# SISTEM DE REGLARE AUTOMATA PENTRU IRIGAREA DE PRECIZIE A PLANTELOR

# Documentul de proiectare

Cuprins

[1. Introducere 3](#_Toc163925174)

[1.1 Scopul documentului 3](#_Toc163925175)

[2. Prezentare generală și abordări de proiectare 4](#_Toc163925176)

[2.1 Prezentare generală 4](#_Toc163925177)

[3. Arhitectura Sistemului și Proiectarea Arhitecturii 5](#_Toc163925178)

[3.1 Vedere logică 6](#_Toc163925179)

[3.2 Arhitectură hardware 7](#_Toc163925180)

[3.3 Arhitectură software 7](#_Toc163925181)

[3.4 Arhitectura informațiilor 8](#_Toc163925182)

[3.5 Arhitectura de comunicații interne 8](#_Toc163925183)

[3.6 Diagrama de arhitectură a sistemului 9](#_Toc163925184)

[4. Proiectarea sistemului 10](#_Toc163925185)

[4.1 Proiectarea bazei de date 10](#_Toc163925186)

[4.2 Conversii de date 11](#_Toc163925187)

[4.3 Interfețe utilizator 11](#_Toc163925188)

[5. Scenarii de utilizare 14](#_Toc163925189)

[6. Proiectare de detaliu 15](#_Toc163925190)

[6.1 Proiectare hardware de detaliu 15](#_Toc163925191)

[6.2 Proiectare software de deatliu 16](#_Toc163925192)

[6.3 Proiectare de detaliu pentru performanța sistemului 16](#_Toc163925193)

[6.4 Proiectare detaliată a comunicațiilor interne (între componente) 17](#_Toc163925194)

## Introducere

### Scopul documentului

Prin intermediul acestui document, dorim să oferim un cadru clar și comprehensiv pentru dezvoltarea și implementarea sistemului de irigare automată. Acesta va conține informații detaliate despre arhitectura sistemului, includerea componentelor și modul în care acestea interacționează între ele. De asemenea, va descrie deciziile de proiectare luate în procesul de dezvoltare, justificările acestora și modalitatea în care acestea susțin obiectivele generale ale proiectului.

Scopul acestui document este de a servi drept sursă principală de referință pentru echipa de dezvoltare și alte părți interesate implicate în proiect. Va oferi o imagine de ansamblu asupra modului în care sistemul va fi proiectat, implementat și integrat în mediul său operațional. În plus, va furniza informații esențiale pentru testarea și validarea sistemului, precum și pentru întreținerea și gestionarea sa pe termen lung.

Documentul SDD este destinat în primul rând managerului de proiect, echipa de dezvoltare și alte părți interesate implicate în proiect, precum și celor care vor utiliza sistemul în final. Prin furnizarea unei descrieri complete și detaliate a arhitecturii și designului sistemului, acest document facilitează înțelegerea și colaborarea între diferitele echipe implicate în proiect, contribuind astfel la succesul general al implementării sistemului de irigare automată.

## Prezentare generală și abordări de proiectare

### Prezentare generală

În această secțiune, vom oferi o prezentare generală a contextului și abordării de bază a proiectării sistemului nostru de irigare automată. Sistemul nostru are ca scop automatizarea procesului de irigare a culturilor agricole, contribuind la optimizarea utilizării resurselor și creșterea randamentului recoltelor.

Arhitectura sistemului constă într-o combinație între hardware și software. Pe partea de hardware, vom utiliza microcontrolere, cum ar fi Arduino Mega și ESP8266, pentru a colecta datele de la senzori și a controla dispozitivele de irigare. Aceste dispozitive vor include valve, pompe și sisteme de distribuție a apei în diferite zone ale serei.

Pe partea de software, vom dezvolta o aplicație mobilă folosită pentru monitorizarea și controlul sistemului de irigare. Această aplicație va interacționa cu serverul nostru backend, care va gestiona datele colectate de la senzori și va coordona operațiunile de irigare în funcție de parametrii predefiniți și de cerințele utilizatorului.

