



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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	ELECTRONICĂ DIGITALĂ – TIDS208						
1.2 Titularul activităților de curs	Prof. dr. ing. Viorel Ariton						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Prof. dr. ing. Viorel Ariton						
1.5 Anul de studiu	II	1.6 Semestrul	II	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					32
Documentare					10
Studiu individual					10
Referate					
Teme casă					12
Proiect					
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					76
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Fundamentele electrtonicii
3.2 de competențe	Utilizarea sistemelor de calcul

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Platforma e-learning Danubius online (Sakai), Internet, sală de curs, video-proiector, laptop.
4.2 de desfășurare a seminarului/laboratorului	Platforma e-learning Danubius online, Internet, laborator, video-proiector. Temele de laborator au termene de predare stabilite de titular și rezolvările sunt notate de la 1 la 10, contând la nota finală în medie ponderată cu nota la examen; ne-predarea unei teme este notată cu 0. Rezolvarea temei se încarcă pe platforma e-learning Danubius online de către fiecare student în secțiunea Assignments proprie, în format text, eventual cu fișiere atașate.

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

5.1 Cunoștințe	C5. Cunoaște în profunzime principiile proiectării logice și ale electronicii digitale, inclusiv arhitecturile pentru prelucrare numerică de semnal, situate la avangarda domeniului. C28. Cunoaște în profunzime principiile dispozitivelor electronice (diode, tranzistoare, circuite integrate analogice) și ale electronicii analogice, ca fundament al înțelegerii critice a componentelor hardware din sistemele informatice.
----------------	--

5.2 Aptitudini	A9. Proiectează și configurează sisteme de calcul bazate pe microprocesoare și procesoare digitale de semnal (DSP), integrând componente hardware și software pentru realizarea de sisteme embedded funcționale.
5.3 Responsabilitate și autonomie	R5. Își asumă responsabilitatea pentru coordonarea activităților tehnice ale unei echipe de dezvoltare software, distribuind sarcinile în mod echitabil, monitorizând progresul și intervenind proactiv pentru rezolvarea blocajelor apărute.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
1. Reprezentarea numerică a datelor. Considerații privind aritmetica binară.	2	Expunerea: descrierea explicația, prelegerea. Postarea cursurilor și a prezentărilor online pe platforma colaborativă Conversația: dezbateri, discuții de fixare și consolidare concepte. Problematizarea: teme și întrebări. Descoperirea: experiment, documentare, dirijată sau independentă	Predarea noțiunilor în cadrul orelor de curs se face în conformitate cu strategiile actuale (prezentări/slidesuri pe platforma de lucru colaborativ), învățământ centrat pe student și pe rezultatul învățării.
2. Coduri numerice.	2		
3. Funcții logice.	2		
4. Reprezentarea fizică a semnalelor digitale.	2		
5. Tipuri și familii de circuite digitale.	2		
6. Porți logice.	2		
7. Decodare.	2		
8. Multiplexoare și demultiplexoare.	4		
9. Bistabile.	2		
10. Numărătoare.	2		
11. Registre de deplasare.	4		
12. Memorii.	2		
Bibliografie obligatorie			
1. Dan Nicula, <i>Electronică Digitală</i> , Carte de învățătură Ed Universității Transilvania din Brașov. 2025			
2. Oniga Ștefan, <i>Circuite digitale</i> - Note de curs Editura U.T.PRESS Cluj-Napoca, 2020			
3. Sorin Hintea, Gabor Csipkes, Doris Csipkes, Paul Farago, Mihaela Cirlugea, <i>Digital Integrated Circuits</i> . Editura Casa Cărții de Știință, Cluj-Napoca, 2017			
4. Gheorghe Toacse, Dan Nicula, <i>Electronică Digitală</i> , Editura Tehnică, 2005			
5. Dan Nicula, <i>Electronica Digitală</i> , Editura Universității TRANSILVANIA din Brașov, 2015			
6. Roger Tokheim, Patrick Hoppe <i>Digital Electronics: Principles And Applications</i> , McGraw Hill, 2022			
7. M. Morris Mano and Michael D. Ciletti, <i>Digital Design</i> , 5th Edition, Pearson, 2014			
8. M. Morris R. Mano and Michael D. Ciletti, <i>Digital Design</i> , 5th ed.: Pearson Education, 2012. [Online]. https://books.google.com.pk/books?id=LhZUngEACAAJ			
9. A. Anand Kumar, <i>Fundamentals of Digital Circuits</i> , 4th Edition, PHI Learning Private Limited, 2016.			
10. Soumitra Kumar Mandal, <i>Digital Electronics</i> , McGraw Hill Education Private Limited, 2016			
11. Neal S. Widmer, Neal Widmer, Gregory L. Moss, Ronald J. Tocci, <i>Digital Systems: Principles and Applications</i> , Pearson, 2017.			
12. B. Wilkinson, <i>Electronică digitală</i> , Ed. Teora, București 2002.			
6.2 Laborator	Nr. ore	Metode de predare	Observații
1. Functionarea circuitelor logice.	2	Exercițiul, discuțiile și dezbateri, modelarea, proiectul, lucrul în grup organizat.	Temele practice și utilizarea noțiunilor la seminar corespund strategiilor actuale ale procesului Bologna: învățământ centrat pe student și pe rezultatul învățării.
2. Functionarea basculante bistabile.	2		
3. Functionarea codificatoarelor și decodificatoarelor.	2		
4. Multiplexoare și demultiplexoare.	2		
5. Numărătoare asincrone și sincrone.	2		
6. Functionarea registrelor de deplasare.	2		
7. Memorii RAM.	2		
Bibliografie obligatorie laborator			
1. Dan Nicula, <i>Electronică Digitală</i> , Carte de învățătură Ed Universității Transilvania din Brașov. 2025			
2. Oniga Ștefan, <i>Circuite digitale</i> - Note de curs Editura U.T.PRESS Cluj-Napoca, 2020			
3. Sorin Hintea, Gabor Csipkes, Doris Csipkes, Paul Farago, Mihaela Cirlugea, <i>Digital Integrated Circuits</i> . Editura Casa Cărții de Știință, Cluj-Napoca, 2017			

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

Tematica acestui curs este în concordanță cu ceea ce este prevăzut în programul de studii la nivel licență al celor mai importante universități din țară și străinătate.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea informației predate la curs, gradul de însușire a limbajului de specialitate, parcurgerea unor referințe bibliografice recomandate-	Participare la curs, discuții. Evaluare finală (examen)	50%
8.5 Seminar/laborator	Frecvența și relevanța intervențiilor sau răspunsurilor. Teme pe parcurs	Înregistrare frecvență Corectitudinea și acuratețea rezolvărilor temelor.	50%
8.6 Standard minim de performanță:			
Teme realizate în proporție de 50%			
Raspunsuri corecte la testul final 50%			

Data completării 18.03.2026	Semnătura titularului de curs Prof. dr. ing. Viorel Ariton	Semnătura titularului de seminar / laborator Prof. dr. ing. Viorel Ariton
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	LIMBAJE FORMALE ȘI AUTOMATE – TIDS210						
1.2 Titularul activităților de curs	Prof. dr. ing. Viorel Ariton						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Prof. dr. ing. Viorel Ariton						
1.5 Anul de studiu	II	1.6 Semestrul	II	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					32
Documentare					10
Studiu individual					10
Referate					
Teme casă					12
Proiect					
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					90
2.10 Numărul de credite					4

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Fundamentele algebrice ale informaticii, Fundamentele programării, Structuri de date
3.2 de competențe	Utilizarea sistemelor de calcul

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Platforma e-learning Danubius online (Sakai), Internet, sală de curs, video-proiector, laptop.
4.2 de desfășurare a seminarului/laboratorului	Platforma e-learning Danubius online, Internet, laborator, video-proiector. Temele de laborator au termene de predare stabilite de titular și rezolvările sunt notate de la 1 la 10, contând la nota finală în medie ponderată cu nota la examen; ne-predarea unei teme este notată cu 0. Rezolvarea temei se încarcă pe platforma e-learning Danubius online de către fiecare student în secțiunea Assignments proprie, în format text, eventual cu fișiere atașate.

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
1. Concepte și formalizare pentru mulțimi și relații. Elemente de logică și formalizare matematice	2	Expunerea: descrierea explicația, prelegerea. Postarea cursurilor si a prezentărilor online pe platforma colaborativă Conversația: in cadrul cursurilor utilizând dialogul, dezbateră, discuții de fixare si consolidare concepte. Problematizarea: teme și întrebări cu răspuns unic sau multiplu in teste de autoevaluare Descoperirea: prin experiment sau prin documentare, dirijată sau independentă a unor noțiuni sau fundamente în cadrul orelor de curs	Predarea noțiunilor in cadrul orelor de curs se face in conformitate cu strategiile actuale utilizând elemente de noutate (prezentări/diapoziitive integrate in platforma de învățământ si lucru colaborativ) in vederea îndeplinirii reperului principal al procesului Bologna unde învățământul este centrat pe student si pe rezultatul învățării.
2. Limbaj și expresii în logica propozițională. Inferențe și demonstrarea automată a teoremelor.	2		
3. Automate Finite Deterministe DFA – concepte, formalizare.	2		
4. Automate Finite Ne-deterministe NFA- concepte, formalizare.	2		
5. Automate cu stivă PD - concepte formalizare. Mașini cu stări finite și ieșiri – Traductori.	2		
6. Limbaje formale. Expresii regulate – regEx. Limbaje regulate. Conversia regEx – NFA și DFA – regEx.	2		
7. Gramatici formale. – concepte, formalizare prin reguli de producție. Gramatici și Limbaje Regulate.	2		
8. Gramatici și Limbaje. Ierarhia Chomsky, probleme de decidabilitate și ambiguitate.	4		
9. Calculabilitate, Decidabilitate. Mașina Turing TM.	2		
10. Complexitate. Parametri ai complexității algoritmilor.	2		
11. Evaluare absolută și relativă a complexității algoritmilor. Notății asimptotice.	4		
12. Clase de complexitate definite prin limbaje și TM.	2		

Bibliografie obligatorie

1. Adrian Crăciun, *Logică computațională*, 2020.
2. Charles E. Hughes, *Computability & Complexity Theory*, University of Central Florida, 2015
3. Michael Sipser, *Introduction to the Theory of Computation Course Technology*, Boston, MA, Third edition, 2013
4. Peter Linz, *An Introduction to Formal languages and Automata*, 6th Ed., Jones and Bartlett Publ., 2017.
5. Viorel Arion, *Automate, Calculabilitate și Complexitate*, Ed. Universitară Danubius, ISBN: 978-606-533-590-5, disponibil pe platforma Danubius Online, 2024.

Bibliografie complementară

6. Adrian Atanasiu, *Limbaje formale si automate*, Ed. Infodata, Cluj 2007.
7. Ananth Kalyanaraman, *Automata and Formal Languages*, Washington State University, 2017.
8. Elaine Rich, *Automata, Computability and Complexity: Theory and Applications*, 2019.
9. John Watrous *Introduction to the Theory of Computing - Lecture notes*, University of Waterloo, 2020.
10. Lucică Iancu, *Logica*, Editura tehnică, colecția Cogito, București, 2008.
11. Michael Levet, *Theory of Computation - Lecture Notes*, College of Charleston, 2020.

6.2 Laborator	Nr. ore	Metode de predare	Observații
1. Probleme de logică în și formalisme pentru calculabilitate.	4	Exercițiul, discuțiile și dezbateră, modelarea, proiectul, lucrul în grup organizat.	Temele practice și utilizarea noțiunilor la seminar corespund strategiilor actuale ale procesului Bologna: învățământ centrat
2. Probleme de Logica Propozițiilor și elemente de complexitate a calculului prin tabele de adevăr.	4		
3. Automate și limbaje regulate recunoscute.	6		
4. Automate stivă și limbaje independente de context.	2		
5. Expresii regulate și cuvinte din limbajul specific.	6		

6. Cazuri de calculabilitate. Complexitate asimptotică.	6	pe student si pe rezultatul învățării.
Bibliografie obligatorie laborator		
1. Adrian Crăciun, Logică computațională, 2020.		
2. Adrian Atanasiu, <i>Limbaje formale si automate</i> , Ed. Infodata, Cluj 2007.		
3. Viorel Ariton, Automate, Calculabilitate și Complexitate, Ed. Universitară Danubius, ISBN: 978-606-533-590-5, disponibil pe platforma Danubius Online, 2024.		

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

Tematica acestui curs este în concordanță cu ceea ce este prevăzut în programul de studii la nivel licență al celor mai importante universități din țară și străinătate.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea informației predate la curs, gradul de însușire a limbajului de specialitate, parcurgerea unor referințe bibliografice recomandate-	Participare la curs, discuții. Evaluare finală (examen)	50%
8.5 Seminar/laborator	Frecvența și relevanța intervențiilor sau răspunsurilor. Teme pe parcurs	Înregistrare frecvență Corectitudinea și acuratețea rezolvărilor temelor.	50%
8.6 Standard minim de performanță:			
Teme realizate in proportie de 50%			
Raspunsuri corecte la testul final 50%			

Data completării 18.03.2026	Semnătura titularului de curs Prof. dr. ing. Viorel Ariton	Semnătura titularului de seminar / laborator Prof. dr. ing. Viorel Ariton
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	METODE NUMERICE – TIDS211						
1.2 Titularul activităților de curs	Lect. Dr. Ion Adelina						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Lect. Dr. Ion Adelina						
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Ob DF

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					12
Referate					8
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	Analiză Matematică I și II (TIDF101, TIDF109) — derivate, integrale, serii Taylor, limite; Algebră Liniară (TIDF102) — matrice, sisteme liniare, valori proprii; Teoria Probabilităților și Statistică Matematică (TIDF201) — distribuții, estimare; Matematici Speciale (TIDF110) — ecuații diferențiale.
3.2 de competențe	Capacitatea de a calcula derivate și integrale; cunoașterea operațiilor cu matrice și a metodelor de rezolvare a sistemelor liniare (Gauss, Cramer); înțelegerea conceptelor de limită, continuitate și convergență a șirurilor; abilitatea de a implementa algoritmi matematici elementari în Python sau C/C++.

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 e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, pseudocodurilor și temelor; recomandare: instalarea Python 3.10+ cu NumPy, SciPy și Matplotlib sau MATLAB Student înainte de curs.
-------------------------------	--

4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; Python 3.10+ cu NumPy, SciPy, Matplotlib și SymPy instalate, sau MATLAB; studenții implementează algoritmi din curs și îi testează pe probleme ingineresti concrete; accesul la Wolfram Alpha sau la documentația SciPy este permis și încurajat pentru verificarea rezultatelor analitice.
--	---

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.
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: Erori în calculul numeric. Surse de erori: erori de rotunjire, erori de trunchiere, erori de modelare. Reprezentarea numerelor în virgulă mobilă (IEEE 754): precizie simplă și dublă, epsilon-mașină. Propagarea erorilor: eroare absolută, relativă și procentuală. Condiționarea unei probleme: numărul de condiționare al unei matrice. Stabilitatea algoritmilor numerici: algoritmi stabili și instabili. Importanța ordinii operațiilor.	2	Prelegere interactivă, exemple numerice	
Tema 2: Rezolvarea ecuațiilor neliniare — metode de izolare și de bisecție. Existența și unicitatea rădăcinilor: teorema valorii intermediare. Izolarea rădăcinilor: metode grafice și analitice. Metoda bisecției: algoritm, convergență garantată, rata de convergență liniară, criteriu de oprire. Metoda regula falsi (false position): variantă îmbunătățită a bisecției. Implementarea în Python cu analiza numărului de iterații necesare.	2	Prelegere, algoritm, analiză convergență	
Tema 3: Rezolvarea ecuațiilor neliniare — metode cu convergență rapidă. Metoda Newton-Raphson: derivarea din seria Taylor, convergență pătratică, condiții de convergență globală. Metoda secantei: aproximarea derivatei, convergență superlineară (ordinul 1.618). Metoda Brent (combinație bisecție + secantă + interpolare pătratică): robustețe și eficiență. Compararea metodelor: număr de	2	Prelegere, comparație metode	

evaluări ale funcției, robustețe, domenii de aplicabilitate.			
Tema 4: Rezolvarea sistemelor liniare — metode directe. Eliminarea gaussiană: algoritmul, complexitate $O(n^3)$. Pivotare parțială și totală: necesitate și implementare. Factorizarea LU: algoritmul Doolittle și Crout, rezolvarea sistemelor cu aceeași matrice și mai mulți termeni liberi. Factorizarea Cholesky pentru matrice simetrice pozitiv definite: complexitate $O(n^3/3)$. Condiționarea sistemelor liniare și efectul erorilor de rotunjire.	2	Prelegere, algoritmi, complexitate	
Tema 5: Rezolvarea sistemelor liniare — metode iterative. Metoda Jacobi: derivare, matrice de iterație, condiție suficientă de convergență (dominanța diagonală). Metoda Gauss-Seidel: îmbunătățire față de Jacobi, convergență mai rapidă. Metoda SOR (Successive Over-Relaxation): parametrul de relaxare optim. Convergența metodelor iterative: raza spectrală a matricei de iterație. Comparatie metode directe vs. iterative: matrice dense vs. rare (sparse).	2	Prelegere, analiză convergență	
Tema 6: Calculul valorilor proprii și al vectorilor proprii. Problema valorilor proprii: definiție, polinomul caracteristic, spațiul propriu. Metoda puterii (power method): algoritmul, convergență, limitări. Metoda puterii inverse (inverse iteration): calculul celei mai mici valori proprii. Deflația. Metoda QR: algoritmul Householder pentru reducerea la forma Hessenberg, iterațiile QR cu deplasare (shift). Aplicații: analiza modală, PageRank, PCA.	2	Prelegere interactivă, aplicații	
Tema 7: Interpolarea polinomială. Problema interpolării: existența și unicitatea polinomului de interpolare Lagrange. Formula de interpolare Lagrange: formulă și complexitate. Forma Newton a polinomului de interpolare: diferențe divizate, algoritmul progresiv. Fenomenul Runge: oscilațiile la marginea intervalului. Nodurile Cebîșev: distribuția optimă a nodurilor pentru minimizarea erorii. Spline cubic: motivație și proprietăți.	2	Prelegere, demonstrații numerice	
Tema 8: Aproximarea funcțiilor — metoda celor mai mici pătrate. Problema aproximării prin metoda celor mai mici pătrate (least squares): formula normală. Regresia liniară și polinomială: deducere prin minimizarea sumei pătratelor reziduurilor. Regresia neliniară: liniarizare și iterații Gauss-Newton. Ajustarea datelor experimentale cu incertitudini: ponderea măsurătorilor. Coeficientul de	2	Prelegere, calcule numerice, Python	

determinare R^2 . Aplicații în Machine Learning: regresia ca optimizare.			
Tema 9: Derivarea și integrarea numerică. Formule de derivare numerică: diferențe finite progresive, retrograde și centrate (ordinul 1 și 2). Eroarea de trunchiere și eroarea de rotunjire în derivarea numerică. Integrare numerică (cuadratură): metoda dreptunghiurilor, trapezelor, Simpson 1/3 și 3/8. Regula lui Richardson pentru extrapolarea la limită (integrare Romberg). Cuadratura gaussiană: noduri și ponderi optime. Integrare numerică în mai multe dimensiuni.	2	Prelegere, calcule numerice	
Tema 10: Rezolvarea numerică a ecuațiilor diferențiale ordinare (EDO). Problema Cauchy: existența și unicitatea soluției. Metoda Euler explicită: derivarea, eroarea locală de trunchiere $O(h^2)$, eroarea globală $O(h)$, stabilitate. Metode Runge-Kutta: RK2, RK4 — derivare prin expansie Taylor, erori locale. Metode cu pas adaptiv: estimarea erorii prin compararea soluțiilor de ordine diferite (RK45 — algoritmul Dormand-Prince). Metode implicite (Euler implicit, Crank-Nicolson): stabilitate pentru ecuații stiff.	2	Prelegere, comparație metode	
Tema 11: Sisteme de EDO și ecuații diferențiale cu valori la limită. Reducerea ecuațiilor de ordin superior la sisteme de ecuații de ordinul 1. Rezolvarea numerică a sistemelor de EDO cu Runge-Kutta vectorizat. Probleme cu valori la limită (BVP): metoda shooting (tragerii), metoda diferențelor finite. Introducere în metoda elementelor finite (FEM): principiul variațional, elemente finite 1D. Aplicații: dinamica mecanică, circuite electrice, transferul de căldură.	2	Prelegere interactivă	
Tema 12: Ecuații cu derivate parțiale (EDP) — metode numerice. Clasificarea EDP: parabolice (difuzie), hiperbolice (undă), eliptice (Laplace, Poisson). Discretizarea cu diferențe finite: grile uniforme, operatorul laplacian discret. Schema explicită pentru ecuația căldurii: condiția de stabilitate CFL. Schema implicită (Crank-Nicolson): stabilitate necondiționată. Ecuația Poisson 2D: rezolvarea cu metode iterative (Gauss-Seidel, SOR). Aplicații: simularea difuziei căldurii, câmpuri electrice.	2	Prelegere interactivă	
Tema 13: Metoda Monte Carlo și simularea stochastică. Principiul metodei Monte Carlo: estimarea integralelor prin eșantionare aleatoare. Generarea numerelor pseudo-aleatoare: generatorul liniar congruențial, Mersenne Twister. Reducerea varianței: eșantionare prin importanță (importance sampling), variabile antitetice. Aplicații	2	Prelegere interactivă, Python	

