



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei	MODELARE SIMULARE – TIDS309						
1.2 Titularul activităților de curs	Conf. Dr. Răileanu Alina-Beatrice						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Conf. Dr. Răileanu Alina-Beatrice						
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Metode Numerice (TIDS211) — ecuații diferențiale, integrare numerică, interpolare; Teoria Sistemelor (TIDS302) — ecuații de stare, funcții de transfer, stabilitate; Teoria Transmisiunii Informației (TIDS212) — procese stohastice, variabile aleatoare, distribuții de probabilitate; Achiziția și Prelucrarea Datelor (TIDS304) — senzori, filtrare, validarea datelor.
3.2 de competențe	Cunoașterea ecuațiilor diferențiale ordinare și a metodelor numerice de rezolvare (Euler, Runge-Kutta); înțelegerea proceselor stohastice și a distribuțiilor de probabilitate uzuale; abilitatea de a programa în Python cu NumPy, SciPy și Matplotlib; familiarizarea cu noțiunile de bază din teoria sistemelor (funcție de transfer, stabilitate).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă pentru demonstrații matematice, videoproiector și acces la internet; platforma Sakai; recomandare: Python 3.10+ cu NumPy, SciPy, Matplotlib, simpy și mesa instalate înainte de curs.
-------------------------------	---

4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; Python 3.10+ cu pachetele NumPy, SciPy, Matplotlib, simpy, mesa, sklearn instalate; studenții pot folosi propriul laptop; acces la MATLAB Online ca alternativă pentru componentele de Simulink.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C3. Cunoaște și aplică în mod critic metode numerice și tehnici de modelare și simulare a sistemelor, cu conștientizarea limitelor și ipotezelor fiecărei abordări. C29. Deține cunoștințe specializate privind teoria sistemelor — sisteme liniare, funcții de transfer, stabilitate, răspuns în frecvență — și aplicarea acestora în modelarea și analiza sistemelor de control și a microsistemelor.
5.2 Aptitudini	A7. Aplică metode numerice și tehnici de modelare matematică pentru simularea și rezolvarea problemelor ingineresti, utilizând instrumente software specializate de calcul numeric. A41. Utilizează tehnici de modelare și simulare avansată (Monte Carlo, simulare bazată pe agenți, emulare hardware) pentru validarea proiectelor ingineresti și evaluarea performanței sistemelor complexe înainte de implementare.
5.3 Responsabilitate și autonomie	R41. Evaluează critic și independent rezultatele simulărilor și modelărilor complexe, asumându-și responsabilitatea pentru validarea modelelor față de date reale și pentru comunicarea transparentă a incertitudinilor și limitelor prognozelor.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în modelare și simulare. Conceptele de model, sistem și simulare. Tipuri de modele: matematice, fizice, conceptuale. Clasificarea simulărilor: cu timp continuu, cu timp discret, cu evenimente discrete, stohastice, deterministe. Procesul de modelare: formulare, implementare, validare, verificare, utilizare. Erori de modelare: erori de abstractizare, de discretizare, numerice, de rotunjire.	2	Prelegere interactivă	
Tema 2: Modelarea sistemelor continue. Ecuații diferențiale ca modele: sisteme mecanice (masa-arc-amortizor), circuite RLC, populații (Lotka-Volterra). Reprezentarea în spațiul stărilor. Simularea numerică a ODE: metoda Euler (explicită și implicită), metoda Runge-Kutta de ordinul 4 (RK4). Eroarea de trunchiere locală și globală. Alegerea pasului de integrare: stabilitate numerică.	2	Prelegere, demonstrații Python	
Tema 3: Modelarea sistemelor de control. Funcțiile de transfer ca modele: polii și zerourile, răspunsul la impuls și la treaptă. Simularea sistemelor în buclă închisă cu un regulator PID: răspunsul tranzitoriu, suprareglajul, stabilitatea. Diagrama bloc și schema de simulink. Modelarea și simularea unui sistem samplat (discret): transformata Z și răspunsul în timp.	2	Prelegere, Python/MATLAB	

<p>Tema 4: Metoda Monte Carlo — fundamente. Generarea numerelor pseudoaleatoare: algoritmul Mersenne Twister, testele de uniformitate. Simularea variabilelor aleatoare prin metoda transformării inverse și metoda respingerii (acceptance-rejection). Estimarea integralelor multiple prin Monte Carlo: eroarea standard $O(1/\sqrt{N})$. Convergența și reducerea varianței: metoda variabilelor antitetice, eșantionarea stratificată.</p>	2	Prelegere interactivă	
<p>Tema 5: Metoda Monte Carlo — aplicații. Estimarea lui π prin metoda geometrică. Simularea unui portofoliu financiar: mișcarea browniană geometrică, Value at Risk (VaR). Simularea propagării erorilor în sisteme de măsurare cu N componente. Metoda Monte Carlo pentru calculul fiabilității unui sistem (serie, paralel, mixt). Algoritmul Metropolis-Hastings: principiul și aplicațiile.</p>	2	Prelegere interactivă, Python	
<p>Tema 6: Simularea cu evenimente discrete (DES). Conceptele DES: entitate, atribut, eveniment, activitate, resursă, coadă. Paradigma de simulare orientată pe evenimente. Ceasul de simulare: avansarea pe eveniment. Bibliotecile Python SimPy și AnyLogic. Modelarea unui sistem de așteptare M/M/1: formulele Erlang, compararea cu simularea. Modele mai complexe: M/G/c, cozi tandem.</p>	2	Prelegere interactivă	
<p>Tema 7: Modele de cozi și rețele de servere. Teoria cozilor: clasificarea Kendall (A/B/c/K/N/D). Sistemul M/M/1: derivarea indicatorilor (L, Lq, W, Wq), condiția de stabilitate $\rho < 1$. Sistemul M/M/c și M/M/∞. Rețele de cozi: teorema produsului Jackson. Modelarea unui sistem de producție cu mai multe stații. Modelarea traficului de rețea cu teoria cozilor.</p>	2	Prelegere interactivă	
<p>Tema 8: Simularea bazată pe agenți (ABM). Conceptele ABM: agent, mediu, interacțiune, emergență. Comparația ABM cu ecuații diferențiale și DES. Platforma Mesa (Python): model, agent, scheduler, colectarea datelor. Modele clasice ABM: jocul vieții Conway, modelul Schelling (segregare), modelul Boid (comportamente de stol), modelul Sugarscape (resurse și agenți). Validarea modelelor ABM.</p>	2	Prelegere interactivă, Python	
<p>Tema 9: Modele stocastice — lanțuri Markov. Procese stohastice: definiție, proprietatea Markov. Lanțuri Markov cu timp discret: matricea de tranziție, distribuția stationary, clasificarea stărilor (recurente, tranzitorii, absorbante). Lanțuri Markov cu timp continuu: generatorul infinitezimal, ecuațiile Kolmogorov, distribuția exponențială a</p>	2	Prelegere interactivă	

timpilor de rezidență. Aplicații: modele de fiabilitate, modele de trafic.			
Tema 10: Simularea sistemelor stocastice. Procesul Poisson: proprietăți, simularea sosirilor. Mișcarea browniană (procesul Wiener): proprietăți, simularea traiectoriilor. Ecuatiile diferențiale stocastice (SDE): ecuația Langevin, schema Euler-Maruyama. Aplicații: modelul Black-Scholes pentru prețul opțiunilor, modelul Ornstein-Uhlenbeck pentru variabile revertive la medie.	2	Prelegere interactivă	
Tema 11: Validarea și verificarea modelelor. Distincția verificare vs. validare. Tehnici de verificare: testarea cu cazuri cunoscute, analiza dimensională, simetria. Tehnici de validare: compararea cu date reale (goodness-of-fit), analiza sensibilității parametrilor, cross-validarea. Metricile de validare: RMSE, MAE, R ² , eroarea relativă. Raportarea incertitudinilor: intervalele de confidență pentru indicatorii simulați.	2	Prelegere interactivă	
Tema 12: Analiza statistică a rezultatelor simulării. Problema replicilor independente: necesitatea rulării mai multor simulări. Intervale de confidență pentru indicatorii simulați (media, deviația standard). Analiza perioadei de încălzire (warm-up period) pentru simulări cu stare staționară: metoda Welch. Tehnici de replicare pentru simulări cu evenimente rare. Planificarea experimentelor de simulare: design factorial, analiza ANOVA.	2	Prelegere interactivă	
Tema 13: Emularea și simularea hardware. Emularea vs. simularea hardware: diferențe conceptuale. Simulatoarele de arhitecturi procesor: gem5, QEMU. Simulatoarele de rețele: NS-3, GNS3. Simulatoarele de sisteme embedded: Wokwi (Arduino/ESP32 în browser). Simulatoarele de circuite: SPICE, LTspice. Rolul simulării în proiectarea chip-urilor: RTL simulation, functional verification.	2	Prelegere interactivă	
Tema 14: Gemeni digitali și tendințe — recapitulare. Conceptul de geamă digital (digital twin): definiție, arhitectura, ciclul de viață. Exemple industriale: gemeni digitali în producție (Siemens), în aviație (NASA), în smart cities. Simularea ca serviciu (Simulation as a Service): platforme cloud. Inteligența artificială și simularea: surrogate models, neural ODEs, physics-informed neural networks. Recapitulare generală.	2	Prelegere, recapitulare	
	Bibliografie obligatorie 1. Law, A.M., Simulation Modeling and Analysis, 5th ed., McGraw-Hill, 2015. 2. Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M., Discrete-Event System Simulation, 5th ed., Pearson, 2009.		

	<p>3. Rubinstein, R.Y., Kroese, D.P., Simulation and the Monte Carlo Method, 3rd ed., Wiley, 2016.</p> <p>Bibliografie complementară</p> <p>4. Ogata, K., Modern Control Engineering, 5th ed., Pearson, 2010 — modelarea sistemelor de control.</p> <p>5. Wilensky, U., Rand, W., An Introduction to Agent-Based Modeling, MIT Press, 2015.</p> <p>Resurse software:</p> <p>6. SimPy Documentation — simulare cu evenimente discrete în Python: https://simpy.readthedocs.io/</p> <p>7. Mesa Documentation — simulare bazată pe agenți în Python: https://mesa.readthedocs.io/</p> <p>8. SciPy Documentation — integrare ODE și metode numerice: https://docs.scipy.org/doc/scipy/reference/integrate.html</p>		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Simularea ODE și sisteme de control. Simularea ecuației masa-arc-amortizor cu <code>scipy.integrate.solve_ivp</code> pentru metodele RK45 și Euler. Compararea erorilor în funcție de pasul de integrare pe o soluție analitică de referință. Simularea unui sistem de ordinul 2 cu regulator PID: trasarea răspunsului la treaptă cu suprareglaj, timp de stabilire și eroarea în regim permanent.	2	Lucrare practică Python	
Lucrarea L2: Metoda Monte Carlo — integrare și estimare. Estimarea integralei unei funcții bidimensionale prin Monte Carlo cu $N = 10^4, 10^5, 10^6$ puncte. Trasarea convergenței erorii vs. N . Simularea fiabilității unui sistem cu 5 componente (mix serie-paralel) cu distribuții exponențiale. Aplicarea metodei variabilelor antitetice: compararea varianței estimatorului față de Monte Carlo simplu.	2	Lucrare practică Python	
Lucrarea L3: Simularea cu evenimente discrete — SimPy. Implementarea unui sistem de așteptare M/M/1 cu SimPy: sosiri Poisson (λ), servicii exponențiale (μ). Colectarea indicatorilor: L_q, W_q , utilizarea serverului. Compararea cu formulele analitice Erlang pentru $\rho = 0.5, 0.7, 0.9$. Extinderea la M/M/3 și analiza impactului numărului de servere.	2	Lucrare practică Python, SimPy	
Lucrarea L4: Simulare bazată pe agenți — Mesa. Implementarea modelului Schelling (segregare rezidențială) cu Mesa: 2 tipuri de agenți, rețea de vecinătate Moore, regula de mutare. Vizualizarea evoluției segregării. Analiza sensibilității față de pragul de toleranță (20%, 30%, 50%). Compararea rezultatelor simulate cu predicțiile din literatura de specialitate.	2	Lucrare practică Python, Mesa	
Lucrarea L5: Lanțuri Markov și simulare stohastică. Simularea unui lanț Markov cu timp discret pentru un model de fiabilitate cu 3 stări (funcțional, degradat, defect): verificarea convergenței la distribuția staționară. Simularea mișcării browniene: $N = 100$ traiectorii, distribuția poziției la timp T . Simularea unei ecuații diferențiale stocastice cu schema Euler-Maruyama.	2	Lucrare practică Python	

Lucrarea L6: Validarea modelelor. Calibrarea unui model de simulare DES față de date reale dintr-un jurnal de sosiri (furnizat): estimarea parametrilor distribuției de sosiri și de serviciu cu MLE. Analiza goodness-of-fit cu testul chi-pătrat și Kolmogorov-Smirnov. Analiza sensibilității: varierea a 2 parametri $\pm 20\%$ și cuantificarea impactului asupra indicatorilor principali.	2	Lucrare practică Python	
Lucrarea L7: Proiect integrat și colocviu. Simularea unui sistem complex ales individual sau dintr-o listă: sistem de producție cu cozi tandem, trafic de rețea, propagarea unui epidemic (SIR), management-ul resurselor unui data center. Incluziunea obligatorie a: modelării matematice, implementării în Python (SimPy sau Mesa sau ODE), validării cu cel puțin o metodă, analizei sensibilității și raportului de incertitudine. Colocviu oral.	2	Proiect individual, colocviu	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. SimPy Documentation — simulare cu evenimente discrete: https://simpy.readthedocs.io/ 2. Mesa Documentation — simulare bazată pe agenți: https://mesa.readthedocs.io/ 3. Law, A.M., Simulation Modeling and Analysis, 5th ed., McGraw-Hill, 2015 — exercițiile practice. <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. SciPy Documentation — integrare ODE, statistici: https://docs.scipy.org/doc/scipy/ 5. Rubinstein, R.Y., Kroese, D.P., Simulation and the Monte Carlo Method, 3rd ed., Wiley, 2016. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. SimPy Documentation: https://simpy.readthedocs.io/ 7. Mesa Documentation: https://mesa.readthedocs.io/ 8. SciPy Documentation: https://docs.scipy.org/doc/scipy/ 9. NetLogo Web — simulare bazată pe agenți online, fără instalare: https://www.netlogoweb.org/ 		

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

<p>Modelarea și Simularea furnizează metodologia universală pentru înțelegerea sistemelor prea complexe sau prea costisitoare pentru a fi experimentate direct: sistemele de producție, rețelele de calculatoare, ecosistemele, sistemele financiare și sistemele de control sunt validate prin simulare înainte de implementare. Disciplina acoperă spectrul complet al tehnicilor — ODE, Monte Carlo, DES, ABM, stocastice — permițând studentului să aleagă abordarea potrivită pentru fiecare tip de problemă.</p> <p>Componenta de laborator urmărește, prin validarea obligatorie a modelelor față de date sau formule analitice de referință, formarea responsabilității pentru comunicarea transparentă a incertitudinilor — direct corelată cu R41 din matricea de corelare, competența care diferențiază un inginer riguros de unul care prezintă rezultatele simulărilor fără a discuta limitele modelului.</p>
--

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea și alegerea tipului de simulare adecvat pentru un sistem dat (ODE, DES, ABM, Monte Carlo,	Examen scris (2 ore): 1 problemă de simulare Monte Carlo (estimarea unui indicator cu N replici, calculul erorii standard și al intervalului de confidență), 1	60%

	stocastică); cunoașterea algoritmilor de integrare numerică și a criteriilor de alegere a pasului; calculul indicatorilor teoriei cozilor (L , L_q , W , W_q) pentru sisteme $M/M/c$; cunoașterea metodologiei de validare a modelelor; cunoașterea proceselor Markov și a lanțurilor cu timp discret.	problemă de teoria cozilor (calculul indicatorilor unui sistem $M/M/c$ pentru parametri dați) și 1 problemă de validare (alegerea și justificarea metodei de validare potrivite pentru un scenariu descris)	
8.5 Seminar/laborator	Corectitudinea implementărilor Python (simulările produc rezultate comparabile cu soluțiile analitice de referință sau cu date reale, cu erori justificate); calitatea analizei sensibilității din proiectul integrat (cel puțin 2 parametri variați sistematic cu interpretarea impactului); prezentarea clară a incertitudinilor și a limitelor modelului la colocviu.	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L7 (50%)	40%

8.6 Standard minim de performanță:

Includerea în proiectul integrat a unei secțiuni de validare care compară rezultatele simulării cu cel puțin o valoare de referință externă (formulă analitică, date din literatură sau date experimentale) ȘI cuantificarea discrepanței printr-o metrică specificată (RMSE, eroare relativă sau test statistic). Proiectele care prezintă rezultate de simulare fără nicio comparație cu o referință externă nu îndeplinesc standardul minim, indiferent de complexitatea implementării — această cerință reflectă direct R41 și diferențiază Modelarea și Simularea de disciplinele de programare din program.

Data completării	Semnătura titularului de curs Conf. Dr. Răileanu Alina-Beatrice	Semnătura titularului de seminar / laborator Conf. Dr. Răileanu Alina-Beatrice
16.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. Dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. Dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei	LIMBAJE FORMALE ȘI TRANSLATOARE – TIDS310						
1.2 Titularul activităților de curs	Lect. Dr. Lițan Daniela-Elena						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Lect. Dr. Lițan Daniela-Elena						
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					44
Documentare					20
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					4
2.9 Total ore pe semestru					100
2.10 Numărul de credite					4

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Matematici Discrete (TIDS205) — logică propozițională și predicativă, teoria mulțimilor, relații, grafuri; Limbaje Formale și Automate (TIDS210) — automate finite, gramatici Chomsky, ierarhia Chomsky; Structuri de Date și Algoritmi (TIDS203) — arbori, grafuri, algoritmi de parcurgere.
3.2 de competențe	Cunoașterea ierarhiei Chomsky și a corespondenței automate-gramatici (regulare, independente de context); capacitatea de a construi un NFA/DFA și de a-l minimiza; familiarizarea cu un limbaj de programare pentru implementarea structurilor de date (arbori, stive, grafuri); înțelegerea conceptului de recursivitate.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă pentru demonstrații formale, videoproiector și acces la internet; platforma Sakai; recomandare: Python 3.10+ cu bibliotecile automata-lib sau JFLAP instalate înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; Python 3.10+ cu bibliotecile automata-lib, PLY (Python Lex-Yacc) sau ANTLR4 instalate; JFLAP ca

	instrument grafic pentru vizualizarea automatelor; studenții pot folosi propriul laptop.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C10. Cunoaște limbajele formale, automatele și principiile translaatoarelor, ca fundament teoretic al proiectării compilatoarelor și al înțelegerii critice a limbajelor de programare moderne.
5.2 Aptitudini	A6. Proiectează și analizează algoritmi eficienți pentru rezolvarea problemelor computaționale complexe, evaluând critic complexitatea temporală și spațială a soluțiilor propuse și alegând structurile de date adecvate.
5.3 Responsabilitate și autonomie	R10. Evaluează critic și independent mai multe soluții tehnice alternative în fața unor probleme ingineresti deschise, selectând și argumentând abordarea optimă pe baza analizei comparative a avantajelor, riscurilor și costurilor.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Recapitulare — limbaje formale și automate. Ierarhia Chomsky: tipurile 0, 1, 2, 3. Automate finite deterministe (DFA) și nedeterministe (NFA): definiție formală, funcția de tranziție extinsă. Teorema Rabin-Scott: echivalența DFA-NFA (construcția subset). Automate cu stivă (PDA): deterministe și nedeterministe, limbajele de tip 2. Mașini Turing: variante, puterea de calcul.	2	Prelegere interactivă	
Tema 2: Expresii regulate și aplicații. Expresii regulate: sintaxa și semantica, operatorii (concatenare, reuniune, Kleene star). Lema de pompare pentru limbaje regulate: demonstrarea non-regularității. Conversia NFA la expresie regulată: algoritmul Arden, eliminarea stărilor. Minimizarea DFA: algoritmul Hopcroft. Expresii regulate în programare: PCRE (Perl Compatible Regular Expressions), motoarele NFA vs. DFA, backtracking.	2	Prelegere interactivă	
Tema 3: Gramatici independente de context (CFG). Gramatici ambigue și dezambiguizarea. Formele normale Chomsky (CNF) și Greibach (GNF): conversia, importanța în parsare. Lema de pompare pentru limbaje independente de context. Automate cu stivă și CFG: echivalența (demonstrarea prin construcție). Proprietățile de închidere ale limbajelor de tip 2.	2	Prelegere, demonstrații formale	
Tema 4: Algoritmi de parsare de sus în jos. Parsarea recursiv-descendentă: implementarea directă a unei gramatici în cod. Gramatici LL(k): condiția First și Follow, tabelul de parsare LL(1). Conflicturile LL(1): recursivitate stângă (eliminarea), ambiguitate (dezambiguizarea). Parsere predictive nerecursive: stiva explicită și tabelul de	2	Prelegere interactivă	

parsare. Limitele parsării LL(1): limbaje care nu admit gramatici LL(1).			
Tema 5: Algoritmi de parsare de jos în sus. Parsare shift-reduce: configurația (stivă, intrare), operațiile shift și reduce. Gramatici LR(0): itemii LR(0), automatele de itemi, tabelele action și goto. Gramatici SLR(1): rezolvarea conflictelor cu Follow. Gramatici LALR(1): fuzionarea stărilor LR(1), diferența față de LR(1) complet. Gramatici LR(1) complet: puterea de recunoaștere. Compararea LL(1), SLR, LALR, LR(1).	2	Prelegere interactivă	
Tema 6: Analiza lexicală. Rolul analizorului lexical (scanner/lexer) în arhitectura unui compilator. Token-uri, lexeme, pattern-uri. Construcția lexerului din expresii regulate: RE → NFA (construcția Thompson) → DFA (construcția subset) → DFA minimizat. Tratarea ambiguității: regula celei mai lungi potriviri, regula priorității. Implementarea cu tabele de tranziție vs. cod direct. Gestionarea erorilor lexicale.	2	Prelegere interactivă	
Tema 7: Analiza sintactică — implementare. Arhitectura generatorului de parsare. Instrumentul ANTLR4: gramatica ANTLR (notația EBNF extinsă), generarea parserului, arborele de parsare vs. AST. Instrumentul PLY (Python Lex-Yacc): definiția tokenilor, regulile de gramatică, acțiunile semantice. Tratarea erorilor sintactice: recuperarea prin panic mode și phrase-level recovery.	2	Prelegere, demonstrații cod	
Tema 8: Analiza semantică. Arborele sintactic abstract (AST): construcția din arborele de parsare concret. Tabela de simboluri: structura, operațiile (inserare, căutare, ștergere), gestiunea domeniilor de vizibilitate (scope). Verificarea tipurilor: sisteme de tipuri statice și dinamice, coerciția implicită, polimorfismul. Analiza semantică în compilatoarele moderne: GCC, Clang/LLVM.	2	Prelegere interactivă	
Tema 9: Reprezentări intermediare. Necesitatea codului intermediar: independența față de arhitectura țintă. Codul de trei adrese (3AC): instrucțiunile, tipurile (assign, binary op, unary op, copy, jump, call). Forma SSA (Static Single Assignment): proprietatea, construcția, avantajele pentru optimizare. Arborele sintactic adnotat. Reprezentarea LLVM IR: modulul, funcțiile, blocurile de bază, instrucțiunile. Compilatorul JIT și interpretarea bytecode.	2	Prelegere interactivă	
Tema 10: Optimizarea codului. Optimizări locale: eliminarea subexpresiilor comune (CSE), propagarea constantelor, folding-ul constantelor, eliminarea codului mort. Optimizări globale: analiza fluxului de date	2	Prelegere interactivă	

(data flow analysis), reaching definitions, live variable analysis. Optimizări pe buclă: code motion, strength reduction, loop unrolling. Optimizări interprocedural: inlining, analiza efectelor secundare.			
Tema 11: Generarea codului. Selecția instrucțiunilor: tree pattern matching, algoritmul BURG. Alocarea registrelor: problema colorării grafului de interferențe (graph coloring), spilling. Planificarea instrucțiunilor (instruction scheduling): exploatarea paralelismului la nivel de instrucțiune (ILP). Generarea codului pentru arhitecturi RISC vs. CISC. Vectorizarea automată: SIMD code generation.	2	Prelegere interactivă	
Tema 12: Teorii ale calculabilității. Problema opririi (Halting Problem): demonstrarea indecidabilității prin diagonalizare. Teorema Rice: orice proprietate non-trivială a funcțiilor Mașinii Turing este indecidabilă. Reduceri: M-reduceri, T-reduceri. Funcții calculabile Turing: teza Church-Turing. Complexitatea computațională: clasele P și NP (revizitare), PSPACE, EXPTIME. Conexiunea cu limbajele formale: ierarhia complexității vs. ierarhia Chomsky.	2	Prelegere interactivă	
Tema 13: Transductoare și aplicații. Transductoare finite (FST): definiție, compunerea transductoarelor. Aplicații NLP: morfologia computațională (analiza și generarea morfologică), normalizarea textului, corectarea ortografică cu distanța Levenshtein. Transductoarele în compilatoare: relocarea adreselor, linking-ul. Gramatici de atribute: gramatici L-atribuite și S-atribuite, evaluarea top-down și bottom-up.	2	Prelegere interactivă	
Tema 14: Limbaje de domeniu specific (DSL) și recapitulare. Limbajele DSL interne (embedded DSL) vs. externe. Exemple de DSL-uri: SQL, regex, Make, Dockerfile, YAML/JSON Schema. Tehnici de implementare a DSL-urilor externe: parsere cu ANTLR4, combinatori de parsere. Metaprogramarea și generatoarele de cod: Jinja2, code generation în LLVM. Recapitulare generală.	2	Prelegere, recapitulare	
	Bibliografie obligatorie 1. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D., Compilers: Principles, Techniques, and Tools (Dragonul Roșu), 2nd ed., Addison-Wesley, 2006. 2. Sipser, M., Introduction to the Theory of Computation, 3rd ed., Cengage, 2012. 3. Hopcroft, J.E., Motwani, R., Ullman, J.D., Introduction to Automata Theory, Languages, and Computation, 3rd ed., Pearson, 2006. Bibliografie complementară 4. Cooper, K., Torczon, L., Engineering a Compiler, 3rd ed., Morgan Kaufmann, 2022. 5. Parr, T., The Definitive ANTLR4 Reference, 2nd ed., Pragmatic Bookshelf, 2013. Resurse software: 6. ANTLR4 Documentation și tutoriale: https://wwwantlr.org/ 7. JFLAP — instrument grafic pentru automate și gramatici: https://www.jflap.org/		

