Txell Vilamajó i Puixeu
Ana Fernández Tejero
PROCESSADORS DIGITALS

SISTEMA DE MONITOREIG I ALIMENTACIÓ DE MASCOTES

OBJECTIU

Volem dur a terme la implementació d'un sistema que permet facilitar el control de l'alimentació de mascotes domèstiques, a través d'una menjadora i d'un abeurador.

El sistema està coordinat a partir d'una pàgina web amb els següents serveis:

Botó de "Afegir menjar" i "Afegir aigua"

- Servomotor
- La pàgina web disposa de dues opcions:
  - Afegir aigua
  - Afegir menjar
- Primer s'haurà de verificar en quin estat es troben els recipients per tal de no sobrepassar la seva capacitat màxima.

Comprovar el nivell de menjar i aigua (capacitat)

- Sensor de pes
- Amb els sensors de pes podem saber amb quina capacitat es troben els recipients. Perquè el sistema ha de tenir establert quin és el pes màxim de cada recipient, és a dir, (quan està completament ple). Aquest pes s'anirà modificant a mesura que la mascota menja/beu.

  Abans d'afegir aigua o menjar el sistema comprovarà que els recipients no es trobin en el seu màxim. En cas que sigui així, dispensarà menjar/aigua durant un període de 5 segons. En canvi, si els recipients estan en la seva capacitat màxima, no es dispensarà aigua ni menjar.

Per exemple, la nostra menjadora té una capacitat de 300 g. Prendrem aquesta mesura com el màxim possible. D'aquesta manera quan l'usuari vulgui afegir menjar al recipient, i per tant faci ús del botó d'afegir menjar que es troba en la pàgina web. S'haurà de comprovar en quin estat es troba el recipient i en funció d'això el sistema actuarà tal com s'ha explicat prèviament. Anàlogament per afegir aigua.

Programar horaris → Permet programar horaris de disposició d'aliment o aigua. Això és útil per poder assegurar-se que a una certa hora del dia els animals disposen d'un cert nivell de menjar i aigua.

Per exemple, programem que es disposi menjar a les 12:30, es guarda aquest horari i, mitjançant un bucle, es va comparant l'hora actual amb la configurada. D'aquesta manera, quan es detecti que les dues hores son iguals s'obrirà el mecanisme per disposar el menjar dins la menjadora.

Estadístiques → Permet a l'usuari poder veure un resum dels nivells d'alimentació de la seva mascota en els últims 15 o 30 dies. També permet veure el menjar i l'aigua consumit diariament.

Per exemple, si el primer dia tenim la menjadora plena al 100%, és a dir, tenim 300 g de pinso dins de la menjadora i al final del dia aquest pes ha disminuït fins a 250g, el programa guarda que el dia 1 la mascota ha menjat 50g de pinso. El mateix fa la resta de dies.

A més a més, s'ha de tenir en compte que, si el 2n dia el nivell de menjar ha passat de 250g a 230g i després l'usuari omple la menjadora fins a 300 g i la mascota torna a buidar-la fins a 260g al final del dia s'ha de tenir en compte que l'animal ha menjat 20g + 40g = 60g.

A part de totes les aplicacions de la web, hem de tenir en compte que tant la menjadora com l'abeurador disposen d'un sensor de moviment que permet controlar que la menjadora i l'abeurador no es reomplen quan la mascota hi està present (així no es tria aigua ni menjar a la cara de la mascota)