Obiectivele noastre de proiectare includ crearea unei interfețe intuitive și ușor de utilizat pentru utilizatorii finali, asigurarea unei performanțe și fiabilități ridicate ale sistemului și optimizarea consumului de resurse, cum ar fi apa și energie. De asemenea, ne propunem să implementăm o arhitectură modulară și scalabilă, care să permită extinderea și adaptarea sistemului în funcție de nevoile și cerințele viitoare.

## Arhitectura Sistemului și Proiectarea Arhitecturii

Arhitectura sistemului este concepută pentru a gestiona colectarea datelor de mediu și controlul sistemului de irigație într-un mod eficient și modular. Sistemul este dezmembrat în mai multe componente principale, fiecare având roluri și responsabilități distincte.

Primul subsistem este dedicat colectării datelor de mediu, inclusiv temperatură și umiditate, folosind senzori specifici. Aceste date sunt apoi transmise către un microcontroler Arduino, care servește ca nucleu al sistemului. Arduino prelucrează și interpretează aceste date și decide când să activeze pompa de irigație și electrovalvele, în funcție de setările predefinite și de logica fuzzy implementată.

Al doilea subsistem este responsabil pentru comunicarea între aplicația mobilă a utilizatorului și sistemul de irigație. Aplicația mobilă permite utilizatorului să monitorizeze starea mediului și să controleze sistemul de irigație de la distanță. Comunicarea între aplicația mobilă și sistemul de irigație este realizată prin intermediul unui API REST, care permite transferul de date între dispozitive.

Arhitectura este proiectată pentru a fi modulară și ușor de extins. Componentele sunt separate și interconectate printr-un API bine definit, permițând adăugarea ulterioară a funcționalităților suplimentare fără a afecta funcționarea existentă a sistemului. De exemplu, adăugarea unor noi tipuri de senzori sau extinderea logicii fuzzy pentru alte culturi agricole poate fi realizată fără a modifica în mod semnificativ restul sistemului.

Pentru implementarea arhitecturii, s-au folosit tipare de proiectare precum arhitectura bazată pe microservicii pentru a izola și decupla componentele și modelul client-server pentru a facilita comunicarea între aplicația mobilă și backend-ul sistemului. Alegerea acestor tipare a fost justificată de nevoia de scalabilitate, flexibilitate și ușurința în dezvoltarea și întreținerea sistemului.

### Vedere logică

O imagine care conține captură de ecran, text, diagramă, pătrat

Descriere generată automat

Fig. 1 – Diagrama de process

### Arhitectură hardware

Sistemul este alcătuit dintr-o varietate de componente hardware esențiale, fiecare având rolul său specific în funcționarea și controlul sistemului. Senzorul DHT11 este utilizat pentru măsurarea temperaturii și umidității în aer, iar senzorul de umiditate a solului este folosit pentru monitorizarea umidității terenului. Senzorul de nivel de apă este utilizat pentru detectarea nivelului apei în rezervorul de irigare, iar pompa de apă este responsabilă pentru pomparea apei către plante. Electrovalvele sunt utilizate pentru controlul fluxului de apă către diferite sectoare ale solului în funcție de necesități.

Arduino Mega 2560 servește drept placa de dezvoltare principală, gestionând și coordonând toate aceste componente hardware. Acesta primește datele de la senzori, ia decizii bazate pe aceste date și controlează pompa și electrovalvele în funcție de cerințele de irigare.

Modulul ESP8266 este utilizat pentru comunicații WiFi, permițând sistemului să comunice cu alte dispozitive sau rețele externe, inclusiv aplicația mobilă pentru controlul și monitorizarea sistemului de irigare de la distanță.

Pentru a asigura alimentarea necesară funcționării componentelor, se folosește o sursă de alimentare 12V, iar tranzistoarele NPN sunt utilizate pentru amplificare și controlul curentului către diferite componente.