6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: DFA, NFA și minimizare. Construirea unui DFA pentru limbajul $\{w \in \{a,b\}^* \mid w \text{ conține subsirul } aba \text{ și simulara pas cu pas. Construcția NFA echivalent și conversia subset la DFA. Minimizarea DFA cu algoritmul Hopcroft. Verificarea cu JFLAP: importarea automatelor și rularea testelor.}$	2	Lucrare practică JFLAP, Python	
Lucrarea L2: Expresii regulate și aplicații. Conversia manuală $RE \rightarrow NFA$ (construcția Thompson) pentru 3 expresii regulate. Verificarea cu automata-lib (Python). Scrierea expresiilor regulate Python (re module) pentru: validarea unui email, extragerea URL-urilor dintr-un text HTML, parsarea unui fișier CSV simplu. Analiza comportamentului backtracking al motorului regex.	2	Lucrare practică Python	
Lucrarea L3: Gramatici CFG și forma normală Chomsky. Construirea gramaticii CFG pentru expresii aritmetice (cu priorități și paranteze). Conversia la CNF: eliminarea producțiilor ϵ , unitare și cu mai mult de 2 simboluri. Verificarea că limbajul generat este același. Demonstrarea ambiguității și propunerea de dezambiguizare.	2	Lucrare practică, demonstrații formale	
Lucrarea L4: Parsare LL(1). Construirea tabelului First și Follow pentru o gramatică LL(1) dată (expresii, instrucțiuni if-else). Construirea tabelului de parsare LL(1) și simularea parserului pe un șir de intrare. Identificarea unui conflict LL(1) (recursivitate stângă) și eliminarea lui. Implementarea unui parser recursiv descendent simplu în Python.	2	Lucrare practică Python	
Lucrarea L5: Parsare SLR și LALR. Construirea automatelor de itemi LR(0) pentru o gramatică simplă. Construirea tabelelor SLR(1) și simularea parserului shift-reduce. Identificarea unui conflict shift-reduce și rezolvarea prin reguli de prioritate. Compararea tabelelor SLR și LALR pe un exemplu simplu.	2	Lucrare practică, demonstrații	
Lucrarea L6: Analizor lexical cu PLY. Implementarea unui scanner complet cu PLY (Python Lex-Yacc) pentru un limbaj de programare simplu: tokenuri pentru cuvinte cheie, identificatori, numere întregi și reale, operatori, comentarii. Testarea pe programe sursă de test. Gestionarea erorilor lexicale și raportarea poziției (linie, coloană).	2	Lucrare practică Python, PLY	
Lucrarea L7: Analizor sintactic cu PLY. Extinderea proiectului din L6 cu un parser PLY pentru expresii aritmetice și instrucțiuni de asignare. Definirea regulilor de precedență și asociativitate a operatorilor. Construirea	2	Lucrare practică Python, PLY	

unui AST simplu. Testarea pe programe sursă care conțin erori sintactice.			
Lucrarea L8: Compilator complet cu ANTLR4 — lexer și parser. Scrierea unei gramatici ANTLR4 pentru un mini-limbaj de programare: variabile, expresii aritmetice, instrucțiuni if-else, while, funcții. Generarea lexerului și parserului. Vizualizarea arborelui de parsare cu grun. Adaptarea gramaticii pentru a elimina ambiguitățile.	2	Lucrare practică Python/Java, ANTLR4	
Lucrarea L9: Visitor și AST cu ANTLR4. Implementarea unui visitor ANTLR4 care construiește un AST simplificat din arborele de parsare concret. Implementarea unui al doilea visitor care evaluează expresii aritmetice cu variabile (tabela de simboluri simplă ca dicționar Python). Testarea pe programe cu expresii complexe.	2	Lucrare practică Python, ANTLR4	
Lucrarea L10: Verificarea tipurilor și tabela de simboluri. Extinderea compilatorului cu un visitor de verificare a tipurilor: detectarea variabilelor nedeclarate, a operațiilor cu tipuri incompatibile, a variabilelor declarate de mai multe ori. Implementarea gestiunii domeniilor de vizibilitate (scopes) cu o stivă de tabele de simboluri.	2	Lucrare practică Python, ANTLR4	
Lucrarea L11: Generarea codului intermediar. Implementarea unui visitor care generează cod de trei adrese (3AC) dintr-un AST. Generarea 3AC pentru: expresii aritmetice, asignări, instrucțiuni if-else (cu etichete și salturi), bucle while. Afișarea și verificarea codului generat pe exemple de programe.	2	Lucrare practică Python	
Lucrarea L12: Optimizări locale. Implementarea în Python a optimizărilor de propagare a constantelor și de folding al constantelor pe reprezentarea 3AC. Implementarea eliminării codului mort (instrucțiuni cu rezultat neutilizat). Compararea codului 3AC înainte și după optimizare pentru 3 programe de test.	2	Lucrare practică Python	
Lucrarea L13: DSL — implementare completă cu ANTLR4. Proiectarea și implementarea unui DSL simplu la alegere: un limbaj de configurare, un limbaj de interogare simplificat sau un limbaj de descriere a regulilor de validare. Gramatica ANTLR4, parserul, interpretatorul (visitor) și testele. Justificarea deciziilor de proiectare ale limbajului.	2	Lucrare practică Python, ANTLR4	
Lucrarea L14: Proiect integrat și colocviu. Prezentarea unui compilator complet pentru mini-limbajul definit pe parcursul seminarelor (L6-L12): lexer, parser, AST, verificare tipuri, generare 3AC, optimizare locală. Demonstrarea live a compilării unui program	2	Proiect individual, colocviu	

de test. Colocviu oral: justificarea deciziilor de proiectare (tipul de parser ales, structura AST, strategia de generare a codului).			
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Parr, T., The Definitive ANTLR4 Reference, 2nd ed., Pragmatic Bookshelf, 2013. 2. ANTLR4 Documentation: https://www.antlr.org/ 3. PLY (Python Lex-Yacc) Documentation: https://www.dabeaz.com/ply/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. Aho, A.V. et al., Compilers: Principles, Techniques, and Tools, 2nd ed., Addison-Wesley, 2006 — exercițiile practice. 5. JFLAP Documentation — instrument grafic pentru automate: https://www.jflap.org/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. ANTLR4 Documentation: https://www.antlr.org/ 7. JFLAP — instrument grafic pentru automate și gramatici: https://www.jflap.org/ 8. PLY Documentation — Python Lex-Yacc: https://www.dabeaz.com/ply/ 9. Regex101 — testarea expresiilor regulate cu explicații: https://regex101.com/ 		

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

Limbae Formale și Translatoare furnizează fundamentul teoretic și practic al construcției compilatoarelor — disciplina care explică de ce un limbaj de programare este proiectat într-un anumit fel, ce limitări matematice există în procesarea limbajelor și cum se transformă codul sursă în instrucțiuni executabile. Înțelegerea acestor fundamente este esențială pentru orice inginer software care proiectează DSL-uri, scrie parsere sau lucrează cu toolchain-uri de compilare. Componenta de laborator urmărește construirea pas cu pas a unui compilator funcțional pentru un mini-limbaj, integrând toate etapele de la analiza lexicală la generarea codului intermediar. Justificarea deciziilor de proiectare la colocviu — de ce s-a ales un anumit tip de parser, cum s-au rezolvat conflictele gramaticale — cultivă direct capacitatea de evaluare critică a alternativelor tehnice, corelată cu R10 din matricea de corelare.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea și aplicarea construcțiilor formale pentru DFA, NFA, CFG, PDA; demonstrarea non-regularității/non-independenței de context cu lema de pompare; construcția tabelor de parsare LL(1) și SLR(1) cu identificarea și rezolvarea conflictelor; cunoașterea etapelor compilatorului și a reprezentărilor intermediare; cunoașterea teoremei Rice și a problemei opririi.	Examen scris (2 ore): 1 problemă de automate formale (construcția DFA + minimizare sau demonstrarea non-regularității cu lema de pompare), 1 problemă de parsare (construcția tabelii LL(1) sau SLR(1) cu identificarea unui conflict și rezolvarea sa) și 1 problemă de compilator (generarea codului 3AC pentru o secvență de cod sursă dată)	50%
8.5 Seminar/laborator	Funcționalitatea compilatorului integrat din L14 (lexer, parser, AST, verificare tipuri,	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și	50%

	generare 3AC funcționale pe programele de test); corectitudinea automatelor și a gramaticilor din lucrările L1-L5; calitatea justificărilor tehnice la colocviu (argumentarea tipului de parser ales, explicarea conflictelor rezolvate și a deciziilor de proiectare a limbajului).	proiect integrat cu colocviu oral individual la L14 (50%)	
--	---	--	--

8.6 Standard minim de performanță:

La colocviu: justificarea orală a tipului de parser implementat — de ce LL(1) sau LALR și nu alternativa — cu menționarea a cel puțin unui conflict gramatical întâlnit și a modului în care a fost rezolvat. La examenul scris: construcția corectă a unui automat formal sau a unui tabel de parsare, cu toate câmpurile completate și conflictele identificate — nu doar enunțarea metodei, ci aplicarea ei completă pe exemplul dat.

Data completării 17.03.2026	Semnătura titularului de curs Lițan Daniela-Elena	Semnătura titularului de seminar / laborator Lițan Daniela-Elena
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei		ARHITECTURI PENTRU PRELUCRARE NUMERICĂ DE SEMNAL – TIDS311					
1.2 Titularul activităților de curs		Conf. Dr. Lacrămă Dan-Laurențiu					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Conf. Dr. Lacrămă Dan-Laurențiu					
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Achiziția și Prelucrarea Datelor (TIDS304) — eșantionare, FFT, filtre FIR/IIR, filtrare Kalman; Arhitectura Sistemelor de Calcul (TIDS214) — organizarea procesorului, pipeline, memorie cache, bus; Electronică Digitală (TIDS209) — aritmetica binară, circuite combinate și secvențiale, FPGA.
3.2 de competențe	Cunoașterea transformatei Fourier și a algoritmului FFT; proiectarea filtrelor FIR și IIR cu scipy.signal; înțelegerea arhitecturii pipeline a unui procesor generic (fetch, decode, execute, writeback); familiarizarea cu aritmetica în virgulă fixă și virgulă mobilă (IEEE 754).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sala de curs cu tabla pentru demonstrații, videoproiector și acces la internet; platforma Sakai; recomandare: Python 3.10+ cu NumPy, SciPy, Matplotlib și MATLAB Online instalate înainte de curs.
-------------------------------	--

4.2 de desfășurare a seminarului/laboratorului	Laborator de informatica cu calculatoare individuale; Python 3.10+ cu NumPy, SciPy, Matplotlib, fixedpoint sau fxpmath; MATLAB cu DSP System Toolbox sau MATLAB Online; acces la simulator QEMU ARM sau placa DSP/FPGA de evaluare (TMS320C6748 sau Zynq) acolo unde este disponibila; studentii pot folosi propriul laptop.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C4. Deține cunoștințe specializate privind arhitecturile sistemelor de calcul, organizarea procesoarelor, ierarhia memoriei și interfețele hardware-software, inclusiv tendințele actuale în proiectarea procesoarelor de semnal digital (DSP). C5. Cunoaște în profunzime principiile proiectării logice și ale electronicii digitale, inclusiv arhitecturile pentru prelucrare numerică de semnal, situate la avangarda domeniului.
5.2 Aptitudini	A4. Elaborează documentație tehnică completă și clară — specificații, diagrame de flux, manuale de utilizare — în conformitate cu standardele ingineresti și cerințele legale aplicabile.
5.3 Responsabilitate și autonomie	R4. Acționează autonom în selectarea și utilizarea tehnologiilor, framework-urilor și instrumentelor software adecvate contextului tehnic, justificând critic alegerile făcute în fața echipei și a beneficiarilor.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în prelucrarea numerica a semnalelor. Semnale continue vs. discrete, esantionare si reconstructie (Shannon-Nyquist). Sisteme LTI discrete: raspuns la impuls, convolutia discreta, ecuatia cu diferente. Transformata Z: definitie, proprietati, ROC (Region of Convergence), transformate uzuale. Relatia transformata Z — transformata Fourier la timp discret (DTFT). Stabilitatea sistemelor discrete.	2	Prelegere interactiva	
Tema 2: Transformata Fourier discreta si FFT. DFT: definitie, proprietati (liniaritate, deplasare circulara, teorema convolutiei circulare). Algoritmul FFT Cooley-Tukey: decimare in timp (DIT), decimare in frecventa (DIF), complexitatea $O(N \log N)$ vs. $O(N^2)$. Zero-padding si rezolutia spectrala. Efecte de spectral leakage si ferestre (Hanning, Hamming, Blackman). Implementarea FFT pe arhitecturi cu paralelism.	2	Prelegere interactiva	
Tema 3: Filtre numerice FIR avansate. Proiectarea FIR prin fereastra: specificatiile (frecventa de taiere, atenuarea benzii de oprire) → alegerea ferestrei → calculul coeficientilor. Faza liniara a filtrelor FIR: importanta pentru aplicatii audio si comunicatii. Proiectarea FIR prin esantionarea frecventei (frequency sampling). Filtre FIR multistadiu: decimare si interpolare (polyphase decomposition). Complexitatea	2	Prelegere interactiva	

computatională: număr de multiplicări și adunări per esanțion.			
Tema 4: Filtre numerice IIR avansate. Proiectarea IIR prin transformare bilineară: prewarping, prototipul analogic Butterworth, Chebyshev I și II, Eliptic. Forma de implementare: directă I, directă II, cascada de secții de ordinul 2 (biquad), paralelă. Sensibilitatea coeficienților la cuantizare: compararea formelor. Implementarea eficientă a biquad-ului pe DSP: acumulator pe 40 de biți în TMS320.	2	Prelegere interactivă	
Tema 5: Aritmetică în virgulă fixă. Reprezentarea în virgulă fixă: formatul Q, Q15, Q31; rangul dinamic, rezoluția. Overflow și saturare: moduri de gestionare. Eroarea de cuantizare a coeficienților: efectul asupra răspunsului în frecvență al filtrului. Eroarea de rotunjire a produselor (round-off noise): analiză statistică. Cicli limita (limit cycles): tipuri (granularity, overflow), condiții de apariție, eliminare. Compararea virgulă fixă vs. virgulă mobilă: viteză, cost, energie.	2	Prelegere interactivă	
Tema 6: Arhitectura procesoarelor DSP. Cerințe specifice DSP: suma produselor (MAC) în un ciclu, adresare în mod circular, adresare bit-reversed pentru FFT. Arhitectura Harvard modificată: memorie de date separată de memoria de program, magistrale paralele. Unitatea MAC pe 32x32 biți cu acumulator pe 40/64 biți. Generatorul de adrese (AGU). Unități de execuție paralele: VLIW (Very Long Instruction Word) în TMS320C6000. Pipeline DSP: conflicte, latență, slot-uri de delay.	2	Prelegere interactivă	
Tema 7: Familia TMS320 Texas Instruments. Evoluția familiei TMS320: C2000 (control), C5000 (putere mică), C6000 (performantă). Arhitectura TMS320C6748: nucleul C674x, unitățile funcționale (.L, .S, .M, .D × 2), registrele, fisierul de registre. Setul de instrucțiuni: instrucțiuni de bază, instrucțiuni SIMD (64-bit), instrucțiuni de control al fluxului. Memoria: memorie internă (L1D, L1P, L2), interfața externă (DDR3, EMIF). Periferice: McASP, McBSP, USB, Ethernet.	2	Prelegere interactivă	
Tema 8: Programarea DSP în C și asamblare. Scrierea codului C optimizat pentru DSP: tipurile de date intrinseci (int16, int32, int64), intrinseciurile compilatorului (intrinsics) pentru TMS320. Directivele pragma: DATA_ALIGN, MUST_ITERATE, UNROLL. Optimizarile compilatorului: loop unrolling, software pipelining. Profilarea codului: cicluri per esanțion, cicluri per cadru. Asamblarea liniară: structura, instrucțiunile de	2	Prelegere interactivă	

baza, paralelismul explicit. Interfata C-asamblare.			
Tema 9: Procesoare DSP multi-core si SIMD. Necesitatea paralelismului: limitele unui singur nucleu DSP. Arhitecturi multi-core DSP: TMS320C6670 (8 nuclee C66x), comunicatia inter-core (MSMC, semafore hardware, EDMA3). Vectorizarea SIMD: instructiunile de 64-biti pe C6000 (DADD, DMPY32, etc.). Arhitectura DSP+ARM (heterogenea): TDA2x (ADAS), Keystone II. Partitionarea algoritmilor intre ARM si DSP.	2	Prelegere interactiva	
Tema 10: FPGA pentru prelucrarea numerica a semnalelor. Arhitectura FPGA: LUT-uri, flip-flop-uri, blocuri DSP (multiplicatoare integrate), memorii BRAM, interfete I/O. Avantajele FPGA fata de DSP: paralelism masiv, latentia determinista, configurabilitate. Dezavantajele: consum mai mare de energie, programare complexa. Implementarea filtrului FIR pe FPGA: arhitectura paralela (latentia minima) vs. arhitectura seriata (suprafata minima). IP core-uri DSP Xilinx: FIR Compiler, FFT IP.	2	Prelegere interactiva	
Tema 11: Limbaje de descriere hardware pentru DSP. VHDL si Verilog: nivel RTL, sinteza. Limbaje de nivel inalt (HLS): Xilinx Vitis HLS (C/C++ → RTL), Intel HLS Compiler. Directivele HLS: PIPELINE, UNROLL, ARRAY_PARTITION — impactul asupra latentei si suprafetei. Implementarea FFT in HLS: streaming vs. burst. Avantajele HLS: portabilitate, timp de proiectare redus. Limitarile HLS: calitatea sintezei vs. RTL manual.	2	Prelegere interactiva	
Tema 12: GPU pentru prelucrarea semnalelor. Arhitectura GPU: SM (Streaming Multiprocessor), CUDA core-uri, memoria ierarhica (registre, shared, global). Modelul de executie CUDA: grid, bloc, thread, warp. Implementarea FFT pe GPU: cuFFT (NVIDIA). Convolutia pe GPU: algoritmul in domeniul frecventa cu cuFFT, algoritmul direct cu shared memory. Comparatia GPU vs. DSP vs. FPGA: throughput, latentia, energie, costul programarii.	2	Prelegere interactiva	
Tema 13: Aplicatii avansate DSP. Procesarea audio in timp real: filtrare, egalizare, reverberatie (algoritmi Schroeder, FDN), anulara ecoului acustic (AEC cu LMS). Comunicatii digitale: filtrele de emisie-receptie (root raised cosine), modulatia OFDM — implementarea pe DSP, sincronizarea. Radarul digital: matched filter, CFAR (Constant False Alarm Rate).	2	Prelegere interactiva	

Compresia audio MPEG/MP3: MDCT, cuantizarea psihoacustica.			
Tema 14: Tendinte si recapitulare. Procesoare DSP dedicate AI: NPU (Neural Processing Unit) — arhitecturi (tensori, MAC array). Edge AI: NVIDIA Jetson, Google Coral TPU, ARM Ethos. DSP in automotive ADAS: procesarea radarului si lidarului. Standardele de programare portabila: OPENCL, SYCL. Tendinte: heterogeneous computing, energy-harvesting DSP pentru IoT. Recapitulare generala.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <p>1. Proakis, J.G., Manolakis, D.G., Digital Signal Processing: Principles, Algorithms and Applications, 4th ed., Pearson, 2006.</p> <p>2. Smith, S.W., The Scientist and Engineer's Guide to Digital Signal Processing, 2nd ed., California Technical Publishing, 1999. Disponibil gratuit la: https://www.dspguide.com/</p> <p>3. Texas Instruments, TMS320C6000 DSP CPU and Instruction Set Reference Guide. Disponibil la: https://www.ti.com/lit/ug/spru189/spru189.pdf</p> <p>Bibliografie complementară</p> <p>4. Oppenheim, A.V., Schafer, R.W., Discrete-Time Signal Processing, 3rd ed., Pearson, 2009.</p> <p>5. Xilinx/AMD, Vitis High-Level Synthesis User Guide (UG1399). Disponibil la: https://docs.xilinx.com/r/en-US/ug1399-vitis-hls</p> <p>Resurse software:</p> <p>6. DSPGuide — carte gratuita de DSP cu exemple: https://www.dspguide.com/</p> <p>7. MathWorks DSP System Toolbox Documentation: https://www.mathworks.com/help/dsp/</p> <p>8. SciPy Signal Processing — referinta: https://docs.scipy.org/doc/scipy/reference/signal.html</p>		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: FFT si analiza spectrala. Implementarea FFT in Python cu numpy.fft pe semnale sinusoidale compuse si cu zgomot. Compararea spectrului cu si fara fereastră Hanning. Demonstrarea efectului zero-padding asupra rezolutiei spectrale. Calculul DTFT teoretic si verificarea cu DFT pentru N mic. Masurarea timpului de executie FFT vs. DFT naiv pentru $N = 2^{10}$ si 2^{14} .	2	Lucrare practica Python	
Lucrarea L2: Proiectarea filtrelor numerice si efectele cuantizarii. Proiectarea unui filtru FIR trece-jos cu fereastră Kaiser si a unui filtru IIR Butterworth de acelasi ordin cu scipy.signal. Compararea raspunsului in frecventa, a fazei si a latentei. Cuantizarea coeficientilor filtrului IIR la Q15 (16 biti) si Q8 (8 biti) cu biblioteca fxpmath. Masurarea degradarii raspunsului in frecventa dupa cuantizare.	2	Lucrare practica Python	
Lucrarea L3: Implementarea eficienta a filtrelor — multistadiu si polyphase. Proiectarea unui lant de decimare cu factor 8: compararea unui singur filtru FIR de decimare cu 8 vs. 3 filtre in cascada (decimare 2-2-2). Masurarea timpului de executie si a numarului de multiplicari per esantion pentru ambele variante. Implementarea unui filtru polyphase cu scipy.signal.upfirdn.	2	Lucrare practica Python	

<p>Lucrarea L4: Profilarea si optimizarea codului C pe arhitectura DSP simulata. Implementarea unui filtru FIR de lungime 64 in C pur (bucla simpla). Compilarea cu optimizarile -O0, -O2, -O3 si masurarea ciclurilor cu perf sau rdts. Rescrierea cu SIMD intrinsics (SSE2/NEON daca disponibil) sau cu unrolling manual. Compararea profilele de executie. Redactarea unui raport tehnic cu diagrama fluxului de date.</p>	2	Lucrare practica C, profilare	
<p>Lucrarea L5: Implementarea FFT pe FPGA (simulare HLS). Implementarea unui butterfly FFT de 8 puncte in C++ cu directive HLS (PIPELINE, UNROLL). Sinteza HLS in Vitis HLS sau intr-un simulator echivalent. Analiza raportului de sinteza: latentă (cicluri), initiation interval, resurse (LUT, DSP48, BRAM). Compararea arhitecturii pipeline cu arhitectura iterativa.</p>	2	Lucrare practica HLS/simulare	
<p>Lucrarea L6: Aplicatie DSP in timp real — egalizator audio. Implementarea unui egalizator grafic pe 5 benzi cu 5 filtre IIR biquad in cascada in Python (sectii Peaking EQ la 60 Hz, 250 Hz, 1 kHz, 4 kHz, 12 kHz). Procesarea unui fisier audio WAV in timp real cu pyaudio. Masurarea timpului de procesare per cadru si verificarea incadrării in bugetul de timp real. Documentarea tehnica: diagrama bloc, specificatiile fiecarui filtru.</p>	2	Lucrare practica Python, audio	
<p>Lucrarea L7: Proiect integrat si colocviu. Proiectarea si implementarea completa a unui sistem DSP la alegere: detector de tonuri DTMF (Goertzel), anulator de ecou simplu (LMS), modulatie/demodulatie BPSK, sau procesarea unui semnal radar (matched filter). Documentatia tehnica: specificatii, arhitectura aleasa (DSP/FPGA/GPU) cu justificarea critica, implementarea Python, profilarea performantei, limitele solutiei. Colocviu oral.</p>	2	Proiect individual, colocviu	

Bibliografie obligatorie laborator

1. Proakis, J.G., Manolakis, D.G., Digital Signal Processing, 4th ed., Pearson, 2006 — exercitiile practice.
2. SciPy Signal Processing Documentation: <https://docs.scipy.org/doc/scipy/reference/signal.html>
3. DSPGuide — carte gratuita cu exemple si cod: <https://www.dspguide.com/>

Bibliografie complementara laborator

4. Oppenheim, A.V., Schaffer, R.W., Discrete-Time Signal Processing, 3rd ed., Pearson, 2009.
5. Texas Instruments DSP Documentation — referinte TMS320: <https://www.ti.com/microcontrollers-mcus/processors/digital-signal-processors/overview.html>

Resurse online gratuite recomandate

6. DSPGuide — carte gratuita DSP: <https://www.dspguide.com/>
7. SciPy Signal Processing: <https://docs.scipy.org/doc/scipy/reference/signal.html>
8. MathWorks DSP System Toolbox: <https://www.mathworks.com/help/dsp/>
9. Vitis HLS User Guide (UG1399): <https://docs.xilinx.com/t/en-US/ug1399-vitis-hls>

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

Arhitecturi pentru Prelucrare Numerică de Semnal integrează cunoașterea algoritmilor DSP cu constrângerile hardware reale: un filtru optim teoretic devine inutilizabil dacă depășește bugetul de cicluri sau produce cicuri limită din cauza cuantizării coeficienților. Această disciplină formează capacitatea de a co-proiecta algoritmul și arhitectura — competența esențială în domenii precum audio, comunicații, ADAS și edge AI.