PREUS I MATERIALS

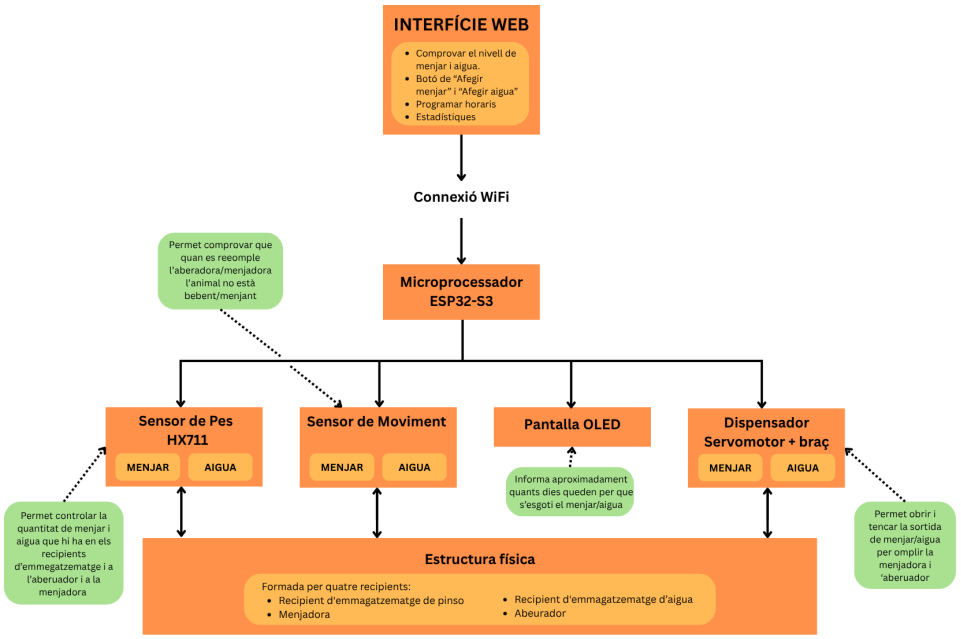
A continuació podeu trobar una taula amb tots els materials utilitzats i el preu corresponent.

MATERIAL	QUANTITAT	PREU
Dispensador de menjar		
Sensor de moviment	x 1	1,20 €
Servomotor	x 1	2 €
Placa HX711	x2	1,75*2=3,5€
Sensor de pes	x 2	0,86 €*2=1,72 €
PREU TOTAL MENJADORA = 8,42 €		
Dispensador d'aigua		
Sensor de moviment	x 1	1,20 €
Servomotor	x 1	2 €
Placa HX711	x2	1,75*2=3,5€
Sensor de pes	x 2	0,86 €*2=1,72 €
PREU TOTAL ABEURADOR = 8,42 €		
Dispositiu global		
Display	x1	2,90€
Protoboard	x1	2,45 €
Cables	x3 (hi ha 3 tipus de cable)	1,60*3= 4,80€
PREU TOTAL DEL PROJECTE = 26,99 €		

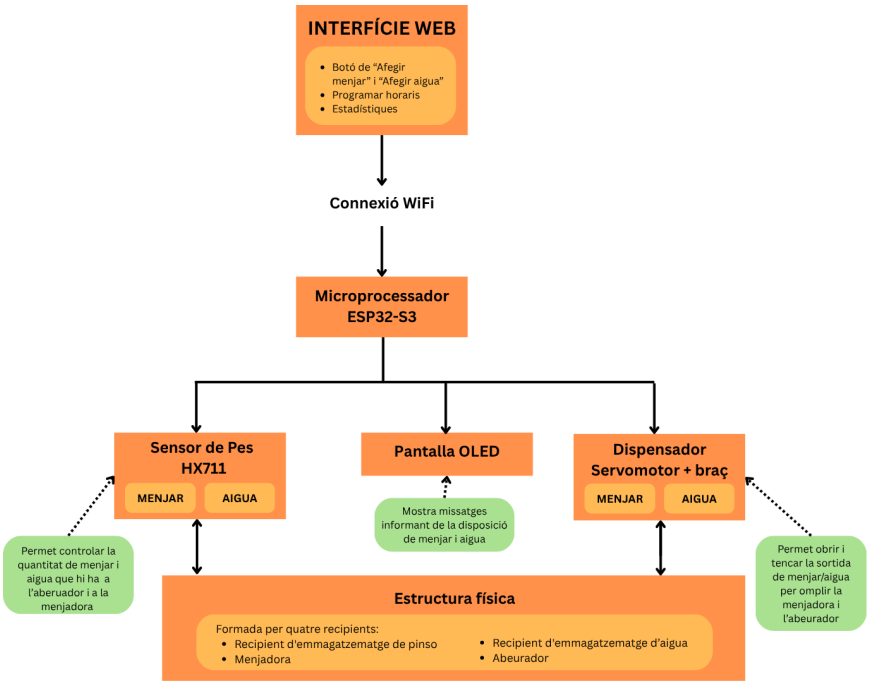
OBS: Podria ser que els gastos d'enviament fessin incrementar una mica el preu final.

DIAGRAMA DE BLOCS:

Inicialment volíem implementar un sistema amb dos sensors de moviment per poder detectar la presència de mascotes i quatre sensors de pes que permeten controlar el nivell de menjar i aigua de la menjadora, l'abeurador i els dos recipients d'emmagatzematge. Així doncs, el diagrama de blocs inicial era el següent:



Un cop vam intentar implementar el sistema però, vam veure que els sensors de moviment eren massa sensibles i detectaven sempre moviment, per tant vam decidir eliminar-los del sistema, ja que impedien que realitzés la seva funció correctement.



