Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti

Facultatea de Electronică, Telecomunicaţii şi

Tehnologia Informaţiei

FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	**Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti**
1.2 Facultatea	**Electronică, Telecomunicaţii şi Tehnologia Informaţiei**
1.3 Departamentul	**Electronică Aplicată şi Ingineria Informaţiei**
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicaţii şi Tehnologii Informaţionale
1.5 Ciclul de studii	Licenţă
1.6 Specializarea	Electronică aplicată

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)				Inginerie software Software Engineering			
2.2 Titularul activităţilor de curs				S.l./Lect. Dr. Ing. Elena Cristina STOICA			
2.3 Titularul activităţilor de seminar / laborator				S.l./Lect. Dr. Ing. Elena Cristina STOICA, S.l./Lect. Dr. George Valentin STOICA			
2.4 Anul de studiu	4	2.5 Semestrul	1	2.6. Tipul de evaluare	V	2.7 Regimul disciplinei	Op
2.8 Tipul disciplinei		S	2.9 Codul disciplinei	04.D.07.O.108		2.10 Tipul de notare	Nota

3. Timpul total estimat (ore pe semestru al activităţilor didactice)

3.1 Număr de ore pe săptămână	3	Din care: 3.2 curs	2	3.3 seminar/laborator	1
3.4 Total ore din planul de învăţământ	42	Din care: 3.5 curs	28	3.6 seminar/laborator	14
Distribuţia fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie şi notiţe Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii şi eseuri					27
Tutorat					0
Examinări					6
Alte activităţi (dacă există):					0

3.7 Total ore studiu individual	33.00
3.8 Total ore pe semestru	75
3.9 Numărul de credite	3

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	Programarea calculatoarelor, Structuri de date si algoritmi, Programare obiect-orientata
4.2 de rezultate ale învăţării	Cunoştinţe de algoritmi si structuri de date, programare într-un limbaj uzual

5. Condiţii necesare pentru desfăşurarea optimă a activităţilor didactice (acolo unde este cazul)

5.1 Curs	Acces Internet si platforma Moodle, sala de curs cu proiector
5.2 Seminar/ Laborator/Proiect	Sala de calculatoare, reţea de comunicaţie. Prezenţa obligatorie la laboratoare (conform regulamentului studiilor universitare de licenta).

6. Obiectiv general *(Se referă la intenţiile profesorilor pentru studenţi, la ceea ce studenţii vor fi învăţaţi în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului ştiinţific abordat, precum şi la rolul pe care acesta îl are în cadrul specializării studiate. Vor fi descrise de o manieră generală tematicile abordate, justificarea includerii cursului în planul de învăţământ al specializării studiate etc.)*

Studiul metodelor inginereşti de rezolvare de probleme reale prin elaborarea de produse software ale aplicatiei dedicate, cu metode, procedee si alte mijloace software organizate puse la dispozitie de experienta de rezolvare de probleme cu software de dezvoltare.

7. Competenţe *(Capacitatea dovedită de a utiliza cunoştinţe, aptitudini şi abilităţi personale, sociale şi/sau metodologice în situaţii de muncă sau de studiu şi pentru dezvoltarea profesională şi personală. Reflectă cerinţele angajatorilor.)*

Specifice	Aplicarea cunoştinţelor, conceptelor şi metodelor elementare privitoare la metodologiile si instrumentele de Inginerie Software. Dezvoltarea sistemelor software complexe: pe baza unor metodologii si instrumentele de Inginerie Software. Proiectarea şi utilizarea sistemelor de calcul şi a reţelelor de calculatoare.
Transversale (generale)	Capacitatea de a asigura planificarea şi managementul proiectelor din domeniul ingineriei informaţiei. Capacitatea de a lua decizii în vederea rezolvării problemelor curente, sau imprevizibile, care apar în procesul de exploatare a sistemelor de calcul. Capacitatea de a se informa şi documenta permanent pentru dezvoltarea personală şi profesională prin citirea literaturii de specialitate. Capacitatea de a comunica şi de a prezenta conţinut tehnic atât în limba română, cât şi în limba engleză. Flexibilitate în utilizarea de noi sisteme şi tehnologii în cadrul unei echipe în care membrii împreună ating un obiectiv bine definit, asumând în acelaşi timp roluri sau sarcini diferite. Capacitatea de a lua decizii în vederea rezolvării problemelor curente

8. Rezultatele învăţării *(Sunt enunţuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învăţării reflectă realizările studentului şi mai puţin intenţiile profesorului. Rezultatele învăţării informează studenţii despre ceea ce se aşteaptă de la ei din punct de vedere al performanţei, pentru a obţine notele şi creditele dorite. Sunt definite în termeni concreţi, folosind verbe similare exemplelor de mai jos şi indică ceea ce se va urmări prin evaluare. Rezultatele învăţării vor fi astfel redactate încât să fie evidenţiată clar relaţia faţă de competenţele definite la punctul 7.)*