Monte Carlo: calculul lui π , integrare multidimensională, simularea prețurilor de opțiuni (modelul Black-Scholes), simularea transportului neutronilor. Simularea evenimentelor discrete (DES).			
Tema 14: Optimizare numerică și recapitulare. Optimizare fără restricții: metoda gradientului descendent, gradientul conjugat, metoda Newton pentru optimizare. Optimizare cu restricții: metoda multiplicatorilor Lagrange, KKT. Algoritmi evolutivi: introducere în algoritmi genetici și simularea recoacerii (simulated annealing). Legătura metodelor numerice cu Machine Learning: gradientul descendent stochastic (SGD), backpropagation. Recapitulare generală.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Burden, R.L., Faires, J.D., Burden, A.M., Numerical Analysis, 10th ed., Cengage Learning, 2015 (retipărit 2021). Quarteroni, A., Saleri, F., Gervasio, P., Scientific Computing with MATLAB and Octave, 4th ed., Springer, 2014 (retipărit 2022). Stolieru, I., Metode Numerice — note de curs, Universitatea Al. I. Cuza, Iași, 2020. Disponibil la: https://www.math.uaic.ro/~stolieru/ <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Trefethen, L.N., Bau, D., Numerical Linear Algebra, 25th Anniversary Edition, SIAM, 2022. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., Numerical Recipes: The Art of Scientific Computing, 3rd ed., Cambridge University Press, 2007 (retipărit 2021). Disponibil la: https://numerical.recipes/ Kincaid, D., Cheney, W., Numerical Analysis: Mathematics of Scientific Computing, 3rd ed., AMS / Brooks Cole, 2002 (retipărit 2009). <p>Resurse software:</p> <ol style="list-style-type: none"> SciPy Documentation — biblioteca Python pentru metode numerice (integrate, optimize, linalg, interpolate): https://docs.scipy.org/doc/scipy/ Numerical Recipes Online — algoritmi și cod sursă în C++/Fortran: https://numerical.recipes/ MIT OpenCourseWare 18.330 — Introduction to Numerical Analysis, note de curs și probleme: https://ocw.mit.edu/courses/18-330-introduction-to-numerical-analysis-spring-2012/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Aritmetică în virgulă mobilă și erori numerice. Explorarea reprezentării IEEE 754 în Python: epsilon-mașină, overflow, underflow, NaN. Demonstrarea necomutativității adunării în virgulă mobilă. Calculul numeric al derivatei prin diferențe finite: analiza erorii de trunchiere și de rotunjire în funcție de pasul h. Identificarea pasului optim. Calculul numărului de condiționare al unei matrice cu <code>numpy.linalg.cond</code> .	2	Lucrare practică Python/NumPy	
Lucrarea L2: Rezolvarea ecuațiilor neliniare. Implementarea în Python a metodelor biseecție, Newton-Raphson și secantă pentru rezolvarea ecuațiilor neliniare. Compararea numărului de iterații și a erorii de convergență pentru fiecare metodă pe aceeași ecuație. Trasarea grafică (matplotlib) a convergenței. Testarea robusteții: comportamentul metodei Newton în	2	Lucrare practică Python	

prezența rădăcinilor multiple sau a derivatei nule.			
Lucrarea L3: Sisteme liniare — metode directe și iterative. Implementarea factorizării LU cu pivotare parțială (de la zero sau cu <code>scipy.linalg.lu</code>) și rezolvarea sistemelor liniare. Compararea cu eliminarea gaussiană fără pivotare pe un sistem slab condiționat. Implementarea metodei Gauss-Seidel și vizualizarea convergenței. Analiza influenței numărului de condiționare asupra erorii soluției.	2	Lucrare practică Python/NumPy	
Lucrarea L4: Interpolare și aproximare prin metoda celor mai mici pătrate. Implementarea interpolării Lagrange și a spline-ului cubic cu <code>scipy.interpolate</code> . Demonstrarea fenomenului Runge cu noduri uniforme vs. noduri Cebîșev pe intervalul $[-1, 1]$. Regresia polinomială cu <code>numpy.polyfit</code> și cu <code>scipy.optimize.curve_fit</code> pentru date cu incertitudini. Calculul și interpretarea R^2 . Aplicație: ajustarea unei curbe de calibrare a unui senzor.	2	Lucrare practică Python	
Lucrarea L5: Integrare numerică și derivare. Implementarea regulii trapezelor compuse și a regulii Simpson 1/3 compuse. Integrarea Romberg cu <code>scipy.integrate.romberg</code> . Cuadratura gaussiană cu <code>scipy.integrate.fixed_quad</code> . Compararea erorilor și a numărului de evaluări ale funcției pentru aceeași precizie. Aplicație: calculul numeric al transformatei Fourier discrete (DFT) pentru un semnal de test.	2	Lucrare practică Python	
Lucrarea L6: Rezolvarea numerică a EDO — metode Runge-Kutta și pas adaptiv. Implementarea metodei Euler explicite și RK4 de la zero pentru o EDO cu soluție analitică cunoscută. Compararea erorii globale pentru diferite valori ale pasului h . Utilizarea <code>scipy.integrate.solve_ivp</code> (RK45) cu toleranțe adaptate. Aplicație: simularea unui circuit RC (ecuație diferențială de ordinul 1) și a unui oscilator armonic amortizat (sistem de EDO de ordinul 2).	2	Lucrare practică Python	
Lucrarea L7: Metoda Monte Carlo și proiect integrat. Estimarea numărului π prin metoda Monte Carlo cu $N \in \{100, 1000, 10000, 100000\}$ puncte — vizualizarea convergenței și a erorii statistice. Calculul unei integrale multidimensionale prin Monte Carlo vs. cuadratura gaussiană: compararea eficienței în funcție de dimensiune. Optimizare cu gradientul descendent: minimizarea unei funcții de cost cu reprezentarea traiectoriei de convergență. Colocviu oral pe baza lucrărilor L1-L7.	2	Lucrare practică Python, colocviu	

Bibliografie obligatorie laborator

1. Burden, R.L., Faires, J.D., Burden, A.M., Numerical Analysis, 10th ed., Cengage, 2021 — exercițiile practice și pseudocodul algoritmilor.
2. SciPy Documentation — referință completă pentru metodele numerice implementate în Python: <https://docs.scipy.org/doc/scipy/>
3. NumPy Documentation — operații cu matrice și algebra liniară în Python: <https://numpy.org/doc/stable/>

Bibliografie complementara laborator

4. Press, W.H. et al., Numerical Recipes: The Art of Scientific Computing, 3rd ed., Cambridge University Press, 2007: <https://numerical.recipes/>
5. Quarteroni, A., Saleri, F., Gervasio, P., Scientific Computing with MATLAB and Octave, 4th ed., Springer, 2022 — exercițiile de laborator.

Resurse online gratuite recomandate

6. SciPy Lectures — tutoriale Python pentru calcul științific (NumPy, SciPy, Matplotlib): <https://scipy-lectures.org/>
7. Wolfram Alpha — verificarea calculelor analitice și numerice: <https://www.wolframalpha.com/>
8. MIT OCW 18.330 — Introduction to Numerical Analysis, seturi de probleme rezolvate: <https://ocw.mit.edu/courses/18-330-introduction-to-numerical-analysis-spring-2012/>
9. Desmos — vizualizarea grafică interactivă a funcțiilor și a algoritmilor iterativi: <https://www.desmos.com/>

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

Metodele Numerice oferă inginerului TI instrumentele matematice esențiale pentru a rezolva probleme computaționale care nu admit soluții analitice exacte — și aceasta descrie marea majoritate a problemelor reale din industrie. Rezolvarea sistemelor liniare mari, interpolarea și aproximarea datelor experimentale, integrarea numerică, simularea ecuațiilor diferențiale și optimizarea sunt operații prezente zi de zi în Machine Learning (antrenarea rețelelor neuronale prin gradient descent), simularea circuitelor (metoda nodurilor cu eliminare gaussiană), grafica 3D (transformări geometrice, rasterizare), procesarea semnalelor (FFT, filtrare numerică) și finanțe computaționale (simulare Monte Carlo).

Conținuturile disciplinei sunt alinate curriculum-ului ACM/IEEE CS2023 (Knowledge Area: Numerical Methods — NM) și standardelor internaționale pentru ingineria computațională. Introducerea metodei Monte Carlo (Tema 13) și a optimizării numerice (Tema 14) conectează disciplina direct cu Machine Learning și cu inteligența artificială — abordări prezente în toate domeniile de specialitate ale programului TI. Componenta aplicativă în Python (SciPy, NumPy) pregătește studentul să utilizeze instrumentele industriale standard, nu doar să cunoască teoretic algoritmi.

Lucrarea L7 — care combină Monte Carlo, optimizare cu gradient descent și colocviu oral pe întregul conținut al laboratorului — urmărește formarea capacității de a evalua critic și independent rezultatele simulărilor: studentul nu doar rulează codul, ci compară metodele, analizează convergența, estimează incertitudinea și comunică transparent limitele modelului numeric față de soluția analitică sau experimentală — direct corelată cu R41 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 principalelor metode numerice din tematica disciplinei: rezolvarea ecuațiilor neliniare (bisecție, Newton-Raphson), rezolvarea sistemelor liniare (LU, Gauss-Seidel),	Colocviu scris (1,5 ore): 1 problemă de rezolvare a unei ecuații neliniare cu analiza convergenței, 1 problemă de rezolvare a unui sistem liniar cu factorizare LU sau Gauss-Seidel, 1 problemă de integrare numerică sau rezolvare EDO cu estimarea erorii și 1 problemă de	50%

	interpolarea (Lagrange, spline), integrarea numerică (Simpson, Romberg), rezolvarea EDO (RK4, pas adaptiv), metoda Monte Carlo; conștientizarea erorilor de trunchiere, de rotunjire și a limitelor fiecărei metode.	interpolare sau Monte Carlo — cu toate calculele intermediare și justificarea alegerii metodei	
8.5 Seminar/laborator	Corectitudinea implementărilor Python din laborator și calitatea rapoartelor (cod funcțional, grafice clare, analiza erorilor și a convergenței); calitatea colocviului oral din L7 — explicarea metodelor implementate, compararea lor și comunicarea transparentă a limitelor; participarea activă la toate lucrările.	Evaluare continuă: implementări Python individuale predate pe Sakai după fiecare lucrare, cu cod, grafice și un paragraf de analiză a erorii (50%) și colocviu oral individual la L7 — explicarea și compararea metodelor implementate pe parcursul semestrului (50%)	50%

8.6 Standard minim de performanță:

Rezolvarea corectă cu calculele complete a cel puțin două din cele patru probleme de la colocviul scris, cu obligativitatea includerii problemei de rezolvare numerică a ecuației neliniare sau a sistemului liniar — aceasta verificând stăpânirea metodelor de bază ale disciplinei. La colocviul oral din L7: prezentarea unui grafic de convergență generat de student (pentru o metodă iterativă implementată pe parcurs) cu explicarea orală a relației dintre pasul h sau numărul de iterații și eroarea observată. Simpla rulare a codului fără analiza erorilor nu îndeplinește standardul minim.

Data completării	Semnătura titularului de curs Lect. Dr. Ion Adelina	Semnătura titularului de seminar / laborator Lect. Dr. Ion Adelina
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	TEORIA TRANSMISIUNII INFORMAȚIEI– TIDS212						
1.2 Titularul activităților de curs	Prof. Dr. Mînză Viorel						
1.3 Titularul activităților de seminar	Lect. Dr. Damian Felicia Anișoara						
1.4 Titularul activităților de laborator							
1.5 Anul de studiu	II	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	5	din care: 2.2 curs	3	2.3 seminar/laborator	2/0
2.4 Total ore din planul de învățământ	70	din care: 2.5 curs	42	2.6 seminar/laborator	28/0
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					52
Documentare					15
Studiu individual					15
Referate					15
Teme casă					3
Proiect					2
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					125
2.10 Numărul de credite					5

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Teoria Probabilităților și Statistică Matematică (TIDF201) — variabile aleatoare, distribuții, entropie informațională; Matematici Discrete (TIDS205) — algebră booleană, coduri; Proiectare Logică (TIDS206) — circuite digitale, codificatoare; Matematici Speciale (TIDF110) — transformata Fourier, semnale și sisteme liniare.
3.2 de competențe	Cunoașterea distribuțiilor de probabilitate discrete și continue și a noțiunii de entropie; înțelegerea transformatei Fourier discrete și a spectrului de frecvență al unui semnal; familiarizarea cu aritmetica modulară și cu polinoamele peste câmpuri finite (GF(2)); abilitatea de a implementa algoritmi matematici în Python.

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 e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, seturilor de probleme și temelor; recomandare: instalarea Python 3.10+ cu NumPy, SciPy și Matplotlib, sau MATLAB, pentru verificarea calculelor numerice din curs.
-------------------------------	--

4.2 de desfășurare a seminarului/laboratorului	Sală de seminar cu acces la calculatoare individuale sau la laptopuri proprii; Python 3.10+ cu NumPy, SciPy și Matplotlib instalate pentru simularea canalelor de comunicație, calculul capacității și implementarea algoritmilor de codare/decodare; studenții rezolvă probleme la tablă și verifică numeric cu Python.
--	--

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: Introducere în teoria informației. Conceptul de informație și comunicație. Sistemul de comunicații: model Shannon (sursă, codor sursă, canal, codor canal, decodor, destinatar). Incertitudinea și măsura informației: definiția lui Shannon a cantității de informație $I(x_i) = \log_2(1/p(x_i))$. Unități de măsură: bit, nat, hartley. Proprietăți ale măsurii informației.	3	Prelegere interactivă, demonstrații	
Tema 2: Entropia informațională. Entropia sursei discrete: $H(X) = -\sum p(x_i) \cdot \log_2 p(x_i)$. Proprietăți: non-negativitate, maxim pentru distribuție uniformă, aditivitate. Entropia condiționată $H(X Y)$ și entropia comună $H(X,Y)$. Informația mutuală $I(X;Y) = H(X) - H(X Y)$: proprietăți și interpretare geometrică. Diagrama Venn a entropiilor. Entropia diferențială pentru surse continue.	3	Prelegere, demonstrații matematice	
Tema 3: Surse de informație discrete și codarea sursă. Sursa Markov: lanțuri Markov de ordinul 1, matricea de tranziție, distribuția staționară, entropia sursei Markov. Teorema codificării sursă (Shannon): lungimea medie minimă a cuvântului de cod. Coduri fără prefix (prefix-free codes): arborele de cod, inegalitatea Kraft-McMillan. Coduri eficiente: codul Shannon-Fano și codul Huffman — construcție și optimalitate.	3	Prelegere, algoritm Huffman	
Tema 4: Codarea aritmetică și comprimarea datelor. Codul Huffman adaptiv: actualizarea arborelui pe parcursul codificării. Codarea aritmetică: principiu, implementare cu precizie finită, eficiență apropiată de entropie. Algoritmii Lempel-Ziv (LZ77, LZ78, LZW):	3	Prelegere interactivă	

principii, dicționar dinamic, aplicații (gzip, deflate, ZIP). Compararea metodelor de compresie: raport de compresie, complexitate, aplicații.			
Tema 5: Canale de comunicații discrete. Canalul discret fără memorie (DMC): definiție, matricea de tranziție a canalului. Canalul binar simetric (BSC): probabilitatea de eroare pe bit. Canalul binar cu ștergeri (BEC). Capacitatea canalului: $C = \max_{p(x)} I(X;Y)$. Calculul capacității pentru BSC și BEC. Teorema fundamentală a lui Shannon (teorema codificării canal): condiția de comunicație fără erori cu rată sub C.	3	Prelegere, demonstrații	
Tema 6: Canale de comunicații continue și capacitatea Gaussian. Canalul AWGN (Additive White Gaussian Noise): modelul matematic. Capacitatea canalului Gaussian: $C = W \cdot \log_2(1 + \text{SNR})$ — formula Shannon-Hartley. Interpretare: lărgime de bandă vs. raport semnal/zgomot. Limitele fundamentale ale comunicației digitale: diagrama waterfall. Eficiența spectrală: bit/s/Hz. Canalul cu bandă limitată și cu putere limitată.	3	Prelegere, calcule numerice	
Tema 7: Coduri detectoare și corectoare de erori — fundamente. Modelul canalului cu erori. Distanța Hamming: definiție și proprietăți. Capacitatea de detecție și corecție a erorilor în funcție de distanța minimă d_{\min} . Codul de repetiție: exemplu simplu, ineficiență. Codul de paritate: detecție simplă. Codul Hamming (7,4): construcție, matricea de verificare a parității H, matricea generatoare G, corecția unui singur bit.	3	Prelegere, demonstrații	
Tema 8: Coduri liniare în bloc. Coduri liniare sistematice (n,k): spațiul cuvintelor de cod ca subspațiu vectorial al $\text{GF}(2)^n$. Matricea generatoare G și matricea de verificare H. Sindromul unui cuvânt recepționat: decodarea bazată pe sindrom. Codul Hamming generalizat ($2^r-1, 2^r-1-r$). Codurile Reed-Muller. Codul Golay (23,12): capacitate de corecție. Limitele Singleton, Hamming (Sphere Packing) și Plotkin.	3	Prelegere, algebra $\text{GF}(2)$	
Tema 9: Coduri ciclice. Codurile ciclice ca ideale în inelul de polinoame $\text{GF}(2)[x]/(x^n-1)$. Polinomul generator $g(x)$ și polinomul de paritate $h(x)$. Implementarea cu registre de deplasare cu feedback (LFSR). Codurile BCH (Bose-Chaudhuri-Hocquenghem): construcție și decodare. Codul CRC (Cyclic Redundancy Check): standard CRC-32, CRC-16 — utilizare în protocoale de comunicație (Ethernet, USB, ZIP).	3	Prelegere interactivă	

<p>Tema 10: Coduri Reed-Solomon și coduri pentru canale burst. Corpul finit Galois $GF(2^m)$: construcție cu polinomul primitiv, aritmetica în $GF(2^m)$. Codurile Reed-Solomon (n,k) peste $GF(2^m)$: parametri, distanța minimă $n-k+1$. Decodarea Berlekamp-Massey. Capacitatea de corectare a erorilor în rafală (burst). Aplicații: CD, DVD, QR codes, comunicații spațiale (Voyager). Codul turbo: introducere și performanțe.</p>	3	Prelegere interactivă	
<p>Tema 11: Modulații digitale. Reprezentarea geometrică a semnalelor: spațiul semnalelor, funcțiile ortogonale de bază. Modulații binare: BPSK, BFSK, BASK — constelații, probabilitatea de eroare. Modulații M-are: QPSK, 8-PSK, 16-QAM, 64-QAM — eficiență spectrală vs. robustețe. Performanța în canalul AWGN: curba P_b vs. E_b/N_0. Diagrama ochi și factorii care o degradează.</p>	3	Prelegere interactivă, constelații	
<p>Tema 12: Codarea canal modernă — coduri turbo și LDPC. Codurile convoluționale: encoder cu registre de deplasare, distanța liberă, algoritmul Viterbi pentru decodare (decoding pe baza traiectoriei de cost minim). Recursivitate și sistematic în codurile convoluționale. Codurile turbo (Berrou, 1993): encodare paralele, decodare iterativă cu algoritmul BCJR, convergența decodării. Coduri LDPC (Low-Density Parity-Check): matrice H rară, decodare sum-product (Belief Propagation).</p>	3	Prelegere interactivă	
<p>Tema 13: Criptografia din perspectiva teoriei informației. Securitatea perfectă (Shannon, 1949): definiție, one-time pad — demonstrarea securității perfecte. Entropia cheii și entropia mesajului. Distanța de unicitate: lungimea minimă a textului cifrat care permite recuperarea cheii. Criptografia cu cheie publică: RSA — bazele aritmetice (modularul invers, teorema Fermat). Protocoale de securitate: TLS/SSL — rolul teoriei informației.</p>	3	Prelegere interactivă	
<p>Tema 14: Teoria informației în sisteme moderne și recapitulare. Capacitatea canalelor cu mai mulți utilizatori: MAC (Multiple Access Channel), BC (Broadcast Channel). Compresia cu pierderi (lossy compression): distorsiunea și funcția rată-distorsiune $D(R)$. Teoria informației în Machine Learning: entropia încrucișată ca funcție de cost, KL-divergența, informația mutuală în rețelele neuronale. Recapitulare generală.</p>	3	Prelegere, recapitulare	

