

# Fire detection and notification system

## Introducere

Proiectul presupune realizarea unui sistem inteligent de detecție a incendiilor, care poate identifica în timp real prezența focului, fumului și a gazelor. Sistemul alertează locuitorii atât local, prin semnale sonore și vizuale, cât și de la distanță, prin trimiterea automată a unei notificări pe telefon. În același timp, datele vor fi transmise către Thingspeak, unde sunt stocate automat și vor putea fi accesate pentru viitoarele analize sau cercetări.

- Scopul proiectului
  - Scopul principal al proiectului este de a contribui la creșterea nivelului de siguranță acasă, la muncă, la școală sau la un restaurant, prin implementarea unui sistem capabil să sesizeze rapid și cele mai mici semne ale unui posibil incendiu, prevenind astfel dezastrele ce orice fel. Acest proiect oferă o soluție ușor de implementat și care poate fi adaptată oricărui tip de locuință, deoarece alertele pot fi percepute foarte ușor de oamenii aflați în clădire în acel moment.
- Ideea de la care am pornit acest proiect
  - Foarte multe locuințe și spații publice nu dispun de sisteme inteligente de detectare a incendiilor. Este o nevoie reală pe care am observat-o de-a lungul timpului, iar dezvoltarea unui proiect ușor de integrat va salva vieți omenești și bunuri personale. Un element principal la care m-am gândit a fost și posibilitatea trimiterii unei notificări pe telefon. Cu toții știm că tehnologia este din ce în ce mai dezvoltată, iar noi devenim dependenți de tot ceea ce oferă ea. Astfel, proiectul pe lângă detectarea unui pericol, oferă și posibilitatea informării la distanță a unui utilizator, accelerând procesul de prevenire a situațiilor nedorite.
- Utilitatea proiectului
  - Consider că acest proiect este util pentru orice persoană, deoarece contribuie la siguranță noastră, prin primirea alertelor în timp real, reducând pierderile de orice fel. Proiectul aduce beneficii reale în viața de zi cu zi, acesta fiind o reală necesitate în orice locuință.

## Descriere generală

Sistemul de detectare a incendiului este construit pe baza plăcii de dezvoltare Arduino UNO și pe baza microcontrollerului ESP8266. Se citesc date de la mai mulți senzori, cum ar fi trei senzori de gaz MQ2, un senzor de temperatură și umiditate DHT11 și un senzor cu infraroșu pentru flacără. Datele vor fi procesate și în cazul în care se observă depășirea pragurilor, vom avea diferite alerte.

Cei trei senzori de gaz vor fi așezați urmând o strategie, astfel dacă cel puțin doi vor identifica o concentrație ridicată de gaz sau fum, se va considera un pericol real care va declanșa alerta. Se va

activa și un buzzer, se va aprinde și led-ul RGB, iar pe LCD se va afișa un mesaj de avertizare.

Pe lângă acești senzori, există și un senzor infraroșu pentru flacără care va detecta apariția unei flăcări, înainte ca fumul să fie sesizat de senzorii MQ2. În acest caz, se consideră o situație de atenționare. Senzorul DHT11 va monitoriza permanent temperatura și umiditatea. Dacă vor apărea valori ridicate și anormale ale temperaturii, acestea vor fi considerate indicii suplimentare de incendiu.

**1. Arduino UNO** : microcontroller clasic folosit pentru citirea senzorilor (analogici și digitali), stabilirea deciziilor locale (ex. aprindere LED, activare buzzer) și trimiterea datelor către ESP8266 (prin UART).

**2. ESP8266** : microcontrollerul Wi-Fi care primește date de la senzori, le procesează și stabilește dacă există o situație de pericol. Va trimite datele către Thingspeak.

**3. 3 x Senzori de gaz MQ2**: detectează fum și gaze inflamabile. Aceștia vor fi plasați în zone strategice pentru a acoperi cât mai mult din suprafață. Vor oferi un semnal analogic ce va reflecta concentrația de gaz.

**4. Senzor infraroșu de flacără**: detectează radiația IR corespunzătoare flăcărilor. Oferă semnal digital. Are posibilitatea de a detecta flacăra, înainte ca ceilalți senzori de gaz să detecteze fumul.