Sistemul este conceput pentru a fi modular și ușor de configurat, cu componente interconectate prin fire jumper și montate pe breadboard-uri pentru testare și modificare rapidă. Această arhitectură permite o integrare flexibilă a componentelor și o adaptare ușoară la cerințele specifice ale fiecărui proiect de irigare.

### Arhitectură software

Sistemul software este compus dintr-o varietate de elemente logice și instrumente care permit dezvoltarea, gestionarea și implementarea funcționalităților sistemului hardware. Printre acestea se numără IDE-urile specifice pentru dezvoltarea platformelor hardware, precum Arduino IDE, care oferă un mediu de dezvoltare dedicat programării și testării de programe pentru plăcile Arduino. IDE-urile precum PyCharm sunt esențiale pentru dezvoltarea în limbajul de programare Python, oferind unelte puternice de editare, depanare și gestionare a proiectelor.

Android Studio este un alt element important, reprezentând mediul oficial de dezvoltare pentru aplicații Android. Acesta integrează un set de instrumente puternice pentru dezvoltarea, testarea și depanarea aplicațiilor mobile, inclusiv un emulator integrat și suport pentru gestionarea proiectelor prin intermediul sistemelor de control al versiunilor.

Django și Flutter sunt framework-uri de dezvoltare a aplicațiilor web, respectiv mobile și web, care oferă un set bogat de funcționalități și unelte pentru dezvoltarea rapidă și eficientă a aplicațiilor. Django este bazat pe limbajul de programare Python și oferă un model robust de dezvoltare MVC, împreună cu o interfață de administrare automată și suport pentru ORM. Flutter utilizează limbajul de programare Dart și un model de programare reactivă pentru dezvoltarea aplicațiilor mobile și web cu o singură bază de cod.

Aceste tehnologii sunt esențiale pentru dezvoltarea și implementarea sistemelor hardware, oferind un mediu de lucru eficient și puternic pentru dezvoltatorii de software. Ele facilitează dezvoltarea rapidă, gestionarea proiectelor și asigurarea calității și performanței aplicațiilor implementate pe sistemele hardware. Integrarea acestor tehnologii în arhitectura sistemului hardware permite o interacțiune eficientă între componentele hardware și software, contribuind la funcționarea și performanța generală a sistemului.

### Arhitectura informațiilor

În cadrul proiectului, vor fi stocate informații referitoare la fiecare dintre cele patru celule ale terenului meu agricol. Aceste informații vor consta în identificatorul unic al fiecărei celule și în datele de temperatură și umiditate măsurate în fiecare celulă.

Identificatorul unic al celulei va fi un număr sau un cod specific, care va permite sistemului să distingă între cele patru zone ale terenului meu agricol.

Datele de temperatură și umiditate vor fi măsurate de senzori plasați în fiecare celulă a terenului. Aceste date vor fi înregistrate și stocate în sistem pentru fiecare celulă în parte.

Modalitatea de colectare a acestor date va fi automată, prin intermediul senzorilor, iar actualizările vor fi făcute în timp real în baza de date a sistemului.

Astfel, datele stocate în sistem vor consta în identificatorul unic al fiecărei celule și în valorile de temperatură și umiditate asociate fiecăreia dintre cele patru zone ale terenului agricol. Aceste informații sunt esențiale pentru monitorizarea și gestionarea corespunzătoare a condițiilor de mediu din fiecare celulă în parte, facilitând luarea deciziilor în ceea ce privește irigarea și îngrijirea culturilor agricole.

### Arhitectura de comunicații interne

Pentru sistemul meu, rețeaua de comunicații este esențială pentru colectarea și transmiterea datelor provenite de la senzori și pentru controlul dispozitivelor conectate. Implementăm o arhitectură de rețea bazată pe o rețea locală (LAN) pentru comunicația internă între dispozitivele sistemului și o rețea de zona largă (WAN) pentru accesul la internet și pentru comunicarea externă.