Componenta de laborator urmărește, prin documentația tehnică cerută la fiecare lucrare și prin justificarea alegerii arhitecturii la colocviu, formarea directă a competenței A4 (elaborarea documentației tehnice în conformitate cu standardele ingineresti) și R4 (justificarea critică a alegerii tehnologiei în fața evaluatorului).

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea arhitecturii procesoarelor DSP (MAC, AGU, pipeline, Harvard modificată, VLIW); cunoașterea efectelor cuantizării coeficienților și a ciclurilor limită în implementările în virgulă fixă; cunoașterea algoritmului FFT Cooley-Tukey (DIT/DIF) și a complexității $O(N \log N)$; cunoașterea criteriilor de alegere a arhitecturii (DSP vs. FPGA vs. GPU) în funcție de latență, putere și cost; cunoașterea directivelor HLS pentru sinteza de filtre pe FPGA.	Examen scris (2 ore): 1 problemă de proiectare DSP (specificarea unui filtru IIR în virgulă fixă cu calculul formatului Q necesar și identificarea riscului de overflow sau ciclu limită), 1 problemă de arhitectură (calculul ciclurilor necesare pentru un filtru FIR de lungime N pe o arhitectură DSP cu MAC și pipeline descrise) și 1 problemă comparativă (alegerea și justificarea arhitecturii optime dintre DSP, FPGA și GPU pentru un scenariu dat)	50%
8.5 Seminar/laborator	Corectitudinea implementărilor Python (filtrele produc răspunsul în frecvență corect, FFT-ul acordă cu DFT-ul de referință); calitatea documentației tehnice a lucrării L4 și a proiectului integrat (diagrama bloc, specificații, profilul de performanță măsurat); calitatea justificării alegerii arhitecturii la colocviu (argumentarea critică bazată pe latență,	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L7 (50%)	50%

	putere si complexitate de implementare).		
8.6 Standard minim de performanță:			
La colocviul din L7: justificarea alegerii arhitecturii (DSP, FPGA sau GPU) pentru aplicatia implementata cu cel putin doua argumente cantitative — de exemplu, numarul de multiplicari per esantion si latentia maxima acceptabila — nu doar preferinte subiective. La examenul scris: calculul corect al formatului Q necesar pentru coeficientii unui filtru dat, cu identificarea explicita a riscului de overflow si a compromisului intre rezolutie si rang dinamic.			

Data completării	Semnătura titularului de curs Conf. Dr. Lacrămă Dan Laurențiu	Semnătura titularului de seminar / laborator Conf. Dr. Lacrămă Dan Laurențiu
18.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. Dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. Dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei	REȚELE LOCALE DE CALCULATOARE – TIDS312						
1.2 Titularul activităților de curs	Lect. Dr. Damian Felicia Anișoara						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Lect. Dr. Damian Felicia Anișoara						
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					44
Documentare					15
Studiu individual					15
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					4
2.9 Total ore pe semestru					100
2.10 Numărul de credite					4

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Rețele de Calculatoare (TIDS301) — modelele OSI/TCP-IP, adresare IPv4/IPv6, rutare, protocoale TCP/UDP, securitate de bază; Arhitectura Sistemelor de Calcul (TIDS214) — interfețe hardware; Sisteme de Operare (TIDS302) — networking în Linux/Windows.
3.2 de competențe	Configurarea adreselor IP statice și dinamice (DHCP); cunoașterea funcționării switch-ului și a VLAN-urilor de bază; familiarizarea cu Cisco Packet Tracer și Wireshark; abilitatea de a depana conectivitatea de bază cu ping, traceroute și ipconfig/ifconfig.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă, videoproiector și acces la internet; platforma Sakai; recomandare: Cisco Packet Tracer instalat și cont Cisco NetAcad activ înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; Cisco Packet Tracer instalat; Wireshark instalat; unde este disponibil, echipamente Cisco

	fizice (switch Catalyst, router ISR) pentru lucrările avansate; studenții pot folosi propriul laptop.
--	---

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C11. Deține cunoștințe specializate privind arhitecturile rețelelor de calculatoare locale și extinse, protocoalele de comunicații (TCP/IP, OSI) și tehnologiile de transmisiune a datelor, cu conștientizarea evoluțiilor recente în domeniu.
5.2 Aptitudini	A11. Proiectează, configurează și administrează rețele de calculatoare locale și extinse (LAN/WAN), implementând protocoale de comunicații, topologii de rețea și soluții de management al traficului de date.
5.3 Responsabilitate și autonomie	R11. Își asumă responsabilitatea pentru migrarea și integrarea sistemelor informatice complexe în infrastructuri cloud sau distribuite, gestionând riscurile tehnice asociate și asigurând continuitatea serviciilor pe parcursul tranziției.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Arhitectura rețelelor LAN enterprise. Modelul ierarhic Cisco: nivelul access, distribution, core. Campus LAN vs. data center LAN. Echipamentele nivelurilor: switch-uri de acces (L2), switch-uri de distribuție (L3), switch-uri de core. Selectarea echipamentelor: criterii tehnice și economice. Evoluția arhitecturii: de la rețele cu hub-uri la switch-uri inteligente L3.	2	Prelegere interactivă	
Tema 2: Switching avansat. Arhitectura internă a switch-ului: planul de date (ASIC, CAM, TCAM), planul de control, planul de management. Tabela CAM: aging, dimensionare. MAC flooding și protecție. Port mirroring (SPAN/RSPAN) pentru monitorizare. Switch-uri stivuibile (stackable): Cisco StackWise, VSS. Forwarding: store-and-forward vs. cut-through.	2	Prelegere interactivă	
Tema 3: VLAN avansat. Revizuire VLAN 802.1Q. VLAN native și securitatea sa. VTP (VLAN Trunking Protocol): moduri server/client/transparent, riscuri. Voice VLAN (VLAN): configurare pentru telefonie IP, CoS. Private VLAN (PVLAN): primar, secundar izolat/community. VLAN Management: best practices — separarea VLAN-ului de management.	2	Prelegere interactivă	
Tema 4: Spanning Tree Protocol avansat. Revizuire STP (802.1D): alegerea root bridge, porturile root/designated/blocked. PVST+ și Rapid PVST+ (802.1w): convergența rapidă, portTypes (edge, root, non-edge). Multiple Spanning Tree (MST) — 802.1s: instanțe, maparea VLAN-uri la instanțe. PortFast,	2	Prelegere interactivă	

BPDUGuard, BPDUFilter, RootGuard, LoopGuard. Atacuri STP și contracarare.			
Tema 5: EtherChannel și redundanța legăturilor. EtherChannel (Link Aggregation — 802.3ad/LACP, PAgP): principii, avantaje, cerințe. Configurarea EtherChannel pe switch-uri Cisco: mod active/passive (LACP), desirable/auto (PAgP). Load balancing: algoritmi (src-mac, dst-mac, src-dst-ip). EtherChannel și STP: reducerea topologiei STP. Depanare EtherChannel: cauze frecvente de erori.	2	Prelegere interactivă	
Tema 6: Rutare inter-VLAN avansată și switch-uri L3. Revizuire router-on-a-stick. Switch L3 (multi-layer): configurarea SVI (Switched Virtual Interface), interfețe rutate. Rutarea OSPF și EIGRP pe switch-uri L3. Hot Standby Router Protocol (HSRP): active/standby, preemption, versiunile 1 și 2. VRRP (Virtual Router Redundancy Protocol). Comparatie HSRP vs. VRRP vs. GLBP.	2	Prelegere interactivă	
Tema 7: Securitatea rețelelor LAN. Port security: maximum MAC, violate modes (shutdown, protect, restrict). DHCP Snooping: trusted/untrusted ports, binding table. Dynamic ARP Inspection (DAI): validarea pachetelor ARP față de binding table DHCP. IP Source Guard: filtrarea pe baza surselor IP autorizate. 802.1X Network Access Control (NAC): RADIUS, EAP, autentificarea porturilor. Atacuri comune la nivel L2 și contracarare.	2	Prelegere interactivă	
Tema 8: QoS în rețelele LAN. Necesitatea QoS: voce, video, date — sensibilitate diferită la latență, jitter, pierderi. Modelele QoS: IntServ (RSVP), DiffServ (DSCP). Câmpurile de marcare: CoS (802.1p în header 802.1Q), DSCP (câmpul DSCP din IP header). Clasificarea și marcarea traficului la access layer. Mecanisme de coadă: FIFO, WFQ, CBWFQ, LLQ. Politică de bandă: policing vs. shaping. Configurarea QoS pe switch-uri Cisco.	2	Prelegere interactivă	
Tema 9: Rețele wireless enterprise (WLAN). Arhitectura WLAN enterprise: autonomous AP vs. lightweight AP + WLC (Wireless LAN Controller). Protocoale LWAPP/CAPWAP: tunelul de control și date. Roaming: Layer 2 roaming, Layer 3 roaming cu mobilitate de client. Planificarea acoperirii radio: site survey, heatmap, overlap de celule. 802.11ac Wave 2 și 802.11ax (Wi-Fi 6): MU-MIMO, OFDMA, BSS Coloring. Securitate WLAN enterprise: WPA3-Enterprise, 802.1X, certificate.	2	Prelegere interactivă	
Tema 10: Rețele LAN pentru data center. Topologia fat-tree (spine-leaf): avantaje față	2	Prelegere interactivă	

de arhitectura ierarhică tradițională pentru data center. Switch-uri spine vs. leaf: roluri, densitate, latență. Protocoale de rutare în data center: OSPF, BGP (underlay). Overlay: VXLAN (Virtual Extensible LAN) — încapsularea L2 în UDP, VTEP (VXLAN Tunnel Endpoint). EVPN (Ethernet VPN) cu MP-BGP ca plane de control pentru VXLAN. Multi-tenancy în data center.			
Tema 11: Monitorizarea și managementul LAN. SNMP v3: autentificare SHA, criptare AES. NetFlow v9/IPFIX: exportul fluxurilor, colectori. Syslog: severitati, centralizarea logurilor. LLDP și CDP: descoperirea topologiei. NTP: sincronizarea ceasului în rețea. Instrumente de monitorizare: Cisco DNA Center, Zabbix, Grafana. Backup și restaurarea configurațiilor: TFTP, FTP, SCP.	2	Prelegere interactivă	
Tema 12: Automatizarea rețelelor LAN. Limitele managementului manual: scalabilitate, erori umane. Cisco IOS REST API. Protocoale de automatizare: NETCONF (RFC 6241) cu YANG models, RESTCONF. Ansible pentru rețele: module ios_command, ios_config. Python și Netmiko/NAPALM: scriptarea configurațiilor Cisco. Infrastructure as Code (IaC): Terraform pentru rețele. Cisco DNA Center ca platform de automatizare.	2	Prelegere interactivă	
Tema 13: Depanarea rețelelor LAN. Metodologia de troubleshooting: modelul OSI layer-by-layer, abordarea divide-and-conquer. Comenzile Cisco IOS pentru depanare: show interfaces, show ip route, show spanning-tree, show vlan, show etherchannel, debug. Wireshark: capturarea pe portul SPAN, filtrarea avansată. Probleme frecvente: STP loops, VLAN mismatch, duplex mismatch, EtherChannel misconfiguration. Documentarea incidentelor.	2	Prelegere interactivă	
Tema 14: Tendințe și recapitulare. SD-Access (Cisco): fabric, underlay, overlay, policy plane. Intent-Based Networking (IBN). Wi-Fi 6E: banda de 6 GHz. 400G Ethernet în data center. Network as Code: GitOps pentru configurații de rețea. Recapitulare generală.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Tanenbaum, A.S., Feamster, N., Wetherall, D.J., Computer Networks, 6th ed., Pearson, 2021. 2. Odom, W., CCNA 200-301 Official Cert Guide, Vol. 1 și Vol. 2, Cisco Press, 2020. 3. Vachon, B., Graziani, R., Accessing the WAN: CCNA Exploration Companion Guide, Cisco Press, 2008. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Donahue, G.A., Network Warrior, 2nd ed., O'Reilly Media, 2011. 5. Lammle, T., CCNA Cisco Certified Network Associate Study Guide, 8th ed., Sybex, 2020. <p>Resurse software:</p> <ol style="list-style-type: none"> 6. Cisco Packet Tracer — simulator de rețele (cont gratuit NetAcad): https://www.netacad.com/courses/packet-tracer 		

	7. Wireshark — analizor de protocol: https://www.wireshark.org/docs/ 8. Cisco IOS Command Reference — referință completă comenzi CLI: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-15-4m-t/products-command-reference-list.html		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Arhitectura ierarhică LAN. Construirea în Packet Tracer a unei rețele campus pe 3 niveluri (access, distribution, core). Configurarea switch-urilor pe fiecare nivel. Verificarea conectivității end-to-end. Analiza traficului în modul simulare: calea unui cadru prin ierarhie.	2	Lucrare practică Packet Tracer	
Lucrarea L2: Switching avansat și port security. Configurarea port security pe switch-uri de acces: maximum MAC, mod shutdown. Testarea prin conectarea unui dispozitiv neautorizat: verificarea intrării în err-disabled. Recuperarea unui port err-disabled. Configurarea port mirroring (SPAN) pentru captura Wireshark.	2	Lucrare practică Packet Tracer	
Lucrarea L3: VLAN avansat — Voice VLAN și VTP. Configurarea Voice VLAN pe porturi de acces pentru telefoane IP simulate în Packet Tracer. Configurarea VTP: un switch server, doi clienți, un transparent. Propagarea VLAN-urilor prin VTP. Demonstrarea riscului VTP: resetarea bazei de date prin introducerea unui nou server cu revision number mai mare.	2	Lucrare practică Packet Tracer	
Lucrarea L4: Rapid PVST+ și PortFast. Configurarea Rapid PVST+ pe o topologie cu 3 switch-uri în triunghi și 3 VLAN-uri. Forțarea root bridge-ului prin priority. Configurarea PortFast și BPDUGuard pe porturile de acces. Testarea convergenței rapide: simularea căderii unui trunk și măsurarea timpului de reconvergență.	2	Lucrare practică Packet Tracer	
Lucrarea L5: EtherChannel. Configurarea EtherChannel cu LACP (mod active/passive) între două switch-uri cu 2 legături fizice. Verificarea cu show etherchannel summary. Testarea load balancing-ului: captură Wireshark pe ambele porturi fizice. Simularea căderii unui port fizic: verificarea continuității traficului.	2	Lucrare practică Packet Tracer	
Lucrarea L6: Switch L3 și HSRP. Configurarea SVI-urilor pe un switch L3 pentru rutare inter-VLAN. Configurarea HSRP între două switch-uri de distribuție: active, standby, virtual IP. Testarea failover-ului: simularea căderii switch-ului active și verificarea preluării de către standby. Configurarea preemption.	2	Lucrare practică Packet Tracer	
Lucrarea L7: Securitate LAN — DHCP Snooping și DAI. Configurarea DHCP Snooping: trusted port (uplink), untrusted	2	Lucrare practică Packet Tracer	

ports (acces). Testarea blocării unui server DHCP rogue pe un port untrusted. Configurarea Dynamic ARP Inspection bazată pe binding table DHCP Snooping. Verificarea respingerii pachetelor ARP false.			
Lucrarea L8: Securitate LAN — 802.1X. Configurarea 802.1X pe un switch Cisco cu server RADIUS simulat în Packet Tracer. Autentificarea unui client cu credențiale corecte și incorecte. Verificarea stării portului: authorized vs. unauthorized. Configurarea VLAN de autentificare (auth-fail VLAN, guest VLAN).	2	Lucrare practică Packet Tracer	
Lucrarea L9: QoS — clasificare și marcarea. Configurarea unui policy-map pe switch-ul de acces pentru clasificarea traficului VoIP (UDP 5004/5060) și marcarea cu DSCP EF (46). Configurarea unui class-map pentru traficul de date și marcarea cu DSCP AF11. Verificarea marcajelor în captura Wireshark: câmpul DSCP din header-ul IP.	2	Lucrare practică Packet Tracer, Wireshark	
Lucrarea L10: WLAN enterprise — Lightweight AP și WLC. Configurarea unui Wireless LAN Controller (WLC) și a unui Lightweight Access Point (LAP) în Packet Tracer. Crearea unui WLAN (SSID) cu securitate WPA2-Personal și WPA2-Enterprise (802.1X). Asocierea unui client wireless. Verificarea datelor de pe WLC: clienți asociați, canal, semnal.	2	Lucrare practică Packet Tracer	
Lucrarea L11: Monitorizarea LAN. Configurarea Syslog pe switch-uri Cisco cu trimiterea logurilor la un server Syslog simulat. Configurarea NTP: un switch ca server NTP, celelalte ca clienți. Verificarea sincronizării. Captură Wireshark a traficului SNMP: GET, RESPONSE. Identificarea datelor din MIB.	2	Lucrare practică Packet Tracer, Wireshark	
Lucrarea L12: Automatizare cu Python și Netmiko. Scrierea unui script Python cu Netmiko care se conectează SSH la 3 switch-uri Cisco (GNS3 sau Packet Tracer) și colectează output-ul comenzilor show interfaces status și show vlan brief. Formatarea datelor în CSV. Extinderea scriptului pentru aplicarea unei configurații de VLAN pe toate switch-urile simultan.	2	Lucrare practică Python, Netmiko	
Lucrarea L13: Depanarea avansată LAN. Simularea și rezolvarea a 5 scenarii de depanare în Packet Tracer: STP loop (trunk lipsă), VLAN mismatch, EtherChannel misconfiguration (duplex diferit), HSRP stuck în standby (priority greșită), port în err-disabled (port security). Documentarea fiecărui caz.	2	Lucrare practică Packet Tracer	

<p>Lucrarea L14: Proiect integrat și colocviu. Proiectarea și implementarea completă a unui campus LAN enterprise în Packet Tracer: arhitectură pe 3 niveluri, 4 VLAN-uri, EtherChannel între distribuție și core, Rapid PVST+, HSRP pe distribuție, DHCP Snooping, DAI, QoS pentru VoIP, WLAN cu WLC. Documentare și colocviu oral.</p>	2	Proiect individual, colocviu	
<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Odom, W., CCNA 200-301 Official Cert Guide, Vol. 1 și Vol. 2, Cisco Press, 2020 — capitolele de switching și securitate LAN. 2. Cisco Packet Tracer — documentație: https://www.netacad.com/courses/packet-tracer 3. Cisco IOS Command Reference — comenzile CLI de configurare: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-15-4m-t/products-command-reference-list.html <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. Donahue, G.A., Network Warrior, 2nd ed., O'Reilly Media, 2011 — scenarii de depanare LAN. 5. Wireshark — ghid utilizare și filtre: https://www.wireshark.org/docs/wsug_html_chunked/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. Cisco Packet Tracer — simulator gratuit: https://www.netacad.com/courses/packet-tracer 7. Wireshark — analizor de protocol: https://www.wireshark.org/ 8. Cisco Learning Network — exerciții și tutoriale switching: https://learningnetwork.cisco.com/ 9. Netmiko Documentation — librărie Python pentru automatizare Cisco: https://netmiko.readthedocs.io/ 			

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

<p>Rețele Locale de Calculatoare aprofundează competențele din Rețele de Calculatoare (TIDS301) prin orientarea spre infrastructura LAN enterprise: switching ierarhic, redundanță (STP avansat, EtherChannel, HSRP), securitate L2 (DHCP Snooping, DAI, 802.1X), QoS și wireless enterprise. Aceasta este infrastructura pe care rulează toate serviciile de business — orice pană în LAN-ul enterprise are impact imediat asupra productivității.</p> <p>Lucrarea L12 (automatizare cu Python și Netmiko) și Tema 12 de curs pregătesc studentul pentru rolul de inginer de rețea modern, care nu configurează manual fiecare dispozitiv, ci scrie scripturi și folosește platforme de automatizare — direct corelat cu A11 și R11 din matricea de corelare. Proiectul integrat din L14 reproduce un campus LAN real cu toate tehnologiile disciplinei.</p>

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea arhitecturii ierarhice LAN (access/distribution/core) și a criteriilor de selecție a echipamentelor; înțelegerea funcționării Rapid PVST+, EtherChannel (LACP) și HSRP; cunoașterea mecanismelor de securitate L2 (Port Security, DHCP	Examen scris (2 ore): 1 problemă de proiectare (selectarea și justificarea echipamentelor pentru o topologie ierarhică dată), 1 problemă de analiză STP (identificarea root bridge-ului, a porturilor blocked și calculul convergenței pentru o topologie cu 3 switch-uri și PVST+) și 1 problemă de securitate sau QoS (configurarea DHCP Snooping/DAI sau clasificarea traficului cu DSCP)	40%

	Snooping, DAI, 802.1X); cunoașterea principiilor QoS (CoS, DSCP, CBWFQ, LLQ); cunoașterea arhitecturii wireless enterprise (WLC, CAPWAP).		
8.5 Seminar/laborator	Funcționalitatea configurațiilor Packet Tracer (EtherChannel activ, HSRP failover demonstrat, DHCP Snooping blocat rogue server, 802.1X autentificare corectă); calitatea scriptului Python din L12 (colectare și configurare funcționale); calitatea proiectului integrat din L14 (toate componentele prezente și funcționale); calitatea prezentării la colocviu.	Evaluare continuă: configurații Packet Tracer și rapoarte predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L14 (50%)	60%
8.6 Standard minim de performanță:			
Demonstrarea funcționalității EtherChannel și a failover-ului HSRP în proiectul din L14 — EtherChannel trebuie să apară cu status P (in-use) în show etherchannel summary, iar HSRP trebuie să preia activ în maxim 10 secunde după simularea căderii switch-ului activ. Spre deosebire de TIDS301 (Rețele de Calculatoare), unde standardul minim era subnetting-ul, la această disciplină verificarea se face pe configurarea redundanței LAN, care este competența centrală specifică rețelelor locale enterprise.			