**5. Senzor DHT11 (temperatură și umiditate)**: trimite date digitale despre temperatura și umiditatea aerului. Valorile neobișnuite indică începutul unui incendiu, iar în acest caz, se va afișa un mesaj de atenționare și se va aprinde becul galben. **6. LCD 1602 cu interfață I2C: afișează valori în timp real: temperatură, umiditate, nivel gaz, stări de alarmă. Permite locatarului să observe local un mesaj de atenționare.** **7. Buzzer: activează o alarmă sonoră când se depășește un anumit prag.** **8. LED RGB: oferă alertare vizuală: \* Verde = totul e normal \* Roșu = atenționare sau pericol incendiu** **Module Software \* Programul principal pe Arduino Uno (scris în Arduino IDE) - Citește valorile de la senzorii de gaz, flacără și DHT11 - Evaluează local starea de pericol și controlează buzzerul, LED-ul RGB și LCD-ul - Trimite datele către ESP8266 prin UART \* ESP8266 - Primește datele de la Arduino - Realizează conexiunea la rețeaua Wi-Fi - Trimite date către Thingspeak**

- Thingspeak – grafice în timp real

1. Primește date în timp real de la ESP8266
2. Oferă stocare centralizată și acces de la distanță

Interacțiunea dintre module

- ESP8266 se conectează la rețeaua Wi-Fi și așteaptă date de la Arduino Uno prin UART.

- Arduino Uno citește valorile de la senzorii conectați:

1. MQ2 (gaz/fum) – semnal analogic
2. IR (flacără) – semnal digital
3. DHT11 (temperatură și umiditate) – semnal digital

- În funcție de pragurile stabilite:

1. Pentru o avertizare (ex. temperatură crescută sau flacără fără gaz): se afișează un mesaj pe LCD și se aprinde LED-ul galben.
2. Pentru un pericol de incendiu (flacără + gaz): se activează buzzerul și LED-ul roșu, se afișează



2. Pini folosiți: A0, A1, A3
3. Senzorii MQ2 generează semnal analogic proporțional cu concentrația gazelor inflamabile. Pini A0, A1 și A3 sunt pini analogici de pe Arduino Uno și permit citirea valorilor folosind conversia ADC.

#### • **Senzor infraroșu de flacără**

1. Pentru identificarea flăcărilor
2. Pin folosit: D10
3. Acest senzor transmite un semnal digital (LOW = flacără detectată). Pinul D10 este configurat ca intrare digitală și

este potrivit pentru a recepționa semnale de tip ON/OFF.

#### • **Senzor DHT11**

1. Pentru monitorizarea temperaturii și umidității aerului
2. Pin folosit: D9
3. DHT11 folosește un protocol digital, gestionat printr-o bibliotecă dedicată (SimpleDHT). Pinul D9 este un pin digital compatibil, potrivit pentru a primi semnalul de la acest senzor.

#### • **LCD 1602 cu interfață I2C**

1. Pentru afișarea locală a valorilor și alertelor
2. Pini folosiți: A4 (SDA), A5 (SCL)
3. Afișajul comunică prin protocolul I2C, care utilizează două fire: SDA (Date) și SCL (Ceas). Pe Arduino Uno, aceste funcții sunt predefinite pe pinii A4 și A5.

#### • **Buzzer pasiv**

1. Pentru alertă sonoră
2. Pin folosit: D5
3. Buzzerul este controlat digital. Pinul D5 este configurat ca ieșire (OUTPUT) și poate genera sunete PWM.

#### • **LED RGB cu anod comun**

1. Pentru alertă vizuală
2. Pini folosiți: D6 (Roșu), D5 (Verde)
3. Roșu care pâlpâie înseamnă avertizare, iar roșu intens înseamnă pericol.
4. Pinul D6 este ideal pentru control PWM (intensitate variabilă) pentru culoarea roșie. LED-ul are anod comun, deci LED-urile se aprind când pinii sunt pe LOW.

#### • **ESP8266 NodeMCU**

1. Pentru conectivitate Wi-Fi și trimitere de date către Thingspeak
2. Pini folosiți: D0 (RX), D1 (TX)
3. Comunicarea dintre Arduino și ESP8266 se realizează prin UART. Pini D0 și D1 corespund portului serial hardware sau software și permit trimiterea/recepționarea de date seriale