În cadrul rețelei LAN, vom avea următoarele componente și conexiuni:

* Senzori: Acestea sunt dispozitivele care colectează datele de temperatură și umiditate din fiecare celulă a terenului agricol. Senzorii vor fi conectați la un microcontroler sau la un sistem înglobat care să prelucreze datele și să le transmită în rețea.
* Arduino Mega 2560: Este placa de dezvoltare utilizată pentru controlul și monitorizarea sistemului. Arduino va fi conectat la senzori pentru a prelua datele și la alte dispozitive pentru a controla pompele și electrovalvele.
* Modul ESP8266 (ESP 8260): Acest modul este utilizat pentru conectarea sistemului la rețea și pentru transmiterea datelor către un server sau către alte dispozitive externe prin WiFi.

În ceea ce privește rețeaua WAN, avem următoarele componente și conexiuni:

* Server: Dacă este cazul, serverul poate fi utilizat pentru stocarea și procesarea datelor colectate de sistem. Serverul poate fi local sau poate fi găzduit în cloud.

### Diagrama de arhitectură a sistemului

O imagine care conține text, captură de ecran, Font, proiectare

Descriere generată automat

Fig. 2 – Diagrama de transmisie a datelor

## Proiectarea sistemului

### Proiectarea bazei de date

În cadrul proiectului nostru, utilizăm Django pentru a gestiona baza de date asociată sistemului nostru. Un aspect esențial al acestui proces este fișierul model, care definește structura entităților noastre și care servește ca șablon pentru crearea automată a tabelelor în baza de date.

Fișierul model reprezintă o componentă centrală a proiectului Django și definește clase Python care reprezintă modelele noastre de date. În cazul nostru, avem o clasă Cell care descrie structura unei celule din sistem.

Structura tabelei este determinată de câmpurile definite în această clasă. În exemplul nostru, avem două câmpuri: humidity și temperature, ambele de tip float. Acestea reprezintă informații relevante pentru celulele din sistem, cum ar fi umiditatea și temperatura.

Un aspect important este că Django gestionează crearea automată a tabelelor în baza de date pe baza definirii claselor modele în fișierul model. La pornirea aplicației Django, acesta verifică structura definită în modele și, dacă este necesar, creează sau actualizează tabelele în baza de date conform acestora.

Acest proces automat de creare a tabelelor elimină nevoia de a scrie manual instrucțiuni SQL pentru crearea tabelelor și face ca dezvoltarea și întreținerea bazei de date să fie mai ușoare și mai eficiente.

În concluzie, structura tabelelor este definită în fișierul model folosind clase Python, iar Django se ocupă de crearea automată a acestora în baza de date. Acest proces abstractizează detaliile tehnice și facilitează gestionarea datelor în cadrul proiectului nostru Django.

#### Obiecte de date și structuri de date rezultante

În ceea ce privește structura tabelei asociate clasei noastre Cell, aceasta este determinată de câmpurile definite în fișierul model. În cazul nostru, avem două câmpuri definite pentru tabela noastră:

Câmpul humidity: Acesta reprezintă umiditatea înregistrată într-o anumită celulă și este de tip float. Acest câmp este opțional (având atributul blank=True) și poate accepta valori nule. De asemenea, este supus unor reguli de validare definite prin validatorii MinValueValidator(0) și MaxValueValidator(100), care asigură că umiditatea se află între 0 și 100 (reprezentând procente).

Câmpul temperature: Acest câmp reprezintă temperatura înregistrată într-o anumită celulă și este, de asemenea, de tip float. Similar cu câmpul humidity, acesta este și el opțional și poate accepta valori nule.

Aceste două câmpuri definesc informațiile esențiale pe care le stocăm în tabela noastră. Prin intermediul acestora, putem înregistra și monitoriza umiditatea și temperatura în fiecare celulă a sistemului nostru. Utilizarea tipului float pentru aceste câmpuri permite stocarea unor valori precise pentru umiditate și temperatură, iar definirea lor ca opționale oferă flexibilitate în cazul în care nu avem date disponibile pentru aceste parametri într-un anumit moment.