Data completării	Semnătura titularului de curs Lect. Dr. Damian Felicia Anișoara	Semnătura titularului de seminar / laborator Lect. Dr. Damian Felicia Anișoara
17.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei	SISTEME ȘI REȚELE DE COMUNICAȚII – TIDS313						
1.2 Titularul activităților de curs	Conf. Dr. Aramă Iulian-Nicușor						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Conf. Dr. Aramă Iulian-Nicușor						
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					44
Documentare					15
Studiu individual					15
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					4
2.9 Total ore pe semestru					100
2.10 Numărul de credite					4

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Teoria Transmisiunii Informației (TIDS212) — entropia Shannon, capacitatea canalului, coduri de sursă; Rețele de Calculatoare (TIDS301) — stiva TCP/IP, protocoale de transport și rețea; Achiziția și Prelucrarea Datelor (TIDS304) — modulații digitale, esanționare, filtrare.
3.2 de competențe	Cunoașterea teoremei Shannon-Nyquist și a notiunilor de bază din teoria informației (entropie, capacitate canal); înțelegerea modului de funcționare al stivei TCP/IP la nivel de protocoale; familiarizarea cu modulațiile digitale (ASK, FSK, PSK) și cu analiza spectrală (FFT); abilitatea de a utiliza Python sau MATLAB pentru prelucrarea semnalelor.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sala de curs cu tablă, videoproiector și acces la internet; platforma Sakai; recomandare: Python 3.10+ cu NumPy, SciPy, Matplotlib și comppy instalate înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; Python 3.10+ cu NumPy, SciPy, Matplotlib, comppy, pyshark sau scapy instalate;

	Wireshark pentru analiza traficului; studentii pot folosi propriul laptop; acces la echipamente SDR (Software Defined Radio, ex. RTL-SDR) acolo unde sunt disponibile.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C12. Cunoaște în profunzime principiile sistemelor și rețelelor de comunicații, inclusiv teoria transmisiunii informației și tehnicile moderne de codare și modulare a semnalelor.
5.2 Aptitudini	A12. Analizează și evaluează performanța sistemelor și rețelelor de comunicații, identificând și rezolvând probleme de funcționare prin aplicarea metodelor de monitorizare și diagnosticare specifice domeniului.
5.3 Responsabilitate și autonomie	R12. Acționează autonom în implementarea și menținerea securității sistemelor informatice, asumându-și responsabilitatea pentru identificarea vulnerabilităților, aplicarea corecțiilor și conformitatea cu standardele de securitate în vigoare.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Sisteme de comunicatii — arhitectura generala. Blocurile unui sistem de comunicatii: sursa, codor de sursa, codor de canal, modulator, canal, demodulator, decodor de canal, decodor de sursa, destinatie. Clasificarea canalelor: cu fir (cablu coaxial, fibra optica, pereche rasucita), fara fir (radio, microunde, satelit, IR). Parametrii de performanta: capacitatea canalului (teorema Shannon), rata de bit, BER (Bit Error Rate), SNR, Eb/N0. Comparatia sistemelor analogice cu cele digitale.	2	Prelegere interactiva	
Tema 2: Modulatii analogice. Modulatia AM (Amplitude Modulation): modulatia de amplitudine standard, DSB-SC (Double Sideband Suppressed Carrier), SSB (Single Sideband). Eficienta spectrala si energetica a variantelor AM. Modulatia FM (Frequency Modulation): indicele de modulatie, spectrul Carson, regula lui Carson. Modulatia PM (Phase Modulation). Demodulatia: detectia de anvelopa, detectia coerenta, discriminatorul de frecventa. Comparatia AM/FM: robustete la zgomot, latime de banda.	2	Prelegere interactiva	
Tema 3: Modulatii digitale in banda de baza. Codificarea liniei (line coding): NRZ-L, NRZ-I, Manchester, AMI, HDB3, 4B/5B, 8B/10B. Spectrul puterii pentru fiecare cod. Filtrele de conformare a impulsurilor (pulse shaping): filtrul raised cosine si root raised cosine — eliminarea ISI (Inter-Symbol Interference). Interferenta intersimbolica: diagrama ochiului (eye diagram), masurarea deschiderii. Egalizarea la receptie.	2	Prelegere interactiva	
Tema 4: Modulatii digitale in banda translata. ASK (Amplitude Shift Keying):	2	Prelegere interactiva	

OOK, 4-ASK. FSK (Frequency Shift Keying): MSK (Minimum Shift Keying), GMSK. PSK: BPSK, QPSK, 8-PSK, DPSK. QAM (Quadrature Amplitude Modulation): 16-QAM, 64-QAM, 256-QAM. Diagramele de constelatie. Performanta in AWGN: formulele BER pentru BPSK, QPSK, QAM. Eficienta spectrala: biti/simbol/Hz. Alegerea modulatiei in functie de SNR si latime de banda.			
Tema 5: Teoria codarii si coduri detectoare/corectoare de erori. Entropia Shannon si codarea sursei: codurile Huffman si Lempel-Ziv. Teorema de codare a canalului (Shannon): capacitatea canalului Gaussian, limita Shannon. Distanta Hamming si puterea de corectie. Coduri de bloc liniare: coduri Hamming (7,4), coduri Reed-Solomon. Coduri ciclice (CRC): polinomul generator, calculul si verificarea CRC. Coduri convolutionale: diagrama de trellis, decodul Viterbi.	2	Prelegere interactiva	
Tema 6: OFDM si sistemele multicarrier. Motivatia OFDM: canalul cu multipath si ISI. Principiul OFDM: subbenzile ortogonale, IFFT/FFT, prefixul ciclic (CP). Spectrul OFDM: lobe laterale si fereastrare. Parametrii OFDM: numarul de subpurtatoare, lungimea CP, intervalul de garda. OFDM in standardele actuale: LTE (4G), NR (5G), 802.11ac/ax (Wi-Fi 5/6), DVB-T2. Dezavantajele OFDM: PAPR (Peak-to-Average Power Ratio), sensibilitatea la offset de frecventa.	2	Prelegere interactiva	
Tema 7: Antene si propagarea undelor radio. Parametrii antenei: castig, directivitate, eficienta, banda, polarizare, pattern de radiatie. Antena dipolului: pattern, impedanta de intrare. Antena Yagi-Uda: principiu, beam-forming. Antena patch (microstrip): aplicatii WiFi si 5G. Propagarea undelor radio: modelul in spatiu liber (Friis), difractia Fresnel, reflexia si dispersia multipla. Modelele de propagare: Okumura-Hata, COST-231, 3GPP. Link budget: calculul marginii de rezerva.	2	Prelegere interactiva	
Tema 8: Sisteme de comunicatii mobile. Arhitectura retelelor mobile: UE, eNodeB/gNodeB, EPC/5GC. Accesul multiplu: FDMA, TDMA, CDMA (spreading, despreading), OFDMA (LTE downlink), SC-FDMA (LTE uplink). Planificarea celulara: reutilizarea frecventei, marimea clusterului, handover. LTE (4G): arhitectura, modulatia (OFDMA/SC-FDMA), MIMO, categoriile de echipament. NR (5G): benzile sub-6 GHz si mmWave, massive MIMO, network slicing, latenta ultra-scazuta.	2	Prelegere interactiva	

<p>Tema 9: Sisteme de comunicatii prin satelit. Orbitale satelitilor: LEO (200-2000 km, Starlink, OneWeb), MEO (GPS, Galileo), GEO (inmarsat, DirecTV). Calculul legaturii satelitare (link budget): puterea EIRP, castigul antenei terestre, path loss, G/T. Transponderile satelit: banda C, Ku, Ka. VSAT (Very Small Aperture Terminal). Sistemele GPS/GNSS: principiul trilatered, pseudo-range, erorile de masurare, DGPS. Starlink si constelatiile LEO: latentă, cobertura.</p>	2	Prelegere interactiva	
<p>Tema 10: Comunicatii optice. Fibra optica: single-mode (SM) si multi-mode (MM), atenuarea (dB/km), dispersia cromatica si de mod de polarizare. Surse si detectoare optice: laser Fabry-Perot, DFB laser, fotodetectoare PIN si APD. Modulatia optica: OOK, DPSK, DP-QPSK pentru sisteme coerente. Multiplexarea WDM (Wavelength Division Multiplexing) si DWDM: capacitati actuale (> 100 Tbit/s). Amplificatoare optice: EDFA. Sisteme coerente de 400G si 800G.</p>	2	Prelegere interactiva	
<p>Tema 11: Securitatea comunicatiilor. Atacuri specifice comunicatiilor: eavesdropping (interceptarea), jamming (bruierea), spoofing (falsificarea sursei), man-in-the-middle. Criptografia la nivel fizic: spread spectrum (FHSS, DSSS) si rezistenta la interceptare si bruiere. Securitatea protoalelor: WPA3, TLS 1.3, IPsec. PKI in comunicatii: certificatele digitale, autoritatea de certificare. Autentificarea in retele mobile: SIM, AKA (Authentication and Key Agreement) in 4G/5G. Vulnerabilitatile IMSI catcher.</p>	2	Prelegere interactiva	
<p>Tema 12: Masurarea si monitorizarea performantei retelelor. Parametrii de performanta: latentă, jitter, throughput, packet loss, disponibilitate (SLA). Masurarea BER in sisteme reale: metode (BERT, pseudo-random sequences). Analizoare de spectru si de protocol: functionare, interpretarea masuratorilor. Monitorizarea retelelor de comunicatii: SNMP, NetFlow, syslog. Testele active si pasive: ping, traceroute, iperf3. Analiza capturii de pachete cu Wireshark: filtrele de displaay, statisticile de performanta.</p>	2	Prelegere interactiva	
<p>Tema 13: Software Defined Radio (SDR) si comunicatii cognitive. Principiul SDR: componentele hardware (RF front-end, ADC/DAC, FPGA/CPU) si implementarea software a modulatiei si demodularii. Platforma GNU Radio: blocurile de prelucrare, grafele de flux de semnal. RTL-SDR: specificatii, aplicatii (receptor FM, ADS-B, trunking). Comunicatii cognitive (CR</p>	2	Prelegere interactiva	

<p>— Cognitive Radio): spectrum sensing, accesul dinamic la spectru, protocolul MAC cognitive. Standardul IEEE 802.22 (WRAN).</p>			
<p>Tema 14: Tendinte si recapitulare. 5G Advanced si 6G: frecventele THz, RIS (Reconfigurable Intelligent Surfaces), comunicatii integrate cu senzorii (ISAC). Comunicatii cuantice: distributia cheilor cuantice (QKD), protocoalele BB84 si E91. Comunicatii pentru IoT: NB-IoT, eMTC, LoRaWAN, Sigfox — comparatia parametrilor. Comunicatiile non-terestre (NTN): LEO sateliti, drone. Recapitulare generala.</p>	2	Prelegere, recapitulare	
<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Haykin, S., Moher, M., Communication Systems, 5th ed., Wiley, 2009. Proakis, J.G., Salehi, M., Digital Communications, 5th ed., McGraw-Hill, 2007. Rappaport, T.S., Wireless Communications: Principles and Practice, 2nd ed., Prentice Hall, 2001. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Sklar, B., Digital Communications: Fundamentals and Applications, 2nd ed., Prentice Hall, 2001. Forouzan, B.A., Data Communications and Networking, 5th ed., McGraw-Hill, 2013. <p>Resurse software:</p> <ol style="list-style-type: none"> GNU Radio Documentation — platforma SDR open-source: https://wiki.gnuradio.org/ CommPy Documentation — comunicatii digitale in Python: https://github.com/veeresht/CommPy SciPy Signal Processing — modulatii si filtrare: https://docs.scipy.org/doc/scipy/reference/signal.html 			
6.2 Laborator	Nr. ore	Metode de predare	Observații
<p>Lucrarea L1: Modulatii analogice — simulare si analiza. Implementarea in Python a modulatiei AM (standard, DSB-SC, SSB) si FM. Trasarea semnalelor in domeniul timp si frecventa (FFT). Masurarea largimii de banda a semnalelor modulate. Simularea demodulatiei AM prin detectia de anvelopa si FM prin discriminatorul de frecventa. Compararea SNR la iesirea demodulatorului pentru AM si FM la acelasi SNR de intrare.</p>	2	Lucrare practica Python	
<p>Lucrarea L2: Modulatii digitale in banda de baza. Implementarea in Python a codificarii NRZ-L, NRZ-I, Manchester si AMI. Trasarea spectrului puterii pentru fiecare cod. Implementarea filtrului raised cosine pentru conformarea impulsurilor. Generarea si analiza diagramei ochiului (eye diagram) cu si fara zgomot AWGN pentru diferite niveluri SNR.</p>	2	Lucrare practica Python	
<p>Lucrarea L3: BPSK, QPSK si QAM — constelatie si BER. Implementarea modulatiei si demodulatiei BPSK, QPSK si 16-QAM cu commpy sau implementare proprie. Trasarea diagramelor de constelatie cu si fara zgomot (SNR = 5, 10, 15 dB). Calculul experimental al BER prin simulare Monte Carlo (N = 10⁵ simboluri). Compararea BER simulate cu formulele teoretice pentru BPSK si QPSK.</p>	2	Lucrare practica Python	

<p>Lucrarea L4: Coduri detectoare si corectoare de erori. Implementarea si testarea codului Hamming (7,4): codarea unui mesaj de 4 biti, injectarea unui bit de eroare, decodarea si corectia. Implementarea CRC-32 pe un bloc de date. Simularea unui cod convolutional cu ratia 1/2 si decodul Viterbi cu commpy. Compararea BER cu si fara codare la acelasi SNR.</p>	2	Lucrare practica Python	
<p>Lucrarea L5: OFDM — implementare si analiza. Implementarea unui transmitator/receptor OFDM cu N=64 subpurtatoare, CP=16 si modulatie QPSK pe fiecare subpurtatoare. Verificarea ortogonalitatii subpurtatoarelor. Simularea canalului cu multipath cu 3 cai si verificarea eficacitatii CP. Calculul BER OFDM in canal AWGN si canal cu multipath. Trasarea spectrogramei semnalului OFDM.</p>	2	Lucrare practica Python	
<p>Lucrarea L6: Link budget si planificarea coberturii radio. Calculul link budget-ului pentru 3 scenarii: legatura WiFi 802.11ac la 5 GHz (distanta 50 m interior), legatura LTE la 1800 MHz (distanta 1 km urban) si legatura LoRa la 868 MHz (distanta 5 km rural). Utilizarea formulei Friis si a modelului Okumura-Hata. Verificarea marginilor de rezerva. Trasarea heatmap-ului RSSI simulat cu un model de propagare simplified.</p>	2	Lucrare practica Python, calcule	
<p>Lucrarea L7: Analiza traficului de retea cu Wireshark. Capturarea traficului pe o interfata de retea cu Wireshark. Analiza unui flux TCP: three-way handshake, fereastră de congestionare, retransmisii. Analiza unui flux UDP (stream video sau VoIP simulat): jitter, packet loss. Filtrarea si exportul statisticilor de performanta. Identificarea unui protocol de comunicatii la nivelul aplicatie (HTTP, DNS, TLS).</p>	2	Lucrare practica Wireshark	
<p>Lucrarea L8: Masurarea performantei retelei. Masurarea latentei si a jitter-ului cu ping si hping3 catre mai multe destinatii. Masurarea throughput-ului TCP si UDP cu iperf3: efectul dimensiunii ferestrei TCP, comparatia TCP vs. UDP. Masurarea packet loss cu iperf3 la diferite rate UDP. Redactarea unui raport de performanta cu interpretarea rezultatelor.</p>	2	Lucrare practica Linux, iperf3	
<p>Lucrarea L9: Securitatea comunicatiilor — atacuri si protectie. Simularea unui atac de jamming simplu (generarea de zgomot pe o frecventa) cu RTL-SDR sau in Python. Demonstrarea rezistentei DSSS la bruij: secventa de spreading cu 11 chips (cod Barker). Analiza unui handshake TLS 1.3 cu Wireshark: identificarea mesajelor ClientHello, ServerHello, Certificate.</p>	2	Lucrare practica Python, Wireshark	

Verificarea unui certificat digital: lantul de incredere, valabilitatea.			
Lucrarea L10: GNU Radio — receptor FM si AM. Construirea unui graf de flux GNU Radio (in simulare sau cu RTL-SDR daca disponibil) pentru un receptor FM: blocurile source, rational resampler, WBFM receive, audio sink. Testarea receptorului pe un semnal FM generat local. Extinderea la demodularea AM. Analiza spectrului semnalului RF captat cu FFT sink.	2	Lucrare practica GNU Radio	
Lucrarea L11: GNU Radio — modulatii digitale. Implementarea unui sistem punct-la-punct in GNU Radio: transmitator BPSK (generarea simbolurilor, conformarea impulsurilor, modulatia) si receptor (sincronizare simbol, demodulatia, decizia). Testarea in loopback. Masurarea BER in functie de SNR configurat in blocul de canal AWGN.	2	Lucrare practica GNU Radio	
Lucrarea L12: Analiza semnalelor radio cu RTL-SDR sau simulare. Receptia si analiza unui semnal ADS-B (transpondere avioane la 1090 MHz) cu RTL-SDR si decodarea mesajelor (identificator ICAO, altitudine, pozitie) cu dump1090 — sau, in lipsa hardware, simularea in Python. Analiza spectrului FM broadcast. Identificarea si clasificarea semnalelor captate dupa caracteristicile spectrale.	2	Lucrare practica RTL-SDR/simulare	
Lucrarea L13: Evaluarea performantei sistemului de comunicatii end-to-end. Proiectarea unui sistem de comunicatii digital complet in Python: QPSK cu filtre raised cosine, codare Hamming, canal AWGN, decodare si demodulatie. Masurarea BER cu si fara codare pentru Eb/N0 intre 0 si 12 dB. Trasarea curbelor BER. Compararea cu limitele teoretice si calculul castigului de codare (coding gain).	2	Lucrare practica Python	
Lucrarea L14: Proiect integrat si colocviu. Implementarea si documentarea unui sistem de comunicatii complet la alegere: modem OFDM, receptor FM cu GNU Radio, sistem cu codare turbo simulata, sau analiza comparativa a modulatorilor (BPSK/QPSK/16-QAM/64-QAM). Documentatia tehnica: diagrama bloc sistem, parametrii alesi si justificarea lor, curbele de performanta (BER vs SNR), limitele identificate. Colocviu oral.	2	Proiect individual, colocviu	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Proakis, J.G., Salehi, M., Digital Communications, 5th ed., McGraw-Hill, 2007 — exercitiile practice si simulari. 2. GNU Radio Documentation: https://wiki.gnuradio.org/ 3. CommPy Documentation — librerie Python pentru comunicatii digitale: https://github.com/veeresht/CommPy <p>Bibliografie complementara laborator</p>		

	<p>4. Haykin, S., Moher, M., Communication Systems, 5th ed., Wiley, 2009.</p> <p>5. Wireshark Documentation — ghid de utilizare si filtre: https://www.wireshark.org/docs/wsug_html_chunked/</p> <p>Resurse online gratuite recomandate</p> <p>6. GNU Radio Documentation si tutoriale: https://wiki.gnuradio.org/</p> <p>7. CommPy — comunicatii digitale in Python: https://github.com/veeresht/CommPy</p> <p>8. Wireshark — analizor de protocol: https://www.wireshark.org/</p> <p>9. RTL-SDR Blog — ghid RTL-SDR si proiecte SDR: https://www.rtl-sdr.com/</p>
--	--

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

Sisteme si Rețele de Comunicatii acopera stratul fizic si de canal al oricarui sistem de comunicatie digitala: de la modulatia OFDM din WiFi si 5G, la codarea corectoare de erori din comunicatiile satelit, pana la securizarea legaturii fizice prin spread spectrum. Aceasta disciplina completeaza perspectiva protocoalelor (de la Rețele de Calculatoare) cu intelegerea profunda a limitelor fizice ale transmisiei si a tehnicilor de abordare a acestora.

Componenta de laborator, care combina simularea Python cu analiza traficului real (Wireshark, GNU Radio/RTL-SDR), formeaza direct competenta A12: analiza si evaluarea performantei sistemelor de comunicatii prin metode de monitorizare si diagnosticare. Tema de securitate din L9 si L11 asigura legatura cu R12, iar documentatia tehnica a proiectului integrat din L14 este evaluata conform standardelor ingineresti.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoasterea si aplicarea formulelor BER pentru BPSK, QPSK si QAM in canal AWGN; calculul link budget-ului pentru o legatura radio (formula Friis, path loss, marja de rezerva); cunoasterea principiului OFDM si a rolului prefixului ciclic; cunoasterea algoritmilor de codare si decodare (Hamming, CRC, Viterbi); cunoasterea principiilor sistemelor de comunicatii mobile (OFDMA, MIMO, handover) si a securitatii la nivel fizic (spread spectrum, WPA3, AKA).	Examen scris (2 ore): 1 problema de modulatie (calculul BER teoretic si comparatia eficientei spectrale pentru doua modulatii date la un SNR specificat), 1 problema de link budget (calculul puterii primite si al marjei de rezerva pentru o legatura radio descrisa) si 1 problema de codare (calculul redundantei unui cod Hamming sau CRC si demonstrarea capabilitatii de corectie)	60%
8.5 Seminar/laborator	Corectitudinea similarilor Python (curbele BER se incadreaza in ± 1 dB fata de valorile teoretice, diagramele de constelatie sunt corecte pentru SNR dat);	Evaluare continua: implementari individuale predate pe Sakai dupa fiecare lucrare (50%) si proiect integrat cu colocviu oral individual la L14 (50%)	40%

	calitatea analizei Wireshark (identificarea corecta a protocoalelor, interpretarea metricilor de performanta); calitatea documentatiei tehnice a proiectului integrat din L14 (diagrama bloc, justificarea parametrilor, curbele BER vs. SNR); calitatea prezentarii la colocviu.		
--	---	--	--

8.6 Standard minim de performanță:

Includerea in proiectul integrat din L14 a curbei BER vs. Eb/N0 obtinuta prin simulare Monte Carlo (minim 10^5 simboluri per punct) cu cel putin un punct de referinta teoretic verificat — aceasta separand un sistem implementat corect de unul cu erori subtile de normalizare sau de calcul al SNR. La examenul scris: calculul complet al link budget-ului (EIRP, path loss, puterea primita si marja de rezerva) pentru un scenariu dat, cu toti pasii intermediari si unitatile corecte (dBm, dB, dBi).

Data completării	Semnătura titularului de curs Conf. Dr. Aramă Iulian-Nicușor	Semnătura titularului de seminar / laborator Conf. Dr. Aramă Iulian-Nicușor
17.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei		TESTAREA SISTEMELOR DE CALCUL – TIDS314					
1.2 Titularul activităților de curs		Lect. Dr. Damian Felicia Anișoara					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Lect. Dr. Damian Felicia Anișoara					
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Ingineria Programelor (TIDS308) — TDD, testare unitara, CI/CD; Baze de Date (TIDS305) — SQL pentru testarea interogărilor si a procedurilor stocate; Rețele de Calculatoare (TIDS301) — protocoale si configuratii pentru testarea de retea.
3.2 de competențe	Cunoasterea ciclului de viata software (Agile, Scrum) si a conceptelor de baza ale testarii (test case, defect, acoperire); experienta scrierii testelor unitare cu JUnit sau pytest; familiarizarea cu Git si GitHub Actions; abilitatea de a scrie si depana cod Python sau Java la nivel mediu.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sala de curs cu tabla, videoproiector si acces la internet; platforma Sakai; recomandare: JDK + JUnit 5 + Maven sau Python + pytest + coverage.py instalate inainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatica cu calculatoare individuale; IDE (IntelliJ IDEA sau VS Code), JUnit 5, pytest, Selenium, JMeter, Postman instalate; cont GitHub activ; studentii pot folosi propriul laptop.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C42. Cunoaște în profunzime principiile testării și validării sistemelor de calcul — testare funcțională, testare de performanță, testare de regresie, verificare formală — cu conștientizarea critică a limitelor detectabilității erorilor.
5.2 Aptitudini	A37. Planifică și execută campanii complete de testare a sistemelor de calcul — de la testarea unitară și de integrare până la testarea de sistem și acceptanță — documentând rezultatele și urmărind remediarea defectelor identificate.
5.3 Responsabilitate și autonomie	R37. Acționează autonom în planificarea și coordonarea activităților de testare a sistemelor de calcul, asumându-și responsabilitatea pentru completitudinea acoperirii testelor și pentru comunicarea clară a rezultatelor către echipa de dezvoltare și beneficiari.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în testarea software. Definiția și scopul testării. Terminologie: defect (bug), eroare, esec. Principiile testării (ISTQB): testarea demonstrează prezența erorilor, testarea exhaustivă este imposibilă, testarea timpurie, aglomerarea defectelor, paradoxul pesticidului, testarea depinde de context, absența erorilor este o iluzie. Costul corectării defectelor în funcție de etapa detectării. Nivelurile de testare: unitar, integrare, sistem, acceptanță.	2	Prelegere interactivă	
Tema 2: Testarea funcțională. Tehnici de proiectare a cazurilor de test bazate pe specificație (black-box): partitionarea echivalenței, analiza valorilor de frontieră, tabelele de decizie, testarea tranzițiilor de stare, testarea cazurilor de utilizare. Tehnici bazate pe structura (white-box): acoperirea instrucțiunilor, acoperirea ramurilor (branch coverage), acoperirea condițiilor, MC/DC (Modified Condition/Decision Coverage — avionics). Complexitatea ciclomatică McCabe.	2	Prelegere interactivă	
Tema 3: Testarea unitară avansată. JUnit 5: adnotările (@Test, @BeforeEach, @AfterEach, @ParameterizedTest, @ExtendWith), aserțiile (assertEquals, assertThrows, assertTimeout). Pytest: fixtures, parametrizare, mock-uri. Testarea excepțiilor și a comportamentelor asincrone. Test doubles: dummy, stub, spy, mock, fake — diferențele (Meszaros). Biblioteca Mockito (Java) și unittest.mock (Python): crearea și verificarea mock-urilor.	2	Prelegere interactivă	
Tema 4: Test-Driven Development și Behavior-Driven Development. TDD: ciclul Red-Green-Refactor, regulile (nu scrie cod de producție fără un test esuat). Avantajele și	2	Prelegere interactivă	

<p>limitele TDD: designul emergent, acoperirea implicita, viteza de dezvoltare. BDD (Behavior-Driven Development): specificatiile Gherkin (Given-When-Then), legatura cu user stories. Cucumber (Java) si Behave (Python): scrierea feature files, implementarea step definitions.</p>			
<p>Tema 5: Testarea de integrare. Tipuri de testare de integrare: Big Bang, top-down (cu stubs), bottom-up (cu drivere), sandwich. Testarea integrarii componentelor: contract testing (Pact), testarea serviciilor REST cu RestAssured sau requests. Testarea bazei de date: testele in-memory (H2, SQLite), testele cu Testcontainers (Docker). Testarea mesageriei: RabbitMQ, Kafka cu embedded brokers.</p>	2	Prelegere interactiva	
<p>Tema 6: Testarea interfetelor web — Selenium. Arhitectura Selenium WebDriver. Selenium + Python/Java: localizarea elementelor (id, name, CSS selector, XPath), actiunile (click, sendKeys, selectare), asteptarile (implicit vs. explicit wait). Page Object Model (POM): structura, avantajele in mentenabilitate. Selenium Grid: executia distribuita. Testarea cross-browser: Chrome, Firefox, Edge.</p>	2	Prelegere interactiva	
<p>Tema 7: Testarea API REST. Postman: colectii, environments, variabile, scripturile de test (JavaScript). Newman: executia Postman din linia de comanda in pipeline CI. Testarea cu requests + pytest (Python): validarea codurilor HTTP, a structurii JSON, a headerelor. Testarea contractelor API cu Pact: consumer-driven contract testing. OpenAPI/Swagger: generarea automata a testelor din specificatie.</p>	2	Prelegere interactiva	
<p>Tema 8: Testarea de performanta. Tipuri: load testing, stress testing, spike testing, soak (endurance) testing, scalability testing. Metricile de performanta: throughput (TPS), response time (medie, percentile p95/p99), error rate, utilizarea resurselor. Apache JMeter: planul de test, thread groups, sampleri HTTP, listeners (aggregate report, response time graph). Grafana + InfluxDB pentru vizualizarea rezultatelor JMeter in timp real.</p>	2	Prelegere interactiva	
<p>Tema 9: Testarea de securitate. Principiile OWASP Top 10 (2021): Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Misconfiguration, Vulnerable Components, Identification and Authentication Failures, Software Data Integrity Failures, Security Logging Failures, SSRF. Testarea manuala vs. automata. SAST (Static Application Security Testing):</p>	2	Prelegere interactiva	