#### • **Fire jumper, rezistențe și breadboard**

1. Pentru conectarea și testarea circuitului

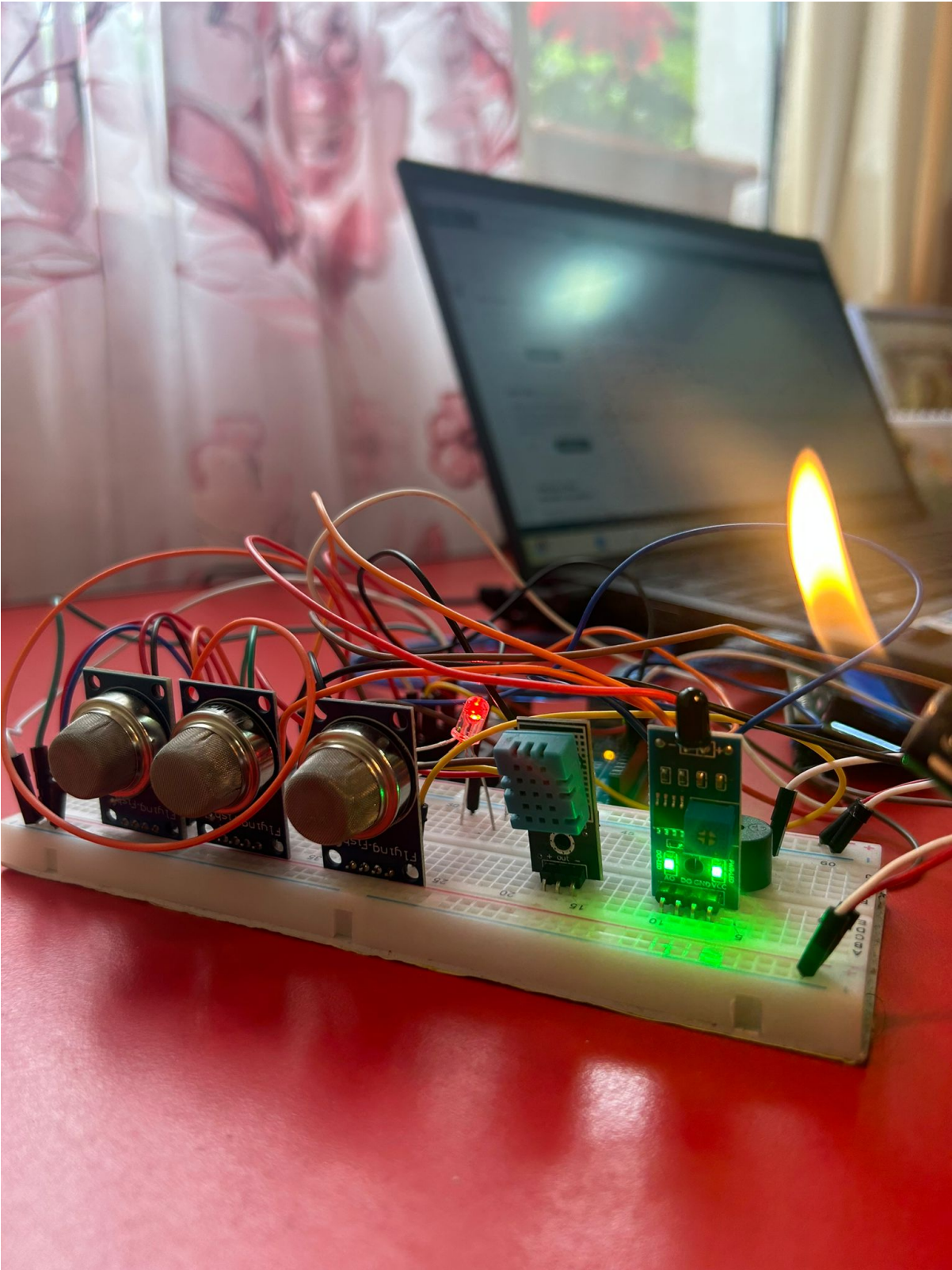
## Procesare și decizii locale

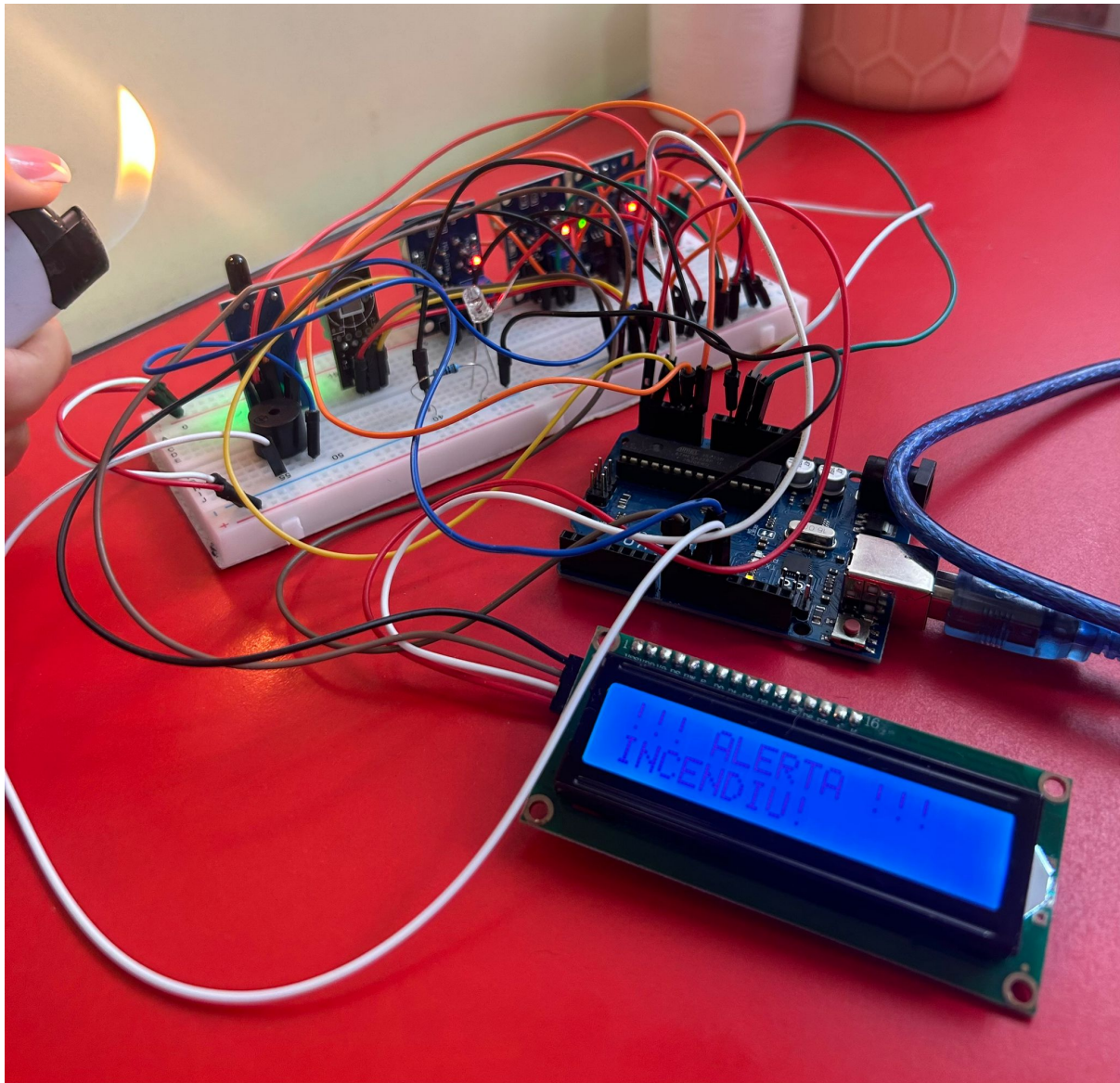
- Arduino preia valorile de la senzori și evaluează starea sistemului:
  1. Dacă se detectează **doar flacără**:
    1. Se afișează avertizare pe LCD
    2. LED roșu aprins care pâlpâie
    3. Se emit semnale sonore scurte cu buzzerul
  1. Dacă se detectează **flacără + minim 2 senzori de gaz activi**:
    1. LED roșu aprins
    2. Buzzer continuu
    3. Mesaj de alertă pe LCD
    4. Pentru a considera incendiu, minim 2 senzori trebuie să aibă răspunsul 'DA'
  1. Dacă **nu există amenințare**:
    1. LED oprit
    2. Buzzer oprit
    3. Afișaj cu temperatură și umiditate
- ESP8266 transmite în timp real informațiile primite de la Arduino către Thingspeak pentru realizarea în timp real a graficelor.

## Schemă electrică



## Imagini componente







## Software Design

### Stadiul actual al implementării software

- Sistemul monitorizează în timp real nivelul de gaze de la trei senzori analogici (pe intrări ADC), temperatura și umiditatea prin senzorul DHT11, precum și prezența flăcării printr-un senzor digital

infraroșu.

- În funcție de datele preluate, se declanșează automat alerte vizuale (LED PWM) și auditive (buzzer PWM), iar starea sistemului este afișată pe un LCD I2C 16×2.