În concluzie, câmpurile humidity și temperature sunt esențiale pentru tabela asociată clasei Cell, deoarece definesc informațiile de interes pentru monitorizarea și gestionarea celulelor din sistemul nostru. Definirea acestora în fișierul model permite Django să creeze automat tabela corespunzătoare în baza de date și să gestioneze eficient datele asociate.

### Conversii de date

În cadrul proiectului Django, conversiile de date sunt gestionate în principal prin intermediul serializerelor, care sunt utilizate pentru a serializa și deserializa obiecte Python în și din formate de date standard, cum ar fi JSON. Aceste conversii sunt esențiale pentru a permite comunicația între diferitele componente ale aplicației și pentru a facilita transferul datelor între client și server.

Serializerul din Django este o componentă cheie care ajută la conversia datelor într-un format care poate fi transmis printr-o rețea sau stocat într-un mediu persistent. În cazul proiectului tău, acesta este utilizat pentru a serializa datele din obiectele Python în format JSON, astfel încât să poată fi trimise către server sau utilizate în alte părți ale aplicației.

Procesul de serializare implică transformarea datelor dintr-un obiect Python într-o reprezentare serializată, cum ar fi un șir JSON. Acest lucru poate implica conversia diferitelor tipuri de date Python în echivalentele lor în JSON și formatarea datelor conform cerințelor specificate în serializer.

Pe de altă parte, deserializarea implică procesul invers, transformând datele din formatul serializat înapoi în obiecte Python. Acest lucru este util atunci când primești date de la client sau din alte surse și dorești să le procesezi sau să le stocăm în baza de date.

Prin intermediul serializerelor din Django, poți controla modul în care datele sunt serializate și deserializate, inclusiv gestionarea conversiilor de date și a validărilor. Acest lucru este important pentru asigurarea integrității datelor și pentru compatibilitatea între diferitele componente ale aplicației.

### Interfețe utilizator

* Ecranul principal:

Pe ecranul principal, utilizatorul va vedea patru pătrate, fiecare reprezentând o zonă de irigare a serei.

Utilizatorul poate interacționa cu aceste pătrate, făcând clic pe ele pentru a vedea datele despre temperatura și umiditatea din fiecare zonă.

În partea dreaptă a ecranului, se va afișa o listă cu temperaturile și umiditățile fiecărei zone, actualizate în timp real.

* Designul pătratelor:

Când utilizatorul face clic pe un pătrat, acesta se va ilumina pentru a indica selecția.

Pătratele vor fi colorate în funcție de condițiile de mediu din fiecare zonă:

Verde pentru condiții optime de irigare.

Roșu pentru condiții nefavorabile.

Portocaliu pentru condiții intermediare.

* Tab-uri pentru vizualizarea datelor:

Interfața va include două tab-uri separate pentru vizualizarea datelor despre temperatură și umiditate.

Primul tab va afișa datele despre temperatură pentru fiecare zonă, iar al doilea tab va afișa datele despre umiditate.

Utilizatorul poate naviga între aceste tab-uri pentru a vizualiza datele relevante.

* Irigarea manuala pentru fiecare patrat:

Pe fiecare patrat de pe ecranul de vizualizare a datelor despre umiditate, utilizatorul va avea amplasat un buton de pornire a pompei. Când utilizatorul apasă butonul, acesta se va schimba automat în "OFF", indicând că acțiunea a fost inițiată și că pompa este în funcțiune (această funcționalitate urmează să fie implementată). Când butonul este apăsat, timer-ul va porni, indicat pentru cât timp este pompa deschisă.

