FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Inginerie Electrică şi Ştiinţa Calculatoarelor
Departamentul	Calculatoare, Electronică şi Automatică
Domeniul de studii	Ingineria sistemelor
Ciclul de studii	Licenţă
Programul de studii	Automatică şi informatică aplicată

2. Date despre disciplină

Denumirea disciplinei	**INGINERIA SISTEMELOR DE PROGRAME**				
Anul de studiu	IV	Semestrul	7	Tipul de evaluare	C
Regimul disciplinei	Categoria formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				DF
	Categoria de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă				DFA

3. **Timpul total estimat** (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	5	Curs	3	Seminar		Laborator	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	70	Curs	42	Seminar		Laborator	28	Proiect	

II Distribuția fondului de timp pe semestru:	ore
II. Studiu individual	52
III. Examinări	3
IV. Alte activități (precizați):	-

Total ore studiu individual (II+III)	55
Total ore pe semestru (I.b+II+III+IV)	125
Numărul de credite	5

4. Competențe specifice acumulate

Competențe profesionale	
Competențe transversale	

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
1. Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale din sisteme automate, sisteme încorporate și inteligente, știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete.	Studentul/absolventul modelează la nivel de sistem și realizează sisteme de control numerice, interfețe și sisteme bazate pe microcontrolere și microprocesoare, folosind - proiectarea hardware – software integrată (codesign) și inginera programării.	Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.

6. **Obiectivele disciplinei** (reieșind din grila competențelor specifice acumulate)

Obiectivul general al disciplinei	• Studentul/Absolventul trebuie să fie capabil să analizeze, să proiecteze și să implementeze soluții software complexe, de calitate, în conformitate cu cerințele considerate, utilizând metodologii adecvate și instrumente moderne, și să își asume responsabilitatea pentru calitatea, testarea și securitatea produselor program dezvoltate în echipă.

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observații
1. Introducere, definiție, istoric, principii de bază, fazele ingineriei programării. Dicționar termeni	3h	expunerea, prelegerea, conversație, studiu de caz, demonstrația	
2. Metodologii de dezvoltare software	3h		
3. Managementul proiectelor software	2h		
4. Ingineria cerințelor	3h		
5. Modelarea sistemelor software			
5.1. Introducere, limbaje de modelare	2h		
5.2. UML. Diagrame structurale. Exemple	3h		
5.3. UML. Diagrame comportamentale. Exemple	3h		
6. Proiectarea sistemelor software	3h		
7. Implementarea sistemelor software			
7.1. Prezentare generală, ghiduri	2h		
7.2. Instrumente software	2h		
7.3. Reutilizare software	2h		
8. Șabloane de proiectare	3h		
9. Verificare și validare. Testare software. Tipuri de testare. Instrumente pentru managementul testării	3h		
10. Calitatea produselor software	2h		
11. Securitatea sistemelor software. Securitatea in ciclul de dezvoltare a unui produs software	3h		
12. Inteligența artificială generativă și ingineria software	3h		

Bibliografie minimală recomandată

1. I. Sommerville, *Software Engineering*, 10th Edition, Pearson, India, 2017
2. M. Richards and N. Ford, *Fundamentals of software architecture: An engineering approach*. Sebastopol, CA: O'Reilly Media, 2020.
3. D. Farley, *Modern software engineering: Doing what works to build better software faster*. Boston, MA: Addison Wesley, 2022.
4. Lara, Letaw. Handbook of Software Engineering Methods. 2024.
5. *, Software Engineering Body of Knowledge (SWEBOK), IEEE Computer Society, 2024, https://www.computer.org/education/bodies-of-knowledge/software-engineering
6. Suportul electronic de curs
7. Resurse online: tutoriale, articole de specialitate

Aplicații (laborator)	Nr. ore	Metode de predare	Observații
1. Analiza stadiului actual în domeniul proiectului de diplomă/ Identificare și documentare tematică pentru a obține minim 2 direcții relevante pentru tema propusă. Familiarizarea cu un instrument specific de management al proiectelor	2	lucrări practice, exerciții, studii de caz, evaluare	
2. Elaborarea specificațiilor cerințelor software pentru proiectul de diplomă	2		
3. Proiectarea interfeței utilizator	2		
4. Proiectarea aplicației. Realizarea diagramelor UML	2		
5. Proiectare API REST. Proiectare arhitectură II (diagrame UML: secvențe, stări, clase).	2		

6. Stabilire temă comuna și echipe de lucru. Elaborarea specificațiilor cerințelor software pentru aplicația de realizat in echipă	2		
7. Proiectare wireframe UI. Proiectare Prototip UI.	2		
8. Proiectarea arhitectura. Realizarea diagramelor UML	2		
9. Planificare implementare versiunea MVP	2		
10. Proiectare baza de date. Implementarea bazei de date. Populare bază de date	2		
11. Implementare schelet aplicație (UI minim integrat cu Backend minim). Dezvoltare module aplicație.	2		
12. Elaborare plan de testare. Testare aplicație. Rafinare implementare. Pregătire aplicație pentru beneficiar	2		
13. Implementarea și execuția unui test automat pentru UI, pentru o aplicație oarecare. Testarea aplicației de către altă echipă	2		
14. Prezentare și evaluare aplicație și documentație elaborată	2		

Bibliografie minimală recomandată

1. Resurse online: tutoriale, materiale bibliografice specifice limbajelor de programare utilizate, articole de specialitate
2. I.Sommerville, *Software Engineering*, 10th Edition, Pearson India, 2017
3. A. Cernian, A.D. Ioniță, *Noțiuni aplicate de inginerie a sistemelor de programe*, Ed. Matrixrom, București, 2009
4. O. Gherghieș, A. Apetrei, *Ingineria programării*, Curs, Iași, 2002
5. I. Odagescu, F. Furtuna, *Ingineria programării*, Ed. ASE, București, 2002
6. Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., *Proiectarea sistemelor informatice. Metode de realizare*, Sylvi, Bucuresti 2001
7. D. Bocu –*Inițiere în ingineria sistemelor soft*, Ed. Albastră, Cluj-Napoca, 2001
8. Suportul electronic de laborator

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Evaluare finală	Înțelegerea critică a teoriilor și principiilor specifice. Capacitatea de utilizare adecvată a termenilor și principiilor din domeniul ingineriei software. Claritatea prezentării aplicației finale și corectitudinea răspunsurilor la întrebări.	Evaluarea cunoștințelor teoretice și practice din tematica studiată în timpul semestrului prin test grilă Evaluarea finală a aplicației practice elaborate.	30 20
Evaluare pe parcurs	Evaluarea capacității de analiză, sinteză și concretizare a cunoștințelor teoretice în identificarea soluțiilor problemei propuse la lucrările de laborator Corectitudinea și calitatea dezvoltării aplicației software și documentației aferente. Corectitudinea planificării, a utilizării instrumentelor, contribuția în echipă	Evaluare continuă a activităților desfășurate	50

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
20.09.2026	Prof.dr.ing. Cristina TURCU	Ș.l.dr.ing. Felicia GÎZĂ

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
23.09.2026	Ș.l.dr.ing. Corneliu BUZDUGA

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
24.09.2026	Conf.dr.ing. Eugen COCA

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
25.09.2026	Prof.dr.ing. Laurențiu-Dan MILICI