ESQUEMA DEL MUNTATGE

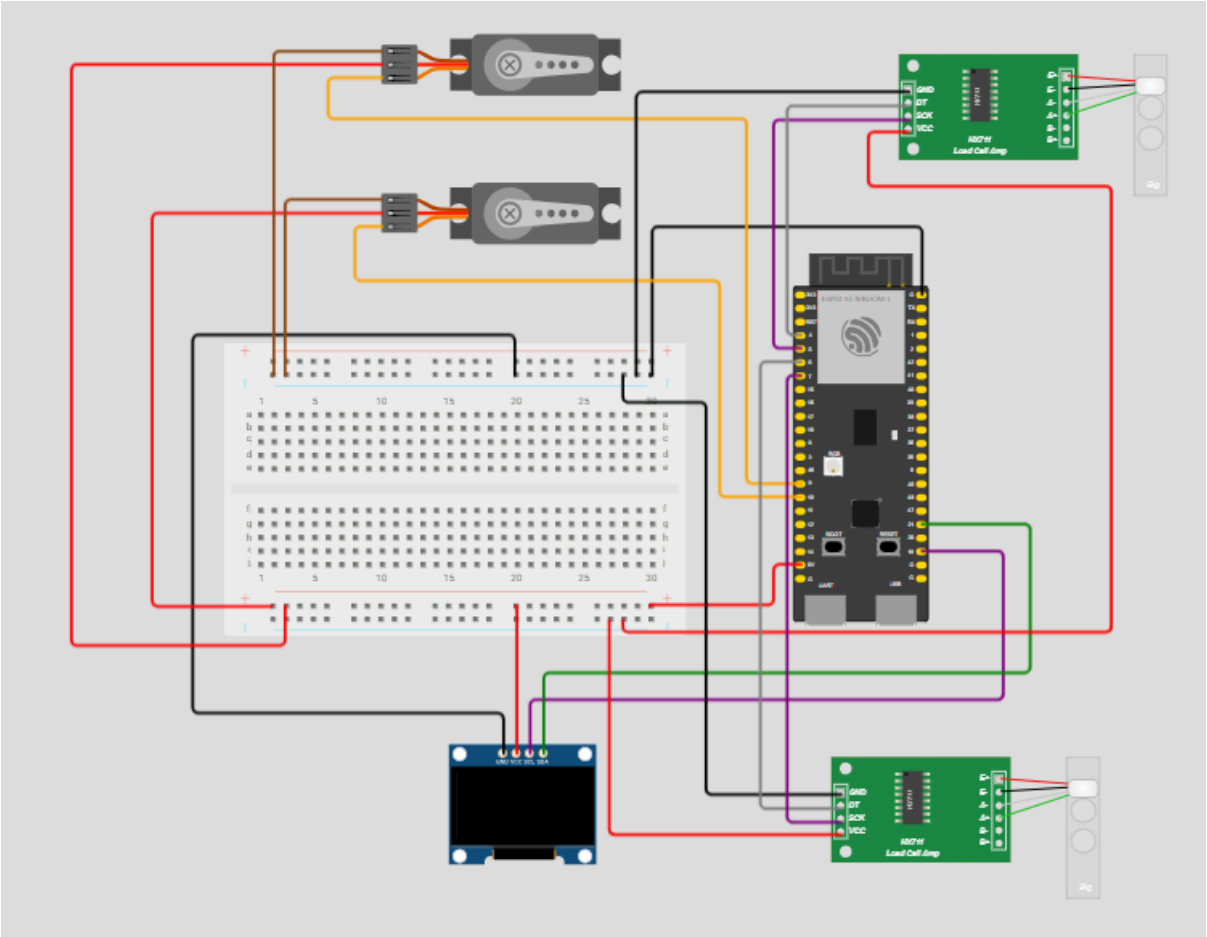