O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automat

Fig. 3 – Interfața care monitorizează umiditatea

O imagine care conține text, captură de ecran, diagramă, proiectare

Descriere generată automat

Fig. 4 – Interfața care monitorizează temperatura

#### Intrări

În cadrul proiectului, mijloacele de intrare sunt esențiale pentru interacțiunea utilizatorului cu sistemul, facilitând controlul și monitorizarea proceselor agricole. Principala formă de intrare constă în comenzile primite de la utilizator pentru pornirea și oprirea pompelor și electrovalvelor, care sunt elemente cheie în sistemul nostru de irigații. Aceste comenzi sunt transmise de utilizator către aplicație prin intermediul unor interfețe grafice simple și intuitive, care permit controlul eficient al activităților din câmp.

Interfețele grafice furnizează ecrane de control și comenzi, unde utilizatorul poate selecta pompele și electrovalvele pe care dorește să le activeze sau să le dezactiveze. Aceste ecrane sunt concepute pentru a fi ușor de înțeles și de utilizat, facilitând utilizatorilor să gestioneze fluxurile de apă în mod eficient în funcție de nevoile specifice ale culturilor și condițiilor meteorologice.

Elementele de date asociate acestor ecrane includ:

* Butonul de pornire/oprire pentru fiecare pompă și electrovalvă disponibilă în sistem.
* Indicatoare vizuale care arată starea curentă a fiecărei pompe și electrovalve (activată sau dezactivată).
* Mesaje de confirmare și de eroare care sunt afișate pentru a confirma acțiunile utilizatorului sau pentru a semnala eventuale probleme sau erori în timpul procesului de comandă.

#### Ieșiri

În cadrul sistemului nostru, ieșirile reprezintă acțiunile sau procesele automate generate de sistem în urma analizei datelor provenite de la senzori. Principala formă de ieșire constă în pornirea automată a pompelor și electrovalvelor, care sunt controlate în funcție de informațiile obținute de la senzorii de monitorizare a condițiilor din câmpul agricol.

Sistemul primește date de la senzori referitoare la nivelul de umiditate, temperatură sau alte parametri relevanți pentru starea culturilor. Pe baza acestor date, algoritmul de control din sistem analizează și interpretează informațiile pentru a determina dacă este necesară activarea sau dezactivarea pompelor și electrovalvelor.

Ieșirile sunt generate automat atunci când sistemul decide că condițiile din câmpul agricol necesită intervenția pentru asigurarea irigației adecvate a culturilor. De exemplu, dacă nivelul de umiditate scade sub un anumit prag critic, sistemul va activa automat pompele și electrovalvele pentru a furniza apă în mod corespunzător culturilor.

Elementele asociate acestor ieșiri includ:

* Algoritmul de control, care interpretează datele de la senzori și decide asupra acțiunilor de ieșire necesare.
* Comenzile de control care sunt transmise automat către pompele și electrovalvele conectate pentru a le porni sau opri în funcție de cerințele de irigație.
* Indicatoare vizuale sau mesaje de confirmare care sunt afișate pentru a notifica utilizatorii cu privire la acțiunile automate întreprinse de sistem.

## Scenarii de utilizare

Pentru a folosi în mod eficient sistemul de irigare automatizat, este necesară o înțelegere detaliată a modului în care acesta funcționează și a tuturor componentelor implicate.

În primul rând, sistemul trebuie conectat la o sursă de alimentare electrică, asigurându-ne că este alimentat corespunzător la priză. Această conexiune va furniza energia necesară funcționării surselor de alimentare de 12V și a altor componente ale sistemului, precum electrovalvele și Arduino-ul.

Senzorii de umiditate și temperatură joacă un rol crucial în funcționarea sistemului. Aceștia monitorizează constant nivelul de umiditate din sol și temperatura mediului ambiant, preluând datele o dată la fiecare 12 ore. Informațiile colectate sunt esențiale pentru determinarea momentului și cantității de apă necesare pentru irigare. De asemenea, aceste date pot fi utile pentru evaluarea stării de sănătate a plantelor și pentru ajustarea parametrilor de irigare.