SonarQube, Checkmarx. DAST (Dynamic AST): OWASP ZAP, Burp Suite. Penetration testing: etapele (reconnaissance, scanning, exploitation, reporting).			
Tema 10: Testarea de regresie si acoperirea codului. Testarea de regresie: selectia si prioritizarea cazurilor de test, costul suita de regresie. Acoperirea codului (code coverage): line, branch, condition, mutation testing (PIT pentru Java, mutmut pentru Python). Analiza raportului de acoperire: interpretarea, limitele (100% acoperire nu garanteaza absentia erorilor). Avertismentele ISTQB despre acoperire. Strategii de reducere a suitei de regresie.	2	Prelegere interactiva	
Tema 11: Managementul defectelor si al testarii. Ciclul de viata al unui defect: nou, asignat, in lucru, rezolvat, verificat, inchis, reopenat. Raportul de defect (bug report): titlul, preconditioniile, pasii de reproducere, comportamentul actual vs. cel asteptat, severitatea, prioritatea. Unelte de tracking: Jira, GitLab Issues. Metrici de testare: densitatea defectelor, eficienta detectiei defectelor (DDE), viteza rezolvarii. Test management: TestRail, Zephyr.	2	Prelegere interactiva	
Tema 12: Testarea in pipeline CI/CD. Integrarea testelor in GitHub Actions: job-urile de test, matricele de testare (Python 3.10/3.11/3.12), cache-ul dependentelor. Quality gates: SonarCloud, pragurile de acoperire si de cod mirositor. Testarea continua: strategia shift-left. Testarea in productie: canary deployments, A/B testing, feature flags. Orchestrarea testelor: paralelizarea cu pytest-xdist sau JUnit Platform. Raportarea automata a rezultatelor.	2	Prelegere interactiva	
Tema 13: Testarea sistemelor embedded si a hardware. Specificul testarii embedded: constrangerile hardware, lipsa unui OS complet, watchdog-uri. Hardware In the Loop (HIL): principiu, avantaje fata de testarea pe sistem real. Software In the Loop (SIL): simularea hardware-ului in cod. JTAG si debugging: breakpoints hardware, trace. Testarea firmware: unit testing cu CUnit, Unity, Google Test adaptat. Testarea interfetelor hardware: GPIO, UART, SPI, I2C — simulatoare si analizatoare logice.	2	Prelegere interactiva	
Tema 14: Verificare formala si recapitulare. Verificarea formala vs. testarea: diferente, complementaritate. Model checking: principiu, spatiul de stari, tool-uri (SPIN, TLA+). Verificarea automata a programelor: Frama-C (C), Dafny (verificare asistata). Proof-based testing: QuickCheck (Haskell),	2	Prelegere, recapitulare	

Hypothesis (Python) — property-based testing. Standardele de testare: ISO/IEC 29119 (Software Testing), DO-178C (avionics). Recapitulare generala.			
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Myers, G.J., Sandler, C., Badgett, T., The Art of Software Testing, 3rd ed., Wiley, 2011. 2. Kaner, C., Bach, J., Pettichord, B., Lessons Learned in Software Testing, Wiley, 2001. 3. ISTQB Foundation Level Syllabus v4.0 (2023). Disponibil gratuit la: https://www.istqb.org/certifications/certified-tester-foundation-level <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Humble, J., Farley, D., Continuous Delivery, Addison-Wesley, 2010. 5. Fowler, M., Mocks Aren't Stubs (articol). Disponibil la: https://martinfowler.com/articles/mocksArentStubs.html <p>Resurse software:</p> <ol style="list-style-type: none"> 6. ISTQB Foundation Level Syllabus v4.0 — document oficial gratuit: https://www.istqb.org/certifications/certified-tester-foundation-level 7. JUnit 5 Documentation: https://junit.org/junit5/docs/current/user-guide/ 8. Apache JMeter Documentation: https://jmeter.apache.org/usermanual/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Testare functionala — partitionare si valori de frontiera. Proiectarea cazurilor de test prin partitionarea echivalentei si analiza valorilor de frontiera pentru o functie de validare a unui formular web (campuri numerice, text, data). Calculul complexitatii ciclomatice pentru un modul dat si proiectarea cazurilor de test pentru acoperirea ramurilor. Documentarea in format IEEE 829.	2	Lucrare practica Java/Python	
Lucrarea L2: Testare unitara cu mock-uri. Scrierea testelor unitare cu JUnit 5 sau pytest pentru un serviciu cu dependente externe (baza de date, API extern). Inlocuirea dependentelor cu mock-uri (Mockito sau unittest.mock). Verificarea interactiunilor (verify). Masurarea acoperirii codului cu JaCoCo sau coverage.py. Analiza raportului si adaugarea testelor pentru ramurile neacoperite.	2	Lucrare practica Java/Python	
Lucrarea L3: BDD cu Cucumber sau Behave. Scrierea a 5 scenarii Gherkin (Given-When-Then) pentru un modul de autentificare sau cos de cumparaturi. Implementarea step definitions in Java (Cucumber) sau Python (Behave). Rularea si interpretarea raportului de test. Refactorizarea step definitions pentru reutilizare.	2	Lucrare practica Java/Python	
Lucrarea L4: Testare API REST cu Postman si pytest. Crearea unei colectii Postman cu 10 cereri pentru un API REST public (ex: JSONPlaceholder sau un API local). Scrierea scripturilor de test JavaScript in Postman: validarea statusului HTTP, a schemei JSON, a timpului de raspuns. Exportul colectiei si rularea cu Newman. Reimplementarea a 3 teste cu requests + pytest.	2	Lucrare practica Postman, Python	
Lucrarea L5: Testare web cu Selenium si Page Object Model. Implementarea Page Object Model pentru un site web de test (ex: The	2	Lucrare practica Selenium, Python/Java	

Internet Herokuapp). Automatizarea a 5 scenarii de test: autentificare, navigare, completarea unui formular, upload fisier, drag-and-drop. Configurarea asteptarilor explicite. Rularea in Chrome si Firefox.			
Lucrarea L6: Testare de performanta cu JMeter. Proiectarea unui plan de test JMeter pentru un API REST cu 50/100/200 utilizatori concurenti. Configurarea ramp-up si a duration. Rularea testului si analiza raportului aggregate report: throughput, p95 response time, error rate. Identificarea bottleneck-ului si redactarea unui raport de performanta cu concluzii si recomandari.	2	Lucrare practica JMeter	
Lucrarea L7: Pipeline CI/CD complet si colocviu. Configurarea unui pipeline GitHub Actions care ruleaza automat: testele unitare, masurarea acoperirii (cu threshold minim 80%), analiza statica (SonarCloud sau Checkstyle), testele de integrare. Interpretarea quality gate-ului. Redactarea unui raport de testare complet (test plan, cazuri de test, rezultate, defecte identificate, concluzie de acceptanta). Colocviu oral.	2	Proiect integrat, colocviu	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> JUnit 5 User Guide: https://junit.org/junit5/docs/current/user-guide/ Pytest Documentation: https://docs.pytest.org/ Apache JMeter User Manual: https://jmeter.apache.org/usermanual/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> Selenium WebDriver Documentation: https://www.selenium.dev/documentation/ OWASP Testing Guide v4.2: https://owasp.org/www-project-web-security-testing-guide/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> JUnit 5 Documentation: https://junit.org/junit5/docs/current/user-guide/ Pytest Documentation: https://docs.pytest.org/ Selenium Documentation: https://www.selenium.dev/documentation/ Apache JMeter Documentation: https://jmeter.apache.org/usermanual/ 		

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

Testarea Sistemelor de Calcul formeaza competentele QA (Quality Assurance) cerute explicit in orice echipa software: scrierea testelor unitare si de integrare, automatizarea testarii web si API, testarea de performanta, integrarea testelor in pipeline CI/CD. Aceste competente sunt complementare Ingineriei Programelor si sustin direct calitatea produselor software livrate.

Raportul de testare complet cerut la colocviu — test plan, cazuri de test documentate, rezultatele executiei, defectele identificate si concluzia de acceptanta — formeaza direct competenta R37: asumarea responsabilitatii pentru completitudinea acoperirii si comunicarea clara a rezultatelor catre echipa si beneficiari.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoasterea si aplicarea tehnicilor de proiectare a cazurilor de test (partitionare echivalenta, valori de	Colocviu oral (20 minute): justificarea deciziilor din raportul de testare al proiectului de laborator — alegerea tehnicilor de testare, interpretarea	30%

	<p>frontiera, tabele de decizie); cunoasterea tipurilor de test doubles (stub, mock, fake) si a modului de utilizare; cunoasterea tipurilor de testare de performanta (load, stress, spike) si a metricilor asociate (p95, throughput); cunoasterea principalelor categorii OWASP Top 10 si a instrumentelor SAST/DAST; cunoasterea conceptului de quality gate in CI/CD.</p>	<p>metricilor de acoperire si de performanta, concluzia de acceptanta — cu o intrebare suplimentara despre un scenariu de testare nou</p>	
8.5 Seminar/laborator	<p>Calitatea raportului de testare predat la L7 (test plan cu obiective, cazuri de test documentate in format IEEE 829, rezultatele executiei, lista defectelor cu severitate si prioritate, concluzia de acceptanta motivata); acoperirea codului de minim 80% verificata in pipeline CI/CD; calitatea testelor Selenium (scenariile ruleaza fara interventie manuala); calitatea analizei de performanta JMeter cu interpretarea percentilelor.</p>	<p>Evaluare continua: livrabile individuale predate pe Sakai dupa fiecare lucrare (50%) si raport de testare complet cu colocviu oral individual la L7 (50%)</p>	70%
8.6 Standard minim de performanță:			
<p>Predarea unui raport de testare care include obligatoriu o sectiune de concluzii de acceptanta — o decizie explicita (acceptat/respins) motivata pe baza metricilor masurate (acoperire, rata de defecte, performanta) si a criteriilor de iesire definite in test plan. Rapoartele care listeaza rezultatele testelor fara o concluzie de acceptanta motivata nu indeplinesc standardul minim, indiferent de numarul cazurilor de test executate — aceasta cerinta reflecta direct R37 si diferentiaza testarea de simpla executare a scripturilor.</p>			

Data completării 16.03.2026	Semnătura titularului de curs Lect. Dr. Damian Felicia Anișoara	Semnătura titularului de seminar / laborator Lect. Dr. Damian Felicia Anișoara
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. Dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. Dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei	ALGORITMI PARALELI ȘI DISTRIBUIȚI – TIDS315						
1.2 Titularul activităților de curs	Lect. Dr. Panfiloiu Gheorghe						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Lect. Dr. Panfiloiu Gheorghe						
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					8
Documentare					2
Studiu individual					2
Referate					2
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					50
2.10 Numărul de credite					2

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Proiectarea Algoritmilor (TIDS307) — analiza complexitatii, divide-et-impera, grafuri; Arhitectura Sistemelor de Calcul (TIDS214) — pipeline, cache, memoria ierarhica, SMP; Sisteme de Operare (TIDS302) — thread-uri, procese, sincronizare, IPC.
3.2 de competențe	Cunoasterea notiunilor de thread si proces si a mecanismelor de sincronizare (mutex, semafor); analiza complexitatii algoritmilor secventiali; programare C++ sau Python la nivel mediu; familiarizarea cu modelul de memorie partajata vs. mesagerie.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sala de curs cu tabla, videoproiector si acces la internet; platforma Sakai; recomandare: Python 3.10+ cu multiprocessing, concurrent.futures, mpi4py si numba instalate inainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatica cu calculatoare individuale (minim 4 nuclee fizice); Python 3.10+ cu multiprocessing, mpi4py, numba, ray; C++ cu

OpenMP si MPI (OpenMPI sau MPICH); studentii pot folosi propriul laptop; acces la un cluster de calcul sau la Google Colab pentru GPU.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C8. Cunoaște în profunzime structurile de date și algoritmi fundamentali, inclusiv algoritmi paraleli și distribuți, cu capacitatea de a analiza critic complexitatea și eficiența soluțiilor propuse.
5.2 Aptitudini	A8. Implementează algoritmi paraleli și distribuți, valorificând arhitecturile moderne de procesoare multi-core și infrastructurile de calcul distribuit pentru optimizarea performanței sistemelor.
5.3 Responsabilitate și autonomie	R8. Răspunde în mod responsabil și transparent față de echipă și beneficiari pentru respectarea termenelor, a bugetului tehnic și a specificațiilor agreeate, comunicând proactiv orice deviere sau risc identificat.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere in calculul paralel si distribuit. Motivatie: limitele procesoadelor single-core (legea lui Moore, peretele de frecventa). Definitii: paralelism vs. concurenta, calcul paralel vs. distribuit. Taxonomia Flynn: SISD, SIMD, MISD, MIMD. Metricile de performanta: speed-up ($S=T1/Tp$), eficienta ($E=S/p$), scalabilitatea. Legea lui Amdahl: limita superioara a speed-up-ului in functie de fractia paralela. Legea lui Gustafson: scalarea problemei cu numarul de procesoare.	2	Prelegere interactiva	
Tema 2: Arhitecturi de procesoare multi-core. Arhitectura SMP (Symmetric MultiProcessing): cache L1/L2/L3 partajat vs. privat, bus de memorie, NUMA (Non-Uniform Memory Access). Coerenta cache-ului: protocolul MESI (Modified, Exclusive, Shared, Invalid), false sharing. Hiper-threading (SMT). Arhitecturi manycore: Intel Xeon Phi, GPUs (NVIDIA CUDA cores, AMD GCUs). Ierarhia de memorie in calcul paralel: registre, cache, RAM, DRAM, stocare.	2	Prelegere interactiva	
Tema 3: Modele de programare paralela — memoria partajata. Modelul de memorie partajata: thread-uri, procese, spatiu de adrese comun. POSIX Threads (pthreads): creare, join, mutex, rwlock, conditii. OpenMP: directivele #pragma omp parallel, for, sections, single, critical, atomic, barrier, reduction. Clauzele OpenMP: private, shared, firstprivate, lastprivate, schedule (static, dynamic, guided). Exemplu: paralelizarea unui produs scalar si a unui sort.	2	Prelegere interactiva	
Tema 4: Sincronizare in sisteme cu memorie partajata. Race conditions: definitie, exemple	2	Prelegere interactiva	

<p>clasice (incrementul nesincronizat, contorul bancar). Primitive de sincronizare: mutex, spinlock, semafor, barrier. Deadlock: conditiile Coffman, detectia si evitarea in contextul paralelismului. Variabilele atomice: operatii compare-and-swap (CAS), fetch-and-add. Lock-free si wait-free data structures: stiva lock-free Treiber, coada Michael-Scott. Memory ordering: sequential consistent, acquire-release, relaxed.</p>			
<p>Tema 5: Paradigme de calcul paralel. Paradigma fork-join: pool de thread-uri, task stealing. Paradigma pipeline: etape, throughput, latentia, echilibrarea incarcarii. Paradigma map-reduce: functia map, functia reduce, combinarea rezultatelor. Paradigma producer-consumer: buffer circular cu sincronizare. Paradigma master-worker: distributia dinamica a sarcinilor. Patterns of Parallel Programming (McCool, Robison, Reinders): MapReduce, Stencil, Scan, Pack.</p>	2	Prelegere interactiva	
<p>Tema 6: MPI — Message Passing Interface. Modelul de comunicare prin mesaje: procese independente, spatii de adrese disjuncte. MPI: initializarea, rang, dimensiunea comunicatorului. Comunicatii punct-la-punct: MPI_Send, MPI_Recv, MPI_Isend, MPI_Irecv (asincron), MPI_Sendrecv. Comunicatii colective: MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Allgather, MPI_Reduce, MPI_Allreduce, MPI_Barrier. Tipuri de date MPI derivate. Comunicatii one-sided (RMA): MPI_Put, MPI_Get.</p>	2	Prelegere interactiva	
<p>Tema 7: Algoritmi paraleli clasici. Suma prefixelor (prefix scan) paralel: algoritmul Blelloch (work-efficient), complexitatea $O(n)$ work si $O(\log n)$ span. Sortarea paralela: Merge Sort paralel, Odd-Even Transposition Sort, Bitonic Sort (pentru GPU). Cautarea binara paralela. Inmultirea matricelor: algoritmul Cannon (MPI), Fox, SUMMA. Reducerile in arbore: aplicatii (suma, max, OR logic). Analiza work si span: legea Brent.</p>	2	Prelegere, demonstratii	
<p>Tema 8: Calcul paralel cu GPU — CUDA. Arhitectura GPU NVIDIA: SMs (Streaming Multiprocessors), CUDA cores, warp (32 thread-uri), scheduler. Modelul de programare CUDA: grid, bloc, thread, indexarea 1D/2D/3D. Ierarhia memoriei CUDA: registre, shared memory (SRAM), L1/L2 cache, global memory (DRAM). Kernel-ul CUDA: sintaxa, lansarea, sincronizarea <code>__syncthreads()</code>. Optimizarea accesului la memorie: memory coalescing, bank conflicts. Exemplu: adunarea vectorilor si produsul matriceal tiled.</p>	2	Prelegere interactiva	

<p>Tema 9: Sisteme distribuite — fundament. Definitia si caracteristicile sistemelor distribuite (Tanenbaum): transparenta, scalabilitatea, toleranta la defecte. Comunicatia distribuita: RPC (Remote Procedure Call), gRPC, REST, mesagerie asincron (RabbitMQ, Kafka). Teorema CAP: consistenta, disponibilitate, toleranta la partitionare — imposibilitatea simultana. Consistentă eventuală. Modele de consistentă: linerizabilitate, consistentă secventială, consistentă cauzală.</p>	2	Prelegere interactiva	
<p>Tema 10: Algoritmi distribuiti de consens si coordonare. Problema acordului distribuit (Byzantine Generals). Algoritmi de leader election: Ring Election (Chang-Roberts), Bully Algorithm. Ceasuri logice Lamport: ordinea evenimentelor in sisteme distribuite fara ceas global. Ceasuri vectoriale: cauzalitatea, compararea evenimentelor. Algoritmi de consens: Paxos (principiu), Raft (log replication, leader election, safety). Aplicatii: etcd, ZooKeeper.</p>	2	Prelegere interactiva	
<p>Tema 11: Sisteme de fisiere distribuite si stocarea distribuita. GFS (Google File System) si HDFS (Hadoop Distributed File System): arhitectura, chunk-uri, replicarea, NameNode/DataNode. Teorema CAP aplicata stocarii: Dynamo (Amazon) — consistent hashing, quorum, vector clocks. Baze de date distribuite: sharding, replicarea master-slave si multi-master, Cassandra (eventual consistency), CockroachDB (strong consistency). Apache Kafka: log distribuit, partitii, consumatori, garantiile de livrare.</p>	2	Prelegere interactiva	
<p>Tema 12: Map-Reduce si frameworks de calcul distribuit. Paradigma Map-Reduce (Google): motivatia, faza map, faza shuffle, faza reduce, toleranta la defecte. Apache Hadoop MapReduce: arhitectura YARN, job scheduling. Apache Spark: RDD (Resilient Distributed Dataset), transformarile (map, filter, flatMap, reduceByKey) vs. actiunile (collect, count, saveAsTextFile), DAG-ul de executie, Spark SQL, MLlib. Apache Flink: stream processing vs. batch, windowing.</p>	2	Prelegere interactiva	
<p>Tema 13: Containere, microservicii si cloud computing. Containerizarea: Docker, namespace-uri Linux, cgroups. Orchestrarea: Kubernetes — pod, deployment, service, ingress, HPA (Horizontal Pod Autoscaler). Microservicii: descompunerea monolitului, comunicatia (REST, gRPC, Kafka), service mesh (Istio). Modelele de cloud: IaaS, PaaS, FaaS (serverless). FaaS: AWS Lambda, Azure</p>	2	Prelegere interactiva	

Functions — cold start, stateless, event-driven. Scalarea orizontala vs. verticala.			
Tema 14: Recapitulare si tendinte. Analiza comparativa: calcul paralel pe multi-core vs. GPU vs. cluster distribuit — cand se alege fiecare. Provocarile programarii paralele: debugging, profiling, determinism. Tendinte: neuromorphic computing, calcul cuantic distribuit, edge computing. Framework-uri moderne: Ray (Python), JAX, PyTorch Distributed. Recapitulare generala: Amdahl, paradigme, MPI, CUDA, sisteme distribuite, consens.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Cormen, T.H. et al., Introduction to Algorithms (CLRS), 4th ed., MIT Press, 2022 — capitolele de algoritmi paraleli. 2. Tanenbaum, A.S., Van Steen, M., Distributed Systems: Principles and Paradigms, 3rd ed., Pearson, 2017. Disponibil gratuit la: https://www.distributed-systems.net/ 3. Kirk, D.B., Hwu, W.W., Programming Massively Parallel Processors, 4th ed., Morgan Kaufmann, 2022. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. McCool, M., Robison, A., Reinders, J., Structured Parallel Programming, Morgan Kaufmann, 2012. 5. Kleppmann, M., Designing Data-Intensive Applications, O'Reilly Media, 2017 — capitolele de sisteme distribuite. <p>Resurse software:</p> <ol style="list-style-type: none"> 6. Distributed Systems — carte gratuita Tanenbaum: https://www.distributed-systems.net/ 7. OpenMP Documentation: https://www.openmp.org/specifications/ 8. CUDA C++ Programming Guide (NVIDIA): https://docs.nvidia.com/cuda/cuda-c-programming-guide/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Paralelism cu Python multiprocessing si OpenMP. Implementarea in Python a unui calcul intensiv (estimarea lui pi prin Monte Carlo cu $N=10^7$) cu multiprocessing.Pool si compararea timpilor cu varianta secventiala. Calculul speed-up si eficientei pentru 1, 2, 4 procese. Implementarea aceluiasi calcul in C++ cu OpenMP si compararea rezultatelor. Verificarea legii lui Amdahl pe rezultatele obtinute.	2	Lucrare practica Python, C++/OpenMP	
Lucrarea L2: Sincronizare si race conditions. Demonstrarea unui race condition in Python cu threading: un contor incrementat concurent de 10 thread-uri, fara si cu Lock. Implementarea unui buffer producator-consumator cu Queue (thread-safe). Demonstrarea deadlock-ului: doua thread-uri care achizitioneaza doua lock-uri in ordine inversa. Rezolvarea prin ordonarea consistenta a lock-urilor.	2	Lucrare practica Python	
Lucrarea L3: MPI — comunicatii punct-la-punct si colective. Scrierea programelor MPI cu mpi4py: calculul sumei unui vector distribuit intre 4 procese cu MPI_Reduce. Inmultirea matricei cu un vector distribuita cu MPI Scatter si MPI Gather. Masurarea	2	Lucrare practica Python, mpi4py	

<p> timpilor de comunicare pentru MPI Bcast cu mesaje de diferite dimensiuni (1 KB, 1 MB, 100 MB) si 2, 4, 8 procese.</p>			
<p> Lucrarea L4: Algoritmi paraleli — prefix scan si sortare. Implementarea prefix scan paralel in Python cu multiprocessing: varianta secventiala $O(n)$ vs. varianta paralela Blelloch $O(\log n)$ span. Implementarea merge sort paralel cu concurrent.futures. Compararea timpilor de executie pentru $n = 10^5, 10^6, 10^7$ elemente. Trasarea curbelor speed-up vs. numarul de procese.</p>	2	Lucrare practica Python	
<p> Lucrarea L5: CUDA sau calcul paralel cu Numba. Implementarea adunarii a doua vectori de dimensiune $n=10^6$ cu Numba @cuda.jit. Implementarea produsului matriceal tiled (16x16 shared memory). Masurarea timpilor GPU vs. CPU (NumPy). Profilarea cu nvprof sau Nsight Compute (sau cu Numba profiler). Analiza ocupantei (occupancy) si a memory coalescing.</p>	2	Lucrare practica Python/Numba, CUDA	
<p> Lucrarea L6: Apache Spark — procesarea distribuita a datelor. Configurarea unui cluster Spark local (PySpark) cu 4 workeri. Procesarea unui fisier CSV de 1 GB: filtrare, agregare (groupBy, agg), join cu un alt dataset. Compararea timpului de executie Spark vs. Pandas pentru operatii echivalente. Analiza planului de executie (explain()) si identificarea shuffle-urilor costisitoare.</p>	2	Lucrare practica PySpark	
<p> Lucrarea L7: Proiect integrat si colocviu. Implementarea unui algoritm la alegere cu doua paradigme diferite: (a) paralel pe multi-core (OpenMP sau Python multiprocessing) si (b) distribuit (MPI sau Spark). Exemple: sortarea externa a unui fisier mare, calculul PageRank, simularea Monte Carlo distribuita, produsul matriceal distribuit. Compararea performantei celor doua variante cu curbe speed-up. Raportul tehnic si colocviu oral.</p>	2	Proiect individual, colocviu	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. mpi4py Documentation: https://mpi4py.readthedocs.io/ 2. Numba CUDA Documentation: https://numba.readthedocs.io/en/stable/cuda/ 3. PySpark Documentation: https://spark.apache.org/docs/latest/api/python/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. OpenMP Documentation: https://www.openmp.org/specifications/ 5. CUDA C++ Programming Guide (NVIDIA): https://docs.nvidia.com/cuda/cuda-c-programming-guide/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. mpi4py Documentation: https://mpi4py.readthedocs.io/ 7. Numba CUDA Documentation: https://numba.readthedocs.io/en/stable/cuda/ 8. PySpark Documentation: https://spark.apache.org/docs/latest/api/python/ 9. Distributed Systems (Tanenbaum) — carte gratuita: https://www.distributed-systems.net/ 		

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

Algoritmi Paraleli și Distribuți acoperă spectrul complet al calculului modern scalabil: de la thread-urile pe un procesor multi-core până la cluster-urile Spark cu sute de noduri. Orice aplicație care trebuie să proceseze volume mari de date sau să răspundă în timp real utilizează una dintre paradigmele acoperite în această disciplină.

