

## FIȘA DISCIPLINEI

### 1. Date despre program

|                                        |                                        |
|----------------------------------------|----------------------------------------|
| 1.1. Instituția de învățământ superior | Universitatea „Tibiscus” din Timișoara |
| 1.2. Facultatea                        | Calculatoare și Informatică Aplicată   |
| 1.3. Departamentul                     | Informatică                            |
| 1.4. Domeniul de studii                | Informatică                            |
| 1.5. Ciclul de studii                  | Master                                 |
| 1.6. Programul de studii/Calificarea   | Administrarea Sistemelor Distribuite   |

### 2. Date despre disciplină

|                                       |                                                            |                |   |                        |   |                          |                               |
|---------------------------------------|------------------------------------------------------------|----------------|---|------------------------|---|--------------------------|-------------------------------|
| 2.1. Denumirea disciplinei            | <b>FUNDAMENTELE PROCESĂRII DISTRIBUITE (FPD) – MASD113</b> |                |   |                        |   |                          |                               |
| 2.2. Titularul activității de curs    | Conf.univ.dr. Alin Munteanu                                |                |   |                        |   |                          |                               |
| 2.3. Titularul activității de seminar | Conf.univ.dr. Alin Munteanu                                |                |   |                        |   |                          |                               |
| 2.4. Anul de studiu                   | 1                                                          | 2.5. Semestrul | 1 | 2.6. Tipul de evaluare | E | 2.7. Regimul disciplinei | DS=sinteză<br>DOB=obligatorie |

### 3. Timpul total estimat

|                                                                                  |     |                    |    |                        |            |
|----------------------------------------------------------------------------------|-----|--------------------|----|------------------------|------------|
| 3.1. Numărul de ore pe săptămână                                                 | 4   | din care 3.2. curs | 2  | 3.3. seminar/laborator | 2          |
| 3.4. Total ore din planul de învățământ                                          | 56  | din care 3.5. curs | 28 | 3.6. seminar/laborator | 28         |
| <b>Distribuția fondului de timp</b>                                              |     |                    |    |                        | <b>Ore</b> |
| Studiul după manual, suport de curs, bibliografie și notițe                      |     |                    |    |                        | 28         |
| Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate |     |                    |    |                        | 30         |
| Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri            |     |                    |    |                        | 28         |
| Tutoriat                                                                         |     |                    |    |                        | 4          |
| Examinări                                                                        |     |                    |    |                        | 4          |
| Alte activități                                                                  |     |                    |    |                        | -          |
| 3.7. Total ore studiu individual                                                 | 94  |                    |    |                        |            |
| 3.8. Total ore pe semestru                                                       | 150 |                    |    |                        |            |
| 3.9. Numărul de credite                                                          | 6   |                    |    |                        |            |

### 4. Precondiții (acolo unde este cazul)

|                    |                                                                                                                                                                                                                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4.1. de curriculum | Programare orientată pe obiect. Structuri de date și algoritmi. Sisteme de operare, Securitate informatică, Administrarea rețelelor locale, Arhitectura sistemelor de calcul. Baze de date                                                                                                          |
| 4.2. de competențe | Utilizarea bazelor teoretice ale informaticii și a rețelelor de calculatoare;<br>Utilizarea limbajelor de programare pentru dezvoltarea aplicațiilor software;<br>Înțelegerea mecanismelor de comunicare între procese și sisteme;<br>Configurarea și utilizarea mediilor de dezvoltare și testare. |

### 5. Condiții (acolo unde este cazul)

|                                     |                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5.1. de desfășurare a cursului      | Sală de curs dotată cu videoproiector, ecran de proiecție; Calculator/laptop pentru cadrul didactic; Acces la internet pentru demonstrații privind arhitecturi distribuite și tehnologii actuale; Acces la platformă e-learning pentru distribuirea materialelor de curs, teme și comunicare; Posibilitatea utilizării unor instrumente software pentru modelarea arhitecturilor distribuite |
| 5.2. de desfășurare a laboratorului | Laborator de informatică dotat cu calculatoare conectate în rețea și acces la internet; Medii de dezvoltare software instalate (ex. Java/Python/C++ sau echivalent); Medii de virtualizare sau containere pentru simularea sistemelor distribuite; Acces la platforme pentru dezvoltare și testare distribuită                                                                               |

## 6. Obiectivele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cunoștințe                    | La finalizarea disciplinei, studentul:<br>Cunoaște conceptele fundamentale ale procesării distribuite și arhitecturilor distribuite;<br>Înțelege modelele de comunicare între procese (mesagerie, RPC, REST);<br>Înțelege problemele specifice mediilor distribuite: consistență, toleranță la erori, latență, scalabilitate;<br>Cunoaște paradigmele moderne de procesare distribuită (cloud computing, microservicii, procesare paralelă). |
| Abilități                     | La finalizarea disciplinei, studentul este capabil să:<br>Proiecteze aplicații distribuite utilizând modele arhitecturale adecvate;<br>Implementeze mecanisme de comunicare între componente distribuite;<br>Integreze servicii și baze de date în arhitecturi distribuite.                                                                                                                                                                  |
| Responsabilitate și autonomie | La finalizarea disciplinei, studentul:<br>Demonstrează autonomie în analiza și rezolvarea problemelor specifice sistemelor distribuite;<br>Aplică bune practici privind securitatea, performanța și fiabilitatea sistemelor;<br>Respectă standardele profesionale și etice în dezvoltarea și administrarea sistemelor informatice distribuite.                                                                                               |

