

FIȘA DISCIPLINEI

Ingineria software

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	de Matematică și Informatică
1.3. Departamentul	de Matematică și Informatică al Liniei Maghiare
1.4. Domeniul de studii	Calculatoare și tehnologia informației
1.5. Ciclu de studii	Licență
1.6. Programul de studii / Calificarea	Ingineria informației
1.7. Forma de învățământ	cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Ingineria software				Codul disciplinei		MLM5011
2.2. Titularul activităților de curs			Lect. dr. SULYOK Csaba					
2.3. Titularul activităților de seminar			Lect. dr. SULYOK Csaba					
2.4. Anul de studiu	2	2.5. Semestrul	4	2.6. Tipul de evaluare	C	2.7. Regimul disciplinei		DD

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

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	3
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					6
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					30
Tutoriat (consiliere profesională)					-
Examinări					4
Alte activități					-
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				55	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	• Metode avansate de programare • Rețelistică • Programare web
4.2. de competențe	nu e cazul

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	• Sală de curs dotat cu videoproiector • Conexiune stabilă de rețea
5.2. de desfășurare a seminarului/ laboratorului	• Sală de seminar echipată cu calculatoare și rețea wireless fiabilă • Sursă de alimentare pentru laptopuri

6. Competențele specifice acumulate¹

Competențe profesionale/esențiale	• C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic • C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date • C2.1 Identificarea metodelor de dezvoltare adecvate pentru sistemele software • C6.4 Efectuarea de măsurători de performanță pentru timpi de răspuns, consum de resurse; stabilirea drepturilor de acces • C6.5 Elaborarea unui proiect rețea
Competențe transversale	• CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională • CT2 Gestionarea eficientă a activităților organizate într-un grup interdisciplinar și dezvoltarea abilității empatice pentru comunicarea interpersonală, relația cu diferite grupuri și cooperarea • CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• descrierea mediilor enterprise • descrierea ciclului de viață al unor proiecte tipice corporative
7.2 Obiectivele specifice	• conceptele ciclului de viață al proiectului: gestiunea cerințelor • instrumente de lucru în echipă: urmărirea versiunilor (version control), gestionarea proiectelor (project management), integrare continuă (continuous integration) • arhitecturi software • strategii de instalare (deployment)

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Săptămâna 1: Introducere	Expuneri, exemple	
Săptămânile 2-3: Urmărirea versiunilor: git & git flow	Expuneri, exemple	
Săptămânile 4-5: Sisteme build și de gestionarea dependențelor: maven, gradle, etc.	Expuneri, exemple	
Săptămâna 6: Cerințe, specificarea cerințelor: diagrame UML de tip use case	Expuneri, exemple	
Săptămâna 7: Managementul proiectelor; metode de lucru: clasic vs. Agile, Scrum	Expuneri, exemple	
Săptămâna 8: Containerizare și orchestrare: Docker	Expuneri, exemple	
Săptămâna 9: Integrare și implementare continuă	Expuneri, exemple	
Săptămânile 10-12: Arhitectura software: arhitecturi multistrat, hexagonale, asincrone, diagrame de componente UML	Expuneri, exemple	
Săptămâna 13: Proiectare, principii SOLID, diagrame de clasă UML, diagrame de secvență	Expuneri, exemple	
Săptămâna 14: Recapitulare	Expuneri, exemple	
Bibliografie: 1. Chacon, S., Straub, B. (2014). <i>Pro Git</i>. Apress. https://git-scm.com/book/en/v2 2. Fowler, M. (2004). <i>UML distilled: A Brief Guide to the Standard Object Modeling Language</i>. Addison-Wesley. 3. Martin, R. C. (2003). <i>Agile Software Development: Principles, Patterns, and Practices</i>. Pearson Education.		

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

4. Turley, F., Rad, N. (2018). <i>Agile Scrum Handbook</i> . Van Haren Publishing. 5. Turnbull, J. (2014). <i>The Docker Book: Containerization Is the New Virtualization</i> . James Turnbull. https://dockerbook.com/ 6. Sommerville, I. (2007). <i>Software Engineering</i> . Addison-Wesley. 7. Fowler, M. (2003). <i>Patterns of Enterprise Application Architecture</i> . Addison-Wesley. https://martinfowler.com/books/ea.html		
8.2 Seminar / laborator	Metode de predare	Observații
Săptămânile 1-4: - seminar: cerințe, diagrame use case - laborator: Git	Exerciții interactive, evaluarea sarcinilor de laborator	
Săptămânile 5-8: - Seminar: UML - Laborator: sisteme de build, Gradle	Exerciții interactive, evaluarea sarcinilor de laborator	
Săptămânile 9-12: - Seminar: Agile + arhitecturi - Laborator: Docker, docker-compose	Exerciții interactive, evaluarea sarcinilor de laborator	
Săptămâna 13-14: - seminar: recapitulare - laborator: integrare continuă, GitLab CI	Exerciții interactive, evaluarea sarcinilor de laborator	
Bibliografie: 1. Chacon, S., Straub, B. (2014). <i>Pro Git</i> . Apress. https://git-scm.com/book/en/v2 2. Fowler, M. (2004). <i>UML distilled: A Brief Guide to the Standard Object Modeling Language</i> . Addison-Wesley. 3. Martin, R. C. (2003). <i>Agile Software Development: Principles, Patterns, and Practices</i> . Pearson Education. 4. Turley, F., Rad, N. (2018). <i>Agile Scrum Handbook</i> . Van Haren Publishing. 5. Turnbull, J. (2014). <i>The Docker Book: Containerization Is the New Virtualization</i> . James Turnbull. https://dockerbook.com/ 6. Sommerville, I. (2007). <i>Software Engineering</i> . Addison-Wesley. 7. Fowler, M. (2003). <i>Patterns of Enterprise Application Architecture</i> . Addison-Wesley. https://martinfowler.com/books/ea.html		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Tematica subiectului arată o mare suprapunere a conținutului de subiecte cu subiecte similare predate în marile universități. - Curriculumul se bazează pe activitatea celor mai apreciați autori internaționali, iar literatura recomandată se bazează și pe cele mai relevante lucrări din domeniu. - Subiectele predate în cadrul cursului sunt necesare pentru plasarea în industria soft, companiile așteptând astfel de cunoștințe.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Dobândirea adecvată a cunoștințelor teoretice prezentate în cursuri	Quiz-uri la cursuri (Q)	30%
		Examen scris (V)	30%
10.5 Seminar/laborator	Punerea în practică a cunoștințelor teoretice învățate	Exerciții de laborator (L)	30%
		Exerciții de seminar (S)	10%
10.6 Standard minim de performanță			
Condițiile minime pentru trecere: ● Rezolvarea și prezentarea cu succes a primilor 3 din cei 4 exerciții de laborator. ● Obținerea a cel puțin 50% din punctajul examenului scris (V). ● Obținerea a cel puțin 50% din punctajul total (Q+V+L+S).			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:
29 aprilie 2025

Semnătura titularului de curs
Lect. dr. SÜLYÖK Csaba

Semnătura titularului de seminar
Lect. dr. SÜLYÖK Csaba

Data avizării în departament:
.....

Semnătura directorului de departament
Conf. dr. ANDRÁS Szilárd Károly

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru Dezvoltare durabilă - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".