	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Mînză, V., Șerbescu, A., Beldiman, L., Șerbescu, A., Théorie des systèmes — éléments fondamentaux, Fondation Académica, România, 2003 — capitolele de teoria semnalelor și a sistemelor, ca fundament al TTI. 2. Cover, T.M., Thomas, J.A., Elements of Information Theory, 2nd ed., Wiley-Interscience, 2006 (retipărit 2022) — referința mondială în teoria informației. 3. Proakis, J.G., Salehi, M., Digital Communications, 5th ed., McGraw-Hill, 2008 . <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Blahut, R.E., Principles and Practice of Information Theory, Addison-Wesley, 1987 (retipărit 2020) — codare algebrică, coduri ciclice și Reed-Solomon. 5. MacKay, D.J.C., Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003. Disponibil gratuit la: https://www.inference.org.uk/mackay/itila/ 6. Moon, T.K., Error Correction Coding: Mathematical Methods and Algorithms, 2nd ed., Wiley-IEEE Press, 2020. <p>Resurse software:</p> <ol style="list-style-type: none"> 7. David MacKay — Information Theory, Inference and Learning Algorithms (carte gratuită, cod sursă inclus): https://www.inference.org.uk/mackay/itila/ 8. MIT OpenCourseWare 6.441 — Information Theory, note de curs și probleme rezolvate: https://ocw.mit.edu/courses/6-441-information-theory-spring-2016/ 9. SciPy Documentation — funcții de entropie și statistică pentru calcul numeric: https://docs.scipy.org/doc/scipy/reference/stats.html 		
6.2 Seminar	Nr. ore	Metode de predare	Observații
<p>Tema S1: Calculul entropiei — exerciții. Calculul entropiei $H(X)$ pentru surse discrete cu distribuții date. Verificarea proprietăților entropiei (maxim pentru distribuție uniformă, nulă pentru surse deterministe). Calculul entropiei condiționate $H(X Y)$ și al informației mutuale $I(X;Y)$ pentru canale simple. Implementare Python: calcul numeric al entropiei cu <code>scipy.stats</code>.</p>	2	Rezolvare probleme, Python	
<p>Tema S2: Coduri Huffman — construcție și analiză. Construirea manuală a arborelui Huffman pentru alfabet cu 4-8 simboluri. Calculul lungimii medii a cuvântului de cod și compararea cu entropia sursei. Demonstrarea optimalității față de alte coduri fără prefix. Implementarea în Python a algoritmului Huffman cu priority queue. Analiza eficienței pentru distribuții variate.</p>	2	Rezolvare probleme, Python	
<p>Tema S3: Capacitatea canalului — calcule practice. Calculul capacității canalului binar simetric (BSC) în funcție de probabilitatea de eroare p: $C(p) = 1 - H(p)$. Calculul capacității canalului binar cu ștergeri (BEC): $C = 1 - \epsilon$. Calculul capacității canalului Gaussian (Shannon-Hartley) pentru valori diferite de SNR și lățime de bandă. Interpretarea diagramei eficiență spectrală vs. SNR.</p>	2	Rezolvare probleme, Python	
<p>Tema S4: Codul Hamming — construcție și corecție. Construirea manuală a codului Hamming (7,4): matricea generatoare G și matricea de verificare H. Codarea unui mesaj de 4 biți. Introducerea unui singur bit eronat și decodarea prin calculul sindromului. Verificarea proprietăților: distanța minimă $d_{min} = 3$, corecția unui bit, detecția a doi biți. Compararea eficienței față de codul de repetiție (3,1).</p>	2	Rezolvare probleme la tablă	

<p>Tema S5: Coduri liniare în bloc — probleme generale. Exerciții sistematice de codare și decodare pentru coduri liniare (n,k) date prin matricea G sau H. Calculul sindromului pentru diferite tipuri de erori (eroi simple, duble, triple) și analiza capacității de corecție. Compararea codurilor cu aceeași rată dar distanțe minime diferite. Implementarea Python a codificatorului și decodatorului bazat pe sindrom.</p>	2	Rezolvare probleme, Python	
<p>Tema S6: Coduri CRC — implementare și analiză. Calculul manual al CRC pentru un mesaj scurt cu polinomul generator CRC-8. Verificarea detecției erorilor simple și a erorilor în rafală. Implementarea în Python a CRC-32 (standard IEEE 802.3) cu registru LFSR. Testarea pe mesaje reale și compararea cu funcția binascii.crc32 din Python. Analiza probabilității de detecție ratată (aliasing).</p>	2	Rezolvare probleme, Python	
<p>Tema S7: Coduri Reed-Solomon — aritmetică în $GF(2^m)$. Construirea câmpului finit $GF(2^3)$ cu polinomul primitiv $p(x) = x^3 + x + 1$: tabela de înmulțire și adunare. Calculul inversului multiplicativ în $GF(2^3)$. Codarea unui mesaj scurt cu codul RS(7,3) peste $GF(2^3)$. Introducerea a două erori și corecția prin calculul sindromului. Discutarea aplicației în codarea QR.</p>	2	Rezolvare probleme	
<p>Tema S8: Modulații digitale — analiza performanței. Trasarea constelațiilor BPSK, QPSK și 16-QAM în Python (matplotlib). Calculul distanței minime dintre punctele constelației în funcție de puterea medie. Calculul probabilității de eroare P_b în funcție de E_b/N_0 pentru BPSK și QPSK. Compararea eficienței spectrale (bit/s/Hz) și a robusteții diferitelor modulații.</p>	2	Rezolvare probleme, Python	
<p>Tema S9: Simularea unui sistem de comunicații digital complet. Implementarea în Python a unui sistem complet: sursă → codare Huffman → modulare BPSK → canal AWGN (zgomot gaussian) → demodulare → decodare. Trasarea curbei BER (Bit Error Rate) în funcție de SNR. Compararea performanței cu și fără codare de canal (codul Hamming (7,4)).</p>	2	Lucrare practică Python	
<p>Tema S10: Coduri convoluționale — encoder și algoritmul Viterbi. Proiectarea unui encoder convoluțional de rată 1/2 cu registre de deplasare. Calculul ieșirilor pentru un mesaj de 8 biți. Trasarea treilisului. Aplicarea algoritmului Viterbi (hard decision) pentru decodarea unui cuvânt cu 2 erori. Compararea cu decodarea prin căutare exhaustivă.</p>	2	Rezolvare probleme la tablă	
<p>Tema S11: Compresia cu pierderi și funcția rată-distorsiune. Cuantizarea scalară uniformă</p>	2	Rezolvare probleme, Python	

<p>și neuniformă: calculul distorsiunii MSE pentru un semnal gaussian. Cuantizarea vectorială: principiul și eficiența față de scalară. Compresia imaginilor JPEG: principiul transformatei DCT și al cuantizării coeficienților. Calculul PSNR (Peak Signal-to-Noise Ratio) pentru diferite rate de compresie.</p>			
<p>Tema S12: Criptografie și teoria informației — exerciții. Demonstrarea practică a securității perfecte a cifrului one-time pad: calculul entropiei cheii și al cheii condiționate pe mesaj. Calculul distanței de unicitate pentru cifrul Caesar și pentru DES (estimare). Analiza entropiei parolelor: calculul entropiei unui șir de n caractere din diferite alfabete. Implicații pentru politicile de securitate.</p>	2	Rezolvare probleme, Python	
<p>Tema S13: Teoria informației în Machine Learning. Calculul entropiei încrucișate $H(p,q) = -\sum p(x) \cdot \log q(x)$ ca funcție de cost pentru clasificare multclasă — comparare cu MSE. Calculul divergenței KL $D_{KL}(P Q)$ pentru distribuții discrete și continue. Demonstrarea că minimizarea entropiei încrucișate echivalează cu maximizarea log-verosimilității. Aplicație Python: antrenarea unui clasificator simplu cu cross-entropy loss.</p>	2	Rezolvare probleme, Python	
<p>Tema S14: Recapitulare și modele de examene. Rezolvarea a 2-3 subiecte model de examen cu toate tipurile de exerciții: calcul entropie și capacitate, construire cod Huffman, codare/decodare Hamming/CRC, calculul probabilității de eroare la modulație. Discutarea greșelilor tipice și a strategiei de abordare. Feedback individual și recomandări.</p>	2	Recapitulare, probleme model	
	<p>Bibliografie obligatorie seminar</p> <ol style="list-style-type: none"> Cover, T.M., Thomas, J.A., Elements of Information Theory, 2nd ed., Wiley, 2006 — exercițiile de la finalul capitolelor. MacKay, D.J.C., Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003 — probleme și soluții: https://www.inference.org.uk/mackay/itila/ Moon, T.K., Error Correction Coding: Mathematical Methods and Algorithms, 2nd ed., Wiley-IEEE Press, 2020 — exercițiile de codare algebrică. <p>Bibliografie complementara seminar</p> <ol style="list-style-type: none"> Blahut, R.E., Principles and Practice of Information Theory, Addison-Wesley, 2020 — exercițiile de coduri ciclice. Proakis, J.G., Salehi, M., Digital Communications, 5th ed., McGraw-Hill, 2021 — problemele de modulații și performanță. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> David MacKay — Information Theory, carte gratuită cu probleme rezolvate și cod Python: https://www.inference.org.uk/mackay/itila/ MIT OCW 6.441 — seturi de probleme rezolvate de teoria informației: https://ocw.mit.edu/courses/6-441-information-theory-spring-2016/ NumPy / SciPy Documentation — calcule de entropie și statistici: https://docs.scipy.org/doc/scipy/reference/stats.html Shannon entropy calculator online — verificarea calculelor de entropie: https://planetcalc.com/2476/ 		

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

Teoria Transmisiunii Informației furnizează fundamentul matematic al întregii inginerii de comunicații digitale: limitele capacității canalului, codarea sursă optimală și codurile corectoare de erori sunt rezultate directe ale teoriei Shannon, prezente zi de zi în protocoalele de rețea, codecurile audio-video, sistemele de stocare și schemele de criptografie. Fără aceste fundamente, inginerul TI nu poate evalua critic de ce un sistem de comunicații funcționează sau eșuează la limita sa teoretică. Disciplina acoperă traseul complet de la măsura informației și entropia Shannon până la modulații digitale moderne și coduri corectoare de ultimă generație (turbo, LDPC), conectând direct cu disciplinele de rețele, securitate informatică și prelucrarea semnalelor din semestrul următor. Seminarul, centrat pe rezolvarea de probleme și pe simularea în Python a sistemelor complete de comunicații, urmărește formarea capacității de a analiza și evalua cantitativ performanța unui sistem real — direct corelată cu A12 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 măsurii informației, entropiei și informației mutuale; stăpânirea algoritmului Huffman și a principiilor de compresie fără pierderi; cunoașterea formulei Shannon-Hartley și a limitelor fundamentale ale capacității canalului; cunoașterea și aplicarea codurilor detectoare și corectoare de erori (Hamming, CRC, Reed-Solomon); cunoașterea modulațiilor digitale principale și a performanței lor.	Examen scris (2 ore): 1 problemă de calcul al entropiei și capacității canalului, 1 problemă de construire a unui cod Huffman sau CRC cu analiza eficienței, 1 problemă de codare/decodare Hamming sau Reed-Solomon (cu identificarea erorilor) și 1 problemă de modulație digitală (constelație, probabilitate de eroare) — toate cu calculele complete	60%
8.5 Seminar/laborator	Corectitudinea rezolvărilor la seminar și calitatea implementărilor Python; calitatea simulării sistemului complet de comunicații din S9 (cod funcțional, curbă BER corectă, analiză comparativă); participarea activă la toate seminarele; progresul demonstrat față de evaluarea inițială.	Evaluare continuă: rezolvări individuale predate pe Sakai după seminarele cu implementare Python (S2, S5, S6, S8, S9, S12, S13) — câte un raport scurt cu cod și analiză (50%) și participare activă și calitatea rezolvărilor la seminarele teoretice (50%)	40%
8.6 Standard minim de performanță:			
Rezolvarea corectă și cu calculele complete la examenul scris a cel puțin două din cele patru probleme, cu obligativitatea de a include problema de calcul al capacității canalului (formula Shannon-Hartley sau capacitatea BSC) — aceasta verificând stăpânirea conceptului central al			

disciplinei. Justificarea valorilor numerice obținute în raport cu limitele fundamentale Shannon (de ce un sistem funcționează sau nu sub capacitatea maximă) este criteriu de diferențiere față de simpla aplicare mecanică a formulelor.

Data completării 19.03.2026	Semnătura titularului de curs Prof. Dr. Mînză Viorel	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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei		TEORIA INFORMAȚIEI ȘI CODURI- TIDS213					
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator / proiect		Prof. Dr. Mînză Viorel					
1.5 Anul de studiu	II	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					
Studiu individual					
Referate					
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					
2.9 Total ore pe semestru					25
2.10 Numărul de credite					1

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Teoria Transmisiunii Informației (TIDS212) — entropie, capacitatea canalului, codare sursă și canal; Teoria Probabilităților și Statistică Matematică (TIDF201) — distribuții, estimare statistică, decizie bayesiană; Metode Numerice (TIDS211) — simulare numerică, Monte Carlo.
3.2 de competențe	Cunoașterea noțiunilor de entropie Shannon, capacitatea canalului și a principiilor codurilor detectoare/corectoare de erori; capacitatea de a aplica estimatori statistici și criterii de decizie bayesiană; abilitatea de a implementa algoritmi în Python (NumPy, SciPy, Matplotlib).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	
4.2 de desfășurare a seminarului/laboratorului	Activitatea de proiect se desfășoară individual, la calculatorul propriu sau în laboratorul de informatică; Python 3.10+ cu NumPy, SciPy și Matplotlib; întâlnirile de îndrumare cu titularul au loc săptămânal (față în față sau online),

	cu prezentarea stadiului și primirea feedback-ului; predarea finală 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	C12. Cunoaște în profunzime principiile sistemelor și rețelelor de comunicații, inclusiv teoria transmisiunii informației și tehnicile moderne de codare și modulare a semnalelor. C41. Deține cunoștințe specializate de estimare și decizie în teoria informației — criterii Bayes, estimatori de maximum likelihood, teoria detecției — cu aplicabilitate în proiectarea sistemelor de comunicații și de recunoaștere a formelor.
5.2 Aptitudini	A36. Aplică metode de estimare statistică și decizie bayesiană în analiza datelor experimentale și în proiectarea sistemelor de detecție, evaluând performanța estimatorilor și selectând criteriile optime de decizie.
5.3 Responsabilitate și autonomie	R36. Își asumă responsabilitatea pentru aplicarea riguroasă a metodelor statistice de estimare și decizie în analiza datelor experimentale, documentând transparent ipotezele statistice și comunicând onest limitele concluziilor formulate. 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 Proiect	Nr. ore	Metode de predare	Observații
P1: Stabilirea temei și planificarea proiectului. Prezentarea temelor disponibile (compresie, codare canal, detecție, criptografie informațională). Alegerea temei individuale, definirea obiectivelor și a cerințelor. Stabilirea planului de lucru pe 14 săptămâni: etape, livrabile, criterii de evaluare.	1	Îndrumare individuală	
P2: Documentare — fundamente teoretice. Studiul bibliografiei specifice temei alese. Identificarea algoritmilor și metodelor relevante. Prezentarea unui rezumat documentar de 1-2 pagini cu principiile teoretice ale temei.	1	Studiu individual, prezentare	
P3: Specificarea sistemului de proiectat. Definirea formală a problemei: intrări, ieșiri, constrângeri, metrici de performanță (BER, raport de compresie, probabilitate de eroare de detecție). Schița arhitecturii soluției.	1	Îndrumare, feedback	
P4: Implementare — modulul de bază I. Implementarea primului modul al proiectului în Python (ex: coder/decoder, estimator, simulator canal). Testarea pe date sintetice simple.	1	Implementare Python, review cod	
P5: Implementare — modulul de bază II. Continuarea implementării: al doilea modul al proiectului. Integrarea cu modulul din P4. Teste de regresie.	1	Implementare Python, review cod	
P6: Validare pe date sintetice. Testarea sistemului complet pe seturi de date sintetice	1	Testare, analiză rezultate	

cu parametri controlați. Verificarea corectitudinii față de rezultatele teoretice cunoscute.			
P7: Analiza performanței — curbe și metrici. Generarea curbelor de performanță (ex: BER vs. SNR, compresie vs. distorsiune, probabilitate de detecție vs. prag). Compararea cu limitele teoretice Shannon sau Cramér-Rao.	1	Analiză, matplotlib	
P8: Prezentare intermediară — stadiul proiectului. Prezentarea orală a stadiului: arhitectura implementată, rezultatele obținute, problemele identificate. Feedback din partea titularului și ajustarea planului.	1	Prezentare, feedback	
P9: Implementare avansată — extensii și optimizări. Implementarea elementelor avansate ale temei (ex: decodare soft-decision, compresie adaptivă, detector optimal Bayes). Optimizarea codului pentru eficiență.	1	Implementare Python	
P10: Validare comparativă. Compararea soluției implementate cu o implementare de referință sau cu rezultatele din literatura de specialitate. Calculul erorii relative și discutarea discrepanțelor.	1	Analiză comparativă	
P11: Simulare Monte Carlo — estimarea incertitudinii. Rularea simulărilor repetate pentru estimarea statistică a metricilor de performanță. Calculul intervalelor de încredere. Analiza sensibilității față de parametrii cheie.	1	Simulare Python, statistică	
P12: Redactarea raportului de proiect. Structura raportului: introducere, fundamente teoretice, proiectare, implementare, rezultate, concluzii. Redactarea secțiunilor principale cu figuri, tabele și referințe bibliografice.	1	Redactare tehnică	
P13: Revizuirea raportului și pregătirea prezentării finale. Corectarea raportului pe baza feedback-ului titularului. Pregătirea prezentării PowerPoint: 10 slide-uri cu arhitectura, curbele de performanță și concluziile.	1	Revizuire, prezentare	
P14: Prezentarea și evaluarea proiectului final. Prezentarea orală a proiectului (10 minute + 5 minute Q&A). Predarea raportului final și a codului sursă pe Sakai. Evaluarea pe baza grilei cu criterii explicite.	1	Prezentare orală, evaluare	

Bibliografie obligatorie proiect

1. Cover, T.M., Thomas, J.A., Elements of Information Theory, 2nd ed., Wiley-Interscience, 2006 (retipărit 2022).
2. MacKay, D.J.C., Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003. Disponibil gratuit la: <https://www.inference.org.uk/mackay/itila/>
3. Moon, T.K., Error Correction Coding: Mathematical Methods and Algorithms, 2nd ed., Wiley-IEEE Press, 2020.

Bibliografie complementara proiect

4. Proakis, J.G., Salehi, M., Digital Communications, 5th ed., McGraw-Hill, 2008 .
5. Kay, S.M., Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory, Prentice Hall, 1993

Resurse online gratuite recomandate

	6. SciPy Documentation — statistică, optimizare și procesarea semnalelor în Python: https://docs.scipy.org/doc/scipy/ 7. David MacKay — Information Theory carte gratuită cu cod sursă: https://www.inference.org.uk/mackay/itila/ 8. MIT OCW 6.441 — Information Theory, probleme și soluții: https://ocw.mit.edu/courses/6-441-information-theory-spring-2016/
--	--

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

Proiectul de Teoria Informației și Coduri integrează competențele dobândite la Teoria Transmisiunii Informației (TIDS212) într-o aplicație practică completă, acoperind traseul de la specificarea teoretică a problemei până la implementarea, simularea și evaluarea critică a soluției. Activitatea de proiect formează autonomia tehnică și responsabilitatea față de validarea riguroasă a rezultatelor. Cele 14 ședințe de îndrumare săptămânală reproduc modul de lucru dintr-un proiect de cercetare sau de dezvoltare: studentul avansează independent între ședințe, prezintă stadiul, primește feedback și ajustează direcția. Prezentarea finală cu Q&A certifică nu doar corectitudinea implementării, ci și capacitatea de a justifica deciziile tehnice și de a comunica transparent limitele soluției.

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	Funcționalitatea și corectitudinea implementării Python (rezultate numerice corecte, cod curat și comentat); calitatea analizei de performanță (curbe generate, comparare cu limitele teoretice, discutarea discrepanțelor); calitatea raportului scris (structură, rigoare, figuri, referințe); prezentarea orală și răspunsurile la întrebări.	Proiect individual evaluat la P14: implementare Python funcțională (40%) + raport tehnic scris predat pe Sakai (40%) + prezentare orală cu Q&A de 15 minute (20%)	100%
8.6 Standard minim de performanță:			
Predarea unui cod Python funcțional care generează cel puțin o curbă de performanță relevantă temei (BER vs. SNR, compresie vs. distorsiune sau probabilitate de detecție vs. prag) ȘI prezentarea la P14 a cel puțin unui rezultat numeric comparat cu limita teoretică corespunzătoare (capacitate Shannon, limita Cramér-Rao sau bound-ul Hamming). Cod care rulează fără a produce rezultate interpretabile nu îndeplinește standardul minim.			