## 7. Conținuturi

| 7.1. Curs                                                                                   | Metode de predare                                             | Observații                                  |
|---------------------------------------------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------|
| 1. Introducere în procesarea distribuită. Definiții, caracteristici, avantaje și provocări  | Expunere interactivă, conversație euristică                   | Corelare cu evoluția sistemelor distribuite |
| 2. Modele arhitecturale distribuite: client-server, peer-to-peer, multi-tier, microservicii | Explicație, analiză comparativă, studiu de caz                | Exemple din aplicații reale                 |
| 3. Comunicare în sisteme distribuite: RPC, RMI, REST, mesagerie                             | Modelare, problematizare, exemplificare                       | Integrare servicii web                      |
| 4. Concurență și sincronizare în sisteme distribuite                                        | Expunere, modelare algoritmică                                | Excludere mutuală și coordonare             |
| 5. Consistența datelor și replicare în sisteme distribuite                                  | Explicație, analiză de scenarii                               | Modele de consistență (strong, eventual)    |
| 6. Toleranța la erori și fiabilitate                                                        | Studiu de caz, analiză arhitecturală                          | Mecanisme de redundanță                     |
| 7. Algoritmi fundamentali în sisteme distribuite (consens, alegerea liderului)              | Modelare formală, problem solving                             | Introducere în Paxos/Raft (conceptual)      |
| 8. Teorema CAP și compromisuri în proiectarea sistemelor distribuite                        | Dezbateri, analiză comparativă                                | Scalabilitate vs. consistență               |
| 9. Procesare paralelă și distribuire a sarcinilor                                           | Expunere, exemplificare                                       | Introducere în map-reduce (conceptual)      |
| 10. Cloud computing și virtualizare                                                         | Explicație, studiu de caz                                     | Modele IaaS, PaaS, SaaS                     |
| 11. Securitate și<br>12. Aspecte etice în procesarea distribuită                            | Conversație euristică, analiză de risc                        | Corelare cu protocoale de securitate        |
| 13. Tendințe actuale: containere, orchestrare, edge computing<br>14. Dezvoltare poirct      | Expunere interactivă, documentare ghidată<br>Discuții proiect | Corelare cu tehnologii moderne              |

### Bibliografie pentru curs

1. Distributed Systems: Principles and Paradigms – Andrew S. Tanenbaum, Maarten van Steen, Prentice Hall, 2007.
2. Designing Data-Intensive Applications – Martin Kleppmann, O'Reilly Media, 2017.
3. Distributed Systems: Concepts and Design – George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, Addison-Wesley, 2011.
4. Principles of Distributed Database Systems – M. Tamer Özsu, Patrick Valduriez, Springer, 2011.
5. Cloud Computing: Concepts, Technology & Architecture – Thomas Erl et al., Prentice Hall, 2013.
6. Distributed Algorithms – Nancy A. Lynch, Morgan Kaufmann.

| 8.2. Seminar/laborator                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Metode de predare/învățare                     | Observații                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|---------------------------------------------------------------|
| 1. Configurarea mediului de lucru distribuit (mașini virtuale / containere)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Demonstrație practică, exercițiu individual    | Introducere în virtualizare / Docker                          |
| 2. Implementarea unui model client-server simplu                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Exercițiu aplicativ, lucru în echipă           | Comunicare prin socket-uri                                    |
| 3. Comunicare prin servicii web (REST API)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Proiect aplicativ, modelare                    | Testare                                                       |
| 4. Implementarea mecanismelor de sincronizare între procese distribuite                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Problem solving, exercițiu practic             | Tratarea condițiilor de concurență                            |
| 5. Replicarea datelor și gestionarea consistenței                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Studiu de caz, simulare                        | Analiza conflictelor                                          |
| 6. Implementarea unui mecanism simplu de alegere a liderului                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Modelare algoritmică, exercițiu aplicativ      | Înțelegerea consensului distribuit                            |
| 7. Simularea toleranței la erori (căderea unui nod)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Simulare, analiză scenarii                     | Redundanță și failover                                        |
| 8. Analiza performanței unui sistem distribuit                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Experiment, măsurare și interpretare           | Latență și scalabilitate                                      |
| 9. Implementarea unei aplicații distribuite pe mai multe noduri                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Învățare prin proiect, lucru în echipă         | Integrare servicii                                            |
| 10. Utilizarea containerelor pentru distribuirea aplicației                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Demonstrație practică, proiect                 | Noțiuni de orchestrare                                        |
| 11. Integrarea unei baze de date într-un sistem distribuit                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Exercițiu aplicativ                            | Gestionarea conexiunilor multiple                             |
| 12. Proiect – Proiectarea unei aplicații distribuite complete                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Învățare prin proiect (Project-Based Learning) | Definirea arhitecturii                                        |
| 13. -14. Proiect – Implementare, testare, optimizare și prezentare finală                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Proiect aplicativ, prezentare și susținere     | Evaluare integrată (funcționalitate + scalabilitate + docum.) |
| <b>Bibliografie pentru laborator</b> <ol style="list-style-type: none"> <li>1. Docker Deep Dive – Nigel Poulton, Leanpub. 2023</li> <li>2. Kubernetes Up &amp; Running – Brendan Burns, Joe Beda, Kelsey Hightower, O'Reilly Media. 2017</li> <li>3. RESTful Web APIs – Leonard Richardson, Mike Amundsen, O'Reilly Media. 2013</li> <li>4. Site Reliability Engineering – Betsy Beyer et al., O'Reilly Media. 2016</li> <li>5. Microservices Patterns – Chris Richardson, Manning Publications. 2018</li> <li>6. Documentații oficiale: Docker Docs, Kubernetes Documentation, Git Documentation (resurse online actualizate permanent).</li> </ol> |                                                |                                                               |