Proiectul integrat din L7 — implementarea aceluiași algoritm cu două paradigme și compararea performanței — formează direct capacitatea de a alege motivat între calcul paralel local și distribuit, corelată cu A8, și capacitatea de a raporta transparent performanța obținută către echipa, corelată cu R8.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea și aplicarea legii lui Amdahl pentru calculul speed-up maxim; cunoașterea directivelor OpenMP esențiale și a clauzelor de sincronizare; cunoașterea primitivelor MPI colective și punct-la-punct; cunoașterea modelului CUDA (grid, bloc, thread, shared memory); cunoașterea teoremei CAP și a algoritmilor de consens (Raft/Paxos la nivel de principiu); cunoașterea paradigmei Map-Reduce și a arhitecturii Spark.	Colocviu oral (20 minute): justificarea alegerii paradigmei de paralelizare din proiectul integrat — de ce multi-core vs. distribuit, care sunt bottleneck-urile identificate, ce speed-up s-a obținut și de ce diferă de predicția Amdahl — cu o întrebare suplimentară despre un scenariu de paralelizare nou	60%
8.5 Seminar/laborator	Corectitudinea implementărilor (rezultatele paralele sunt identice cu varianta secvențială de referință); calitatea curbelor speed-up din proiectul integrat (măsurate corect, cu variantele secvențiale și paralele rulate în condiții identice); calitatea interpretării în raportul tehnic (explicarea discrepantei față de legea Amdahl); calitatea prezentării la colocviu.	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L7 (50%)	40%
8.6 Standard minim de performanță:			
Includerea în raportul de la L7 a curbei speed-up măsurată pentru cel puțin 3 configurații (1, 2 și 4 procese/nuclee), însoțită de aplicarea legii lui Amdahl pentru estimarea fracției paralele a			

implementării. Implementările care produc rezultate incorecte sau care nu includ măsurați de performanță comparabile cu varianta secvențială nu îndeplinesc standardul minim, indiferent de complexitatea codului.

Data completării 16.03.2026	Semnătura titularului de curs Lect. Dr. Panfiloiu Gheorghe	Semnătura titularului de seminar / laborator Lect. Dr. Panfiloiu Gheorghe
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. Dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. Dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei		ALGORITMI PARALELI ȘI DISTRIBUIȚI - PROIECT-TIDS316					
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator / proiect		Lect. Dr. Lițan Daniela -Elena					
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	P	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	1	din care: 2.2 curs		2.3 seminar/laborator/proiect	0/0/1
2.4 Total ore din planul de învățământ	14	din care: 2.5 curs		2.6 seminar/laborator/proiect	0/0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					
Practică					
Elaborare proiect diplomă					
2.8 Activități individuale					11
Documentare					4
Studiu individual					3
Referate					
Teme casă					
Proiect					2
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					25
2.10 Numărul de credite					1

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Algoritmi Paraleli si Distribuiti (TIDS315) — legea Amdahl, OpenMP, MPI, CUDA, Spark, sisteme distribuite; Proiectarea Algoritmilor (TIDS307) — analiza complexitatii, structuri de date avansate.
3.2 de competențe	Capacitatea de a implementa si rula un program paralel cu Python multiprocessing sau OpenMP; cunoasterea structurii unui program MPI (rang, communicate, scatter/gather); familiarizarea cu Spark sau Ray; abilitatea de a masura si interpreta speed-up-ul unui program paralel.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	
4.2 de desfășurare a seminarului/laboratorului	Activitatea de proiect se desfasoara individual, la calculatorul propriu sau in laboratorul de informatica; Python cu multiprocessing, mpi4py, numba, ray si PySpark instalate; intalnirile de indrumare saptamanale au loc fata in fata sau

	online, cu prezentarea stadiului si primirea feedback-ului; predarea finala pe platforma Sakai.
--	---

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C8. Cunoaște în profunzime structurile de date și algoritmi fundamentali, inclusiv algoritmi paraleli și distribuți, cu capacitatea de a analiza critic complexitatea și eficiența soluțiilor propuse.
5.2 Aptitudini	A8. Implementează algoritmi paraleli și distribuți, valorificând arhitecturile moderne de procesoare multi-core și infrastructurile de calcul distribuit pentru optimizarea performanței sistemelor.
5.3 Responsabilitate și autonomie	R8. Răspunde în mod responsabil și transparent față de echipă și beneficiari pentru respectarea termenelor, a bugetului tehnic și a specificațiilor agreeate, comunicând proactiv orice deviere sau risc identificat.

6. Conținuturi

6.1 Proiect	Nr. ore	Metode de predare	Observații
P1: Alegerea temei si definirea cerintelor. Prezentarea categoriilor de proiecte: algoritmi paraleli pe multi-core (sortare, cautare, calcul numeric), algoritmi distribuiti (MapReduce, procesare grafica de grafuri), simulari Monte Carlo distribuite, procesarea distribuita a datelor cu Spark. Alegerea temei individuale si redactarea documentului de cerinte: problema, datele de intrare/iesire, metrica de performanta.	1	Indrumare individuala	
P2: Analiza algoritmului secvential de referinta. Implementarea varianta secventiala completa. Analiza complexitatii: O mare pentru timp si spatiu. Profilarea executiei cu cProfile (Python) sau perf (Linux) pentru identificarea bottleneck-urilor. Calculul fractiei teoretic paralele a algoritmului. Documentarea bazei de referinta.	1	Indrumare, feedback	
P3: Proiectarea solutiei paralele. Alegerea paradigmei de paralelizare: fork-join (multiprocessing), pipeline, map-reduce, master-worker. Identificarea sarcinilor independente si a dependentelor (graful de dependente). Alegerea granularitatii: sarcini fine vs. grosiere — compromisul intre overhead de comunicatie si grad de paralelism. Estimarea speed-up-ului teoretic (Amdahl) pentru numarul de nuclee disponibil.	1	Indrumare, feedback	
P4: Implementarea versiunii paralele pe multi-core. Implementarea cu Python multiprocessing (Pool, Process, Queue) sau C++/OpenMP, conform alegerii din P3. Testarea corectitudinii: compararea rezultatelor cu varianta secventiala pe seturi de date de test. Prima masurare a speed-up-ului pentru 1 si 2 procese/nuclee.	1	Implementare, indrumare	
P5: Masurarea si imbunatatirea performantei multi-core. Masurarea speed-up-ului pentru 1,	1	Implementare, profilare	

2 si 4 procese/nuclee pe seturi de date de diferite dimensiuni. Identificarea si rezolvarea problemelor de performanta: false sharing, overhead de sincronizare, dezechilibrare de sarcina (load imbalance). Aplicarea unei optimizari documentate si re-masurarea.			
P6: Implementarea versiunii distribuite — MPI sau Spark. Implementarea celei de-a doua variante paralele: cu mpi4py (distributia datelor, comunicatii colective) sau PySpark (RDD-uri, transformari, actiuni). Testarea corectitudinii pe 2-4 procese/noduri. Identificarea diferentelor de implementare fata de varianta multi-core.	1	Implementare, indrumare	
P7: Masurarea performantei distribuite. Masurarea timpilor de executie pentru varianta distribuita pe 1, 2 si 4 procese MPI sau noduri Spark. Masurarea overhead-ului de comunicare: cata parte din timp este transferul de date. Compararea cu varianta multi-core: care este mai rapida si de ce. Trasarea primelor curbe speed-up comparative.	1	Implementare, masuratori	
P8: Prezentare intermediara. Prezentarea in 10 minute a stadiului proiectului: problema, varianta secventiala (complexitate, profil), varianta multi-core (implementare, speed-up), varianta distribuita (implementare, prime masuratori). Feedback din partea titularului pe corectitudinea implementarilor si pe directia de imbunatatire.	1	Prezentare intermediara	
P9: Optimizarea variantei distribuite. Aplicarea optimizarilor identificate in urma prezentarii intermediare: reducerea comunicatiei prin colectie de date locale, partitionarea eficienta (Spark), evitarea shuffle-urilor inutile. Re-masurarea si compararea cu versiunea neoptimizata. Documentarea optimizarilor aplicate.	1	Implementare, optimizare	
P10: Analiza comparativa si interpretarea rezultatelor. Completarea tuturor masurarilor: speed-up multi-core si distribuit pentru 1, 2, 4 procese pe seturi de date mici, medii si mari. Calculul eficientei ($E=S/p$) pentru fiecare configuratie. Aplicarea legii Amdahl pentru estimarea fractiei paralele a implementarii. Identificarea discrepantei intre speed-up teoretic si real si explicarea cauzelor.	1	Analiza, documentare	
P11: Scalabilitatea — strong scaling si weak scaling. Masurarea scalabilitatii tari (strong scaling): timp de executie cu aceeasi problema, mai multi procesori. Masurarea scalabilitatii slabe (weak scaling, legea Gustafson): marimea problemei creste proportional cu numarul de procesori. Trasarea curbelor si interpretarea: ce tip de scalabilitate se obtine si de ce.	1	Masuratori, analiza	

P12: Redactarea documentatiei tehnice. Structura documentatiei: introducere (problema si motivatia), analiza algoritmului secvential (complexitate, profil), proiectarea solutiei paralele (paradigma aleasa, justificare), implementarile (multi-core si distribuita, pseudocode), rezultatele (tabele si grafice speed-up/eficienta), concluzii (limitele solutiei, directii de imbunatatire). Inserarea graficelor de performanta.	1	Redactare tehnica	
P13: Revizuirea si finalizarea. Revizuirea documentatiei pe baza feedback-ului de la P12. Curatarea codului sursă: comentarii, structurare, README cu instructiuni de rulare. Testarea finala pe toate configuratiile. Pregatirea prezentarii orale: diapozitive, demonstratie live a rularii proiectului.	1	Revizuire, finalizare	
P14: Prezentarea si evaluarea proiectului final. Prezentare orala (10 minute + 5 minute Q&A): problema, deciziile de proiectare (paradigma aleasa si justificarea), demonstratie live a rularii pe cel putin 2 configuratii, curbele de performanta. Predarea documentatiei si a codului sursă pe Sakai.	1	Prezentare orala, evaluare	
	<p>Bibliografie obligatorie proiect</p> <ol style="list-style-type: none"> 1. Cormen, T.H. et al., Introduction to Algorithms (CLRS), 4th ed., MIT Press, 2022. 2. Tanenbaum, A.S., Van Steen, M., Distributed Systems: Principles and Paradigms, 3rd ed., Pearson, 2017. Disponibil gratuit la: https://www.distributed-systems.net/ 3. mpi4py Documentation: https://mpi4py.readthedocs.io/ <p>Bibliografie complementara proiect</p> <ol style="list-style-type: none"> 4. Kirk, D.B., Hwu, W.W., Programming Massively Parallel Processors, 4th ed., Morgan Kaufmann, 2022. 5. PySpark Documentation: https://spark.apache.org/docs/latest/api/python/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. Distributed Systems (Tanenbaum) — carte gratuita: https://www.distributed-systems.net/ 7. mpi4py Documentation: https://mpi4py.readthedocs.io/ 8. PySpark Documentation: https://spark.apache.org/docs/latest/api/python/ 9. Ray Documentation — framework distribuit Python: https://docs.ray.io/ 		

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

Proiectul integreaza competentele din TIDS315 intr-un ciclu complet de proiectare: analiza algoritmului secvential, implementarea a doua variante paralele (multi-core si distribuita), masurarea si compararea performantei, si documentarea tehnica. Parcurgerea autonoma a acestui ciclu reproduce contextul unui proiect de optimizare real din industrie.

Analiza comparativa ceruta (strong scaling, weak scaling, eficienta) si documentarea transparenta a discrepantei fata de predictia Amdahl formeaza direct competenta R8: responsabilitatea si transparenta in comunicarea performantei catre echipa si beneficiari.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs			

8.5 seminar/laborator/proiect	Corectitudinea ambelor implementari paralele (rezultatele identice cu varianta secventiala de referinta pe toate seturile de test); calitatea si completitudinea curbelor de performanta (speed-up si eficienta pentru 1, 2 si 4 procese, pe cel puțin 2 dimensiuni ale problemei); calitatea interpretarii rezultatelor in documentatie (discrepanta fata de Amdahl explicata, tipul de scalabilitate identificat); calitatea prezentarii orale si a demonstratiei live.	Proiect individual evaluat la P14: demonstratie live a rularii (30%) + documentatie predata pe Sakai (40%) + prezentare orala cu Q&A de 15 minute (30%)	100%
----------------------------------	---	---	-------------

8.6 Standard minim de performanță:

Predarea a doua implementari paralele distincte (una multi-core si una distribuita) care produc rezultate corecte — verificate prin compararea cu varianta secventiala — si includerea curbei speed-up masurate pentru cel puțin 3 configuratii. Un proiect cu o singura paradigma de paralelizare sau fara masuratori de performanta nu indeplineste standardul minim, indiferent de calitatea documentatiei.

Data completării 19.03.2026	Semnătura titularului de curs	Semnătura titularului de seminar / laborator/proiect Lect. Dr. Lițan Daniela-Elena
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei				PRACTICĂ III– TIDS318			
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator / proiect / practică				Prof. Dr. Ariton Viorel			
1.5 Anul de studiu	II	1.6 Semestrul	2	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână		din care: 2.2 curs		2.3 seminar/laborator/proiect	
2.4 Total ore din planul de învățământ	60 ore	din care: 2.5 curs		2.6 seminar/laborator/proiect./ practică	60 ore
Distribuția fondului de timp					ore
2.7 Activități monitorizate					60
Practică					60
Elaborare proiect diplomă					
2.8 Activități individuale					15
Documentare					4
Studiu individual					
Referate					
Teme casă					
Proiect					7
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Practica II (TIDS317) — experiența anterioară de practică; Algoritmi Paraleli și Distribuți (TIDS315); Testarea Sistemelor de Calcul (TIDS314); Ingineria Programelor (TIDS308); Baze de Date (TIDS305).
3.2 de competențe	Capacitatea de a lucra autonom pe sarcini de complexitate medie-ridică; experiența cu Git și fluxul de lucru agil dobândite la Practica II; cunoașterea disciplinelor din An III; abilitatea de a redacta documentație tehnică și de a comunica progresul tehnic echipei.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	
4.2 de desfășurare a seminarului/laboratorului	Practica se desfășoară la organizația gazdă (companie parteneră sau instituție publică), în ritm de aproximativ 4-5 ore pe săptămână, pe parcursul celor 14 săptămâni ale semestrului (60 ore total). Față de Practica II, studentul preia responsabilități mai mari: contribuții la funcționalități de complexitate ridicată,

	participare activă la decizii tehnice, mentorizarea unor colegi mai tineri. Activitățile se desfășoară la sediul organizației sau parțial în regim remote, cu acordul tutorelui.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C25. Posedă cunoștințe integrate și critice asupra domeniului tehnologiei informației ca întreg, cu capacitatea de a identifica conexiunile dintre subdomeniile disciplinei, de a evalua soluții tehnice alternative și de a formula direcții de cercetare sau dezvoltare originale, la avangarda domeniului.
5.2 Aptitudini	A25. Gestionează proiecte de inginerie software de complexitate medie și ridicată, planificând activitățile, alocând resursele, monitorizând progresul și aplicând metode agile pentru livrarea la termen a produselor software de calitate.
5.3 Responsabilitate și autonomie	R25. Își asumă în mod conștient și responsabil rolul de profesionist în domeniul tehnologiei informației ca agent al transformării digitale a societății, contribuind la dezvoltarea durabilă a comunității prin soluții informatice inovatoare, sigure și etice.

6. Conținuturi

6.1 Practică	Nr. ore	Metode de predare	Observații
Săptămânile 1-2: Integrare și preluarea responsabilităților. Reluarea contactului cu organizația gazdă (sau integrarea într-o nouă organizație). Cunoașterea echipei, a proiectului și a obiectivelor semestrului. Identificarea sarcinilor de complexitate ridicată care vor fi asumate. Configurarea mediului de lucru și revizuirea arhitecturii proiectului.	8	Activitate la organizația gazdă	
Săptămânile 3-5: Implementare — faza 1. Implementarea independentă a funcționalităților alocate. Aplicarea bunelor practici de inginerie software: design patterns, testare automată, documentare. Participarea activă la ceremoniile agile: sprint planning, daily standup, retrospectivă. Documentarea săptămânală a activităților în caietul de practică.	12	Activitate la organizația gazdă	
Săptămânile 6-8: Implementare — faza 2 și colaborare tehnică. Continuarea implementării. Participarea la decizii tehnice ale echipei: alegerea arhitecturii, soluționarea problemelor tehnice deschise. Contribuții la documentația internă a proiectului: diagrame, specificații, README. Deschiderea și revizuirea de pull request-uri, integrarea feedback-ului.	12	Activitate la organizația gazdă	
Săptămânile 9-11: Contribuții avansate și calitate. Asumarea unor sarcini transversale: optimizarea performanței, refactorizarea modulelor existente, creșterea acoperirii cu teste. Participarea la activități de code review pentru alți membri ai echipei. Identificarea și	12	Activitate la organizația gazdă	

comunicarea proactivă a riscurilor tehnice identificate în proiect.			
Săptămânile 12-13: Finalizare și documentare. Finalizarea tuturor sarcinilor asumate. Redactarea documentației tehnice finale: descrierea contribuțiilor, arhitectura componentelor implementate, instrucțiuni de utilizare. Redactarea secțiunilor principale ale raportului de practică: contextul organizației, activitățile desfășurate, competențele dobândite.	8	Activitate la organizația gazdă	
Săptămâna 14: Evaluare finală și predare. Prezentarea contribuțiilor în fața echipei din organizație. Obținerea evaluării și a semnăturii tutorelui. Completarea caietului de practică cu reflecția personală. Predarea raportului final și a caietului de practică pe platforma Sakai.	8	Activitate la organizația gazdă	
	<p>Bibliografie obligatorie practică</p> <ol style="list-style-type: none"> 1. Regulamentul de practică al Universității Internaționale Danubius — disponibil pe platforma Sakai. 2. Convenția de practică individuală — model disponibil la secretariatul facultății. 3. Ghidul de redactare a caietului și a raportului de practică — disponibil pe platforma Sakai. <p>Bibliografie complementara practică</p> <ol style="list-style-type: none"> 4. Sommerville, I., Software Engineering, 10th ed., Pearson, 2015 — capitolele privind managementul proiectelor și lucrul în echipă. 5. Kim, G., Humble, J., Behr, P., Forsgren, N., The DevOps Handbook, 2nd ed., IT Revolution Press, 2021. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. GitHub Documentation — ghid complet Git și colaborare: https://docs.github.com/ 7. Atlassian Git Tutorials — fluxuri de lucru în echipă: https://www.atlassian.com/git/tutorials 8. Scrum Guide — ghid oficial Scrum gratuit: https://scrumguides.org/ 		

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

Practica III plasează studentul din ultimul an de studiu într-un rol profesional cu responsabilități sporite față de practicile anterioare: contribuții la funcționalități de complexitate ridicată, participare la decizii tehnice și activități de code review. Cele 60 de ore distribuite pe parcursul semestrului permit integrarea și aplicarea competențelor acumulate de-a lungul întregului program de studiu. Asumarea unui rol activ în echipă, comunicarea proactivă a riscurilor tehnice și documentarea riguroasă a contribuțiilor formează direct competențele A25 și R25 din matricea de corelare.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs			
8.5 Proiect / practică	Prezența și activitatea la organizația gazdă (60 ore confirmate în caietul de practică cu semnăturile tutorelui, săptămână cu săptămână); calitatea contribuțiilor tehnice de	Evaluare pe baza: caietului de practică completat și semnat de tutore (40%) + raportul final predat pe Sakai (30%) + evaluarea tutorelui din organizație (30%). Nota finală se acordă după confirmarea celor 60 de ore.	100%

	complexitate ridicată (cel puțin două funcționalități finalizate sau o contribuție transversală confirmată de tutore); calitatea și completitudinea caietului de practică; calitatea raportului final.		
--	--	--	--

8.6 Standard minim de performanță:

Confirmarea celor 60 de ore de practică prin caietul de practică completat săptămânal și semnat de tutorele din organizație, însoțit de cel puțin două contribuții tehnice de complexitate ridicată documentate. Un caiet de practică fără semnătura tutorelui sau care descrie activități fără relevanță tehnică nu îndeplinește standardul minim, indiferent de calitatea raportului final.

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar / laborator/proiect/practică Prof. Dr. Ariton Viorel
17.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. Dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. Dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei		ELEMENTE DE GRAFICĂ PE CALCULATOR – TIDS321					
1.2 Titularul activităților de curs		Conf. Dr. Bucea Radu Manea Țonis					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Conf. Dr. Bucea Radu Manea Țonis					
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Op DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Matematici Discrete (TIDS205) — algebră liniară (vectori, matrice, transformări), geometrie analitică; Programare Orientată pe Obiecte (TIDS204) — C++ sau Python la nivel mediu; Achiziția și Prelucrarea Datelor (TIDS304) — reprezentarea semnalelor, transformata Fourier.
3.2 de competențe	Cunoașterea algebrei liniare: înmulțirea matricelor, vectori în 3D, produsul vectorial; familiarizarea cu cel puțin un limbaj de programare (C++ sau Python); înțelegerea sistemelor de coordonate 2D și 3D; abilitatea de a instala și utiliza o bibliotecă grafică (OpenGL sau Pygame).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă pentru demonstrații matematice, videoproiector și acces la internet; platforma Sakai; recomandare: Python cu Pygame și PyOpenGL sau C++ cu GLFW/GLEW instalate înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; plăci grafice cu suport OpenGL 3.3+; Python cu Pygame, PyOpenGL, NumPy și Pillow

	instalate; alternativ C++ cu GLFW, GLEW și GLM; IDE (VS Code sau Visual Studio); studenții pot folosi propriul laptop.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C18. Deține cunoștințe specializate de prelucrare a imaginilor și grafică pe calculator, inclusiv tehnici de recunoaștere a formelor și viziune artificială, situate la avangarda aplicațiilor ingineresti actuale.
5.2 Aptitudini	A18. Aplică tehnici de prelucrare a imaginilor și viziune artificială pentru dezvoltarea de aplicații ingineresti, utilizând biblioteci și framework-uri specializate de procesare a semnalelor vizuale.
5.3 Responsabilitate și autonomie	R18. Își asumă responsabilitatea pentru impactul social și etic al soluțiilor informatice dezvoltate, evaluând critic consecințele tehnologice asupra utilizatorilor, comunității și mediului și propunând măsuri de minimizare a riscurilor identificate.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în grafica pe calculator. Istoricul graficii pe calculator. Domeniile de aplicare: jocuri video, CAD, simulare, vizualizare medicală, film de animație. Pipeline-ul grafic 2D și 3D: geometrie, rasterizare, shading, output. Sisteme de coordonate: ecran, normalizat, lume, obiect. Modele de culoare: RGB, CMYK, HSV, spațiile de culoare sRGB și liniar.	2	Prelegere interactivă	
Tema 2: Transformări 2D. Translația, scalarea și rotația în 2D: formulele matriceale. Coordonatele omogene: necesitate și avantaje. Matricea de transformare 2D în coordonate omogene (3×3). Compoziția transformărilor: ordinea de aplicare. Transformarea inversă. Rotația față de un punct arbitrar (compoziție T-R-T ⁻¹).	2	Prelegere, demonstrații matematice	
Tema 3: Transformări 3D și proiecții. Translația, scalarea și rotația în 3D (matrici 4×4 în coordonate omogene). Rotații Euler: gimbal lock, quaternioni — definiție și avantaje. Proiecția ortografică: matricea, utilizările (CAD, jocuri 2.5D). Proiecția perspectivă: camera pinhole, matricea de perspectivă, frustrumul de vizualizare.	2	Prelegere, demonstrații matematice	
Tema 4: Pipeline-ul grafic OpenGL. Arhitectura OpenGL 3.3+ core profile: VBO, VAO, EBO. Vertex shader și fragment shader: limbajul GLSL. Etapele pipeline-ului: vertex processing, primitive assembly, rasterization, fragment processing, output merger. Compilarea și linkarea shader-elor. Trimiterea datelor din CPU în GPU: uniform variables, attribute.	2	Prelegere interactivă	
Tema 5: Iluminare și materiale. Modelul de iluminare Phong: componentele ambientală, difuză și speculară. Normale la suprafețe:	2	Prelegere interactivă	

calculul și transformarea corectă (matricea normală = transpusa inversului matricei model). Tipuri de lumini: punctuală, direcțională, spot. Implementarea Phong în GLSL. Modelul Blinn-Phong: diferența față de Phong. Materiale: proprietățile ka, kd, ks, shininess.			
Tema 6: Texturare. Coordonate UV: definiție, unwrapping. Texture mapping în OpenGL: crearea texturii, binding, parametri de filtrare (GL_NEAREST, GL_LINEAR, mipmapping). Filtrarea bilineară și trilineară. Wrapping: GL_REPEAT, GL_CLAMP_TO_EDGE. Multi-texturing: combinarea mai multor texturi în fragment shader. Normal mapping: ideea și implementarea.	2	Prelegere interactivă	
Tema 7: Culori și spații de culoare. Percepția vizuală umană: fotoreceptori, curbele de sensibilitate spectrală. Spațiul CIE XYZ: diagramul de cromaticitate. Conversia RGB ↔ XYZ. Gamma correction: spațiul sRGB, codarea gamma 2.2, importanța pentru pipeline-ul liniar. HDR (High Dynamic Range): reprezentarea în float16/float32, tone mapping (Reinhard, ACES). Compositing: canalul alpha, blending (over operator).	2	Prelegere interactivă	
Tema 8: Curbe și suprafețe. Curbe Bézier: puncte de control, polinomul Bernstein, algoritmul De Casteljau, gradul curbei. Curbe B-spline: vectorul nodal, proprietățile locale. NURBS: ponderile raționale, importanța în CAD. Suprafețe Bézier bicu bice: grila de control, evaluarea. Suprafețe de rotație și de extrudare.	2	Prelegere interactivă	
Tema 9: Modele 3D și meshuri. Reprezentarea meshurilor: liste de vârfuri și fețe, winged-edge, half-edge. Formatul OBJ: structura, grupuri, materiale MTL. Formatul glTF: standardul modern pentru web și gaming. Încărcarea meshurilor în OpenGL: utilizarea bibliotecii Assimp. Calculul normalelor per vertex. Subdiviziunea suprafețelor: algoritmul Catmull-Clark.	2	Prelegere interactivă	
Tema 10: Prelucrarea imaginilor — fundamente. Imaginea digitală: pixeli, rezoluție, adâncimea de culoare. Operații punctuale: luminozitate, contrast, histograma, egalizarea histogramei. Operații pe vecinătate: convoluția discretă 2D, kernel-uri (blur, sharpen, emboss). Filtrul Gaussian: separabilitatea. Filtrul median: reducerea zgomotului impulsiv. Biblioteca OpenCV: structura Mat, operații de bază.	2	Prelegere interactivă	
Tema 11: Detectarea marginilor și segmentarea. Gradientul imaginii: operatorii	2	Prelegere interactivă	