Data completării 19.03.2026	Semnătura titularului de curs	Semnătura titularului de seminar / laborator/proiect Prof. Dr. Mînză Viorel
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	ARHITECTURA SISTEMELOR DE CALCUL – TIDS214						
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	II	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	3	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	42	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					44
Documentare					15
Studiu individual					10
Referate					10
Teme casă					5
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	Proiectare Logică (TIDS206) — circuite combinaționale și secvențiale, mașini de stare, Verilog; Electronică Digitală (TIDS209) — memorii, conversie ADC/DAC, interfețe seriale; Structuri de Date și Algoritmi (TIDS203) — complexitate algoritmică, grafuri.
3.2 de competențe	Cunoașterea funcționării circuitelor logice combinaționale și secvențiale; înțelegerea organizării memoriei (SRAM, DRAM, Flash) și a ierarhiei de memorie; abilitatea de a citi și interpreta o schemă bloc a unui sistem digital; familiarizarea cu un limbaj de asamblare simplu sau cu arhitectura unui microcontroller.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoprojector, tablă albă și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor; recomandare: instalarea simulatorului RISC-V (RARS sau Venus) și a compilatorului GCC cross pentru RISC-V înainte de curs.
-------------------------------	--

4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; simulatorul RISC-V RARS instalat; acces la compilatorul GCC cross pentru RISC-V; simulatorul gem5 sau QEMU pentru exercițiile 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	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).
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: Abstracții și performanță. Niveluri de abstractizare în sisteme de calcul: hardware, ISA, microarhitectură, sistem de operare, aplicație. Legea lui Amdahl: speedup, fracție paralelizabilă. Metricile de performanță: MIPS, MFLOPS, CPI, timpul de execuție. Benchmarks: SPEC CPU2017. Tendințe în tehnologia VLSI: Legea lui Moore, Dennard scaling și limitele lor. Consumul de putere: putere dinamică și statică.	3	Prelegere interactivă	
Tema 2: Setul de instrucțiuni RISC-V. Filozofia RISC vs. CISC. Arhitectura RISC-V: registre (32×64-bit), tipuri de instrucțiuni (R, I, S, B, U, J), convenția de apel (ABI). Instrucțiuni aritmetice, logice, de transfer, de ramificare și de salt. Moduri de adresare. Apelul de proceduri: stiva, frame pointer, registre salvate de apelant/apelat. Comparatie cu x86-64 și ARM AArch64.	3	Prelegere, cod RISC-V	
Tema 3: Aritmetică pentru calculatoare. Reprezentarea numerelor întregi: complement față de 2, depășire (overflow). Adunare și scădere în hardware: sumator ripple-carry, sumator carry-lookahead. Înmulțirea întregilor: algoritm secvențial și hardware dedicat. Împărțirea întregilor: algoritm cu restabilire. Reprezentarea în virgulă mobilă (IEEE 754): format simplu și dublu, operații, excepții (NaN, Inf, denormale).	3	Prelegere, demonstrații	
Tema 4: Procesorul — datapath-ul monociclu. Construirea pas cu pas a unui procesor RISC-V monociclu: unitatea de control, ALU, fișierul de registre, memoria de instrucțiuni și de date. Implementarea instrucțiunilor R-type, I-type (load), S-type (store) și B-type (branch).	3	Prelegere, scheme bloc	

Schema bloc completă. Analiza timpului de propagare și stabilirea frecvenței maxime de ceas.			
Tema 5: Procesorul pipeline. Motivația pipelineului: îmbunătățirea throughput-ului. Etapele pipeline RISC-V: IF, ID, EX, MEM, WB. Registrele de pipeline. Hazarduri structurale, de date și de control: detectare și rezolvare prin forwarding, stall și branch prediction. Analiza performanței pipeline: CPI, speedup ideal vs. real. Excepții și întreruperi în pipeline.	3	Prelegere, diagrame pipeline	
Tema 6: Hazarduri și tehnici avansate de pipeline. Forwarding complet: toate cazurile EX-EX, MEM-EX, MEM-MEM. Hazardul load-use: stall-ul de un ciclu. Predicția ramificărilor: statică (always taken, always not taken) și dinamică (bimodal, two-level). Speculative execution. Pipeline-uri superscalare in-order. Introducere în execuția out-of-order: reorder buffer, register renaming.	3	Prelegere interactivă	
Tema 7: Ierarhia de memorie — cache. Principiul localității spațiale și temporale. Cache direct-mapped: index, tag, valid bit, calculul hitului. Cache set-associative și fully-associative. Politici de înlocuire: LRU, random, FIFO. Politici de scriere: write-through și write-back, write-allocate. Rata de miss și categoriile (3C: compulsory, capacity, conflict). Calculul AMAT (Average Memory Access Time).	3	Prelegere, calcule numerice	
Tema 8: Ierarhia de memorie — cache multinivel și DRAM. Cache L1, L2, L3: parametri tipici în procesoarele moderne. Coerența cache-ului în sisteme multiprocesor: protocolul MESI. Memoria DRAM: organizare, refresh, timingul RAS/CAS. SDRAM, DDR SDRAM: burst mode, rate de transfer. Memoria virtuală: spații de adrese, paginare, page table, TLB. Page fault și swapping.	3	Prelegere interactivă	
Tema 9: Memoria virtuală. Paginarea: pagini de dimensiune fixă, page table entry (PTE), biți de protecție și stare. Traducerea adreselor virtuale în fizice: TLB (Translation Lookaside Buffer), TLB miss. Page table ierarhică (multi-level). Memoria virtuală și cache-ul: VIPT, PIPT. Segmentarea. Protecția memoriei: inele de privilegiu, supervisor mode vs. user mode.	3	Prelegere, exemple	
Tema 10: Stocarea și sistemul de I/O. Ierarhia stocării: cache, DRAM, SSD (NAND Flash), HDD. Organizarea HDD: sectoare, piste, cilindri, timingul accesului. SSD: controller, wear leveling, garbage collection, performanță	3	Prelegere interactivă	

vs. HDD. Magistrale: AXI, PCIe, SATA, USB — topologie, protocol, lățime de bandă. DMA (Direct Memory Access): funcționare și avantaje. Întreruperi: controllerul de întreruperi, vectorul de întreruperi.			
Tema 11: Arhitecturi paralele — multicore și ILP. Paralelism la nivel de instrucțiune (ILP): superscalar, VLIW, execuție out-of-order. Procesoare multicore: SMP (Symmetric Multiprocessing), coerența cache MESI/MOESI. Paralelism la nivel de date: SIMD, extensii vectoriale (SSE, AVX, RISC-V V). Paralelism la nivel de fir: SMT (Simultaneous Multithreading), Hyper-Threading Intel. Amdahl revisited pentru sisteme multicore.	3	Prelegere interactivă	
Tema 12: GPU și arhitecturi specializate (DSP, AI). Arhitectura GPU: SM (Streaming Multiprocessor), SIMT (Single Instruction Multiple Threads), modelul CUDA. Comparatie CPU vs. GPU: latență vs. throughput. Procesoare DSP: arhitectura Harvard modificată, unitate MAC, adresare circulară — TI C6000, SHARC. Acceleratoarele AI: Google TPU (Tensor Processing Unit) — arhitectura systolic array. Tendințe: chiplet design, 3D stacking.	3	Prelegere interactivă	
Tema 13: Sistemul de operare și interfața hardware-software. Modurile de execuție RISC-V: M-mode, S-mode, U-mode. Registrele CSR (Control and Status Registers): mstatus, mtvec, mepc, mcause. Tratarea excepțiilor și întreruperilor în hardware. Boot sequence: BIOS/UEFI, bootloader, kernel. Apelurile de sistem (syscall): mecanismul trap, tabelul de syscall-uri. Virtualizarea hardware: hypervisor, VM exit/entry.	3	Prelegere interactivă	
Tema 14: Tendințe actuale și recapitulare. Sfârșitul Legii lui Moore și Dennard scaling: implicații pentru arhitecturile viitoare. Arhitecturi domain-specific (DSA): TPU, NPU, FPGA în cloud. RISC-V în industrie: adopție, ecosistem. Arhitecturi heterogene: big.LITTLE, ARM DynamIQ. Securitatea arhitecturii: Spectre, Meltdown — vulnerabilități micro-arhitecturale și mitigări. Recapitulare generală.	3	Prelegere, recapitulare	
	Bibliografie obligatorie 1. Patterson, D.A., Hennessy, J.L., Computer Organization and Design: RISC-V Edition, 2nd ed., Morgan Kaufmann, 2020. 2. Hennessy, J.L., Patterson, D.A., Computer Architecture: A Quantitative Approach, 6th ed., Morgan Kaufmann, 2017. 3. Harris, D., Harris, S., Digital Design and Computer Architecture: RISC-V Edition, Morgan Kaufmann, 2022. Bibliografie complementară 4. Bryant, R.E., O'Hallaron, D.R., Computer Systems: A Programmer's Perspective, 3rd ed., Pearson, 2015.		

	<p>5. Waterman, A., Asanovic, K. (eds.), The RISC-V Instruction Set Manual, Volume I: Unprivileged Architecture, 2019. Disponibil la: https://riscv.org/technical/specifications/</p> <p>Resurse software:</p> <p>6. RARS — RISC-V Assembler and Runtime Simulator, simulator gratuit: https://github.com/TheThirdOne/rars</p> <p>7. RISC-V International — specificații oficiale ISA și resurse: https://riscv.org/technical/specifications/</p> <p>8. MIT OCW 6.004 — Computation Structures, note de curs și probleme: https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/</p>		
6.2 Laborator	Nr. ore	Metode de predare	Observații
<p>Lucrarea L1: Programare în limbaj de asamblare RISC-V — fundamente. Familiarizarea cu simulatorul RARS. Scrierea, asamblarea și rularea primelor programe RISC-V: afișarea unui șir, calculul sumei a n numere, implementarea funcției factorial recursiv. Analiza registrelor și a stivei după fiecare instrucțiune.</p>	2	Lucrare practică RARS	
<p>Lucrarea L2: Aritmetică și reprezentarea datelor. Implementarea în RISC-V a operațiilor aritmetice pe întregi cu semn și fără semn: detectarea overflow-ului, împărțirea cu semn vs. fără semn. Conversia manuală float IEEE 754 și verificarea cu instrucțiunile RISC-V în virgulă mobilă. Analiza reprezentării unui struct în memorie: padding, aliniament.</p>	2	Lucrare practică RARS	
<p>Lucrarea L3: Simularea unui procesor monociclu. Implementarea în Python sau Logisim a unui procesor RISC-V monociclu simplificat care execută un subset de 8 instrucțiuni (add, sub, and, or, lw, sw, beq, jal). Testarea cu un program de test complet. Calculul timpului de execuție în funcție de frecvența de ceas.</p>	2	Lucrare practică Python/Logisim	
<p>Lucrarea L4: Analiza performanței pipeline. Simularea unui pipeline RISC-V cu 5 etape în Python: implementarea forwarding-ului și detecției hazardurilor de date. Generarea diagramei pipeline (timp vs. etapă) pentru o secvență de instrucțiuni dată. Calculul CPI real față de CPI ideal și analiza impactului load-use hazard.</p>	2	Lucrare practică Python	
<p>Lucrarea L5: Simularea cache-ului. Implementarea în Python a unui simulator de cache direct-mapped și set-associative cu politici LRU și random. Rularea pe un traseu de accese din benchmark SPEC. Calculul ratei de hit și al AMAT pentru diferite configurații. Compararea grafică a politicilor de înlocuire.</p>	2	Lucrare practică Python	
<p>Lucrarea L6: Memoria virtuală și TLB. Implementarea în Python a unui simulator de traducere adrese virtuale în fizice cu page table pe două niveluri și TLB cu 8 intrări (LRU). Analiza ratei de TLB miss pentru diferite programe de acces. Simularea unui page fault și a procesului de swapping.</p>	2	Lucrare practică Python	

<p>Lucrarea L7: Proiect integrat — evaluarea unui sistem de calcul și colocviu. Configurarea și rularea unui benchmark simplu (sorting, matrix multiply) pe simulatorul gem5 sau pe RARS pentru diferite configurații de cache (dimensiune, asociativitate). Documentarea comparativă: tabele de performanță, grafice CPI vs. configurație cache, justificarea configurației optime. Colocviu oral pe baza lucrărilor L1-L7.</p>	2	Benchmark, documentare, colocviu	
<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Patterson, D.A., Hennessy, J.L., Computer Organization and Design: RISC-V Edition, 2nd ed., Morgan Kaufmann, 2020 — exercițiile practice și appendix-ul RISC-V. 2. RARS — documentație și exemple: https://github.com/TheThirdOne/rars 3. RISC-V ISA Specification — referință pentru toate instrucțiunile: https://riscv.org/technical/specifications/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. Harris, D., Harris, S., Digital Design and Computer Architecture: RISC-V Edition, Morgan Kaufmann, 2022 — laboratoarele de proiectare hardware. 5. Bryant, R.E., O'Hallaron, D.R., Computer Systems: A Programmer's Perspective, 3rd ed., Pearson, 2015 — capitolele de cache și memorie virtuală. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. RARS — simulator RISC-V gratuit cu interfață grafică: https://github.com/TheThirdOne/rars 7. Venus — simulator RISC-V online (fără instalare): https://venus.cs61c.org/ 8. MIT OCW 6.004 — laboratoare de Computation Structures: https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/ 			

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

Arhitectura Sistemelor de Calcul răspunde întrebării fundamentale a oricărui inginer TI: cum funcționează de fapt calculatorul pe care rulează software-ul pe care îl scrie. Înțelegerea ierarhiei de memorie, a pipeline-ului, a hazardurilor și a arhitecturilor paralele este indispensabilă pentru scrierea de cod performant, pentru alegerea arhitecturii hardware potrivite unui proiect embedded sau cloud și pentru înțelegerea vulnerabilităților de securitate la nivel microarhitectural (Spectre, Meltdown). Componenta de laborator, bazată pe simulatorul RISC-V și pe implementarea în Python a subsistemelor (procesor monociclu, pipeline, cache, memorie virtuală), urmărește construirea înțelegerii de jos în sus — nu memorarea specificațiilor, ci proiectarea și testarea fiecărui subsistem. Colocviul oral din L7 verifică capacitatea de a justifica alegeri arhitecturale concrete în fața unor cerințe date — direct corelată cu R4 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 arhitecturii RISC-V (formate de instrucțiuni, moduri de adresare, convenția de apel); înțelegerea funcționării procesorului pipeline (etape, forwarding, hazarduri); cunoașterea ierarhiei de memorie (cache, DRAM, memorie virtuală, TLB);	Examen scris (2 ore): 1 problemă de programare RISC-V sau analiză pipeline (diagrama etapelor, identificarea hazardurilor), 1 problemă de cache (calculul AMAT, rata de hit), 1 problemă de memorie virtuală (traducerea adresei, TLB) și 1 problemă de performanță (legea lui Amdahl, CPI) — cu calculele complete	40%

	cunoașterea arhitecturilor paralele (multicore, GPU, DSP) și a tendințelor actuale.		
8.5 Seminar/laborator	Corectitudinea implementărilor Python și a programelor RISC-V din laborator; calitatea documentației lucrărilor (scheme bloc, grafice, justificarea deciziilor); calitatea colocviului oral din L7 — justificarea configurației optime de cache pe baza rezultatelor benchmark-ului; participarea activă la toate lucrările.	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și colocviu oral individual la L7 — prezentarea și justificarea comparativă a configurațiilor de cache testate (50%)	60%
8.6 Standard minim de performanță:			
Rezolvarea corectă cu calculele complete la examenul scris a cel puțin două probleme, cu obligativitatea includerii problemei de pipeline (diagrama etapelor cu identificarea corectă a hazardurilor și a soluției de forwarding sau stall) — aceasta verificând înțelegerea profundă a componentei centrale a disciplinei. La colocviul oral din L7: prezentarea unui tabel comparativ de performanță pentru cel puțin 3 configurații de cache diferite cu justificarea alegerii optime.			

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar / laborator
18.03.2026	Conf. Dr. Răileanu Alina-Beatrice	Conf. Dr. Răileanu Alina-Beatrice
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	PROIECTAREA CU MICROPROCESOARE – TIDS215						
1.2 Titularul activităților de curs	Prof. Dr. Mînză Viorel						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Prof. Dr. Mînză Viorel						
1.5 Anul de studiu	II	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	3	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	42	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					44
Documentare					13
Studiu individual					12
Referate					10
Teme casă					5
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	Arhitectura Sistemelor de Calcul (TIDS214) — organizarea procesorului, ierarhia memoriei, pipeline, interfețe hardware-software; Electronică Digitală (TIDS209) — convertoare ADC/DAC, interfețe seriale SPI/I2C/UART; Proiectare Logică (TIDS206) — circuite combinaționale și secvențiale, HDL.
3.2 de competențe	Cunoașterea arhitecturii unui microcontroller (CPU, memorie Flash/RAM, periferice integrate); abilitatea de a scrie și depana programe C pentru sisteme embedded; cunoașterea funcționării interfețelor seriale (SPI, I2C, UART) și a convertoarelor ADC/DAC; familiarizarea cu utilizarea unui osciloscop și a unui analizor logic.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector, tablă albă și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor; recomandare: instalarea STM32CubeIDE sau PlatformIO cu VS Code înainte de curs, și a mediului Arduino IDE pentru demonstrații rapide.
4.2 de desfășurare a seminarului/laboratorului	Laborator de electronică cu: plăci de dezvoltare STM32 (Nucleo-F401RE sau similar) sau Arduino Mega/ESP32, surse de alimentare, osciloscop

	digitale, analizoare logice (Saleae sau similar), senzori și actuatore (DHT22, HC-SR04, servo, motor DC cu driver L298N), breadboard-uri și cabluri; calculatoare cu STM32CubeIDE și PlatformIO instalate.
--	--

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).
5.2 Aptitudini	A9. Proiectează și configurează sisteme de calcul bazate pe microprocesoare și procesoare digitale de semnal (DSP), integrând componente hardware și software pentru realizarea de sisteme embedded funcționale.
5.3 Responsabilitate și autonomie	R9. Gestionează autonom situații tehnice complexe și imprevizibile — defecțiuni critice de sistem, vulnerabilități de securitate, degradări de performanță — adoptând noi abordări strategice și luând decizii rapide și documentate.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în sistemele embedded. Definiția și caracteristicile sistemelor embedded: constrângeri de resurse, timp real, fiabilitate. Clasificare: bare-metal vs. RTOS, sisteme hard vs. soft real-time. Ciclul de dezvoltare embedded: specificare, proiectare hardware/software, implementare, testare, producție. Arhitectura generică a unui microcontroller: CPU, Flash, SRAM, periferice, interconectare AHB/APB.	3	Prelegere interactivă	
Tema 2: Microcontrolere ARM Cortex-M. Familia ARM Cortex-M: M0, M0+, M3, M4, M7, M33 — diferențe și domenii de aplicabilitate. Arhitectura internă Cortex-M4: pipeline 3 etape, unitate FPU, instrucțiuni DSP. Familia STM32: linii de produse (F0-F7, H7, L0-L5), criterii de selecție (putere, periferice, memorie, preț). Registrele de bază: R0-R12, SP, LR, PC, xPSR. Modurile de operare: Thread/Handler, Privileged/Unprivileged.	3	Prelegere interactivă	
Tema 3: GPIO și gestionarea întreruperilor. Configurarea GPIO: moduri (input, output push-pull/open-drain, alternate function, analog), viteza, pull-up/pull-down. Registrele GPIO STM32: MODER, OTYPER, OSPEEDR, PUPDR, IDR, ODR, BSRR. Sistemul de întreruperi NVIC: priorități, grupuri de priorități, preemption. EXTI (External Interrupt): configurare pentru flanc pozitiv/negativ, debouncing software. Latența de intrare în întrerupere.	3	Prelegere, cod C, scheme	
Tema 4: Timere și generarea semnalelor PWM. Timerele STM32: timer de bază, timer general, timer avansat — structură și registre	3	Prelegere, cod C	

cheie (PSC, ARR, CCR). Modurile de operare: upcounting, downcounting, center-aligned. Generarea PWM: modul PWM1/PWM2, dead time pentru control motor. Captura de intrare (input capture): măsurarea frecvenței și a perioadei. Output compare: generarea impulsurilor precise. Aplicații: controlul intensității LED (dimmer), controlul servomotoarelor.			
Tema 5: Conversoarele ADC și DAC. ADC STM32: arhitectura SAR pe 12 biți, canale multiplexate, conversie single/continuous/scan. Triggering: software, timer, extern. DMA transfer pentru ADC: eliminarea overhead-ului CPU. Calibrarea ADC-ului. Tehnici de filtrare software: medie mobilă, filtru median. DAC STM32: pe 12 biți, triggering, generarea formelor de undă cu DMA. Erori de conversie și tehnici de compensare.	3	Prelegere interactivă	
Tema 6: Interfețe seriale — UART, SPI, I2C. UART: cadrul de date, baudrate, paritate, flow control RTS/CTS, DMA transfer. SPI: master/slave, CPOL/CPHA, full-duplex, DMA. I2C: adresare 7/10 biți, START/STOP, ACK/NACK, clock stretching, multi-master, adresare generală. Compararea protocoalelor: viteză, distanță, număr fire, complexitate. Depanarea cu analizorul logic. Comunicarea cu senzori și periferice comune: display OLED (I2C), SD card (SPI), GPS (UART).	3	Prelegere interactivă	
Tema 7: DMA și eficiența transferurilor de date. Principiul DMA: transferuri fără intervenția CPU. Controllerul DMA STM32: canale, stream-uri, priorități, moduri (normal, circular). Configurarea DMA pentru ADC, UART, SPI, I2C: registrele de configurare, flagurile de stare, întreruperile de transfer complet și eroare. Transferuri memorie-memorie. Impactul DMA asupra performanței: reducerea latentei, creșterea throughput-ului. Hazarduri: coerența cache-ului, accesul simultan la bus.	3	Prelegere, cod C	
Tema 8: Gestionarea puterii și modurile low-power. Consumul de energie în sistemele embedded: CPU, periferice, memoria Flash. Modurile low-power STM32: Sleep, Stop, Standby — comparație latență wake-up vs. consum. RTC (Real-Time Clock): funcționare în Standby, alarme, wake-up timer. Optimizarea consumului: reducerea frecvenței ceasului (voltage scaling), oprirea periferelor neutilizate, instrucțiunea WFI/WFE. Alegerea sursei de ceas: HSI, HSE, PLL. Aplicații IoT alimentate cu baterie.	3	Prelegere interactivă	