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

Conținuturile disciplinei sunt elaborate în concordanță cu direcțiile actuale de dezvoltare din domeniul sistemelor distribuite, cloud computing, microserviciilor, procesării paralele și infrastructurilor scalabile. Tematica referitoare la modele arhitecturale distribuite, comunicare inter-proces, consistență a datelor, algoritmi de consens, toleranță la erori și scalabilitate reflectă standardele academice și direcțiile de cercetare promovate de comunitatea științifică în domeniul informaticii distribuite.

Prin abordarea teoretico-aplicativă și realizarea proiectului integrator, disciplina contribuie la formarea competențelor necesare pentru roluri precum: arhitect software, inginer sisteme distribuite, dezvoltator backend distribuit, inginer DevOps sau administrator infrastructuri cloud.

## 9. Evaluare

| Tip de activitate | Criterii de evaluare                                                                                                             | Metode de evaluare                                       | Pondere din nota finală |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|-------------------------|
| 9.1. Curs         | Cunoașterea conceptelor fundamentale; înțelegerea modelelor distribuite; capacitatea de analiză a arhitecturilor și algoritmilor | Examen scris (întrebări teoretice + probleme aplicative) | 40%                     |
| 9.2. Laborator    | Implementarea mecanismelor distribuite; capacitatea de integrare a componentelor; analiza performanței și scalabilității         | Evaluare continuă (teme, exerciții practice)             | 20%                     |

| Tip de activitate | Criterii de evaluare                                                                                                        | Metode de evaluare                  | Pondere din nota finală |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------|-------------------------------------|-------------------------|
|                   | Proiect final:<br>Proiectarea și implementarea unei aplicații distribuite complete; documentație; prezentare și argumentare | Proiect aplicativ + susținere orală | 40%                     |

### 9.3. Standard minim de performanță

Pentru promovarea disciplinei cu minim nota 5, studentul trebuie să demonstreze:

- Însușirea conceptelor fundamentale privind procesarea distribuită;
- Înțelegerea modelelor arhitecturale și a mecanismelor de comunicare;
- Implementarea unei aplicații distribuite simple, funcționale;
- Proiect final integrator - Corectitudinea arhitecturii. Modularitatea și claritatea codului. Calitatea documentației.
- Capacitatea de susținere și argumentare;
- Obținerea a minimum 50% din punctajul alocat.

Standard pentru nota 7–8 (nivel bun / foarte bun)

Studentul demonstrează:

- Înțelegerea clară a conceptelor fundamentale ale procesării distribuite;
- Capacitatea de analiză comparativă a modelelor arhitecturale;
- Rezolvarea corectă a problemelor aplicative cu grad mediu de dificultate;
- Obținerea unui punctaj între 70–85% la evaluare;
- Proiectarea unei arhitecturi distribuite coerente;
- Realizarea unei analize elementare de performanță;
- Documentație tehnică bine structurată;
- Proiect funcțional, stabil și argumentat corect.

Standard pentru nota 9–10 (nivel excelent / performanță avansată – master)

- Înțelegerea aprofundată a conceptelor și algoritmilor distribuiți;
- Explicarea mecanismelor de consens (conceptual: Paxos/Raft);
- Rezolvarea problemelor complexe și formularea de soluții optimizate;
- Obținerea unui punctaj peste 85% la evaluare;
- Proiectarea unei arhitecturi scalabile și bine fundamentate;
- Documentație tehnică riguroasă și structurată profesional;
- Capacitatea de argumentare critică și justificare tehnică a deciziilor arhitecturale.

**Data completării**

20.09.2025

**Titular de disciplină**

*Conf.univ.dr. Alin-Daniel MUNTEANU*

**Data avizării în departament**

26.09.2025

**Director de departament**

*Conf.univ.dr. Victoria IORDAN*