Cunoştinţe	*Rezultatul asimilării de informaţii prin învăţare. Cunoştinţele reprezintă ansamblul de fapte, principii, teorii şi practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice şi/sau faptice.* **Enumeră** cele mai importante etape care au marcat dezvoltarea domeniului Ingineriei Software. **Defineşte** noţiuni specifice domeniului de Inginerie Software. **Descrie** noţiuni de bază si organizarea etapelor unui proiect software **Cunoştinţele acumulate** în cadrul orelor de curs si laborator pot fi folosite în proiectarea si implementarea proiectelor software

Aptitudini	*Capacitatea de a aplica cunoştinţe şi de a utiliza know-how pentru a duce la îndeplinire sarcini şi a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive şi creative) sau practice (implicând dexteritate manuală şi utilizarea de metode, materiale, unelte şi instrumente).* **Selectează** şi grupează informaţii relevante din domeniul ingineriei software. **Utilizează** argumentat principii specifice în vederea proiectării şi implementării proiectelor software folosind metodologii specifice ingineriei software **Lucrează** productiv în echipă. **Verifică** experimental soluţii identificate. **Rezolvă** aplicaţii practice. **Analizează** şi compară mai multe modele de date existente in domeniul ingineriei software **Identifică** soluţii şi elaborează planuri de proiectare a aplicatiilor software. **Formulează** concluzii la implementările realizate. **Argumentează** soluţiile identificate precum şi modurile de rezolvare a problemelor de proiectare si implementare a aplicatiilor software.
Responsabilitate şi autonomie	*Capacitatea cursantului de a aplica în mod autonom şi responsabil cunoştinţele şi aptitudinile sale.* **Selectează** surse bibliografice potrivite şi le analizează. **Respectă** principiile de etică academică, citând corect sursele bibliografice utilizate. **Demonstrează** receptivitate pentru contexte noi de învăţare. **Manifestă** colaborare cu ceilalţi colegi şi cadre didactice în desfăşurarea activităţilor didactice Demonstrează autonomie în organizarea situaţiei/contextului de învăţare sau a situaţiei problemă de rezolvat **Manifestă** responsabilitate socială prin implicarea activă în viaţa socială studenţească/implicare în evenimentele din comunitatea academică **Promovează/contribuie** prin soluţii noi, aferente domeniului de specialitate pentru a îmbunătăţi calitatea vieţii sociale. **Conştientizează** valoarea contribuţiei sale în domeniul ingineriei la identificarea de soluţii viabile/sustenabile care să rezolve probleme din viaţa socială şi economică (responsabilitate socială). **Aplică** principii de etică/deontologie profesională în analiza impactului tehnologic al soluţiilor propuse în domeniul de specialitate asupra mediului înconjurător. Analizează şi valorifică oportunităţi de afaceri/de dezvoltare antreprenorială în domeniul de specialitate. **Demonstrează** abilităţi de management al situaţiilor din viaţa reală (gestionarea timpului colaborare vs. conflict).

9. Metode de predare *(Se vor avea în vedere metode care să asigure predarea centrată pe student. Se va descrie modul în care se asigură participarea studenţilor la stabilirea propriului parcurs de învăţare, cum se identifică eventualele rămâneri în urmă şi ce măsuri remediale se adoptă în astfel de cazuri.)*

Prezentarea prelegerilor de curs se face în amfiteatru cu facilităţi multimedia sau online în platforma Teams. Materialele de curs sunt: notele şi prezentările de curs. Toate materialele sunt disponibile în format electronic pe platforma „Moodle” a facultăţii.

10. Conţinuturi

CURS		
Capitolul	**Conţinutul**	**Nr. ore**

1	1. Necesitatea ingineriei software, istoric, definiţii. Proiectarea inginerească, ciclu de proiectare, specificitatea proiectării software. Cicluri de viaţă al software-lui, tipuri de abordări în rezolvarea de probleme.	4
2	2. Analiza de sistem, definirea cerinţelor, niveluri formale de specificare a cerinţelor, documente cu cerinţe, evoluţie cerinţe, etape ale fazei de specificare, structura documentul de definire a cerinţelor şi de specificare a cerinţelor. Modelarea de sistem, context de sistem, analiza punctului de vedere, descriere de model sistem software. Modelare obiectuală	6
3	3. Proiectare software. Proiectare structurată: definiţii, teorema structurii, componente elementare, structuri de control de bază. Tehnici de descriere a programelor. Rafinarea programelor. Proiectarea funcţională: diagrame de flux de date şi diagrame de structură, dicţionar de date, sinteza diagramelor de structură.	6
4	4. Modelare obiectuală. UML Limbaj unificat de modelare. Organizare a activităţii în echipa software. Stil de programare.	4
5	5. Implementare, verificare şi validare de programe. Calitate a proiectării software. Metrici software. Software reutilizabil	4
6	6. Verificare finala	4
	Total:	28
Bibliografie: (1) A. Sommerville – “Software engineering”, ed. VIII, Addison Wesley, 2007		