<p>Tema 9: RTOS — fundamente. Necesitatea RTOS-ului: limitările programării bare-metal în sisteme complexe. Concepte RTOS: task, scheduler (preemptiv vs. cooperativ), prioritate, stare (ready, running, blocked, suspended). FreeRTOS: arhitectura, portare pe STM32, crearea și gestionarea task-urilor. Sincronizare: semafoare binare și de numărare, mutex, queue. Gestionarea timpului: vTaskDelay, tickuri, timere software. Analiza temporală: worst-case execution time (WCET).</p>	3	Prelegere interactivă	
<p>Tema 10: Comunicații fără fir pentru sisteme embedded. Bluetooth Low Energy (BLE): arhitectura stack, profile GATT, advertising, connection interval. WiFi pe microcontroller: ESP32 — arhitectura dual-core, stack TCP/IP lwIP, conexiunea la rețea. Protocolul MQTT: broker, topic, QoS, aplicații IoT. LoRa/LoRaWAN: modulația chirp spread spectrum, range vs. data rate, aplicații LPWAN. Zigbee: rețele mesh, profilul ZHA. Alegerea protocolului în funcție de cerințe.</p>	3	Prelegere interactivă	
<p>Tema 11: Bootloader și actualizarea firmware-ului (OTA). Structura memoriei Flash: sectoare, pagini, protecție la scriere. Bootloader-ul intern STM32: activare, protocoale suportate (UART, USB, I2C, SPI, CAN). Bootloader personalizat: structura, tabla de vectori, relocarea în RAM, saltul la aplicație. OTA (Over-The-Air) update: mecanismul dual-bank, verificarea integrității cu CRC sau SHA, rollback la firmware anterior. Semnarea firmware-ului pentru securitate.</p>	3	Prelegere interactivă	
<p>Tema 12: Securitate în sistemele embedded. Suprafața de atac a unui sistem embedded: interfețe fizice (JTAG, UART debug), comunicații wireless, firmware. Protecția memoriei Flash: read-out protection (RDP), write protection. Atacuri side-channel: power analysis, timing attacks. Elemente de criptografie hardware: RNG hardware, accelerator AES. Secure boot: lanțul de încredere, verificarea semnăturii la pornire. TrustZone pe Cortex-M33: lumea securizată vs. nesecurizată.</p>	3	Prelegere interactivă	
<p>Tema 13: Depanarea și testarea sistemelor embedded. Tehnici de depanare hardware: osciloscop, analizor logic, multimetru. Depanare software: JTAG/SWD cu GDB, breakpoints hardware și software, watchpoints. Sisteme de logging și tracing: ITM (Instrumentation Trace Macrocell), SWO, Segger RTT. Testarea unitară pe hardware: frameworks (Unity, CMock).</p>	3	Prelegere interactivă	

Testarea de integrare: hardware-in-the-loop (HIL). Analiza consumului de putere cu probe dedicate.			
Tema 14: Proiectarea pentru producție și recapitulare. Trecerea de la prototip la produs: selecția componentelor (disponibilitate, preț, lifecycle), proiectarea PCB-ului pentru EMC, certificări (CE, FCC, RoHS). Documentația tehnică: schema bloc, specificații hardware/software, manualul de utilizare — conform standardelor. Managementul versiunilor firmware: semantic versioning, changelog. Recapitulare generală.	3	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> White, E., Making Embedded Systems: Design Patterns for Great Software, 2nd ed., O'Reilly Media, 2024. Patterson, D.A., Hennessy, J.L., Computer Organization and Design: RISC-V Edition, 2nd ed., Morgan Kaufmann, 2020. STMicroelectronics, Reference Manual RM0368 — STM32F401 advanced ARM-based 32-bit MCUs. Disponibil la: https://www.st.com/resource/en/reference_manual/rm0368-stm32f401xbc-and-stm32f401xde-advanced-armbased-32bit-mcus-stmicroelectronics.pdf <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Erika Erika Hagberg, James A. Langbridge, Elecia White, Phillip Johnston, Embedded Artistry. Embedded Systems Architecture: Explore architectural concepts, pragmatic design patterns, and best practices to produce robust embedded solutions, 2nd ed., Packt Publishing, 2023. FreeRTOS — documentație oficială și API reference: https://www.freertos.org/Documentation/RTOS_book.html <p>Resurse software:</p> <ol style="list-style-type: none"> STM32CubeIDE — mediu de dezvoltare gratuit STMicroelectronics: https://www.st.com/en/development-tools/stm32cubeide.html PlatformIO — mediu de dezvoltare cross-platform pentru sisteme embedded: https://platformio.org/ ARM Developer Documentation — documentație Cortex-M: https://developer.arm.com/documentation/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Primul program embedded — GPIO și întreruperi. Configurarea mediului STM32CubeIDE sau PlatformIO. Programul blink cu HAL și fără HAL (acces direct la registre). Configurarea unui buton cu întrerupere EXTI și debouncing software. Măsurarea timpului de răspuns la întrerupere cu osciloscopul.	2	Lucrare practică STM32	
Lucrarea L2: Timere și PWM. Configurarea unui timer pentru generarea PWM pe 3 canale. Controlul intensității unui LED RGB cu PWM. Varierea ciclică a intensității (efect breathing). Controlul unui servomotor (0°-180°) prin modificarea duty cycle-ului. Verificarea formelor de undă PWM cu osciloscopul.	2	Lucrare practică STM32	
Lucrarea L3: ADC, DAC și DMA. Citirea unui potențiomtru cu ADC în modul continuu + DMA circular buffer. Filtrarea digitală a valorilor ADC (medie mobilă pe 8 eșantioane). Generarea unui semnal sinusoidal cu DAC + DMA folosind un tabel de 256	2	Lucrare practică STM32	

valori. Vizualizarea semnalului generat cu osciloscopul.			
Lucrarea L4: Comunicații seriale — UART, SPI, I2C. Comunicația UART cu PC (terminal serial): trimiterea valorilor ADC în format text la 115200 baud. Citirea unui senzor de temperatură/umiditate prin I2C (ex: SHT31 sau BMP280). Comunicația SPI cu un display OLED SSD1306 sau cu un modul SD card.	2	Lucrare practică STM32	
Lucrarea L5: Moduri low-power și RTC. Implementarea unui sistem de monitorizare cu ciclu sleep: citire senzor → procesare → transmisie UART → intrare Stop mode. Configurarea RTC cu alarma pentru wake-up periodic (ex: la fiecare 10 secunde). Măsurarea consumului de curent în fiecare mod cu multimetrul de precizie. Compararea consumului: active vs. sleep vs. stop.	2	Lucrare practică STM32	
Lucrarea L6: FreeRTOS — tasks, queue și semafoare. Crearea a 3 task-uri cu priorități diferite: citire senzor (high), procesare date (medium), afișare UART (low). Comunicarea între task-uri prin queue. Sincronizarea accesului la resursa partajată (periferică I2C) cu mutex. Analiza timpilor de execuție cu GPIO toggling și osciloscop.	2	Lucrare practică FreeRTOS	
Lucrarea L7: Proiect integrat și colocviu. Implementarea unui sistem embedded complet la alegere: stație meteo cu afișaj și logging pe SD card, sistem de control al unui motor DC cu feedback de turație, dispozitiv IoT cu transmisie MQTT prin WiFi (ESP32) sau BLE. Prezentarea schemei bloc, a codului și a măsurătorilor. Colocviu oral.	2	Proiect individual, colocviu	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> White, E., Making Embedded Systems, 2nd ed., O'Reilly Media, 2024 — capitolele de arhitectură software și depanare. STMicroelectronics, documentație HAL și LL pentru STM32F4: https://www.st.com/en/embedded-software/stm32cube4.html FreeRTOS — documentație API și ghid de utilizare: https://www.freertos.org/Documentation/RTOS_book.html <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> Patterson, D.A., Hennessy, J.L., Computer Organization and Design: RISC-V Edition, 2nd ed., Morgan Kaufmann, 2020 — capitolele de interfețe hardware-software. PlatformIO Documentation — configurare proiecte embedded: https://docs.platformio.org/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> STM32CubeIDE — mediu de dezvoltare gratuit: https://www.st.com/en/development-tools/stm32cubeide.html PlatformIO — suport multi-framework pentru embedded: https://platformio.org/ Embedded Artistry — articole tehnice despre arhitectura software embedded: https://embeddedartistry.com/ 		

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

Proiectarea cu Microprocesoare transformă cunoștințele teoretice din Arhitectura Sistemelor de Calcul și Electronică Digitală în competența practică de a construi sisteme embedded funcționale — de la configurarea periferelor până la integrarea unui RTOS și implementarea comunicațiilor fără

fir. Această disciplină acoperă tocmai stratul în care hardware-ul și software-ul se întâlnesc efectiv, strat omniprezent în industria IoT, automotive, medical și industrială.

Structura laboratorului urmează o progresie deliberată — de la GPIO și întreruperi la DMA, RTOS și comunicații wireless — culminând cu un proiect integrat care reproduce un ciclu real de dezvoltare embedded. Colocviul oral din L7 verifică nu doar funcționalitatea sistemului, ci și capacitatea de a gestiona autonom situații tehnice imprevizibile și de a documenta și justifica deciziile tehnice luate — direct corelată cu R9 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 arhitecturii interne a microcontrollerelor ARM Cortex-M (GPIO, timere, ADC/DAC, DMA, interfețe seriale); înțelegerea principiilor RTOS (task, scheduler, semafoare, queue); cunoașterea protocoalelor de comunicație wireless (BLE, WiFi/MQTT, LoRa); cunoașterea mecanismelor de securitate și a procesului de actualizare OTA.	Examen scris (2 ore): 1 problemă de configurare a unui periferic STM32 (timer PWM sau ADC cu DMA — calculul registrelor PSC, ARR, CCR), 1 problemă de analiză a unui sistem RTOS (identificarea condițiilor de race condition sau deadlock), 1 problemă de selecție și justificare a unui protocol de comunicație pentru un scenariu dat și 1 problemă de calcul al consumului de energie	60%
8.5 Seminar/laborator	Funcționalitatea sistemelor implementate în laborator (circuitul lucrează conform specificației); calitatea codului C (structură, comentarii, gestionarea erorilor); calitatea măsurătorilor documentate (forme de undă osciloscop, timingul semnalelor analizate); calitatea proiectului integrat din L7 și justificările tehnice la colocviul oral.	Evaluare continuă: lucrări individuale predate pe Sakai după fiecare ședință — cod sursă + raport scurt cu măsurătorile efectuate (50%) și proiect integrat cu colocviu oral individual la L7 (50%)	40%
8.6 Standard minim de performanță:			
Demonstrarea funcționalității unui sistem embedded real la colocviul din L7 — sistemul trebuie să ruleze pe hardware fizic (nu în simulator), să utilizeze cel puțin două periferice diferite și să aibă un mecanism de gestionare a erorilor documentat. La examenul scris: calculul corect și complet al registrelor de configurare pentru problema de periferic (PSC, ARR, CCR pentru timer sau echivalentul pentru ADC) — aceasta verificând înțelegerea profundă a mapării software-hardware, nu memorarea API-ului HAL.			

Data completării 18.03.2026	Semnătura titularului de curs Conf. Dr. Răileanu Alina- Beatrice	Semnătura titularului de seminar / laborator Conf. Dr. Răileanu Alina- Beatrice
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei				PROIECTAREA CU MICROPROCESOARE– TIDS216			
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator / proiect				Prof. Dr. Mînză Viorel			
1.5 Anul de studiu	II	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					
Studiu individual					
Referate					
Teme casă					
Proiect					
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	Proiectarea cu Microprocesoare (TIDS215) — configurarea microcontrollerelor ARM Cortex-M, periferice (GPIO, timere, ADC/DAC, DMA, interfețe seriale), RTOS, comunicații fără fir; Arhitectura Sistemelor de Calcul (TIDS214) — organizarea procesorului, ierarhia memoriei, interfețe hardware-software.
3.2 de competențe	Capacitatea de a configura și programa un microcontroller STM32 sau echivalent (GPIO, timere, ADC, comunicații seriale); cunoașterea principiilor FreeRTOS (task, semafoare, queue); abilitatea de a scrie documentație tehnică clară (schemă bloc, specificații, cod comentat); familiarizarea cu instrumentele de depanare (osciloscop, analizor logic).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	
4.2 de desfășurare a seminarului/laboratorului	Activitatea de proiect se desfășoară individual, la calculatorul propriu sau în laboratorul de electronică embedded; STM32CubeIDE sau PlatformIO instalate;

	placa de dezvoltare STM32 sau echivalent disponibilă; întâlnirile de îndrumare cu titularul au loc săptămânal, față în față sau online, cu prezentarea stadiului și primirea feedback-ului; predarea finală 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	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).
5.2 Aptitudini	A9. Proiectează și configurează sisteme de calcul bazate pe microprocesoare și procesoare digitale de semnal (DSP), integrând componente hardware și software pentru realizarea de sisteme embedded funcționale.
5.3 Responsabilitate și autonomie	R9. Gestionează autonom situații tehnice complexe și imprevizibile — defecțiuni critice de sistem, vulnerabilități de securitate, degradări de performanță — adoptând noi abordări strategice și luând decizii rapide și documentate.

6. Conținuturi

6.1 Proiect	Nr. ore	Metode de predare	Observații
P1: Alegerea temei și planificarea. Prezentarea categoriilor de proiecte disponibile: sisteme de monitorizare, sisteme de control, dispozitive IoT, instrumente de măsurare. Alegerea temei individuale, definirea cerințelor hardware și software. Stabilirea planului de lucru: etape, livrabile, criterii de evaluare.	1	Îndrumare individuală	
P2: Specificarea sistemului. Redactarea specificației tehnice: diagrama de context, schema bloc, lista componentelor hardware, interfețele externe. Definirea metricilor de performanță (timp de răspuns, consum curent, precizie). Identificarea riscurilor tehnice și a soluțiilor alternative.	1	Îndrumare, feedback	
P3: Documentare și selecția componentelor. Studiul foilor de catalog pentru componentele alese (microcontroller, senzori, actuatoare). Verificarea compatibilității electrice: niveluri logice, curenți maximi, tensiuni de alimentare. Estimarea consumului total de curent.	1	Studiu individual, prezentare	
P4: Implementare hardware — prototip pe breadboard. Realizarea prototipului pe breadboard: conectarea microcontrollerului cu senzorii și actuatoarele. Verificarea conexiunilor cu multimetrul. Programarea unui firmware minimal de test (blink, UART hello world).	1	Implementare hardware	
P5: Implementare software — drivere și periferice. Implementarea driverelor pentru componentele hardware alese: citire senzor, comandă actuator, comunicație serială. Structurarea codului în module cu interfețe clare. Testarea fiecărui driver izolat.	1	Implementare C, review cod	
P6: Integrarea subsistemelor. Integrarea driverelor într-un sistem funcțional complet.	1	Implementare, depanare	

Testarea fluxului principal de date: senzor → procesare → actuator/afișare. Identificarea și rezolvarea problemelor de timing, overflow sau interferențe.			
P7: Prezentare intermediară. Prezentarea orală a stadiului: schema bloc implementată, codul sursă, demonstrație live a funcționalității de bază. Feedback din partea titularului și ajustarea planului pentru a doua jumătate a semestrului.	1	Prezentare, feedback	
P8: Implementare avansată — funcționalități suplimentare. Adăugarea funcționalităților avansate definite în specificație: comunicație wireless, logging pe SD card, interfață utilizator, mod low-power. Integrarea cu subsistemele existente.	1	Implementare C	
P9: Testare și validare. Testarea sistemului complet față de cerințele din specificație: verificarea fiecărei cerințe funcționale. Măsurarea parametrilor de performanță cu instrumente (osciloscop, multimetru). Documentarea rezultatelor.	1	Testare, măsurători	
P10: Robustețe și gestionarea erorilor. Identificarea și tratarea cazurilor de eroare: timeout la comunicație, valori senzor în afara domeniului, defecțiuni hardware simulabile. Implementarea watchdog-ului și a mecanismelor de recuperare. Testarea comportamentului la condiții limită.	1	Implementare, testare	
P11: Optimizarea consumului de energie. Profilarea consumului de curent în diferite stări de funcționare. Implementarea modurilor low-power (Sleep, Stop) cu wake-up prin eveniment. Compararea consumului înainte și după optimizare cu multimetrul de precizie.	1	Implementare, măsurători	
P12: Redactarea raportului tehnic. Structura raportului: introducere, specificații, arhitectura hardware/software, implementare, rezultate, concluzii. Inserarea schemelor bloc, a fragmentelor de cod relevante și a graficelor de performanță.	1	Redactare tehnică	
P13: Revizuirea raportului și pregătirea prezentării. Corectarea raportului pe baza feedback-ului titularului. Pregătirea prezentării finale: schema bloc, demonstrație live, rezultate măsurate, concluzii. Repetarea Q&A cu colegii.	1	Revizuire, prezentare	
P14: Prezentarea și evaluarea proiectului final. Prezentarea orală (10 minute + 5 minute Q&A): demonstrație live a sistemului funcțional pe hardware real, prezentarea arhitecturii și a deciziilor tehnice. Predarea raportului și a codului sursă pe Sakai.	1	Prezentare orală, evaluare	
	Bibliografie obligatorie proiect 1. White, E., Making Embedded Systems: Design Patterns for Great Software, 2nd ed., O'Reilly Media, 2024.		

	<p>2. STMicroelectronics, Reference Manual RM0368 — STM32F401 advanced ARM-based 32-bit MCUs. Disponibil la: https://www.st.com/resource/en/reference_manual/rm0368-stm32f401xbc-and-stm32f401xde-advanced-armbased-32bit-mcus-stmicroelectronics.pdf</p> <p>3. FreeRTOS — documentație oficială și API reference: https://www.freertos.org/Documentation/RTOS_book.html</p> <p>Bibliografie complementara proiect</p> <p>4. Patterson, D.A., Hennessy, J.L., Computer Organization and Design: RISC-V Edition, 2nd ed., Morgan Kaufmann, 2020 — capitolele de interfețe hardware-software.</p> <p>5. STM32CubeF4 HAL Driver Documentation — referință completă a driverelor HAL: https://www.st.com/en/embedded-software/stm32cubef4.html</p> <p>Resurse online gratuite recomandate</p> <p>6. STM32CubeIDE — mediu de dezvoltare gratuit STMicroelectronics: https://www.st.com/en/development-tools/stm32cubeide.html</p> <p>7. PlatformIO — mediu de dezvoltare cross-platform pentru sisteme embedded: https://platformio.org/</p> <p>8. Embedded Artistry — resurse tehnice pentru arhitectura software embedded: https://embeddedartistry.com/</p>
--	---

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

<p>Proiectul de Proiectare cu Microprocesoare integrează competențele din disciplina omonimă (TIDS215) într-un sistem embedded complet — de la specificarea cerințelor hardware/software până la testarea pe hardware real și prezentarea orală a deciziilor tehnice. Studentul parcurge în mod autonom întregul ciclu de dezvoltare embedded, reproducând contextul unui proiect real din industrie.</p> <p>Ședințele săptămânale de îndrumare formează capacitatea de a gestiona autonom situații tehnice imprevizibile — blocaje de depanare, incompatibilități hardware, timinguri critice — și de a lua decizii rapide și documentate, direct corelată cu R9 din matricea de corelare. Demonstrația live din P14 pe hardware fizic este criteriul central de evaluare.</p>

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	<p>Funcționalitatea sistemului embedded pe hardware real (sistem pornit, periferice operaționale, fluxul principal de date demonstrat live); calitatea arhitecturii software (modularitate, gestionarea erorilor, comentarii); calitatea documentației tehnice (schemă bloc, specificații, raport); calitatea prezentării orale și a răspunsurilor la întrebări.</p>	<p>Proiect individual evaluat la P14: demonstrație live pe hardware real (40%) + raport tehnic predat pe Sakai (40%) + prezentare orală cu Q&A de 15 minute (20%)</p>	100%

8.6 Standard minim de performanță:

<p>Demonstrarea funcționalității sistemului embedded pe hardware fizic real la P14 — sistemul trebuie să pornească autonom (fără intervenția debugger-ului), să integreze cel puțin doi senzori sau actuatori și să trateze cel puțin un caz de eroare documentat în raport. Un sistem care funcționează</p>
--

exclusiv în simulare sau care nu are mecanism de gestionare a erorilor nu îndeplinește standardul minim.