### Bibliotecile utilizate

Bibliotecă	Motivare utilizare
Wire.h	Comunicație I2C cu LCD-ul.
LiquidCrystal_I2C.h	Permite controlul LCD-ului cu doar 2 fire, economisind pini.
SimpleDHT.h	Acces rapid la temperatura și umiditate de la senzorul DHT11.

### Elementul de noutate

- Utilizarea a **3 senzori de gaz** simultan pentru o detecție mai precisă.
- **Corelarea logică** a prezenței flăcării și concentrației de gaz pentru stabilirea gradului de pericol.
- **Semnalizare adaptivă PWM** (LED + buzzer) în funcție de gravitatea situației.
- **Interfață informativă în timp real** pe un afișaj I2C LCD.
- **Trimiterea datelor în Thingspeak**

### Funcționalități din laborator

Funcționalitate	Laborator	Implementare în proiect
ADC	Lab 4	Citirea senzorilor de gaz pe PA0, PA1, PA3
PWM	Lab 2	Buzzer și LED controlate cu Timer0 (OCR0A/B)
I2C	Lab 3	Afișaj LCD I2C 16×2 cu `LiquidCrystal_I2C`

### Scheletul proiectului și interacțiuni

- `initPWM()` - inițializează Timer0 pentru LED și buzzer;
  - `initFlameSensor()` - configurare pin senzor flacără;
  - `ADC_init()` și `ADC_read(chan)` - pentru senzorii de gaz;
  - `actualizeazaStare()` - analizează datele senzorilor;
  - `afisareLCD()` - afișează mesajul corespunzător;
  - `semnalizarePWM()` - generează semnale sonore și vizuale.
- Testarea s-a făcut cu o brichetă, observând comportamentul în timp real pe ecran și prin buzzer/LED.

### Calibrarea senzorilor

- **Senzori de gaz** calibrați prin testare în aer curat și la contactul cu flacăra.
  - Praguri alese: ``> 600`` pentru detecție gaz.
- **Senzor flacără**: valoare logică ``LOW`` la detectare - inversată software:
  - `uint8_t flacara = !(PINB & (1 << IRSENSOR_PIN));`

### Optimizări implementate

- Reutilizarea Timer0 pentru controlul simultan al buzzerului și LED-ului prin PWM pe canale separate (OCR0A și OCR0B), economisind resurse hardware.
- Acces direct la registre ADC și PWM, evitând funcții de nivel înalt.

## Rezultate Obținute

Analizând datele colectate în ThingSpeak, observ că senzorii de gaz (Gas1, Gas2, Gas3) reacționează corect, înregistrând creșteri semnificative atunci când sunt expuși la concentrații mai mari de gaz. Senzorul de flacără răspunde precis la prezența focului, schimbând starea între 0 și 1, ceea ce arată că partea de detecție funcționează bine.



## Concluzii

Prin realizarea acestui proiect am reușit să construiesc un sistem funcțional de monitorizare a incendiilor, bazat pe Arduino. Am folosit senzori de gaz, un senzor de flacără și un senzor DHT11 pentru temperatură și umiditate, astfel încât să pot detecta rapid orice situație de risc. Pentru mine, a fost foarte important ca sistemul să ofere feedback local. Astfel, am implementat semnalizare vizuală pe un afișaj LCD și LED, dar și semnalizare sonoră cu un buzzer. În plus, mi-am dorit ca datele colectate să nu rămână doar local, ci să fie transmise către internet, motiv pentru care am integrat un modul ESP și am folosit platforma ThingSpeak pentru vizualizare online.

## Download

Codul sursă [https://github.com/anamariapotoceanu/PM\\_PROIECT/](https://github.com/anamariapotoceanu/PM_PROIECT/) și Video demo: <https://www.youtube.com/shorts/su3ioFB6500>

## Bibliografie/Resurse

### Resurse Software

- [https://www.espressif.com/sites/default/files/documentation/4b-esp8266\\_at\\_command\\_examples\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/4b-esp8266_at_command_examples_en.pdf)
- [https://docs.espressif.com/projects/esp-at/en/latest/esp32/AT\\_Command\\_Set/Basic\\_AT\\_Commands.html](https://docs.espressif.com/projects/esp-at/en/latest/esp32/AT_Command_Set/Basic_AT_Commands.html)
- [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)
- <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024>

### Resurse Hardware

- <https://www.tinkercad.com/things/41eMA4Gu44g/editel>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2025/eradu/ana\\_maria.potoceanu](http://ocw.cs.pub.ro/courses/pm/prj2025/eradu/ana_maria.potoceanu)



Last update: **2025/05/29 23:23**