mira si les mostres estan dintre del rang d’error. 6. Si ho està guarda els valors per fer una mitjana. 7. Puja la mitjana al servidor *Firebase*. | * 1. Rep l’opció rebre el temps d’espera.   2. Envia el valor.   3. Torna a esperar en el punt dos. | | |

|  |  |
| --- | --- |
| Nom: | Entrar a l’aplicació per veure les dades |
| Descripció: | Un usuari acaba de baixar l’aplicació i accedeix per visualitzar el programa. |
| Actors: | Usuari final |
| Pre-condició: | El usuari acaba d’instal·lar l’app. |
| Post-condició: | Surt de l’aplicació al terminar de veure les dades. |
| |  |  | | --- | --- | | Flux Normal | Flux Alternatiu | | 1. Inicia el programa. 2. Veu el logo de l’aplicació amb la col·laboració de les empreses. 3. Es registra al *Firebase* com usuari anònim. 4. Introdueix dades bàsics. 5. L’aplicació connecta amb la base de dades. 6. Extreu per dades per visualitzar. 7. L’usuari surt del programa. |  | | |

|  |  |
| --- | --- |
| Nom: | Entrar a l’aplicació per omplir un formulari |
| Descripció: | L’usuari que esta registrar prèviament, vol omplir un qüestionari de mobilitat. |
| Actors: | Usuari final. |
| Pre-condició: | Ja registrat en l’aplicació. |
| Post-condició: | Veure les dades introduïdes. |
| |  |  | | --- | --- | | Flux Normal | Flux Alternatiu | | 1. Inicia el programa. 2. Veu el logo de l’aplicació amb la col·laboració de les empreses. 3. Obre el formulari. 4. Omple el formulari. 5. Visualitza com s’envia el formulari en la pantalla del principi. |  | | |

## Diagrama de classes

## Proves

Proves amb *Arduino*, crear mini programes per veure la funcionalitat dels mòduls sensors.

Proves del servidor i *Arduino*, enviar, rebre les dades i contendre els possibles errors de dades falses.

Proves del servidor i l’aplicació, enviar, rebre les dades.

# Conclusió