Data completării 19.03.2026	Semnătura titularului de curs	Semnătura titularului de seminar / laborator/proiect Prof. Dr. Mînză Viorel
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	LIMBA STRĂINĂ III – TIDC207						
1.2 Titularul activităților de curs	Conf. Dr. Cheșcă Alina-Beatrice						
1.3 Titularul activităților de seminar	Conf. Dr. Cheșcă Alina-Beatrice						
1.4 Titularul activităților de laborator							
1.5 Anul de studiu	II	1.6 Semestrul	2	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Ob DC

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	1/0
2.4 Total ore din planul de învățământ	28	din care: 2.5 curs	14	2.6 seminar/laborator	14/0
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					22
Documentare					5
Studiu individual					5
Referate					5
Teme casă					3
Proiect					2
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	Limba Străină I-III (TIDF107, TIDC115, TIDC207) — nivel C1 consolidat: comunicare tehnică avansată, prezentări pentru audiențe mixte, scriere academică, interviuri tehnice, comunicare în echipe multiculturale.
3.2 de competențe	Nivel C1 CEFR consolidat: capacitatea de a susține o prezentare tehnică complexă de cel puțin 10 minute cu Q&A, de a redacta un articol sau un raport tehnic la standard academic și de a comunica fluent și spontan în orice situație profesională; vocabularul tehnic IT acumulat în semestrul I-III.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector și acces la internet; platforma e-learning Danubius Online (Sakai); acces la articole tehnice IEEE Spectrum, ACM Communications, MIT Technology Review și la resurse Cambridge/Oxford nivel C1.
4.2 de desfășurare a seminarului/laboratorului	Sală de seminar cu acces la internet; activitățile includ dezbateri tehnice structurate, negocieri simulate, prezentări de cercetare, atelier de scriere a lucrării de licență în engleză și peer review reciproc; studenții sunt încurajați să utilizeze dispozitivele proprii.

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

5.1 Cunoștințe	Posedă cunoștințe de limbă engleză la nivel C1 (CEFR), cu vocabular tehnic specializat extins și nuanțat pentru toate domeniile programului TI: inginerie software, sisteme embedded, rețele, securitate, inteligență artificială; cunoaște convențiile stilistice ale genurilor de text academic și profesional de înalt nivel (teze, articole de conferință, rapoarte de cercetare, propuneri de proiect) și este familiarizat cu registrele specifice comunicării în medii academice și industriale internaționale.
5.2 Aptitudini	Comunică fluent și spontan în engleză în toate situațiile profesionale și academice, fără efort evident de formulare; elaborează texte tehnice complexe (articole, teze, propuneri de proiect, rapoarte de cercetare) cu structură clară, argumentare riguroasă și stil academic precis; susține prezentări tehnice de cercetare pentru audiențe specializate, gestionând eficient întrebările dificile și adaptând discursul în timp real la reacțiile audienței.
5.3 Responsabilitate și autonomie	Utilizează limba engleză cu deplină autonomie și responsabilitate profesională, asumându-și răspunderea pentru acuratețea și integritatea conținutului tehnic și academic produs; promovează o cultură a comunicării clare și etice în toate formele de interacțiune profesională — de la colaborarea în echipe de cercetare internaționale până la redactarea documentației tehnice publice; respectă cu consecvență normele de integritate academică în citare, parafrază și atribuire a ideilor.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Research presentations — structură și livrare. Structura unei prezentări de cercetare (conference paper presentation): motivation, related work, methodology, results, conclusions, Q&A. Diferențe față de prezentarea de produs: audiență specializată, nivelul de detaliu tehnic, citarea lucrărilor conexe. Tehnici de gestionare a întrebărilor dificile.	2	Prelegere interactivă, analiză video	
Tema 2: Academic writing — articolul tehnic complet. Structura IMRAD: Introduction, Methods, Results, Discussion. Scrierea introducerii: motivarea problemei, gap-ul din literatură, contribuția proprie. Secțiunea Related Work: sinteza critică a surselor, poziționarea față de literatura existentă. Limbajul hedging academic: limiting claims, expressing uncertainty.	2	Prelegere interactivă, analiză texte	
Tema 3: Grant proposals și propuneri de proiect. Structura unei propuneri de finanțare (research proposal, project proposal): problem statement, objectives, methodology, expected outcomes, budget justification, timeline. Limbajul persuasiv academic. Analiza unei propuneri reale de proiect Horizon Europe sau NSF.	2	Prelegere interactivă, analiză texte	
Tema 4: Negotiation și comunicare în proiecte complexe. Negocierea tehnico-comercială în engleză: cerințe, constrângeri, compromisuri,	2	Prelegere interactivă, studii de caz	

BATNA. Comunicarea în proiecte Agile internaționale: sprint reviews, retrospectives, stakeholder updates. Gestionarea conflictelor tehnice în echipe distribuite. Comunicarea veștilor proaste (bad news delivery) în context profesional.			
Tema 5: Technical documentation la nivel avansat. Documentația de arhitectură (Architecture Decision Records — ADR). API documentation: stilul tehnic precis, exemplele de cod, gestionarea versiunilor. Open source contribution: scrierea și revizuirea pull request-urilor, comentariile de cod în engleză. Documentarea proceselor de securitate și a post-mortem-urilor publice.	2	Prelegere interactivă, exemple autentice	
Tema 6: English for job seeking — nivel avansat. Scrisoarea de intenție (cover letter) pentru poziții senior în IT: personalizare, evidențierea contribuțiilor concrete. Pregătirea portofoliului tehnic public: GitHub README, articole pe blog tehnic, contribuții open source. Salary negotiation avansat: anchoring, counter-offers, benefits negotiation.	2	Prelegere, exerciții practice	
Tema 7: Writing the Bachelor's thesis in English. Structura lucrării de licență în engleză: abstract, introducere, capitole, concluzii, bibliografie. Stilul academic formal: eliminarea colocvialismelor, precizia terminologică, neutralitatea tonului. Citarea în stilul IEEE și APA: note de subsol, lista de referințe, evitarea plagiatului.	1	Prelegere interactivă, exerciții	
Tema 8: Sinteză și pregătire pentru colocviu. Recapitularea structurilor lingvistice și a vocabularului tehnic C1 acumulat în semestrul 4. Autoevaluare cu grila CEFR C1. Strategii pentru colocviu: prezentarea de cercetare și Q&A. Feedback final.	1	Recapitulare, autoevaluare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Swales, J.M., Feak, C.B., Academic Writing for Graduate Students, 3rd ed., University of Michigan Press, 2012. Glasman-Deal, H., Science Research Writing for Non-Native Speakers of English, Imperial College Press, 2010. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Wallwork, A., English for Writing Research Papers, 2nd ed., Springer, 2016. Alred, G.J., Brusaw, C.T., Oliu, W.E., Handbook of Technical Writing, 12th ed., Bedford / St. Martin's, 2020. Mascull, B., Business Vocabulary in Use: Advanced, 3rd ed., Cambridge University Press, 2017. <p>Resurse software:</p> <ol style="list-style-type: none"> IEEE Spectrum — articole tehnice autentice nivel C1: https://spectrum.ieee.org/ MIT Technology Review — reportaje de cercetare avansată în engleză: https://www.technologyreview.com/ ACM Digital Library — articole de conferință și jurnal pentru lectură academică: https://dl.acm.org/ 		
6.2 Seminar	Nr. ore	Metode de predare	Observații
Tema S1: Conference presentation — rondul I. Fiecare student susține o prezentare de cercetare de 5 minute pe o temă IT proprie, în	2	Prezentări individuale, peer feedback	

format conferință. Feedback structurat: structura argumentului, claritatea contribuției proprii, gestionarea Q&A. Compararea cu prezentările de tip LS3.			
Tema S2: Writing workshop — Introduction și Related Work. Atelier: redactarea unei secțiuni Introduction de 200 de cuvinte pentru o temă de cercetare dată. Corectarea și discutarea erorilor tipice la nivel C1: imprecizie argumentativă, lipsa poziționării față de literatura existentă, tonul incorect. Peer review în perechi.	2	Atelier scriere, peer review	
Tema S3: Grant proposal workshop. Redactarea unui problem statement de 150 de cuvinte pentru o propunere de proiect fictivă în domeniul IT. Analiza unui fragment dintr-o propunere Horizon Europe reală: identificarea structurii și a tehnicilor persuasive. Reformularea unui obiectiv vag în obiectiv SMART.	2	Atelier scriere, analiză text	
Tema S4: Negotiation simulation. Simularea unei negocieri tehnico-comerciale între două echipe: o echipă de dezvoltatori și un client cu cerințe contradictorii. Feedback pe vocabularul de negociere, asertivitate, tehnici de compromis. Debrief colectiv.	2	Joc de rol, debrief	
Tema S5: Advanced mock interview — senior positions. Simularea unui interviu pentru o poziție de senior software engineer sau tech lead: system design questions (design a URL shortener, design a chat application), leadership questions, salary negotiation. Feedback individualizat.	2	Simulare interviu, feedback individual	
Tema S6: Bachelor's thesis writing workshop. Atelier: redactarea abstractului lucrării de licență (200 de cuvinte: context, problemă, metodă, rezultate, concluzie). Corectarea reciprocă cu grila de criterii IEEE. Discutarea citărilor și a secțiunii de bibliografie.	2	Atelier scriere, peer review	
Tema S7: Open source contribution simulation. Redactarea unui pull request description complet în engleză pentru o modificare de cod fictivă: what, why, how, testing done. Scrierea unui issue report detaliat. Revizuirea unui PR al colegului cu comentarii constructive în engleză.	1	Exerciții practice, peer review	
Tema S8: Prezentări finale și evaluare portofoliu. Fiecare student susține prezentarea finală de portofoliu: 7 minute pe o temă de cercetare aleasă, cu Q&A de 3 minute. Evaluare pe grilă CEFR C1 cu criterii explicite. Autoevaluare finală.	1	Prezentări individuale, evaluare finală	

	<p>Bibliografie obligatorie seminar</p> <ol style="list-style-type: none"> Swales, J.M., Feak, C.B., Academic Writing for Graduate Students, 3rd ed., University of Michigan Press, 2012 — exercițiile de Introduction și Related Work. Glasman-Deal, H., Science Research Writing for Non-Native Speakers of English, Imperial College Press, 2010 — exercițiile de scriere academică. <p>Bibliografie complementara seminar</p> <ol style="list-style-type: none"> Wallwork, A., English for Writing Research Papers, 2nd ed., Springer, 2016. Alred, G.J., Brusaw, C.T., Oliu, W.E., Handbook of Technical Writing, 12th ed., Bedford / St. Martin's, 2020. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> TED Talks (Technology) — prezentări autentice C1 cu subtitrare: https://www.ted.com/topics/technology ACM Digital Library — articole academice tehnice pentru lectură și analiză: https://dl.acm.org/ Grammarly Blog — ghiduri de scriere academică și profesională avansată: https://www.grammarly.com/blog/ IEEE Author Center — ghiduri oficiale pentru scrierea articolelor tehnice: https://ieeauthorcenter.ieee.org/
--	--

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

<p>Limba Străină IV marchează finalizarea traseului de patru semestre de formare lingvistică și atingerea nivelului C1 CEFR — nivelul solicitat pentru publicarea în reviste internaționale, participarea la conferințe de specialitate și integrarea în echipe de cercetare sau de dezvoltare globale. Conținuturile din semestrul 4 — scrierea articolului academic, propunerea de proiect, negocierea tehnică și prezentarea de cercetare — sunt competențe solicitate direct în activitatea profesională a unui inginer TI senior sau cercetător.</p> <p>Atelierul de redactare a lucrării de licență în engleză (Temele 7 și S6) are o valoare practică imediată: studentul poate folosi direct produsele acestor ateliere în elaborarea propriei lucrări de absolvire. Prezentarea finală de portofoliu din S8, extinsă la 7 minute față de 5 minute în semestrul anterior, certifică progresul pe tot parcursul celor patru semestre și capacitatea de comunicare autonomă la nivel C1.</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 structurii și a convențiilor genurilor academice avansate (articol tehnic IMRAD, grant proposal, ADR, abstract de lucrare de licență); însușirea vocabularului tehnic IT la nivel C1; cunoașterea tehnicilor de negociere tehnică și de prezentare de cercetare.	Colocviu scris (1 oră): redactarea unui abstract academic de 200 de cuvinte pentru o temă de cercetare IT dată și răspunsul la 10 întrebări de vocabular tehnic avansat și structuri argumentative din tematica cursului	40%
8.5 Seminar/laborator	Calitatea participării active la seminar; progresul demonstrat față de semestrul III (S1 vs. S8); calitatea temelor de scriere academică predate (Introduction, abstract, grant proposal fragment, pull request	Evaluare continuă: participare activă și teme scrise de seminar (50%) și prezentare finală de portofoliu la S8 — 7 minute pe o temă de cercetare aleasă, cu Q&A de 3 minute, evaluată pe grilă CEFR C1 (50%)	60%

	description); calitatea prezentării finale de portofoliu din S8.		
8.6 Standard minim de performanță:			
Susținerea prezentării finale de portofoliu la S8 cu un discurs fluent și structurat de minimum 7 minute în engleză pe o temă de cercetare IT, demonstrând capacitatea de a poziționa contribuția proprie față de literatura existentă (cel puțin o referință citată și comentată) ȘI obținerea unui punctaj de minimum 50% la colocviul scris. Față de semestrul III, prezentarea din S8 impune obligatoriu o secțiune de Related Work cu cel puțin două lucrări citate și comentate. Predarea a cel puțin 5 din 8 teme de seminar scrise constituie condiție obligatorie pentru accesul la colocviu.			

Data completării	Semnătura titularului de curs Conf. Dr. Cheșcă Alina-Beatrice	Semnătura titularului de seminar / laborator Conf. Dr. Cheșcă Alina-Beatrice
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei		PRACTICĂ 1– TIDS218					
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator / proiect / practică		Lect. Dr. Ion Adelina					
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					5
Studiu individual					
Referate					
Teme casă					
Proiect					8
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	Toate disciplinele din An I și An II, Semestrul 1: Programare calculator I și II, Structuri de Date și Algoritmi, Programare Orientată pe Obiecte, Proiectare Logică, Electronică Digitală, Arhitectura Sistemelor de Calcul, Matematici Discrete, Metode Numerice.
3.2 de competențe	Capacitatea de a implementa programe funcționale în cel puțin un limbaj de programare (C, C++, Java sau Python); cunoașterea principiilor de bază ale proiectării software (modularitate, interfețe, documentare); abilitatea de a lucra cu instrumente standard de dezvoltare (IDE, version control Git, linie de comandă); cunoașterea noțiunilor elementare de rețele și sisteme de operare.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	
4.2 de desfășurare a seminarului/laboratorului	Stagiul de practică se desfășoară la o companie parteneră sau în laboratoarele universității, sub îndrumarea unui tutore desemnat de companie și a

	coordonatorului de practică din partea universității. Studentul este integrat în echipa de lucru și participă la activitățile reale ale organizației gazdă. Comunicarea cu coordonatorul universitar se face săptămânal, față în față sau online.
--	---

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	R2. Gestionează independent ciclul complet de dezvoltare al unui produs software — de la analiza cerințelor până la testare și livrare — demonstrând capacitatea de a anticipa riscurile tehnice și de a propune soluții alternative viabile.

6. Conținuturi

6.1 Practică	Nr. ore	Metode de predare	Observații
Săpt. 1-2: Integrarea în organizație. Familiarizarea cu organizația gazdă: structura echipelor, procesele interne, instrumentele de lucru (IDE, version control, ticketing system, CI/CD). Configurarea mediului de lucru. Înțelegerea proiectului curent al echipei și a rolului studentului în cadrul acestuia.	12	Activitate practică la locul de practică	
Săpt. 3-4: Participarea la activitățile echipei. Contribuția la activitățile curente ale echipei: implementarea unor funcționalități de complexitate redusă, rezolvarea unor bug-uri, scrierea de teste, revizuirea documentației. Participarea la ședințele de lucru (daily standup, sprint planning sau echivalent).	12	Activitate practică la locul de practică	
Săpt. 5-6: Sarcini de complexitate medie. Implementarea unor sarcini tehnice de complexitate medie, stabilite împreună cu tutorele: noi funcționalități, refactorizare, integrare cu servicii externe. Respectarea standardelor de calitate ale echipei (code review, teste unitare, documentare).	12	Activitate practică la locul de practică	
Săpt. 7-8: Contribuție independentă. Asumarea și finalizarea autonomă a cel puțin unei sarcini tehnice complete — de la analiză și proiectare până la implementare, testare și livrare. Documentarea soluției implementate în conformitate cu standardele echipei.	12	Activitate practică la locul de practică	
Săpt. 9-10: Finalizare și raportare. Finalizarea sarcinilor asumate. Redactarea raportului de practică: descrierea organizației, a activităților desfășurate, a competențelor formate și a contribuțiilor tehnice concrete. Pregătirea prezentării pentru colocviul de practică.	12	Activitate practică, redactare raport	

	<p>Bibliografie obligatorie practică</p> <ol style="list-style-type: none"> 1. Spolsky, J., Smart and Gets Things Done, Apress, 2007 — introducere în mediul de lucru din industria software. 2. Lopp, M., Managing Humans: Biting and Humorous Tales of a Software Engineering Manager, 3rd ed., Apress, 2016. 3. Convenția-cadru de practică între universitate și organizația gazdă — documentul semnat de toate părțile înainte de începerea stagiului <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 4. Atlassian — ghiduri practice pentru lucrul în echipe software (Agile, Scrum, Jira): https://www.atlassian.com/agile 5. Git Documentation — referință completă pentru utilizarea version control: https://git-scm.com/doc 6. IEEE Code of Ethics — codul de etică profesională în inginerie: https://www.ieee.org/about/corporate/governance/p7-8.html
--	---

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

Practica I reprezintă primul contact direct al studentului cu mediul profesional real din industria IT. Cele 60 de ore de stagiul la o companie parteneră sau în laboratoarele universității transformă competențele tehnice acumulate în primii doi ani de studiu într-o experiență de lucru concretă, cu cerințe reale, termene reale și standarde de calitate profesionale.

Integrarea în echipa organizației gazdă formează competențe transversale esențiale care nu pot fi dobândite exclusiv în mediul academic: comunicarea în contexte profesionale, înțelegerea proceselor de dezvoltare software în producție, gestionarea autonomă a unui ciclu complet de sarcină tehnică și asumarea responsabilității față de calitatea livrabilelor — direct corelate cu R2 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 participarea activă la toate cele 60 de ore de stagiul (confirmată prin foaia de prezență semnată de tutorele din companie); calitatea și completitudinea raportului de practică (descrierea organizației, activităților, competențelor formate, contribuțiilor tehnice concrete); calitatea prezentării orale la colocviu și capacitatea de a descrie și justifica deciziile tehnice luate în stagiul.	Raport de practică predat pe Sakai cu cel puțin 3 zile înainte de colocviu (50%) și colocviu oral individual — prezentarea activității din stagiul, cu Q&A de 10 minute, în fața coordonatorului de practică (50%)	100%

8.6 Standard minim de performanță:

Efectuarea integrală a celor 60 de ore de stagiul (confirmate prin foaia de prezență semnată de tutorele organizației gazdă) ȘI predarea raportului de practică cu descrierea a cel puțin unei contribuții tehnice concrete realizate în stagiul — cu specificarea sarcinii, a soluției implementate și a rezultatului obținut. Un raport care descrie doar organizația fără a documenta activitatea tehnică proprie nu îndeplinește standardul minim.

Data completării 19.03.2026	Semnătura titularului de curs	Semnătura titularului de seminar / laborator/proiect/practică Lect. Dr. Ion Adelina
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei		EDUCAȚIE FIZICĂ ȘI SPORT I V– TIDC221					
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Prof. Dr. Ploesteanu Constantin					
1.5 Anul de studiu	II	1.6 Semestrul	2	1.7 Tipul de evaluare	A/R	1.8 Regimul disciplinei	FAC

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	0/1
2.4 Total ore din planul de învățământ	14	din care: 2.5 curs	-	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					
Practică					
Elaborare proiect diplomă					
2.8 Activități individuale					11
Documentare					3
Studiu individual					3
Referate					
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					5
2.9 Total ore pe semestru					25
2.10 Numărul de credite					1

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Educație Fizică și Sport I-III (TIDF108, TIDC116, TIDC208) — deprinderi motrice consolidate la nivel avansat; cunoașterea principiilor fiziologice ale antrenamentului; condiție fizică evaluată și îmbunătățită pe parcursul primilor doi ani de studiu.
3.2 de competențe	Participare regulată la activitățile din semestrul I-III; nivel de condiție fizică consolidat; cunoașterea și aplicarea metodelor de antrenament funcțional (HIIT, circuit training) introduse în semestrul III; absența contraindicațiilor medicale pentru efort fizic.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	
4.2 de desfășurare a seminarului/laboratorului	Sală de sport sau teren de sport al universității; echipament sportiv adecvat (tricou, pantalon scurt, încălțăminte sport); materiale sportive specifice ramurilor practicate; prezența în echipament complet este obligatorie.