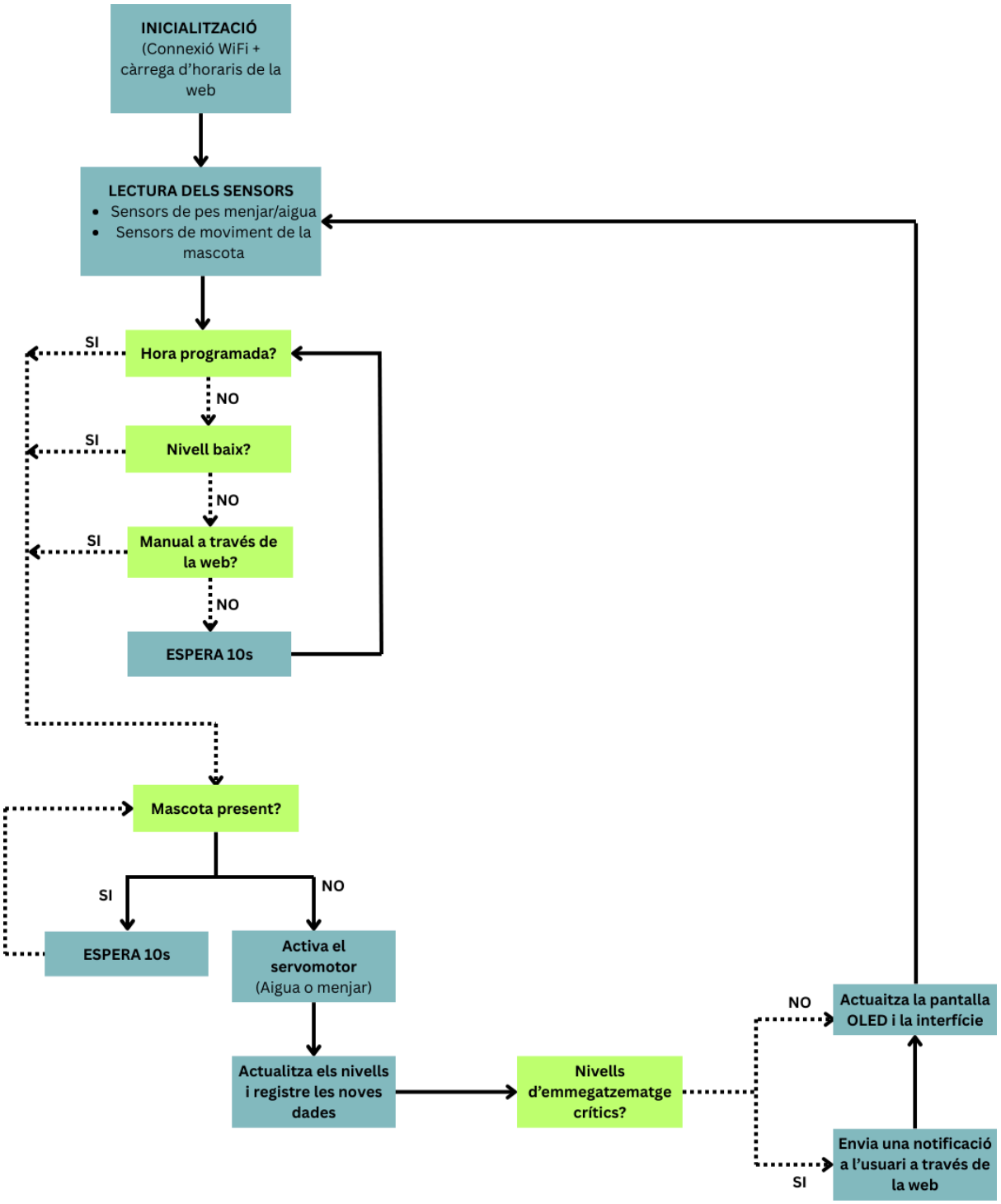
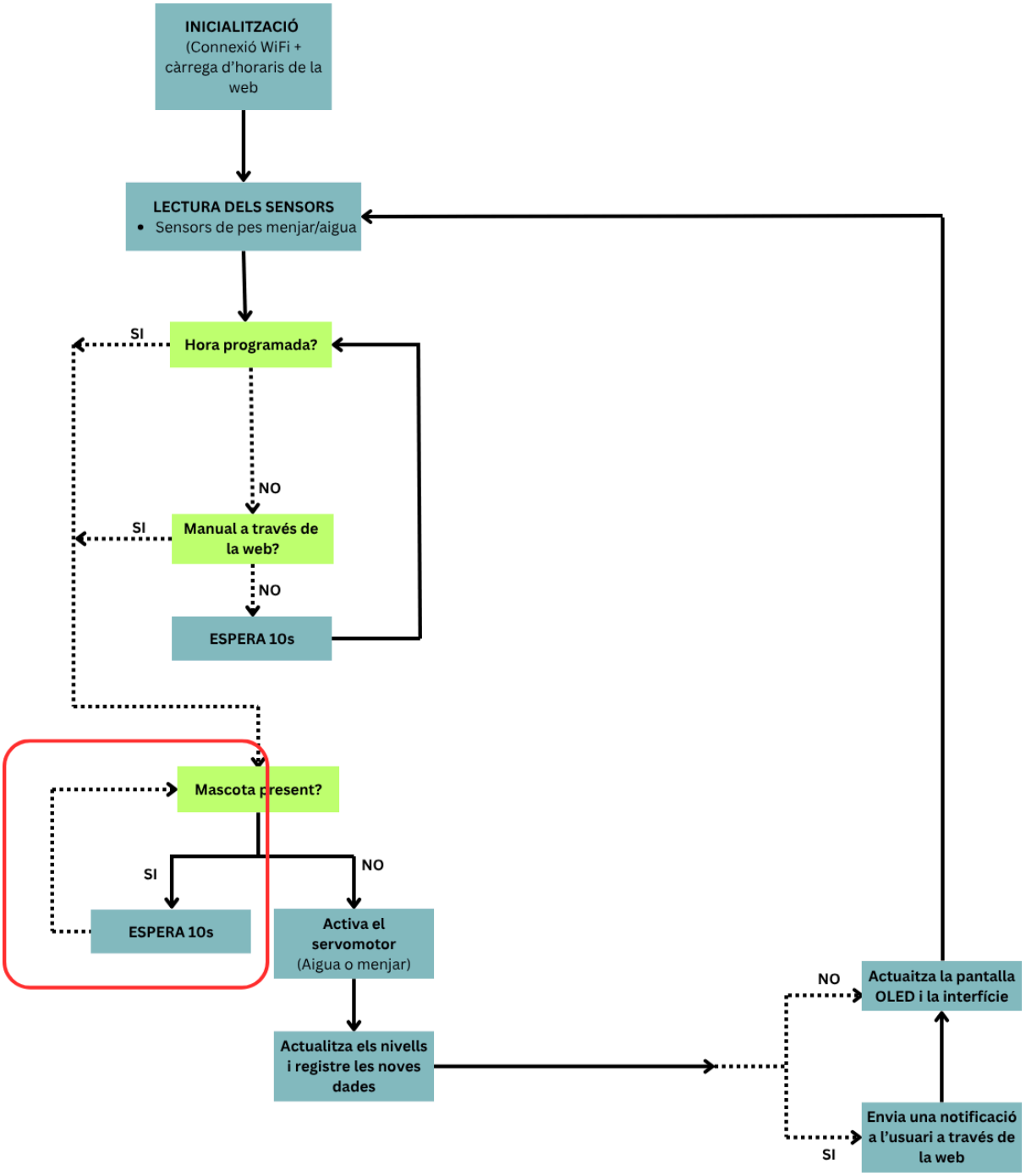