LABORATOR		
Nr. crt.	**Conţinutul**	**Nr. ore**
1	Diagrame UML de clase	2
2	Diagrame UML de tip activitate	2
3	Diagrame UML de stare	2
4	Diagrame UML de interacţiune	2
5	Diagrame UML „use case”	2
6	Planificare. Utilitarul Microsoft Project	2
7	Colocviu	2
	Total:	14
Bibliografie: (1) A. Sommerville – “Software engineering”, ed. VIII, Addison Wesley, 2007		

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală

11.4 Curs	- cunoaşterea noţiunilor teoretice fundamentale; - cunoaşterea modului de aplicare a teoriei la probleme specifice; - analiza diferenţială a instrumentelorşi metodelor teoretice.	Test scris si oral de verificare, acoperă întreaga materie, realizând o sinteză între parcurgerea teoretică comparativă a materiei şi explicitarea prin exerciţii şi probleme a modelelor de aplicaţie.	50%
11.5 Seminar/laborator/proiect	- cunoaşterea tipurilor de metodologii si utilizarea acestora in rezolvarea unei probleme date; - cunoaşterea modului de utilizare a unor unelte pentru dezvoltarea, de aplicatii software	Colocviu final de laborator, cuprinzând o componentă teoretică şi o componentă practică. Componenta teoretică este verificată prin test; componenta practică este evaluată prin verificarea modului de rezolvare (implementare, testare, funcţionare) de către student a unei probleme practice.	50%
11.6 Condiţii de promovare			
Obţinerea a cel puţin 50% din punctaj			

12. Coroborarea conţinutului disciplinei cu aşteptările reprezentanţilor angajatorilor şi asociaţiilor profesionale reprezentative din domeniul aferent programului, precum şi cu stadiul actual al cunoaşterii în domeniul ştiinţific abordat şi practicile în instituţii de învăţământ superior din Spaţiul European al Învăţământului Superior (SEÎS)

Industria are o cerere importantă de ingineri calificaţi, cu specializări legate de proiectarea si întreţinerea retelelor de calculatoare si a aplicaţiilor de sistem pentru sisteme distribuite, inclusiv pentru aplicatii in Internet – cloud computing, pentru care este necesara cunoaşterea detaliata a structurii pe niveluri, a protocoalelor si a serviciilor reţelelor de calculatoare si a reţelelor de calculatoare interconectate; dezvoltarea actuala de sisteme dedicate complexe, inclusiv echipamente de comunicaţii sau mobile, includ obligatoriu cunoaşterea problemelor si a soluţiilor acestora în cadrul reţelelor de calculatoare. Programa cursului răspunde concret acestor cerinţe actuale de dezvoltare şi evoluţie, subscrise planului EUROPA 2020, a serviciilor din domeniul Inginerie Electronica. În contextul progresului tehnologic actual al tehnologiei informatiei, domeniile de activitate vizate cuprind toate aspectele vieţii economice si sociale in care se urmăreşte trecerea spre societatea bazata pe cunoaştere, în care experienţa umană este externalizata in sisteme independente, dedicate unor domenii diverse, de la sisteme utilitare curente, la domeniul telecomunicaţii, domeniul militar, domeniul de securitate (sisteme de supraveghere), domeniul automatizărilor industriale (sisteme de inspecţie produse), robotică (sisteme de interfaţare om-maşină) şi altele.

Se asigură astfel absolvenţilor competenţe adecvate cu necesităţile calificărilor actuale şi o pregătire ştiinţifică şi tehnică moderne, de calitate şi competitive, care să le permită angajarea rapidă după absolvire, fiind perfect încadrat în politica Universităţii Politehnica din Bucureşti, atât din punctul de vedere al conţinutului şi structurii, cât şi din punctul de vedere al aptitudinilor şi deschiderii internaţionale oferite studenţilor.

Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti

Facultatea de Electronică, Telecomunicaţii şi

Tehnologia Informaţiei

Data completării	Titular de curs	Titular(i) de aplicaţii
21.09.2026	S.l./Lect. Dr. Ing. Elena Cristina STOICA	S.l./Lect. Dr. Ing. Elena Cristina STOICA
		S.l./Lect. Dr. George Valentin STOICA

Data avizării în departament	Director de departament
22.09.2026	Conf. Dr. Bogdan Cristian Florea

Data aprobării în Consiliul Facultăţii	Decan
22.09.2026	Prof. Dr. Mihnea Udrea