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

5.1 Cunoștințe	Cunoaște la nivel expert principiile planificării unui program personal de antrenament pe termen lung: periodizare anuală, ciclizarea sarcinii de antrenament, metodele de refacere și prevenire a supraantrenamentului; posedă cunoștințe avansate de nutriție sportivă, psihologie sportivă (motivație, gestionarea stresului competițional) și de prim ajutor în activitățile sportive.
5.2 Aptitudini	Execută cu măiestrie tehnică structuri motrice complexe specifice ramurilor sportive practicate pe parcursul celor patru semestre; elaborează și implementează independent un program personal de antrenament săptămânal adaptat obiectivelor proprii de sănătate și fitness; asumă roluri de lider în activitățile sportive de echipă, organizând și arbitrând competiții inter-grupe cu regulament complet.
5.3 Responsabilitate și autonomie	Promovează activ în cadrul comunității studențești un stil de viață sănătos bazat pe activitate fizică regulată, alimentație echilibrată și odihnă suficientă; își asumă responsabilitatea pentru menținerea pe termen lung a nivelului de condiție fizică dobândit pe parcursul programului de studii, planificând autonom activitatea fizică după finalizarea cursurilor universitare; acționează ca model de comportament fair-play și sportivitate față de colegii mai tineri.

6. Conținuturi

6.1 Laborator	Nr. ore	Metode de predare	Observații
Șed. 1-2: Evaluare inițială și planificare personală. Testare complexă finală de program: testul Cooper, flotări, abdomene, săritură în lungime de pe loc, flexibilitate. Compararea cu rezultatele din semestrul I. Elaborarea planului personal de activitate fizică pentru anul următor.	2	Testare practică, planificare individuală	
Șed. 3-4: Antrenament funcțional avansat. Circuit training de intensitate ridicată cu exerciții compuse: thruster, clean and press cu kettlebell sau halteră ușoară, box jump, pull-up asistat, Turkish get-up. Executarea unui protocol de 4 runde complete cu monitorizarea frecvenței cardiace.	2	Activitate practică intensivă	
Șed. 5-6: Sportul preferat — nivel avansat. Continuarea sportului ales de studenți prin vot (baschet, volei, futsal sau alt sport disponibil) la nivel avansat: tactică de echipă complexă, sistem de joc, arbitraj reciproc, managementul situațiilor de criză în joc (time-out, schimbări). Turneu scurt inter-echipe.	2	Joc bilateral avansat, arbitraj	
Șed. 7-8: Mindfulness și refacere activă avansată. Tehnici avansate de yoga terapeutică pentru sportivi: posturi de inversie (Shoulderstand, Headstand asistat), pranayama avansată (Nadi Shodhana, Kapalabhati). Stretching fascial: tehnica foam roller pentru principalele lanțuri musculare. Crioterapie și termoterapie: principii și aplicare.	2	Activitate practică, recuperare	
Șed. 9-10: Sporturi de agilitate și coordonare. Antrenament de agilitate cu scheme de mișcare complexe: ladder drill, cone drill, pro-	2	Activitate practică	

agility shuttle. Sporturi de rachetă la nivel introductiv: badminton sau squash — tehnica de bază, serviciu, forehand, backhand, mișcare pe teren. Mini-turneu cu tablou simplu.			
Șed. 11-12: Autoorganizarea activității sportive. Studenții organizează complet o mini-competiție sportivă: stabilesc regulamentul adaptat, formează echipele, asigură arbitrajul și evidența rezultatelor. Debrief colectiv: ce a funcționat, ce ar fi putut fi îmbunătățit. Accent pe leadership și comunicare sportivă.	2	Organizare autonomă, leadership	
Șed. 13-14: Evaluare finală și planul de viitor. Retestare cu bateriile de teste din ședințele 1-2. Calculul progresului față de semestrul I (4 semestre de EFS). Prezentarea planului personal de activitate fizică elaborat în ședințele 1-2. Feedback individual și recomandări pentru menținerea stilului de viață activ după absolvire.	2	Testare finală, prezentare plan, feedback	
<p>Bibliografie obligatorie laborator</p> <p>1. Bumpa, T.O., Buzzichelli, C., Periodization: Theory and Methodology of Training, 6th ed., Human Kinetics, 2019.</p> <p>2. Drăgan, I. (coord.), Medicina sportivă, Ed. Medicală, București, 2002.</p> <p>3. Regulamentele oficiale FIBA (baschet), FIVB (volei) și FIFA/Futsal — versiunile în vigoare: https://www.fiba.basketball/ / https://www.fivb.com/ / https://www.fifa.com</p> <p>Bibliografie complementara laborator</p> <p>4. McArdle, W.D., Katch, F.I., Katch, V.L., Exercise Physiology: Nutrition, Energy, and Human Performance, 9th ed., Wolters Kluwer, 2022.</p> <p>5. Ploșteanu, C., Educație fizică și sport — îndrumar metodic, Ed. Universitară Danubius, Galați.</p> <p>Resurse online gratuite recomandate</p> <p>6. American College of Sports Medicine (ACSM) — ghiduri de activitate fizică și condiție fizică: https://www.acsm.org/</p> <p>7. National Strength and Conditioning Association (NSCA) — resurse de antrenament funcțional: https://www.nsc.com/</p> <p>8. Darebee — programe gratuite de antrenament funcțional și calisthenics: https://darebee.com/</p>			

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

Educația Fizică și Sport IV marchează finalizarea celor patru semestre de formare fizică din program. Obiectivul final nu mai este dobândirea unor deprinderi sportive, ci consolidarea unui stil de viață activ sustenabil pe termen lung — singura garanție a menținerii sănătății fizice și mentale în contextul profesional sedentar specific ingineriei IT.

Compararea rezultatelor testelor finale cu cele din semestrul I oferă fiecărui student o imagine obiectivă a progresului realizat în doi ani de activitate fizică regulată, iar elaborarea planului personal de activitate fizică pentru perioada de după absolvire transformă disciplina dintr-o obligație curriculară într-un proiect de viitor asumat.

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	Prezența și participarea activă la minimum 10 din 14 ședințe (75%); progresul demonstrat la	Evaluare continuă: prezență (40%) + progres la testul de condiție fizică din ședința 13-14 față de ședința 1-2 (30%) +	100%

	testul final față de testul inițial din ședințele 1-2; calitatea planului personal de activitate fizică prezentat în ședința 14; comportamentul fair-play și asumarea rolului de organizator/arbitru în ședințele 11-12; prezența în echipament sportiv complet.	calitatea planului personal de activitate fizică prezentat (20%) + comportament și implicare (10%). Nota finală: Admis / Respins	
--	--	--	--

8.6 Standard minim de performanță:

Prezența la minimum 10 din 14 ședințe ȘI predarea în ședința 14 a unui plan personal de activitate fizică pentru 12 luni post-absolvire — cu minim 3 activități săptămânale specificate, obiective SMART și metode de monitorizare a progresului. Un plan generic fără specificitate (ex. 'voi face sport de 3 ori pe săptămână') nu îndeplinește standardul minim. Față de semestrul III, criteriul de evaluare nu mai este progresul față de testul inițial, ci dovada unui proiect personal de activitate fizică asumat și planificat.

Data completării 17.03.2026	Semnătura titularului de curs	Semnătura titularului de seminar / laborator Prof. Dr. Ploesteanu Constantin
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei		DECIZIE ȘI ESTIMARE ÎN TEORIA INFORMAȚIEI – TIDS222					
1.2 Titularul activităților de curs		Prof. Dr. Ariton Viorel					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Prof. Dr. Ariton Viorel					
1.5 Anul de studiu	II	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	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					13
Referate					7
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	Teoria Probabilităților și Statistică Matematică (TIDF201) — distribuții de probabilitate, estimare statistică clasică, teste de ipoteză; Teoria Transmisiunii Informației (TIDS212) — entropie, capacitate canal, modulații digitale, zgomot gaussian; Metode Numerice (TIDS211) — optimizare numerică, simulare Monte Carlo.
3.2 de competențe	Cunoașterea distribuțiilor de probabilitate uzuale (Gaussian, Poisson, Bernoulli, Rayleigh) și a proprietăților lor; înțelegerea conceptelor de estimator nedepășat, consistent și eficient; familiarizarea cu modelul canalului AWGN și cu probabilitatea de eroare la modulații digitale; abilitatea de a implementa algoritmi statistici în Python.

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 pentru distribuirea materialelor; recomandare: Python 3.10+ cu NumPy, SciPy și Matplotlib instalate înainte de curs.
-------------------------------	---

4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; Python 3.10+ cu NumPy, SciPy, Matplotlib și scikit-learn; studenții implementează estimatorii și detectoarele din curs și le verifică numeric prin simulare Monte Carlo; acces la MATLAB sau Octave ca alternativă.
--	---

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

5.1 Cunoștințe	C41. Deține cunoștințe specializate de estimare și decizie în teoria informației — criterii Bayes, estimatori de maximum likelihood, teoria detecției — cu aplicabilitate în proiectarea sistemelor de comunicații și de recunoaștere a formelor.
5.2 Aptitudini	A36. Aplică metode de estimare statistică și decizie bayesiană în analiza datelor experimentale și în proiectarea sistemelor de detecție, evaluând performanța estimatorilor și selectând criteriile optime de decizie.
5.3 Responsabilitate și autonomie	R36. Își asumă responsabilitatea pentru aplicarea riguroasă a metodelor statistice de estimare și decizie în analiza datelor experimentale, documentând transparent ipotezele statistice și comunicând onest limitele concluziilor formulate.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Estimarea statistică — cadru general. Problema estimării: parametru necunoscut, model statistic, observație. Calitățile unui estimator: nedepășare (unbiasedness), consistență, eficiență. Varianța unui estimator și eroarea medie pătratică (MSE). Estimatorul suficient: teorema factorizării Fisher-Neyman. Familia exponențială de distribuții.	2	Prelegere interactivă	
Tema 2: Limita Cramér-Rao (CRLB). Matricea informației Fisher: definiție pentru parametru scalar și vectorial. Inegalitatea Cramér-Rao: $MSE \geq 1/I(\theta)$. Estimatorul eficient: atingerea CRLB. CRLB pentru distribuția gaussiană: estimarea mediei și a varianței. CRLB asimptotică. Aplicații: estimarea frecvenței unui semnal, estimarea SNR.	2	Prelegere, demonstrații matematice	
Tema 3: Estimatorul MVUE și statistici complete. Estimatorul de varianță minimă nedepășat (MVUE): teorema Rao-Blackwell. Statistici complete: familia exponențială. Teorema Lehmann-Scheffé: construirea MVUE din statistici complete suficiente. Exemple: MVUE pentru media și varianța distribuției gaussiene, pentru parametrul Poisson.	2	Prelegere, demonstrații	
Tema 4: Estimatorul de maximum likelihood (MLE). Principiul maximum likelihood: funcția de verosimilitate, log-verosimilitate. Ecuațiile de scor (score equations). Proprietăți ale MLE: consistență, asimptotică normalitate, asimptotică eficiență. MLE pentru distribuții uzuale: gaussiană, exponențială, Poisson.	2	Prelegere interactivă	

MLE numeric: algoritmul Newton-Raphson, EM.			
Tema 5: Estimarea bayesiană. Probabilitatea a priori și a posteriori: formula Bayes. Funcția de pierdere (loss function): pierdere pătratică, absolută, 0-1. Estimatorul Bayes optim: estimatorul a posteriori mediu (MMSE) și MAP (Maximum A Posteriori). Estimatorul Bayes pentru distribuția gaussiană: formula de estimare lineară bayesiană. Comparatie MLE vs. MAP vs. MMSE.	2	Prelegere, comparații	
Tema 6: Estimatorul liniar optim — filtrul Wiener. Estimarea liniară: estimatorul BLUE (Best Linear Unbiased Estimator). Filtrul Wiener discret: derivarea ecuațiilor Wiener-Hopf, soluția în domeniul frecvență. Filtrul Wiener cauzal și non-cauzal. Aplicații: filtrarea zgomotului, predicția liniară, deconvoluția. Limita filtrului Wiener și motivația filtrului Kalman.	2	Prelegere interactivă	
Tema 7: Filtrul Kalman. Modelul de sistem dinamic stochastic: ecuația de stare și ecuația de observație (model liniar gaussian). Filtrul Kalman recursiv: predicția stării și actualizarea după observație. Câștigul Kalman și matricea de covarianță a erorii. Filtrul Kalman staționar. Extensii: filtrul Kalman extins (EKF) pentru sisteme neliniare. Aplicații: tracking, navigație.	2	Prelegere interactivă	
Tema 8: Teoria detecției — cadru general. Problema detecției: ipoteze H_0 și H_1 , regula de decizie. Probabilitatea de detecție PD și probabilitatea de alarmă falsă PFA. Curba ROC (Receiver Operating Characteristic). Criteriul Neyman-Pearson: maximizarea PD pentru PFA fixat — lema Neyman-Pearson. Testul raportului de verosimilitate (LRT).	2	Prelegere, demonstrații	
Tema 9: Detectorul optim în canal AWGN. Semnalul în zgomot alb gaussian aditiv (AWGN): modelul matematic. Detectorul optim: regula deciziei ML pentru semnale cunoscute. Probabilitatea de eroare pentru BPSK și QPSK: funcția Q. Detectorul de energie (power detector). Detecția unui semnal cu parametri necunoscuți: testul GLRT (Generalized LRT).	2	Prelegere, calcule numerice	
Tema 10: Detecția semnalelor cu parametri necunoscuți. Testul raportului de verosimilitate generalizat (GLRT): principiu și proprietăți. Detectorul CFAR (Constant False Alarm Rate): variante OS-CFAR, CA-CFAR. Detecția cu parametri necunoscuți: frecvență și fază necunoscute. Aplicații: detecția radarului, sonarului, detecția anomaliilor în rețele.	2	Prelegere interactivă	

<p>Tema 11: Clasificare și recunoașterea formelor. Clasificatorul Bayes optim: regula deciziei MAP pentru clasificare multclasă. Frontiera de decizie pentru distribuții gaussiene: clasificatorul LDA (Linear Discriminant Analysis) și QDA. Probabilitatea de eroare la clasificare: limita Bayes. Clasificatorul k-NN: distanța Mahalanobis. Evaluarea clasificatorilor: matricea de confuzie, ROC, AUC.</p>	2	Prelegere interactivă	
<p>Tema 12: Estimarea spectrală. Spectrul de putere și densitatea spectrală de putere (PSD). Estimatorul periodogram (Schuster): proprietăți, biasul și varianța. Periodogramul Welch: suprapunere și ferestre de ponderare. Metode parametrice de estimare spectrală: modelul AR (autoregresiv), metoda Yule-Walker. Estimarea frecvenței unui semnal sinusoidal: MUSIC, ESPRIT.</p>	2	Prelegere interactivă	
<p>Tema 13: Testarea ipotezelor statistice. Testul Student (t-test): pentru medie, pentru diferența a două medii. Testul chi-pătrat: pentru varianță, pentru bonitate (goodness-of-fit). Testul ANOVA (analiza varianței). Valoarea p și nivelul de semnificație. Erori de tip I și tip II: puterea testului. Testele neparametrice: Wilcoxon, Mann-Whitney. Corecția Bonferroni pentru teste multiple.</p>	2	Prelegere interactivă	
<p>Tema 14: Aplicații în sisteme de comunicații și AI — recapitulare. Estimatorul ML pentru sincronizare în receptoare digitale: estimarea întârzierii de timp (timing recovery), a frecvenței (carrier recovery). Detecția anomaliilor cu metode statistice: Isolation Forest, detecția bazată pe distribuție. Estimarea bayesiană în rețele neuronale (variational inference). Recapitulare generală.</p>	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Kay, S.M., Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory, Prentice-Hall, 1993. 2. Kay, S.M., Fundamentals of Statistical Signal Processing, Vol. II: Detection Theory, Prentice-Hall, 1998. 3. Cover, T.M., Thomas, J.A., Elements of Information Theory, 2nd ed., Wiley-Interscience, 2006. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Bishop, C.M., Pattern Recognition and Machine Learning, Springer, 2006. Disponibil gratuit la: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf 5. Poor, H.V., An Introduction to Signal Detection and Estimation, 2nd ed., Springer, 1994. <p>Resurse software:</p> <ol style="list-style-type: none"> 6. SciPy Documentation — statistică, estimare și testare de ipoteză în Python: https://docs.scipy.org/doc/scipy/reference/stats.html 7. MIT OpenCourseWare 6.437 — Inference and Information, note de curs: https://ocw.mit.edu/courses/6-437-inference-and-information-spring-2003/ 8. Bishop — Pattern Recognition and Machine Learning (PDF gratuit): https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf 		
6.2 Laborator	Nr. ore	Metode de predare	Observații

<p>Lucrarea L1: CRLB și eficiența estimatorilor. Calculul numeric al matricei informației Fisher pentru distribuții uzuale (gaussiană, Poisson, exponențială). Verificarea prin simulare Monte Carlo că MLE atinge CRLB asimptotic: generarea de $N=1000$ eșantioane, calculul varianței MLE și compararea cu $1/I(\theta)$. Trasarea curbei varianță vs. N.</p>	2	Lucrare practică Python/SciPy	
<p>Lucrarea L2: Maximum Likelihood și estimarea bayesiană. Implementarea MLE pentru frecvența unui semnal sinusoidal în zgomot gaussian: funcția de verosimilitate, maximizarea cu <code>scipy.optimize</code>. Estimarea bayesiană MAP cu prior gaussian pentru același parametru. Compararea MLE, MAP și MMSE ca funcție de SNR prin simulare Monte Carlo.</p>	2	Lucrare practică Python	
<p>Lucrarea L3: Filtrul Kalman. Implementarea filtrului Kalman pentru urmărirea unui obiect în mișcare uniformă: modelul de stare (poziție, viteză), modelul de observație cu zgomot gaussian. Simularea traiectoriei cu perturbații. Vizualizarea estimatei Kalman față de traiectoria reală. Analiza influenței parametrilor Q și R asupra performanței.</p>	2	Lucrare practică Python	
<p>Lucrarea L4: Curba ROC și criteriul Neyman-Pearson. Implementarea detectorului optim (LRT) pentru detectarea unui semnal BPSK în canal AWGN. Trasarea curbei ROC prin varierea pragului de decizie pentru SNR de -5, 0, 5 dB. Verificarea analitică a PD și PFA cu funcția Q. Compararea curba ROC simulată cu cea teoretică.</p>	2	Lucrare practică Python	
<p>Lucrarea L5: Detectorul GLRT și CFAR. Implementarea detectorului de energie (energy detector) pentru un semnal cu amplitudine necunoscută în canal AWGN. Implementarea CA-CFAR (Cell Averaging CFAR) pentru detecția unui semnal într-un fundal de zgomot neuniform. Verificarea proprietății CFAR: PFA constantă indiferent de puterea zgomotului.</p>	2	Lucrare practică Python	
<p>Lucrarea L6: Clasificatorul Bayes și LDA. Implementarea clasificatorului Bayes optim pentru două clase cu distribuții gaussiene bidimensionale. Trasarea frontierei de decizie. Compararea cu LDA (<code>sklearn.discriminant_analysis.LinearDiscriminantAnalysis</code>) și QDA pe date cu covariante egale și diferite. Calculul matricei de confuzie și al curbei ROC.</p>	2	Lucrare practică Python/scikit-learn	
<p>Lucrarea L7: Estimarea spectrală și proiect integrat. Implementarea periodogramului Welch cu ferestre Hanning pentru estimarea PSD a unui semnal de test (sinusoide + zgomot). Implementarea metodei Yule-</p>	2	Lucrare practică Python, colocviu	

Walker pentru estimarea spectrală AR. Proiect integrat: detectarea anomaliilor statistice într-un set de date de rețea (timp de răspuns, pachete pierdute) cu un detector bazat pe criteriul Bayes. Colocviu oral.			
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Kay, S.M., Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory, Prentice-Hall, 1993 — exercițiile practice. 2. Kay, S.M., Fundamentals of Statistical Signal Processing, Vol. II: Detection Theory, Prentice-Hall, 1998 — exercițiile practice. 3. SciPy Documentation — statistică și procesarea semnalelor în Python: https://docs.scipy.org/doc/scipy/reference/stats.html <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. Bishop, C.M., Pattern Recognition and Machine Learning, Springer, 2006 — capitolele de clasificare bayesiană. 5. scikit-learn Documentation — clasificare, regresie și evaluarea modelelor: https://scikit-learn.org/stable/documentation.html <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. SciPy — statistică și procesarea semnalelor: https://docs.scipy.org/doc/scipy/reference/stats.html 7. scikit-learn — algoritmi de Machine Learning în Python: https://scikit-learn.org/stable/ 8. Wolfram MathWorld — referință pentru distribuții de probabilitate: https://mathworld.wolfram.com/topics/ProbabilityDistributions.html 		

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

Decizia și Estimarea în Teoria Informației furnizează cadrul matematic riguros pentru extragerea informației din date zgomotoase — competența centrală a oricărui sistem inteligent: receptor de comunicații, detector radar, sistem de recunoaștere a vorbirii, algoritm de Machine Learning sau sistem de detecție a anomaliilor. Disciplina stabilește limitele teoretice fundamentale ale performanței — CRLB pentru estimare, limita Bayes pentru clasificare, curba ROC pentru detecție — față de care orice soluție practică trebuie evaluată.