DIAGRAMA DE FLUX

Inicialment volíem implementar el següent diagrama de blocs, però degut a les complicacions que hem tingut amb els sensors de moviment i els sensors de pes ha hagut de patir algunes modificacions.



Així doncs el diagrama utilitzat és el següent:



DESENVOLUPAMENT DEL PROJECTE

Per poder començar el projecte crearem una classe, com les que hem estudiat a l'assignatura de estructura de dades i orientació a objectes, per cada component del sistema. Això ens permetrà definir el comportament de cada component del sistema.

> Una **classe** és un **model o plantilla** que defineix com serà un objecte: les **dades (atributs)** que conté i les **accions (mètodes)** que pot fer.

Així doncs, definierm una classe per cada mòdul del nostre programa (sensor, pantalla, servomotor...), ja que cada un té un comportament independent depenent de la seva funcionalitat.

Fer una classe per cada component ens permet:

- Separar les responsabilitats
- Reutilitzar funcions → Podem crear dos objectes de la mateixa classe per controlar el menjar o l'aigua
- Facilita el manteniment

Així doncs crearem una classe pels següents components:

- PesSensor → Classe on defenirem els mètodes que ens permetran definir el comportament del Sensor de Pes. Els mètodes que volem implementar son els següents:
  - iniciar_sensor_pes → Funció que inicia el sensor i el calibra.
  - Comprovar_nivell → Definirem el nivell de capacitat màxima (ex: 300g) i a partir d'aquest, prenent el valor de mesura del sensor de pes, retornar el % de menjar que hi ha a la menjadora/aberuadora.
  - llegir_pes → Ens retorna la mesura que pren el sensor de pes.
  - %_pes → Funció que a partir d'un percentatge que li arriba com a paràmetre (d'un dels botons de afegir menjar fins a x%), troba l'equivalent en pes d'aquest %.