Sobel și Prewitt. Detectorul de margini Canny: pașii (Gaussian blur, gradient, NMS, double threshold, hysteresis). Transformata Hough pentru linii și cercuri. Segmentarea prin prag (thresholding): Otsu. Segmentarea prin regiuni: creșterea regiunilor. Morfologie matematică: eroziunea și dilatarea.			
Tema 12: Caracteristici și recunoaștere vizuală. Detectoare de caracteristici: Harris corner detector. Descriptori SIFT: orientare, histograma gradientilor. Descriptori SURF și ORB (mai rapid). Potrivirea caracteristicilor: brute-force, FLANN, filtrarea Lowe's ratio test. Omografia: calculul cu RANSAC. Aplicație: panorame foto, AR markers.	2	Prelegere interactivă	
Tema 13: Rețele neuronale convoluționale pentru grafică. Rețelele CNN: stratul convoluțional, pooling, fully connected. Clasificarea imaginilor: AlexNet, VGG, ResNet — evoluția arhitecturilor. Transfer learning: fine-tuning cu PyTorch/TensorFlow. Detecția obiectelor: YOLO, principiul. Segmentarea semantică: U-Net. Impactul etic al viziunii artificiale: bias în datele de antrenament, deepfakes.	2	Prelegere interactivă	
Tema 14: Rendering avansat și recapitulare. Ray tracing: principiul, ray-sphere intersection, reflexie, refracție, umbre. Comparatie rasterizare vs. ray tracing: viteză, calitate, hardware dedicat (NVIDIA RTX). Path tracing: ecuația de rendering, sampling Monte Carlo. Tehnici real-time moderne: ambient occlusion (SSAO), reflecții (SSR), global illumination aproximat (DDGI). Recapitulare generală.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Shirley, P., Marschner, S., Fundamentals of Computer Graphics, 4th ed., CRC Press, 2015. 2. Angel, E., Shreiner, D., Interactive Computer Graphics: A Top-Down Approach with WebGL, 7th ed., Pearson, 2014. 3. Bradski, G., Kaehler, A., Learning OpenCV 3, O'Reilly Media, 2016. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Foley, J.D. et al., Computer Graphics: Principles and Practice, 3rd ed., Addison-Wesley, 2013. 5. Szeliski, R., Computer Vision: Algorithms and Applications, 2nd ed., Springer, 2022. Disponibil gratuit la: https://szeliski.org/Book/ <p>Resurse software:</p> <ol style="list-style-type: none"> 6. OpenGL Documentation — referință completă OpenGL: https://www.opengl.org/documentation/ 7. LearnOpenGL — tutoriale moderne OpenGL 3.3+: https://learnopengl.com/ 8. OpenCV Documentation — prelucrarea imaginilor: https://docs.opencv.org/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Transformări 2D în Pygame. Implementarea în Python/Pygame a transformărilor 2D (translație, rotație, scalare) prin înmulțire de matrice 3×3 cu NumPy. Animarea unui obiect 2D compus (braț robotic cu 2 segmente) prin compoziția	2	Lucrare practică Python/Pygame	

transformărilor. Vizualizarea interactivă prin tastatură.			
Lucrarea L2: Transformări 3D și proiecții în OpenGL. Configurarea unui proiect OpenGL 3.3 core profile. Implementarea transformărilor 3D cu GLM (model, view, projection). Orbita camerei în jurul unui cub colorat. Comutarea între proiecția perspectivă și ortografică.	2	Lucrare practică Python/PyOpenGL	
Lucrarea L3: Shader-e și iluminare Phong. Scrierea vertex shader-ului și fragment shader-ului GLSL pentru modelul Phong. Implementarea luminii direcționale și punctuale. Modificarea interactivă a parametrilor materialului (ambientalul, difuzia, specularul, shininess). Compararea Phong vs. Blinn-Phong.	2	Lucrare practică OpenGL/GLSL	
Lucrarea L4: Texturare și încărcarea modelelor 3D. Aplicarea unei texturi UV pe un cub și pe o sferă. Configurarea parametrilor de filtrare și wrapping. Încărcarea unui model OBJ cu Assimp și afișarea lui cu textură. Normal mapping pe un plan: efectul vizual față de geometrie plată.	2	Lucrare practică OpenGL	
Lucrarea L5: Prelucrarea imaginilor cu OpenCV. Operații de bază: citire/scriere imagine, conversii spații de culoare (BGR/RGB/HSV). Egalizarea histogramei. Filtre de convoluție (Gaussian blur, Sobel, Laplacian). Detectia marginilor Canny. Detectarea și afișarea conturilor cu findContours.	2	Lucrare practică Python/OpenCV	
Lucrarea L6: Detectarea caracteristicilor și potrivirea imaginilor. Detectarea colțurilor Harris și a caracteristicilor ORB cu OpenCV. Potrivirea caracteristicilor între două imagini cu brute-force matcher și filtrul Lowe's ratio test. Calculul omografiei cu RANSAC. Proiectarea (warping) unei imagini peste alta.	2	Lucrare practică Python/OpenCV	
Lucrarea L7: Proiect integrat și colocviu. Implementarea unei aplicații grafice la alegere: (a) un mini-motor 3D cu iluminare, texturare și model OBJ încărcat sau (b) un pipeline de viziune artificială (detectie de obiecte sau reconstrucție panoramă). Documentarea: descrierea algoritmilor folosiți, capturile de ecran, analiza impactului etic al aplicației. Colocviu oral.	2	Proiect individual, colocviu	

Bibliografie obligatorie laborator

1. LearnOpenGL — tutoriale complete OpenGL moderne: <https://learnopengl.com/>
2. OpenCV Documentation — referință Python/C++: <https://docs.opencv.org/>
3. Shirley, P., Marschner, S., Fundamentals of Computer Graphics, 4th ed., CRC Press, 2015.

Bibliografie complementara laborator

4. Bradski, G., Kaehler, A., Learning OpenCV 3, O'Reilly Media, 2016.
5. Szeliski, R., Computer Vision: Algorithms and Applications, 2nd ed., Springer, 2022. Disponibil gratuit la: <https://szeliski.org/Book/>

Resurse online gratuite recomandate

6. LearnOpenGL — tutoriale OpenGL 3.3+: <https://learnopengl.com/>
7. OpenCV Documentation: <https://docs.opencv.org/>
8. The Book of Shaders — tutoriale interactive GLSL: <https://thebookofshaders.com/>
9. Shadertoy — mediu online de experimentare cu shader-e: <https://www.shadertoy.com/>

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

Grafica pe Calculator și Prelucrarea Imaginilor acoperă două domenii complementare cu aplicații industriale directe: motoarele grafice 3D (jocuri, simulare, CAD) și viziunea artificială (inspecție industrială, conducere autonomă, medicină). Aceste competențe sunt solicitate explicit în industria software din România și din UE.

Analiza impactului etic cerută la colocviu — consecințele deepfake-urilor, biasul din sistemele de recunoaștere facială, impactul viziunii artificiale asupra intimității — formează direct competența R18, care diferențiază disciplina de un simplu curs tehnic de programare grafică.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea și aplicarea transformărilor geometrice 2D și 3D cu matrice în coordonate omogene; cunoașterea pipeline-ului grafic OpenGL și scrierea shader-elor GLSL de bază; cunoașterea modelului de iluminare Phong și a tehnicilor de texturare; cunoașterea operațiilor fundamentale de prelucrare a imaginilor (histogramă, convoluție, Canny); cunoașterea descriptorilor de caracteristici (ORB, SIFT) și potrivirea lor; evaluarea impactului etic al aplicațiilor de viziune artificială.	Colocviu oral (20 minute): prezentarea și justificarea deciziilor tehnice din proiectul integrat (alegerea algoritmilor, trade-off-urile performanță/calitate) și a analizei de impact etic a aplicației, cu o întrebare suplimentară din materia de curs	40%
8.5 Seminar/laborator	Funcționalitatea aplicației grafice din proiectul integrat (rulează fără erori, produce rezultatul vizual corect); calitatea codului (shader-ele sunt corecte, pipeline-ul OpenGL este configurat corect sau prelucrările OpenCV produc rezultatele așteptate);	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L7 (50%)	60%

	calitatea secțiunii de impact etic din documentație (cel puțin două consecințe identificate cu măsuri propuse de minimizare); calitatea prezentării la colocviu.		
--	--	--	--

8.6 Standard minim de performanță:

Predarea unui proiect integrat funcțional care rulează fără erori și produce un rezultat vizual demonstrabil (scena 3D afișată corect sau pipeline-ul de viziune artificială detectează/procesează corect pe imaginile de test), însoțit de o secțiune de impact etic de cel puțin un paragraf. Un proiect care compilează dar nu produce output vizual corect sau care nu include analiza de impact nu îndeplinesc standardul minim, indiferent de calitatea codului sursă.

Data completării	Semnătura titularului de curs Conf. Dr. Bucea Radu Manea Țonis	Semnătura titularului de seminar / laborator Conf. Dr. Bucea Radu Manea Țonis
17.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei		PRELUCRARE GRAFICĂ – TIDS322					
1.2 Titularul activităților de curs		Conf. Dr. Bucea Radu Manea Țonis					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Conf. Dr. Bucea Radu Manea Țonis					
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Op DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Achiziția și Prelucrarea Datelor (TIDS304) — eșantionare, transformata Fourier, filtre digitale; Proiectarea Algoritmilor (TIDS307) — complexitate, structuri de date; Ingineria Programelor (TIDS308) — utilizarea bibliotecilor și framework-urilor software.
3.2 de competențe	Cunoașterea reprezentării digitale a imaginilor (pixeli, canale, adâncimea de culoare); familiarizarea cu Python la nivel mediu și cu NumPy; înțelegerea noțiunii de convoluție; cunoașterea de bază a rețelelor neuronale (structură, propagare înainte, backpropagation).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă, videoproiector și acces la internet; platforma Sakai; recomandare: Python 3.10+ cu OpenCV, NumPy, Matplotlib, scikit-image, PyTorch și Jupyter Notebook instalate înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale sau acces la Google Colab (cu GPU gratuit); Python cu OpenCV, scikit-image, PyTorch,

torchvision și Albumentations instalate; studenții pot folosi propriul laptop sau Google Colab pentru lucrările de deep learning.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C18. Deține cunoștințe specializate de prelucrare a imaginilor și grafică pe calculator, inclusiv tehnici de recunoaștere a formelor și viziune artificială, situate la avangarda aplicațiilor ingineresti actuale.
5.2 Aptitudini	A18. Aplică tehnici de prelucrare a imaginilor și viziune artificială pentru dezvoltarea de aplicații ingineresti, utilizând biblioteci și framework-uri specializate de procesare a semnalelor vizuale.
5.3 Responsabilitate și autonomie	R18. Își asumă responsabilitatea pentru impactul social și etic al soluțiilor informatice dezvoltate, evaluând critic consecințele tehnologice asupra utilizatorilor, comunității și mediului și propunând măsuri de minimizare a riscurilor identificate.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Fundamente ale prelucrării imaginilor. Imaginea digitală: pixeli, canale, adâncimea de culoare, rezoluție. Spații de culoare: RGB, grayscale, HSV, LAB — conversia și utilizările. Histograma: interpretarea, egalizarea, specificarea histogramei. Operații geometrice: redimensionare (interpolările nearest, bilinear, bicubic), rotație, crop, flip.	2	Prelegere interactivă	
Tema 2: Filtrare spațială. Convoluția discretă 2D: kernelul, padding-ul, stride-ul. Filtre de netezire: box blur, Gaussian blur (kernel separabil). Filtre de accentuare: unsharp masking, filtrul Laplacian. Filtrarea mediană: robustețea la zgomot impulsiv. Filtrul bilateral: netezire cu păstrarea muchiilor. Implementarea eficientă cu FFT pentru kerneluri mari.	2	Prelegere interactivă	
Tema 3: Analiza frecvențelor în imagini. Transformata Fourier 2D: spectrul de magnitudine și de fază. Interpretarea componentelor de frecvență: frecvențele joase (contur global) vs. înalte (detalii, margini). Filtrare în domeniul frecvență: trece-jos, trece-sus, band-pass (filtrul Butterworth). Teorema convoluției: echivalența convoluției spațiale cu înmulțirea în frecvență. Filtrele Gabor pentru texturi.	2	Prelegere interactivă	
Tema 4: Detecția marginilor și a colțurilor. Operatorii de gradient: Sobel, Prewitt, Schar. Detectorul Canny: netezire Gaussian, gradient, NMS, dublu prag, hysteresis. Detectorul Harris: matricea de autocorelare, răspunsul R, NMS. FAST (Features from Accelerated Segment Test): viteza de detecție. Compararea detecția marginilor vs. detectarea caracteristicilor pentru diferite aplicații.	2	Prelegere interactivă	

Tema 5: Morfologie matematică. Elementul structurant: forme (disc, pătrat, cruce), dimensiuni. Eroziunea și dilatarea: efectele vizuale, proprietățile algebrice. Deschiderea și închiderea morfologică: eliminarea zgomotului, umplerea golurilor. Gradientul morfologic. Hit-or-miss transform. Aplicații: eliminarea zgomotului în imagini binare, extragerea scheletului.	2	Prelegere interactivă	
Tema 6: Segmentarea imaginilor. Segmentarea prin prag: globală (Otsu), locală/adaptivă. Segmentarea bazată pe regiuni: creșterea regiunilor, watershed. Segmentarea bazată pe clustering: k-means pe spațiul LAB. Contoarele active (snakes): funcționala de energie, minimizarea. Evaluarea segmentării: IoU (Intersection over Union), Dice coefficient. Aplicații: segmentarea celulelor în microscopie.	2	Prelegere interactivă	
Tema 7: Descriptori și recunoașterea formelor. Descriptori de contur: momentele Hu, descriptori Fourier ai conturului. Descriptori de textură: GLCM (Gray-Level Co-occurrence Matrix) — energia, contrastul, corelația, omogenitatea. Descriptori locali SIFT: octave, scara, orientarea, histograma gradientilor. Descriptori ORB: binar, rapid, utilizare în aplicații real-time.	2	Prelegere interactivă	
Tema 8: Clasificarea imaginilor cu CNN. Arhitectura CNN: strat convoluțional, pooling, batch normalization, dropout, fully connected. Funcțiile de activare: ReLU, Leaky ReLU, GELU. Arhitecturi clasice: LeNet-5, AlexNet, VGG-16, ResNet-50 (conexiunile reziduale). Funcția de pierdere cross-entropy. Optimizatorii: SGD cu momentum, Adam. Augmentarea datelor: flip, rotație, color jitter, cutout.	2	Prelegere interactivă	
Tema 9: Transfer learning și fine-tuning. Conceptul de transfer learning: reutilizarea caracteristicilor extrase. Feature extraction: înghețarea straturilor convoluționale, antrenarea doar a capului clasificator. Fine-tuning: dezghețarea progresivă a straturilor, learning rate diferențiat. Pretrained models în PyTorch: torchvision.models (ResNet, EfficientNet, ViT). Evaluarea modelului: acuratețe, precision, recall, F1, confusion matrix.	2	Prelegere interactivă, PyTorch	
Tema 10: Detecția obiectelor. Formularea problemei: clasificare + localizare (bounding box). Algoritmi clasici: Sliding window + SVM + HOG. Detecția modernă: R-CNN, Fast R-CNN, Faster R-CNN (Region Proposal Network). Detecția one-stage: YOLO v5/v8 — arhitectura, anchor boxes, NMS. Metrici:	2	Prelegere interactivă	

mAP (mean Average Precision), IoU threshold. Aplicații: detecția fețelor, numerelor de înmatriculare, defectelor industriale.			
Tema 11: Segmentarea semantică și de instanță. Segmentarea semantică: FCN (Fully Convolutional Network), encoder-decoder U-Net, DeepLab v3+. Segmentarea de instanță: Mask R-CNN. Segmentarea panoramică. Evaluarea: mean IoU per clasă. Setul de date COCO, Pascal VOC. Aplicații: conducerea autonomă (segmentarea drumului), analiza imaginilor medicale (segmentarea tumorilor).	2	Prelegere interactivă	
Tema 12: Generarea și sinteza imaginilor. Autoencoderul variațional (VAE): spațiul latent, reparametrizarea, pierderea ELBO. Rețelele generative adversariale (GAN): discriminatorul, generatorul, funcția de pierdere minimax. GAN-uri condiționate (cGAN): generarea controlată. Aplicații: super-rezoluție (SRGAN), transferul de stil neural, completarea imaginilor (inpainting). Modelele de difuzie (DDPM): principiul și aplicațiile actuale.	2	Prelegere interactivă	
Tema 13: Prelucrarea video și fluxuri în timp real. Compresia video: I-frame, P-frame, B-frame, codec H.264/H.265. Citirea și scrierea video cu OpenCV: VideoCapture, VideoWriter. Subtracția fondului: MOG2, KNN background subtractor. Urmărirea obiectelor: algoritmi KCF, CSRT, algoritmul Sort. Fluxul optic dens: algoritmul Farneback. Prelucrarea video în timp real: optimizarea pipeline-ului pentru latență scăzută.	2	Prelegere interactivă	
Tema 14: Impactul etic al viziunii artificiale și recapitulare. Biasul în sistemele de recunoaștere facială: studii de caz (Buolamwini & Gebru 2018). Deepfake-urile: tehnicile de generare și metodele de detecție. Supravegherea în masă: implicațiile legale (GDPR, AI Act). Confidențialitatea datelor de antrenament. Principiile AI responsabil: echitate, transparență, responsabilitate, robustețe. Recapitulare generală.	2	Prelegere, discuție etică	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Gonzalez, R.C., Woods, R.E., Digital Image Processing, 4th ed., Pearson, 2018. Szeliski, R., Computer Vision: Algorithms and Applications, 2nd ed., Springer, 2022. Disponibil gratuit la: https://szeliski.org/Book/ Bradski, G., Kaehler, A., Learning OpenCV 3, O'Reilly Media, 2016. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Goodfellow, I., Bengio, Y., Courville, A., Deep Learning, MIT Press, 2016. Disponibil gratuit la: https://www.deeplearningbook.org/ Prince, S.J.D., Understanding Deep Learning, MIT Press, 2023. Disponibil gratuit la: https://udlbook.github.io/udlbook/ <p>Resurse software:</p> <ol style="list-style-type: none"> OpenCV Documentation — prelucrarea imaginilor și viziune artificială: https://docs.opencv.org/ PyTorch Documentation — deep learning pentru imagini: https://pytorch.org/docs/ 		

6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Prelucrarea clasică a imaginilor cu OpenCV. Operații de bază: conversii de culoare, histogramă, egalizare. Filtrare Gaussian și mediană. Detecția marginilor Canny. Operații morfologice: eroziune, dilatare, deschidere, închidere. Aplicarea pe un set de imagini de test cu diferite niveluri de zgomot.	2	Lucrare practică Python/OpenCV	
Lucrarea L2: Segmentare și analiză de formă. Segmentarea prin prag Otsu și thresholding adaptiv. Detectarea conturilor cu findContours. Calculul caracteristicilor conturului: aria, perimetrul, bounding box, momentele Hu. Aplicație: numărarea și clasificarea obiectelor într-o imagine de inspecție industrială simulată.	2	Lucrare practică Python/OpenCV	
Lucrarea L3: Descriptori locali și potrivirea imaginilor. Extragerea descriptorilor ORB și SIFT cu OpenCV. Potrivirea cu BFMatcher și FLANN, filtrarea cu raportul Lowe. Calculul omografiei cu RANSAC. Aplicații: detectarea unui obiect în mai multe perspective, asamblarea unei panorame din 2 imagini.	2	Lucrare practică Python/OpenCV	
Lucrarea L4: Clasificarea imaginilor cu CNN — transfer learning. Fine-tuning al unui model ResNet-18 pretrained (torchvision) pe un dataset de 4-5 clase. Configurarea DataLoader cu Albumentations (augmentare). Antrenarea pe Google Colab (GPU). Evaluarea pe setul de test: confusion matrix, F1 per clasă. Vizualizarea activărilor (Grad-CAM).	2	Lucrare practică PyTorch/Colab	
Lucrarea L5: Detecția obiectelor cu YOLO. Configurarea și rularea YOLOv8 (Ultralytics) pe un set de imagini de test. Antrenarea pe un dataset personalizat de 2 clase (minim 100 imagini adnotate cu Roboflow). Evaluarea modelului: mAP@50 și mAP@50-95. Inferență pe video cu afișarea bounding box-urilor.	2	Lucrare practică Python/YOLOv8	
Lucrarea L6: Segmentarea semantică cu U-Net. Implementarea sau utilizarea unui model U-Net pretrained pentru segmentarea unui dataset medical (ex: DRIVE — retina) sau de satelit. Calculul IoU și Dice coefficient pe setul de validare. Vizualizarea suprapunerii predicției față de masca de referință (ground truth).	2	Lucrare practică PyTorch/Colab	
Lucrarea L7: Proiect integrat și colocviu. Implementarea unui pipeline complet de prelucrare grafică pentru o problemă aleasă: (a) inspecție vizuală automată (detecția defectelor) sau (b) segmentarea și clasificarea unui set de imagini medicale sau industriale.	2	Proiect individual, colocviu	

Documentarea: descrierea pipeline-ului, metricile de evaluare, analiza erorilor, impactul etic al aplicației. Colocviu oral.			
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. OpenCV Documentation: https://docs.opencv.org/ 2. PyTorch Documentation și tutoriale: https://pytorch.org/docs/ 3. Gonzalez, R.C., Woods, R.E., Digital Image Processing, 4th ed., Pearson, 2018. <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. scikit-image Documentation: https://scikit-image.org/docs/ 5. Ultralytics YOLOv8 Documentation: https://docs.ultralytics.com/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. OpenCV Documentation: https://docs.opencv.org/ 7. PyTorch Documentation: https://pytorch.org/docs/ 8. Ultralytics YOLOv8: https://docs.ultralytics.com/ 9. Google Colab — mediu gratuit cu GPU pentru deep learning: https://colab.research.google.com/ 		

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

Prelucrarea Grafică acoperă spectrul complet al viziunii artificiale moderne: de la filtrarea clasică și segmentarea morfologică până la detecția obiectelor cu YOLO și segmentarea semantică cu U-Net. Față de Elemente de Grafică pe Calculator (TIDS321), care pune accentul pe randarea 3D și pipeline-ul grafic, această disciplină se orientează spre analiza și înțelegerea conținutului imaginilor — competențe direct aplicabile în inspecția vizuală automată, medicina digitală și vehiculele autonome. Tema 14 și secțiunea de impact etic din proiectul integrat formează direct R18 — responsabilitatea pentru consecințele sociale ale sistemelor de viziune artificială, cu referire la studii de caz reale (bias în recunoașterea facială, deepfake-uri, supravegherea în masă).

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea și aplicarea filtrelor spațiale și în domeniul frecvenței; cunoașterea tehnicilor de segmentare (Otsu, watershed, morfologie); cunoașterea descriptorilor locali (ORB, SIFT) și a potrivirii lor; cunoașterea arhitecturii CNN (ResNet, U-Net, YOLO) și a principiilor transfer learning; evaluarea modelelor cu IoU, mAP, Dice; identificarea și descrierea a cel puțin două riscuri etice ale viziunii artificiale.	Colocviu oral (20 minute): prezentarea pipeline-ului din proiectul integrat, justificarea alegerii metodelor, analiza erorilor modelului și a impactului etic al aplicației, cu o întrebare suplimentară din materia de curs	40%
8.5 Seminar/laborator	Funcționalitatea pipeline-ului din proiectul integrat (rulează pe setul de test și produce predicții	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L7 (50%)	60%

	<p>evaluate cu metrici corecte); calitatea implementărilor din laborator (codul este corect, rezultatele sunt interpretate, nu doar afișate); calitatea secțiunii de impact etic din documentație (cel puțin două riscuri identificate cu măsuri propuse); calitatea prezentării la colocviu.</p>		
--	---	--	--

8.6 Standard minim de performanță:

Predarea unui proiect integrat cu un pipeline funcțional evaluat pe un set de test real cu cel puțin o metrică cantitativă (IoU, mAP sau F1) calculată corect și raportată, însoțit de o secțiune de impact etic de cel puțin un paragraf cu riscuri concrete și măsuri propuse. Un pipeline care nu produce metrici cuantificabile sau o documentație fără analiza etică nu îndeplinesc standardul minim — această cerință diferențiază disciplina de simpla executare de scripturi OpenCV.