Componenta de laborator, bazată pe simulare Monte Carlo în Python, urmărește formarea capacității de a evalua cantitativ performanța estimatorilor și a detectoarelor față de limitele teoretice și de a documenta transparent ipotezele statistice utilizate — direct corelate cu A36 și R36 din matricea de corelare. Proiectul integrat din L7, aplicat pe date reale de rețea, certifică transferul competenței din domeniul teoretic în practica ingineriei TI.

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 riguroasă a estimatorilor MLE, MAP și MMSE; cunoașterea CRLB și a condiției de eficiență; înțelegerea filtrului Kalman și a ecuațiilor Wiener-Hopf; cunoașterea și aplicarea criteriului Neyman-Pearson, a testului LRT și a curbei ROC; cunoașterea clasificatorului Bayes	Examen scris (2 ore): 1 problemă de calcul CRLB și verificarea eficienței unui estimator dat, 1 problemă de construire a testului LRT cu calculul PD și PFA și 1 problemă de estimare bayesiană sau filtru Kalman — cu toate calculele complete și justificarea ipotezelor statistice utilizate	60%

	optim și a testelor statistice clasice.		
8.5 Seminar/laborator	Corectitudinea implementărilor Python (simulări Monte Carlo cu număr de realizări suficient, grafice clare, compararea rezultatelor simulate cu valorile teoretice); calitatea proiectului integrat din L7 (detector funcțional pe date reale, documentarea ipotezelor, analiza limitelor); prezentarea și justificările la colocviul oral din L7.	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:			
Rezolvarea corectă cu calculele complete la examenul scris a cel puțin două probleme, cu obligativitatea includerii problemei de CRLB — aceasta verificând stăpânirea conceptului fundamental al disciplinei. La colocviul oral din L7: prezentarea curbei ROC simulate pentru detectorul implementat, cu explicarea orală a relației PD-PFA și a influenței SNR asupra performanței. Un detector care produce o curbă ROC sub diagonala aleatoare (PD mai mic decât PFA) sau fără varierea pragului nu îndeplinește standardul minim.			

Data completării	Semnătura titularului de curs Prof. Dr. Ariton Viorel	Semnătura titularului de seminar / laborator Prof. Dr. Ariton Viorel
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: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	DIDACTICA SPECIALIZĂRII – TIDFAC224						
1.2 Titularul activităților de curs	Conf. Dr. Beteringhe Adrian						
1.3 Titularul activităților de seminar	Conf. Dr. Beteringhe Adrian						
1.4 Titularul activităților de laborator							
1.5 Anul de studiu	II	1.6 Semestrul	2	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	FAC

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	2/0
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	28/0
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					69
Documentare					20
Studiu individual					20
Referate					20
Teme casă					5
Proiect					
Alte activități (inclusive colocviu, examen)					4
2.9 Total ore pe semestru					125
2.10 Numărul de credite					5

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Pedagogie I (TIDFAC122) și Pedagogie II (TIDFAC223) — proiectarea didactică, metode didactice, evaluare, managementul clasei; Psihologia Educației (TIDFAC121) — teorii ale învățării, motivație, diferențe individuale; toate disciplinele de specialitate din An I și II — cunoașterea conținuturilor de predat.
3.2 de competențe	Capacitatea de a proiecta o lecție completă cu obiective operaționale formulate corect; cunoașterea principalelor metode didactice și a criteriilor de selecție; familiarizarea cu documentele curriculare (programă școlară, manual) pentru disciplinele informatică sau matematică din liceu; experiența cel puțin a două sesiuni de microteaching din Pedagogie II.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector, tablă albă și acces la internet; platforma Sakai; studenții vor aduce la curs manualele și programele școlare de informatică și matematică pentru liceu pentru analiză și proiectare didactică aplicată.
4.2 de desfășurare a seminarului/laboratorului	Sală de seminar cu videoproiector și acces la internet; lecțiile simulate (microteaching) se susțin în fața grupei; studenții utilizează mijloace didactice

	reale (tablă, fișe de lucru, prezentări, software educațional) pentru a reproduce cât mai fidel contextul unei clase reale.
--	---

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

5.1 Cunoștințe	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.
5.3 Responsabilitate și autonomie	R36. Își asumă responsabilitatea pentru aplicarea riguroasă a metodelor statistice de estimare și decizie în analiza datelor experimentale, documentând transparent ipotezele statistice și comunicând onest limitele concluziilor formulate.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Didactica informaticii — specificul disciplinei. Particularitățile predării informaticii și matematicii față de alte discipline: gândire algoritmică, abstractizare, rezolvare de probleme. Programa școlară de informatică pentru liceu: structura, competențele generale și specifice, conținuturile pe clase. Manualele alternative: criterii de analiză și selecție. Locul informaticii în curriculumul național.	2	Prelegere interactivă, analiză documente	
Tema 2: Proiectarea unității de învățare. Structura unității de învățare pentru informatică: competențe specifice vizate, conținuturi, activități de învățare, resurse, evaluare. Corelarea cu calendarul și cu orarul. Exemple de unități de învățare pentru: algoritmi și structuri de date, programare orientată pe obiecte, baze de date, rețele. Greșeli frecvente în proiectare.	2	Prelegere, modele, analiză	
Tema 3: Predarea conceptelor algoritmice. Dificultăți specifice în predarea algoritmicii: recursivitate, complexitate, corectitudine. Strategii de predare: reprezentarea prin pseudocod, organigramă, limbaj natural. Trasarea execuției (trase) ca metodă de verificare. Alegerea limbajului de programare pentru predare: C++, Python, Java — avantaje și dezavantaje pedagogice. Progresivitatea introducerii conceptelor.	2	Prelegere interactivă, demonstrații	
Tema 4: Metode didactice specifice informaticii. Demonstrația cu calculatorul: planificare, execuție, gestionarea erorilor în timp real. Lucrarea practică la calculator: organizare, fișa de lucru, ajutorul diferențiat. Problematizarea în informatică: prezentarea unei probleme cu soluții multiple. Pair	2	Prelegere interactivă	

programming ca metodă didactică. Algoritmul de grup (jigsaw) pentru concepte complexe.			
Tema 5: Evaluarea în informatică. Specificul evaluării în informatică: testele grilă, problemele de algoritmi, proiectele software. Evaluarea automată: platforme online judge (pbinfo.ro, infoarena.ro, LeetCode), avantaje și limitări pedagogice. Rubricile pentru evaluarea proiectelor software. Evaluarea complet orală: prezentarea și apărarea unui proiect. Plagiatul în proiectele de informatică: detecție și prevenire.	2	Prelegere interactivă	
Tema 6: Proiectarea lecției de algoritmică. Lecția de predare a unui algoritm nou: structura, introducerea motivațională, prezentarea algoritmului, exemplificarea, exerciții ghidate, exerciții independente. Lecția de rezolvare de probleme: selectarea problemelor cu dificultate progresivă. Lecția de laborator: organizarea timpului, fișa de lucru, circulația profesorului în clasă. Lecția de recapitulare în informatică.	2	Prelegere, modele de proiecte	
Tema 7: Predarea programării orientate pe obiecte (OOP). Dificultăți specifice ale predării OOP: abstracția clasei, moștenirea, polimorfismul. Secvența didactică recomandată: clase simple, obiecte, moștenire, interfețe. Analogii și metafore pentru explicarea conceptelor OOP. Proiectele de dimensiune medie ca instrument de evaluare a competenței OOP. Erori tipice ale elevilor în OOP și strategii de remediere.	2	Prelegere interactivă, demonstrații	
Tema 8: Predarea bazelor de date și a rețelelor. Predarea modelului relațional: tabele, chei, relații — analogii cu situații reale. Predarea SQL: ordinea introducerii instrucțiunilor (SELECT simplu → JOIN → subinterogări). Proiectarea unei baze de date ca proiect de semestru. Predarea rețelelor de calculatoare: modele, protocoale, adresare. Simulatoare de rețele: Cisco Packet Tracer ca instrument didactic.	2	Prelegere interactivă	
Tema 9: Managementul clasei la informatică. Specificul managementului clasei de calculator: calculator ca distractor, verificarea lucrului independent vs. copiatul. Organizarea lucrului în echipă la proiecte software. Gestionarea diferențelor de nivel în clasă: elevi avansați și elevi cu dificultăți simultane. Motivarea elevilor pentru informatică: olimpiade, hackathoane, proiecte reale. Relația cu părinții în privința utilizării calculatorului.	2	Prelegere interactivă, studii de caz	
Tema 10: Utilizarea platformelor și a instrumentelor digitale. Platforme de e-learning pentru informatică: Google	2	Prelegere interactivă, demonstrații	

Classroom, Moodle cu plugin-uri de cod, Replit Teams. Sisteme online judge pentru teme automat evaluate: pbinfo.ro, infoarena.ro. Instrumente de colaborare la cod: GitHub Classroom. Prezentările interactive: Mentimeter, Kahoot pentru recapitulare. Videoclipurile didactice: screencasting eficient pentru demonstrații de cod.			
Tema 11: Diferențierea instruirii în informatică. Elevi supradotați în informatică: programe de îmbogățire, olimpiadele de informatică, proiectele avansate. Elevii cu dificultăți: decompoziția sarcinilor complexe, fișele de algoritm pas-cu-pas, programarea în perechi cu un elev mai avansat. Diferențierea prin produs: același obiectiv, dar livrabile diferite ca complexitate. Instruirea adaptivă cu platforme inteligente.	2	Prelegere interactivă	
Tema 12: Predarea matematicii — conexiuni cu informatica. Predarea matematicii discrete: logică, mulțimi, combinatorică, grafuri — elemente comune cu informatica. Interdisciplinaritatea informatică-matematică: proiecte comune. Predarea algoritmilor matematici: cel mai mare divizor comun, descompunerea în factori, numerele prime — secvența didactică. Calculul și estimarea — legătura cu analiza complexității.	2	Prelegere interactivă	
Tema 13: Cercetarea didactică aplicată la informatică. Cercetarea-acțiune în predarea informaticii: exemplu de studiu. Analiza erorilor tipice ale elevilor la algoritmică: metodologia de cercetare. Conceperea unui chestionar de feedback pentru o lecție de informatică. Utilizarea datelor de pe platformele online judge pentru analiza progresului. Publicarea rezultatelor: articolul de didactică a informaticii.	2	Prelegere interactivă	
Tema 14: Practica pedagogică — pregătire și recapitulare. Organizarea practicii pedagogice: cerințe, documente, tutorele din școală, raportul de practică. Asistența la lecțiile tutorelui: fișa de observare a lecției. Susținerea lecțiilor proprii: tipuri de lecții, cerințe de proiectare. Portofoliul de practică pedagogică: componente și criterii de evaluare. Recapitulare generală.	2	Prelegere, pregătire practică	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Cucuș, C., Pedagogie, ed. a III-a, Ed. Polirom, Iași, 2014. 2. Ionescu, M., Bocoș, M. (coord.), Tratat de didactică modernă, Ed. Paralela 45, Pitești, 2017. 3. Programa școlară de informatică pentru liceu, clasele IX-XII — Ministerul Educației, versiunea în vigoare. Disponibilă la: https://www.edu.ro/programe-scolare <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Winslow, L.E., Programming pedagogy — a psychological overview, SIGCSE Bulletin, 28(3), 1996 — specificul predării programării. 5. Gries, D., The Science of Programming, Springer, 1981 — metodologia predării algoritmicii formale. 		

	<p>6. Robins, A., Rountree, J., Rountree, N., Learning and Teaching Programming: A Review and Discussion, Computer Science Education, 13(2), 2003.</p> <p>Resurse software:</p> <p>7. pbinfo.ro — platformă națională de probleme de informatică pentru liceu: https://www.pbinfo.ro/</p> <p>8. Ministerul Educației — programe școlare și auxiliare curriculare: https://www.edu.ro/programe-scolare</p> <p>9. Cisco Packet Tracer — simulator de rețele pentru predare (gratuit cu cont NetAcad): https://www.netacad.com/courses/packet-tracer</p>		
6.2 Seminar	Nr. ore	Metode de predare	Observații
S1: Analiza programei de informatică. Analiza detaliată a programei școlare de informatică pentru o clasă la alegere (IX, X, XI sau XII): identificarea competențelor specifice, a conținuturilor, a sugestiilor metodologice. Compararea cu o programă din alt sistem european.	2	Analiză documente, dezbateri	
S2: Proiectarea unității de învățare. Elaborarea completă a unei unități de învățare de 4-6 ore pentru o temă de informatică aleasă de student. Completarea tabelului unității: competențe specifice, activități de învățare, resurse, evaluare. Prezentare și feedback.	2	Atelier, peer review	
S3: Proiectul de lecție — algoritmică. Elaborarea unui proiect de lecție pentru predarea unui algoritm (ex: sortare prin selecție, BFS, recursivitate simplă). Proiectarea secvenței: motivare, prezentare algoritm, trasare execuție, exerciții progresive. Feedback de grup.	2	Atelier, peer review	
S4: Microteaching I — demonstrație cu calculatorul. Fiecare student susține o demonstrație de cod de 7 minute: explică un algoritm sau un concept de programare la calculator, în fața grupei. Feedback structurat pe grilă: claritate, gestionarea erorilor, ritm, interacțiunea cu clasa.	2	Microteaching, feedback pe grilă	
S5: Proiectul de lecție — OOP sau baze de date. Elaborarea unui proiect de lecție pentru o temă OOP (ex: moștenire) sau baze de date (ex: instrucțiunea JOIN). Proiectarea fișei de lucru pentru elevi cu exerciții progresive. Feedback de grup.	2	Atelier, peer review	
S6: Evaluarea în informatică — instrumente. Elaborarea unui test de evaluare sumativă de 50 de minute pentru o unitate de învățare: 2 itemi grilă, 1 problemă de algoritm cu subpuncte, 1 problemă de implementare. Elaborarea baremului detaliat. Feedback pe claritatea cerințelor.	2	Atelier, peer review	
S7: Platforme didactice pentru informatică. Sesiune practică: configurarea unui curs în Google Classroom sau Moodle cu o temă automată de cod (Replit sau similar). Analiza platformei pbinfo.ro ca instrument de evaluare continuă. Compararea cu infoarena.ro. Avantaje și limitări pedagogice ale online judge.	2	Atelier practic, analiză critică	

S8: Diferențierea instruirii — aplicație practică. Proiectarea aceleiași lecții (ex: recursivitate) la trei niveluri: bază, mediu, avansat. Elaborarea a trei fișe de lucru diferite. Prezentarea și compararea soluțiilor de diferențiere. Debrief: cum gestionezi simultan trei niveluri în aceeași clasă.	2	Atelier, dezbateri	
S9: Managementul clasei de calculator. Analiza unor scenarii de management al clasei specifice orelor de informatică: elev care lucrează la altceva, copiatul de pe calculator al colegului, elevul mult mai avansat care se plictisește. Soluții practice discutate și evaluate în grup.	2	Studii de caz, joc de rol	
S10: Fișa de observare a lecției. Completarea și analiza unei fișe de observare a lecției pe baza unei înregistrări video a unei lecții de informatică (disponibilă public). Identificarea punctelor forte și a aspectelor de îmbunătățit. Pregătirea pentru asistența la lecții în cadrul practicii pedagogice.	2	Analiză video, dezbateri	
S11: Microteaching II — lecție cu elevi activi. Fiecare student susține o lecție de 10 minute utilizând o metodă care implică activ elevii (rezolvare de problemă în perechi, joc de rol debugging, problematizare). Compararea cu Microteaching I. Feedback extins.	2	Microteaching, feedback comparativ	
S12: Portofoliul de practică pedagogică. Prezentarea cerințelor portofoliului de practică pedagogică: componente obligatorii, criterii de evaluare, termene. Elaborarea fișei de prezentare a școlii de practică. Simularea completării unui jurnal reflexiv de practică.	2	Pregătire practică, atelier	
S13: Interdisciplinaritate informatică-matematică. Proiectarea unui proiect interdisciplinar informatică-matematică: o problemă matematică rezolvată prin programare (ex: șirul Fibonacci, algoritmul lui Euclid, grafuri și teoria grafurilor). Elaborarea fișei proiectului și a criteriilor de evaluare.	2	Atelier, dezbateri	
S14: Prezentare finală — portofoliu didactic. Fiecare student prezintă portofoliul didactic al semestrului: unitate de învățare, proiecte de lecție, instrumente de evaluare, reflecție personală asupra microteaching-urilor. Evaluare pe grilă cu criterii explicite. Feedback final.	2	Prezentare portofoliu, evaluare finală	

Bibliografie obligatorie seminar

1. Cucos, C., Pedagogie, ed. a III-a, Ed. Polirom, Iași, 2014 — capitolele aplicative.
 2. Ionescu, M., Bocoș, M. (coord.), Tratat de didactică modernă, Ed. Paralela 45, Pitești, 2017 — proiectarea didactică.
 3. Programa școlară de informatică — Ministerul Educației: <https://www.edu.ro/programe-scolare>
- Bibliografie complementara seminar**
4. Robins, A., Rountree, J., Rountree, N., Learning and Teaching Programming: A Review and Discussion, Computer Science Education, 13(2), 2003.

	<p>5. Winslow, L.E., Programming pedagogy — a psychological overview, SIGCSE Bulletin, 28(3), 1996.</p> <p>Resurse online gratuite recomandate</p> <p>6. pbinfo.ro — platformă de probleme de informatică pentru liceu: https://www.pbinfo.ro/</p> <p>7. infoarena.ro — platformă de algoritmică și online judge: https://infoarena.ro/</p> <p>8. Ministerul Educației — programe școlare și resurse curriculare: https://www.edu.ro/programe-scolare</p> <p>9. CS Unplugged — activități de informatică fără calculator pentru introducerea conceptelor: https://csunplugged.org/</p>
--	---

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

Didactica Specializării aplică principiile generale din Pedagogie II la contextul specific al predării informaticii, matematicii și disciplinelor tehnice. Disciplina pregătește studentul pentru practica pedagogică și, ulterior, pentru eventualul rol de profesor de informatică sau de formator tehnic — competențe direct aplicabile și în activitățile de training și onboarding din industria software. Seminarul, centrat pe proiectare aplicată și microteaching pentru discipline tehnice, urmărește formarea progresivă a competenței de predare prin exercițiu direct, feedback structurat și portofoliu reflexiv — direct corelate cu A39 din matricea de corelare. Dimensiunea de management al clasei de calculator și de utilizare a platformelor online judge răspunde specificului predării informaticii în contextul digital actual.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea specificului didacticii informaticii: particularitățile predării algoritmicii, OOP, bazelor de date și rețelelor; cunoașterea structurii programei școlare și a documentelor de proiectare (unitate de învățare, proiect de lecție); cunoașterea metodelor de evaluare specifice informaticii (online judge, proiecte software, rubrici); cunoașterea platformelor didactice pentru informatică.	Examen scris (1,5 ore): 1 problemă de proiectare a unității de învățare (completarea tabelului pentru o temă dată), 1 problemă de elaborare a unui proiect de lecție (secvența didactică și instrumentul de evaluare) și 1 subiect de analiză a unei situații didactice specifice informaticii (ex: cum predai recursivitatea sau cum gestionezi o clasă cu niveluri diferite)	50%
8.5 Seminar/laborator	Calitatea portofoliului didactic elaborat pe semestru (unitate de învățare, minim două proiecte de lecție, instrument de evaluare); progresul demonstrat în cele două sesiuni de microteaching (S4 vs. S11); participarea activă la ateliere și dezbateri;	Evaluare continuă: portofoliu didactic predat pe Sakai la finalul semestrului (60%) și prezentare finală la S14 cu Q&A de 10 minute (40%)	50%

	calitatea prezentării finale la S14.		
8.6 Standard minim de performanță:			
Predarea unui portofoliu care conține obligatoriu o unitate de învățare completă pentru o temă de informatică sau matematică ȘI cel puțin un proiect de lecție cu secvența didactică detaliată și instrumentul de evaluare aferent. Spre deosebire de Pedagogie II, proiectele de lecție trebuie să fie specifice disciplinei de specialitate (informatică sau matematică), nu generice — un proiect de lecție fără conținut tehnic specific nu îndeplinește standardul minim.			

Data completării	Semnătura titularului de curs Conf. Dr. Beteringhe Adrian	Semnătura titularului de seminar / laborator Conf. Dr. Beteringhe Adrian
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		