Procesul de irigare este gestionat inteligent, folosind o logică fuzzy, care permite sistemului să ia decizii în funcție de parametrii specifici ai fiecărei culturi selectate de către utilizator. Aceasta înseamnă că sistemul poate adapta nivelul și frecvența irigării în funcție de nevoile individuale ale plantelor, asigurându-se că acestea primesc cantitatea optimă de apă pentru o creștere sănătoasă.

Utilizatorul are posibilitatea să intervină manual în procesul de irigare, având acces la butoane dedicate pentru activarea sau oprirea pompei de irigare în fiecare zonă. Acest lucru oferă utilizatorului controlul complet asupra sistemului și posibilitatea de a interveni în mod direct în funcționarea acestuia în funcție de necesități sau preferințe.

Datele colectate de senzori sunt transmise către un server central folosind un modul WiFi ESP8266. Serverul procesează și stochează aceste date într-o bază de date, iar aplicația mobilă le preia și le afișează utilizatorului într-un format ușor de înțeles și accesibil. Acest lucru permite utilizatorului să monitorizeze starea de irigare și să vizualizeze istoricul datelor pentru a evalua performanța sistemului și nevoile plantelor.

## Proiectare de detaliu

### Proiectare hardware de detaliu

Pentru proiectul, detaliez componentele hardware utilizate în sistemul de monitorizare și control al irigației agricole.

* Sursa de alimentare de 12V: Utilizăm o sursă de alimentare cu tensiune de 12V pentru a alimenta întregul sistem. Această sursă trebuie să furnizeze suficientă putere pentru a susține funcționarea continuă a tuturor componentelor conectate la sistem.
* Electrovalve de 12V: Electrovalvele de 12V sunt utilizate pentru a controla fluxul de apă către fiecare zonă de irigație în funcție de necesitățile culturilor. Aceste electrovalve sunt conectate la sistem și sunt activate și dezactivate automat în funcție de datele obținute de la senzori.
* Tranzistor: Un tranzistor este utilizat pentru a controla automat electrovalvele din Arduino. Arduino emite semnale digitale către tranzistor, care reglează fluxul de curent către electrovalve, permițându-le să se deschidă și să se închidă în mod corespunzător.
* Arduino: Arduino este creierul sistemului nostru, care primește date de la senzori și decide asupra acțiunilor de control, cum ar fi activarea sau dezactivarea electrovalvelor. Arduino utilizează un algoritm de control pentru a interpreta datele de la senzori și pentru a emite comenzile corespunzătoare către electrovalve.
* Senzor DHT11: Senzorul DHT11 este utilizat pentru a măsura temperatura și umiditatea aerului în zona de irigație. Acest senzor furnizează date precise despre condițiile meteorologice locale, care sunt importante pentru programarea corectă a sistemului de irigație.
* Senzor de umiditate a solului: Acest senzor este utilizat pentru a monitoriza umiditatea solului în fiecare zonă de irigație. Datele obținute de la acest senzor sunt utilizate pentru a determina când este necesar să se activeze electrovalvele pentru a furniza apă culturilor.
* Senzor de nivel de apă: Acest senzor este utilizat pentru a monitoriza nivelul apei în rezervorul de stocare a apei pentru irigație. El poate detecta dacă nivelul apei este suficient de ridicat pentru a asigura o irigare corespunzătoare a culturilor sau dacă este necesară alimentarea cu apă suplimentară.
* Pompă de 5V: Pompa de 5V este utilizată pentru a pompa apa din rezervor către zonele de irigație atunci când este necesar. Ea este controlată de către Arduino și este activată și dezactivată în funcție de datele obținute de la senzorii de umiditate a solului și de nivelul apei.

### Proiectare software de deatliu

Pentru a proiecta și descrie serviciile software ale sistemului, este crucial să detaliem fiecare aspect al funcționării lor. Serviciul de control al irigației agricole, clasificat ca o aplicație, are scopul de a automatiza procesul de irigare în funcție de datele obținute de la senzorii de umiditate a solului și nivelul apei. Printre cerințele principale ale acestui serviciu se numără asigurarea unei irigații corespunzătoare, adaptată nevoilor culturilor și condițiilor de mediu.