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar / laborator
17.03.2026	Conf. Dr. Bucea Radu Manea Țonis	Conf. Dr. Bucea Radu Manea Țonis
Data avizării în Consiliul Departamentului	Semnătura directorului DIT	
24.03.2026	Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT	
07.04.2026	Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei	TEHNICI DE COMUNICARE – TIDFAC323						
1.2 Titularul activităților de curs	Conf. Dr. Bucea Radu Manea Țonis						
1.3 Titularul activităților de seminar	Conf. Dr. Bucea Radu Manea Țonis						
1.4 Titularul activităților de laborator							
1.5 Anul de studiu	III	1.6 Semestrul	2	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	FAC

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

2.1 Număr de ore pe săptămână	2	din care: 2.2 curs	1	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	28	din care: 2.5 curs	14	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					22
Documentare					5
Studiu individual					8
Referate					7
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					50
2.10 Numărul de credite					2

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Nu există condiții formale de curriculum — disciplina este facultativă și accesibilă tuturor studenților din An III, indiferent de parcursul anterior.
3.2 de competențe	Capacitatea de a exprima ideea tehnică în scris în limba română la nivel academic; experiența prezentării unor rezultate tehnice (referate, proiecte); familiarizarea cu instrumente de prezentare (PowerPoint, Google Slides sau echivalent).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector; platforma Sakai pentru distribuirea materialelor; recomandare: instalarea unui instrument de prezentare (PowerPoint, Google Slides, Canva) și a unui instrument de editare colaborativă (Google Docs sau Word Online) înainte de prima ședință de curs.
4.2 de desfășurare a seminarului/laboratorului	Sală seminar cu aranjament care favorizează discuția (semicerc sau grupuri); acces la videoproiector și la microfon portabil pentru simularea prezentărilor; platforma Sakai pentru predarea lucrărilor scrise; studenții

	sunt încurajați să aducă propriul laptop sau tabletă pentru activitățile interactive.
--	---

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C43. Posedă cunoștințe privind dezvoltarea aplicațiilor mobile native și hibride — ciclul de viață al aplicației, gestionarea resurselor hardware ale dispozitivului, publicarea în magazine de aplicații — și principiile DevOps aplicate în livrarea continuă a software-ului mobil.
5.2 Aptitudini	A43. Elaborează și susține prezentări tehnice și demonstrații practice pentru audiențe cu profiluri mixte — tehnic și non-tehnic — adaptând conținutul, nivelul de detaliu și suportul vizual la specificul publicului țintă.
5.3 Responsabilitate și autonomie	R24. Contribuie în mod responsabil la transferul de cunoștințe tehnice specializate către beneficiari, colegi și parteneri, adaptând nivelul de detaliu tehnic al comunicării la profilul și nevoile interlocutorilor. R43. Își asumă responsabilitatea pentru calitatea și eficiența comunicării tehnice în toate formele sale — scrisă, orală, vizuală — adaptând mesajul la audiență, asigurând acuratețea informației transmise și promovând o cultură a transparenței profesionale în cadrul echipei și față de beneficiari.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Comunicarea în mediul profesional TI. Specificul comunicării tehnice: precizie, concizie, audiență. Tipuri de comunicare în industria software: documentație, e-mail, cod review, prezentări, rapoarte. Barierele de comunicare între tehnicieni și non-tehnicieni. Modelul Shannon-Weaver adaptat la comunicarea profesională.	1	Prelegere interactivă	
Tema 2: Comunicarea verbală și non-verbală. Componente ale comunicării orale: tonul, ritmul, volumul, articularea. Comunicarea non-verbală: postura, gesturile, contactul vizual, spațiul personal. Ascultarea activă: tehnicile, importanța în proiectele agile. Adaptarea stilului verbal la interlocutor: client, manager, coleg tehnic.	1	Prelegere interactivă	
Tema 3: Comunicarea scrisă tehnică — principii. Principiile textului tehnic: claritate, corectitudine, coerență, concizie. Structura unui document tehnic: introducere, conținut, concluzie, referințe. Registrul formal vs. informal în scris. Greșelile frecvente în comunicarea scrisă a inginerilor software: jargonul excesiv, ambiguitatea, lipsa structurii.	1	Prelegere interactivă	
Tema 4: E-mailul și comunicarea asincronă. Structura unui e-mail profesional: subiectul, salutul, corpul, încheierea, semnătura. Regulile de etichetă în e-mail: Reply vs. Reply All, CC vs. BCC, urgența falsă. Comunicarea în Slack și Microsoft Teams: canalele, thread-urile, eticheta mesajelor. Gestionarea volumului de comunicări digitale.	1	Prelegere interactivă	

<p>Tema 5: Documentarea tehnică. Tipuri de documentație software: README, API docs, ghid de utilizare, arhitecturală, runbook. Principiile documentației eficiente: docs-as-code, versionarea. Instrumente: Markdown, Confluence, GitHub Wiki, Notion. Documentarea codului: JavaDoc, docstrings Python. Greșeli frecvente: documentația depășită, lipsa exemplurilor.</p>	1	Prelegere interactivă	
<p>Tema 6: Prezentarea tehnică — structură și conținut. Structura unei prezentări tehnice: agenda, contextul, problema, soluția, demonstrația, concluziile, Q&A. Adaptarea conținutului la audiență: prezentarea pentru stakeholders vs. pentru colegi tehnici. Regula 10-20-30 (Guy Kawasaki): 10 slide-uri, 20 minute, font 30. Narativul tehnic: povestea din spatele datelor.</p>	1	Prelegere interactivă	
<p>Tema 7: Suportul vizual și designul slide-urilor. Principiile designului vizual: ierarhia, contrastul, alinierea, apropierea (CRAP). Greșelile clasice în slide-uri: text dens, fonturi mici, culori discordante, animații distragătoare. Vizualizarea datelor: alegerea tipului de grafic, etichetarea axelor, mesajul principal vizibil. Instrumente: PowerPoint, Google Slides, Canva, Marp (Markdown).</p>	1	Prelegere interactivă	
<p>Tema 8: Comunicarea în echipele agile. Ceremoniile Scrum ca exerciții de comunicare: daily standup (3 întrebări, 15 minute), sprint review (demonstrarea live), retrospectiva (ce a mers, ce nu, ce îmbunătățim). Comunicarea în code review: tonul constructiv, specificitatea feedback-ului. Scrierea user stories și a criteriilor de acceptanță.</p>	1	Prelegere interactivă	
<p>Tema 9: Comunicarea cu clienții și stakeholder-ii. Identificarea stakeholder-ilor: matricea putere-interes. Managementul așteptărilor: under-promise, over-deliver. Comunicarea veștilor proaste: structura SBI (Situation-Behavior-Impact). Demo-urile pentru clienți: ce să arăți și ce să eviți. Gestionarea întrebărilor dificile în prezentări.</p>	1	Prelegere interactivă	
<p>Tema 10: Comunicarea interculturală în TI. Dimensiunile culturale Hofstede aplicate la echipele distribuite: individualism/colectivism, distanța față de putere, evitarea incertitudinii. Echipele remote și offshore: provocările de comunicare, fuserile orare, instrumentele (Zoom, Meet, Teams). Eticheta videoconferințelor: fundalul, microfonul, camera, punctualitatea.</p>	1	Prelegere interactivă	
<p>Tema 11: Negocierea și gestionarea conflictelor. Tipuri de conflicte în echipele tehnice: conflict de sarcini vs. conflict</p>	1	Prelegere interactivă	

interpersonal. Stilurile de negociere: competitiv, colaborativ, compromis, acomodare, evitare. Negocierea cerințelor cu clientul: BATNA, ZOPA. Tehnica DESC pentru feedback negativ: Describe, Express, Specify, Consequences.			
Tema 12: Comunicarea scrisă — rapoarte și specificații. Structura unui raport tehnic: sumarul executiv, metodologia, rezultatele, concluziile, recomandările. Specificațiile de cerințe software (SRS): IEEE 830, cerințele funcționale și non-funcționale, criteriile de acceptanță. Studiul de caz tehnic: structura, scopul, publicul.	1	Prelegere interactivă	
Tema 13: Comunicarea în situații dificile. Comunicarea în crize tehnice: incidente de producție, outage-uri — structura post-mortem-ului (timeline, root cause, acțiuni). Transmiterea veștilor proaste: terminarea unui proiect, raportarea unui defect critic. Comunicarea publică a companiei în situații de criză (exemple din industria tech).	1	Prelegere interactivă	
Tema 14: Portofoliul de comunicare și recapitulare. Construirea brandului personal: profilul LinkedIn pentru un inginer software, contribuțiile open-source ca instrument de comunicare, articolele tehnice (blog, Medium, dev.to). CV-ul tehnic: ce include și ce evită un recruiter tehnic. Interviu tehnic ca formă de comunicare. Recapitulare generală.	1	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Lannon, J.M., Gurak, L.J., Technical Communication, 15th ed., Pearson, 2020. Locker, K., Kienzler, D., Business and Administrative Communication, 12th ed., McGraw-Hill, 2018. Markel, M., Selber, S., Technical Communication, 13th ed., Bedford/St. Martin's, 2021. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Anderson, P.V., Technical Communication: A Reader-Centered Approach, 9th ed., Cengage, 2018. Reynolds, G., Presentation Zen: Simple Ideas on Presentation Design and Delivery, 3rd ed., New Riders, 2019. <p>Resurse software:</p> <ol style="list-style-type: none"> Google Technical Writing Courses — cursuri gratuite de scriere tehnică: https://developers.google.com/tech-writing Hemingway Editor — instrument online de claritate a textului: https://hemingwayapp.com/ Canva — instrument de prezentare vizuală: https://www.canva.com/ 		
6.2 Seminar	Nr. ore	Metode de predare	Observații
Seminarul S1: Prezentarea participanților. Exercițiu de prezentare de 2 minute: cine ești, ce faci tehnic, ce vrei să îmbunătățești în comunicare. Feedback structurat din partea grupului. Identificarea punctelor forte și a ariilor de dezvoltare pentru fiecare participant.	1	Seminar interactiv	
Seminarul S2: Analiza e-mailurilor profesionale. Analiza în grup a 5 e-mailuri reale (anonimizate) din mediul corporate TI: identificarea greșelilor, reformularea subiectelor, restructurarea textului. Exercițiu	1	Seminar interactiv	

individual: scrierea unui e-mail de escaladare a unui incident tehnic.			
Seminarul S3: Scrierea documentației tehnice. Atelier de scriere: fiecare student documentează o funcție/API la alegere conform unui template dat (README sau docstring). Peer review în perechi: claritatea, completitudinea, exemplele. Discutarea versiunilor revizuite.	1	Atelier de scriere	
Seminarul S4: Designul slide-urilor. Analiza comparativă a două versiuni ale aceluiași set de slide-uri (slabă vs. bună). Exercițiu practic: redesenarea a 3 slide-uri supraaglomerate cu text în Canva sau PowerPoint. Prezentarea și discutarea rezultatelor.	1	Atelier practic	
Seminarul S5: Prezentare tehnică de 5 minute — runda 1. Fiecare student susține o prezentare de 5 minute pe un subiect tehnic ales (tehnologie, proiect, concept). Audiența completează o fișă de feedback structurat. Discuție despre aspectele de îmbunătățit.	1	Prezentări individuale	
Seminarul S6: Comunicarea în daily standup și code review. Simularea daily standup-ului (3 întrebări, 2 minute per persoană). Exercițiu de code review scris: scrierea a 5 comentarii constructive pe un fragment de cod dat, evitând tonul critic. Discuția diferenței dintre feedback tehnic și feedback personal.	1	Simulare, exercițiu practic	
Seminarul S7: Comunicarea cu non-tehnicieni. Exercițiu: explicarea unui concept tehnic complex (ex: REST API, baza de date relațională, containerizare Docker) unui public non-tehnic în maximum 3 minute, fără jargon. Feedback din grupul de colegi care simulează publicul non-tehnic.	1	Exercițiu comunicare adaptată	
Seminarul S8: Negocierea cerințelor — joc de rol. Joc de rol în perechi: un student joacă rolul clientului cu cerințe vagi sau contradictorii, celălalt joacă rolul inginerului care clarifică, prioritizează și negociază. Debriefing în grup: ce tehnici au funcționat.	1	Joc de rol, debrief	
Seminarul S9: Scrierea unui raport tehnic scurt. Atelier: fiecare student redactează un sumar executiv de 300 de cuvinte pentru un proiect tehnic propriu. Peer review pe claritate, structură și mesajul principal. Discutarea greșelilor frecvente.	1	Atelier de scriere	
Seminarul S10: Comunicarea în crize tehnice. Studiu de caz: analiza unui post-mortem public real (de ex. GitLab, Cloudflare). Exercițiu: scrierea unui post-mortem fictiv pentru un incident tehnic dat, respectând structura standard (timeline, root cause, lecții învățate).	1	Studiu de caz, atelier	
Seminarul S11: Comunicarea interculturală — studii de caz. Analiza a 3 scenarii de	1	Discuție, studii de caz	

comunicare în echipe distribuite (România–India, România–SUA, echipă complet remote). Identificarea surselor de conflict cultural. Formularea regulilor de echipă pentru o comunicare eficientă în contexte multiculturale.			
Seminarul S12: Prezentare tehnică de 5 minute — runda 2. A doua rundă de prezentări individuale, aplicând feedback-ul primit la S5. Autoevaluarea progresului față de prima prezentare. Feedback comparativ din partea grupului.	1	Prezentări individuale	
Seminarul S13: Portofoliul de comunicare — LinkedIn și CV. Analiza comparativă a profilurilor LinkedIn ale studenților: secțiunile About, Experience, Skills. Exercițiu: rescrierea secțiunii About a profilului propriu în stil profesional tehnic. Discutarea CV-ului tehnic: ordinea informațiilor, proiectele relevante.	1	Atelier practic	
Seminarul S14: Evaluare finală și reflecție. Fiecare student prezintă în 3 minute cea mai importantă lecție din disciplină și un exemplu concret de aplicare în activitatea sa. Autoevaluarea competențelor de comunicare la finalul cursului față de începutul lui. Feedback colectiv față de titular.	1	Prezentări scurte, reflecție	
	<p>Bibliografie obligatorie seminar</p> <ol style="list-style-type: none"> 1. Lannon, J.M., Gurak, L.J., Technical Communication, 15th ed., Pearson, 2020. 2. Reynolds, G., Presentation Zen, 3rd ed., New Riders, 2019. 3. Google Technical Writing Courses — scriere tehnică gratuită: https://developers.google.com/tech-writing <p>Bibliografie complementara seminar</p> <ol style="list-style-type: none"> 4. Anderson, P.V., Technical Communication: A Reader-Centered Approach, 9th ed., Cengage, 2018. 5. Hemingway Editor — claritatea textului: https://hemingwayapp.com/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. Google Technical Writing Courses: https://developers.google.com/tech-writing 7. Hemingway Editor: https://hemingwayapp.com/ 8. Canva: https://www.canva.com/ 9. Marp — prezentări Markdown: https://marp.app/ 		

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

Tehnicile de Comunicare răspund uneia dintre cele mai frecvente observații ale angajatorilor din industria TI: inginerii cu competențe tehnice solide întâmpină dificultăți în comunicarea cu clienții, managementul și echipele multidisciplinare. Disciplina formează direct A43 — capacitatea de a elabora și susține prezentări tehnice adaptate la audiențe mixte — și R43, responsabilitatea pentru calitatea comunicării profesionale.

Formatul alternant curs-seminar permite aplicarea imediată a fiecărui concept predat: prezentările simulate, atelierile de scriere și jocurile de rol din seminar transformă cunoașterea teoretică în competență demonstrabilă.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea principiilor comunicării tehnice scrise și orale; cunoașterea structurii documentelor tehnice (e-mail profesional, raport, SRS, post-mortem, README); cunoașterea principiilor designului vizual pentru prezentări; cunoașterea tehnicilor de comunicare în echipe agile (standup, code review, retrospectivă); cunoașterea dimensiunilor comunicării interculturale în echipele distribuite.	Colocviu oral: susținerea unei prezentări tehnice de 7 minute pe un subiect ales de student, urmată de 5 minute de Q&A cu titularul și cu un coleg desemnat ca audiență non-tehnică — evaluând adaptarea conținutului la audiență, calitatea slide-urilor și claritatea mesajului	50%
8.5 Seminar/laborator	Calitatea celor două prezentări de seminar (S5 și S12): progresul demonstrat față de prima rundă; calitatea lucrărilor scrise predate la seminar (e-mailul de la S2, documentul tehnic de la S3, sumarul executiv de la S9, post-mortem-ul de la S10); participarea activă și calitatea feedback-ului oferit colegilor în activitățile de peer review.	Evaluare continuă: lucrări scrise predate pe Sakai după seminarele S2, S3, S9 și S10 (40%) și cele două prezentări orale din S5 și S12 (60%, ponderea mai mare pentru S12)	50%
8.6 Standard minim de performanță:			
Susținerea prezentării tehnice de 7 minute din colocviu fără a citi de pe slide-uri și cu demonstrarea adaptării conținutului la audiența mixtă (un slide sau un pasaj explicit formulat pentru non-tehnicieni). O prezentare în care tot conținutul este tehnic fără nicio adaptare pentru audiența non-tehnică sau care este citită integral de pe slide-uri nu îndeplinește standardul minim, indiferent de calitatea conținutului informațional.			

Data completării 16.03.2026	Semnătura titularului de curs Conf. Dr. Bucea Radu Manea Țonis	Semnătura titularului de seminar / laborator Conf. Dr. Bucea Radu Manea Țonis
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. Dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. Dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2028-2029

1. Date despre disciplină

1.1 Denumirea disciplinei	PRACTICĂ PEDAGOGICĂ ÎN ÎNV. PREUNIVERSITAR OBLIGATORIU (1) – TIDFAC326						
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Conf. Dr. Beteringhe Adrian						
1.5 Anul de studiu	III	1.6 Semestrul	1	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	FAC

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs		2.3 seminar/laborator	0/3
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs		2.6 seminar/laborator	0/42
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Psihologia Educației (disciplina din modulul psihopedagogic); Pedagogie I — Teoria și Metodologia Curriculum-ului; Didactica Specialității — Informatică; Instruire Asistată de Calculator (TIDFAC325).
3.2 de competențe	Cunoașterea principiilor de bază ale psihologiei adolescentului și ale teoriilor învățării; familiarizarea cu noțiunile de proiectare didactică (obiective, competențe, lecție, evaluare); capacitatea de a preda un conținut tehnic simplu de informatică; acreditarea pentru Modulul Psihopedagogic Nivel I în curs de parcurgere.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	
4.2 de desfășurare a seminarului/laboratorului	Practica pedagogică se desfășoară la o școală gimnazială sau liceu partener, sub îndrumarea unui profesor-mentor și a coordonatorului de practică din universitate. Studentul asistă la lecții, susține lecții asistate și

lecții independente, redactează proiecte didactice și rapoarte de observație. Documentele obligatorii se predau pe platforma Sakai.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C35. Posedă cunoștințe din domeniul psihologiei educației și al pedagogiei — teorii ale învățării, metode didactice, proiectare curriculară — necesare activității de instruire și transfer de cunoștințe tehnice specializate. C36. Cunoaște principiile didacticii specializării și ale managementului clasei, inclusiv strategii de evaluare formativă și summativă, adaptate predării disciplinelor tehnice din învățământul preuniversitar și universitar.
5.2 Aptitudini	A39. Aplică tehnici pedagogice și didactice adecvate în activitățile de predare, îndrumare și mentoring tehnic, proiectând activități de învățare diferențiată și evaluând progresul cursanților prin metode formative și sumative. A40. Gestionează clasa de elevi în contexte de educație tehnică, aplicând strategii de motivare, rezolvare a conflictelor și diferențiere a instruirii, asigurând un climat favorabil învățării disciplinelor informatice.
5.3 Responsabilitate și autonomie	R39. Acționează cu responsabilitate pedagogică și etică în toate activitățile didactice, respectând principiile echității, incluziunii și respectului față de diversitatea cursanților, și sesizând prompt situațiile care necesită intervenție specializată. R40. Gestionează autonom clasa de elevi în activitățile practice de informatică, asumând responsabilitatea pentru securitatea digitală a mediului de lucru al elevilor, respectarea regulamentelor școlare și prevenirea comportamentelor inadecvate în mediul online.

6. Conținuturi

6.1 Laborator	Nr. ore	Metode de predare	Observații
Activitatea 1: Observarea lecțiilor. Asistarea la 3 lecții de informatică predate de profesorul-mentor. Completarea fișei de observație: obiectivele lecției, metodele utilizate, managementul timpului, interacțiunea cu elevii, evaluarea. Discuția post-lecție cu mentorul: identificarea punctelor forte și a aspectelor de îmbunătățit.	6	Observare, fișă de observație	
Activitatea 2: Analiza documentelor curriculare. Studiul programei școlare de informatică pentru clasa și nivelul alocat. Analiza manualului și a auxiliarelor. Identificarea competențelor generale și specifice. Redactarea unui proiect didactic complet pentru o lecție de tip predare de cunostinte noi, cu toate rubricile (obiective, metode, mijloace, evaluare, desfășurarea lecției).	6	Activitate individuală, proiect didactic	
Activitatea 3: Lecție asistată nr. 1. Susținerea primei lecții asistate în fața clasei, cu proiectul didactic avizat de mentor. Utilizarea a cel puțin două metode didactice diferite (expunerea și demonstrația, sau problematizarea și exercițiul). Autoevaluarea după lecție pe baza fișei de autoevaluare. Feedback structurat din partea mentorului.	6	Lecție asistată, autoevaluare	

Activitatea 4: Proiectarea și susținerea lecției de recapitulare. Redactarea unui proiect didactic pentru o lecție de recapitulare și sistematizare. Utilizarea metodelor active: lucrul în echipe, brainstorming, harta conceptuală sau quiz digital (Kahoot/Quizlet). Susținerea lecției. Analiza participării elevilor și a climatului clasei.	6	Lecție asistată, metode active	
Activitatea 5: Lecție de evaluare — proiectare și aplicare. Proiectarea unui instrument de evaluare formativă: test cu itemi obiectivi și semiobiectivi, adaptat competențelor urmărite. Aplicarea testului. Corectarea și calculul rezultatelor. Completarea catalogului. Redactarea unui raport de analiză a rezultatelor cu identificarea lacunelor și a măsurilor de remediere.	6	Lecție de evaluare, instrument de evaluare	
Activitatea 6: Lecție independentă. Susținerea unei lecții complet independente (fără intervenția mentorului decât în situații de urgență), pe un conținut la alegere din programa clasei. Utilizarea platformei Sakai sau a unui instrument digital ales. Gestionarea autonomă a situațiilor neprevăzute. Evaluarea lecției de către mentor cu fișa de evaluare a practicii pedagogice.	6	Lecție independentă	
Activitatea 7: Portofoliu de practică și evaluare finală. Finalizarea portofoliului de practică: fișele de observație (Activitatea 1), proiectele didactice ale tuturor lecțiilor susținute, instrumentul de evaluare (Activitatea 5), raportul de analiză, jurnalul reflexiv (jurnal săptămânal de reflecție profesională). Prezentarea portofoliului în fața comisiei. Evaluarea finală de către mentor și coordonatorul universitar.	6	Portofoliu, evaluare finală	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> Cucoș, C., Pedagogie, ed. a III-a revăzută și adăugită, Polirom, 2014. Ionescu, M., Radu, I., Didactica modernă, ed. a II-a, Dacia, 2001. Programele școlare de informatică și TIC — Ministerul Educației, disponibile la: https://www.edu.ro/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> Molan, V., Didactica disciplinelor din domeniul limbă și comunicare în învățământul primar, MINIPED, 2012. Neacșu, I., Instruire și învățare, ed. a II-a, Ed. Științifică, 1999. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> Programe școlare — Ministerul Educației: https://www.edu.ro/ Kahoot — platformă de quiz educațional: https://kahoot.com/ Quizlet — instrumente de învățare interactivă: https://quizlet.com/ 		

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

Practica Pedagogică în Învățământul Preuniversitar (1) formează competențele de predare a informaticii în context școlar real, complement esențial pentru studenții care urmează Modulul Psihopedagogic Nivel I. Disciplina asigură trecerea de la cunoașterea pedagogică teoretică la aplicarea ei în fața unei clase reale, cu toată complexitatea gestionării colectivului de elevi.

Portofoliul de practică — cu proiecte didactice, instrumente de evaluare și jurnalul reflexiv — formează direct A39, A40, R39 și R40, cultivând responsabilitatea pedagogică față de calitatea instruirii, securitatea digitală a elevilor și echitatea în evaluare.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	-	-	-
8.5 Seminar/laborator	Calitatea proiectelor didactice (obiective corecte, metodele alese adecvate conținutului, instrumentul de evaluare aliniat la obiective); calitatea celor 3 lecții susținute (lecția asistată 1, lecția de recapitulare, lecția independentă) evaluate de mentor cu fișa de evaluare; completitudinea portofoliului de practică (toate documentele obligatorii prezente și datate); calitatea jurnalului reflexiv.	Evaluare de către mentorul din școală și coordonatorul universitar: fișele de evaluare a lecțiilor (50%) + portofoliul complet predat pe Sakai (50%)	100%
8.6 Standard minim de performanță:			
Susținerea a cel puțin 3 lecții complete (din care una independentă) cu proiecte didactice avizate de mentor și predarea portofoliului de practică complet. Un student care nu a susținut lecția independentă sau care predă un portofoliu fără proiectele didactice și jurnalul reflexiv nu îndeplinește standardul minim, indiferent de calitatea fișelor de observație.			

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar / laborator Conf. Dr. Beteringhe Adrian
16.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. Dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. Dr. Beteringhe Adrian	
07.04.2026		