- Dispensador
  - iniciar_servo → Funció que permet iniciar el servomotor.
  - Obrir → Funció que, quan es crida, fa que el servomotor obri la sortida de menjar/aigua.
  - Tancar→ Funció que, quan es crida, fa que el servomotor tanqui la sortida de menjar/aigua.

- OledPantalla
    - iniciar_pantalla ➜ Mètode que parmet iniciar la pantalla oled, és a dir, la inicia i mostra un petit missatge de benvinguda.
    - mostrar_missatge ➜ Mètode que permet mostrar per la pantalla OLED un text que li entra com a paràmetre des del main.

- Horaris
    - es_hora ➜ Funció que compara l'hora d'entrada a través de la web amb l'hora actual i retorna si és l'hora de dispensar menjar o no (boleà).

IDEA ORIGINAL:
Per exemple, **quan l'usuari clic el botó d'afegir menjar fins al 50%** el main principal crida al mètode %_pes que retorna el pes total que ha d'assolir la menjadora, a continuació, es crea un bucle que s'executa mentres mascota_present=*false*, de manera que ens assegurem d'omplir la menjadora quan la mascota no està menjant. Dins del bucle es crida al mètode Obrir de la classe Dispensador. Mentres aquest mètode s'està duent a terme es va cridant en bucle a la funció llegir_pes de la classe PesSensor i, quan s'assoleix el pes desitjat, és a dir, quan s'assoleix el pes que ha retornat la funció %_pes de la classe PesSensor, es crida la funció Tancar de la classe Dispensador.

Si la primera condició no s'ha complert, és a dir, mascota_present=*true*, haurem d'esperar 10s i tornar a comprovar la presència de la mascota.

IDEA DESENVOLUPADA:
Per exemple, **quan l'usuari clic el botó d'afegir menja** el main principal crida al mètode Obrir mentres comprova que no se superi la capacitat màxima establerta (300g). Si no se supera la capacitat maxima, espera 5s i a continuació crida al mètode Tancar. En el cas de detectar que recipient està completament ple, es crida el mètode Tancar immediatament sense esperar els 5 segons. A més a més s'actualitzen les estadístiques guardant a la memòria l'acció realitzada.

Per tant, necessitem crear una web interactiva que permeti que, si l'usuari selecciona l'opció d'afegir menjar, s'activin els mecanismes necessaris per dur a terme l'acció desitjada. Aquesta web serà creada a partir del nostre microprocessador ESP32-S3 que, després de crear un punt wifi ens proporcionarà l'adreça IP de la web. Per poder accedir-hi se'ns demanarà el nom d'usuari i contrasenya que hem definit en el codi. En el nostre cas:

- Usuari ➜ admin
- Contrasenya ➜ 1234

Un cop creades les 4 classes necessàries hem pogut ajuntar-les en el main, creant un objecte del tipus necessari per cada component:

- 2 servomotors ➜ 2 objectes de la classe Dispensador
- 1 pantalla ➜ 1 objecte de la classe PantallaLCD
- 2 sensors de pes ➜ 2 objectes de la classe PesSensor
- 1 horari ➜ 1 objecte de la classe horaris.

A partir d'aquests objectes i dels seus mètodes corresponents definits en el seu corresponent .cpp, hem pogut implementar totes les funcions en el main de manera més senzilla i fàcil de comprendre.

També hem necessitat que l'ESP32-S3 es connectés a internet per tal de poder obtenir l'hora actual i comparar-la amb la configurada als horaris.

POSSIBLES MILLORES

Ens hauría agrada poder implementar de manera correcta els sensors de pes per tal de poder controlar en tot moment la capacitat de menjar/aigua disposada a cada recipient. D'aquesta manera també hauríem pogut controlar la quantitat de menjar/aigua que hi ha dins dels recipients d'emmagatzematge i poder avisar a través de la pantalla quan s'ha d'omplir aquests recipients. Això no ha estat possible ja que creiem que vam comprar uns sensors de pes defectuosos que no prenen mesures de manera exacte i que tenen connexions molt febles.

També ens hauria agradat implementar els sensors de moviment que no hem pogut utilitzar ja que eren massa sensibles i detectaven sempre la presència de mascotes, fet que impedía que el sistema complet funciones correctament.