Pentru a funcționa eficient, serviciul utilizează structuri de date interne pentru stocarea informațiilor provenite de la senzori și pentru a controla dispozitivele de irigare. Respectarea constrângerilor, cum ar fi limitele și cerințele de funcționare ale pompelor și electrovalvelor, este crucială pentru a evita supra-irigarea sau sub-irigarea culturilor. Integrarea acestui serviciu implică utilizarea componentelor hardware, cum ar fi senzorii de umiditate a solului și de nivelul apei, pompele și electrovalvele, controlate de un microcontroler sau Arduino.

Interacțiunea cu dispozitivele hardware și modulele de senzori este esențială pentru obținerea datelor necesare și pentru a controla funcționarea corectă a pompelor și electrovalvelor. Procesarea datelor de la senzori este efectuată pentru a determina necesitatea de irigare, iar serviciul activează sau dezactivează automat pompele și electrovalvele în funcție de aceste date. În plus, serviciul furnizează o interfață de utilizator sau un API pentru controlul și monitorizarea sistemului, facilitând interacțiunea utilizatorului cu funcționalitățile acestuia. Această integrare și funcționalitate completă a serviciului de control al irigației agricole reprezintă un element esențial al sistemului nostru global.

### Proiectare de detaliu pentru performanța sistemului

Pentru a asigura performanța și fiabilitatea sistemului hardware, este esențial să avem o proiectare detaliată care să satisfacă cerințele de capacitate și disponibilitate. Cerințele noastre includ capacitatea de a gestiona un volum mare de date din senzori și de a asigura disponibilitatea sistemului în mod continuu pentru a evita întreruperile în operațiunile agricole.

Pentru a îndeplini aceste cerințe, am identificat și evaluat performanțele așteptate ale fiecărei componente hardware, inclusiv microcontrolerul Arduino, senzorii de umiditate a solului și a nivelului apei, pompele și electrovalvele. De asemenea, am estimat capacitatea și volumul de date pe care trebuie să le manipuleze sistemul în timp real, având în vedere frecvența citirilor de la senzori și necesitățile de control ale dispozitivelor de irigare.

Pentru a îndeplini cerințele de performanță, am proiectat hardware-ul pentru a optimiza viteza de procesare a datelor și capacitatea de răspuns la comenzi. Am ales componente care sunt capabile să manipuleze volumul estimat de date într-un timp rezonabil și să asigure o comunicare eficientă între ele

În ceea ce privește fiabilitatea, am luat în considerare punctele de eșec potențiale ale fiecărei componente și am implementat măsuri de rezervă sau redundanță pentru a minimiza impactul unui posibil eșec. De asemenea, am proiectat un sistem de backup, recuperare și arhivare pentru a proteja datele și pentru a asigura o funcționare continuă în caz de evenimente neprevăzute.

### Proiectare detaliată a comunicațiilor interne (între componente)

Senzorii colectează datele și le transmit către placa Arduino folosind o conexiune bit cu bit. Aceste date sunt apoi transmise prin porturile de comunicație RX și TX ale plăcii Arduino către modulul ESP, care le transmite prin Wi-Fi către un server utilizând metoda PUT. Datele sunt prelucrate la nivelul serverului și transmise prin HTTPS către aplicația mobilă pentru afișare.

În plus, aplicația mobilă primește comenzi de la utilizator pentru controlul pompei și al electrovalvelor. Aceste comenzi sunt transmise prin HTTPS înapoi către server, care le transmite apoi prin Wi-Fi și protocolul UART către placa Arduino. Placa Arduino procesează comanda și controlează pompa și electrovalvele în consecință. Astfel, diagrama evidențiază întregul flux de date în sistem, inclusiv interacțiunea bidirecțională între aplicație și