

UNIVERSITATEA TEHNICĂ DIN CLUJ NAPOCA
CENTRUL UNIVERSITAR NORD DIN BAIA MARE
FACULTATEA DE ȘTIINȚE
[INFORMATICĂ]/ [MATEMATICĂ- INFORMATICĂ]

LUCRARE DE LICENȚĂ / DISERTAȚIE

COORDONATOR ȘTIINȚIFIC:

ABSOLVENT:

2025

UNIVERSITATEA TEHNICĂ DIN CLUJ NAPOCA
CENTRUL UNIVERSITAR NORD DIN BAIA MARE

FACULTATEA DE ȘTIINȚE
[INFORMATICĂ]/ [MATEMA TICĂ- INFORMATICĂ]

LUCRARE DE LICENȚĂ /
DISERTAȚIE
[TITLU LUCRARE]

COORDONATOR ȘTIINȚIFIC:

ABSOLVENT:

2025

Recomandări privind formatul lucrării

Lucrarea trebuie să fie redactată folosind un editor specializat pentru lucrări științifice, cum ar fi LaTeX sau Microsoft Word (pentru ecuații, se poate folosi Microsoft Equation). Se recomandă să se folosească fontul Times New Roman, mărimea 12 pt, cu spațiere de 1.5 rânduri și margini de 2.5 cm pe toate laturile. Textul va fi justificat (cu excepția titlurilor capitolelor, care pot fi aliniate central, și a etichetelor din tabele și figuri).

Numerotarea secțiunilor și subsecțiunilor va urma o structură ierarhică: secțiune, subsecțiune (de exemplu, 3.1 pentru subsecțiunea 1 din secțiunea 3).

Paginile vor fi numerotate începând cu pagina de titlu, dar numărul paginii va apărea doar după secțiunea "Introducere", în subsolul paginii.

Definițiile, propozițiile matematice, ecuațiile, diagrama, figurile și graficele trebuie numerotate și etichetate corespunzător, pentru a fi ușor de referit. De exemplu, figura a doua din secțiunea 3 va fi numerotată „Figura 3.2”, iar ecuația a treia din secțiunea 4 va fi numerotată „Ecuația 4.3”.

Fiecare figură va avea un număr, un titlu și o scurtă descriere legată de conținutul acesteia.

La sfârșitul lucrării se pot adăuga Anexele (acestea nu se numără și nu contează la numărul de pagini al lucrării) vor include materiale suplimentare relevante, cum ar fi figuri, demonstrații detaliate, secvențe de cod, tabele, fișiere de configurare sau planurile de testare.

Lucrarea trebuie să aibă un MINIM de 40 de pagini.

Prezentarea lucrării de licență/disertație

Prezentarea este la fel de importantă ca redactarea acesteia. Înainte de prezentare, absolventul va fi anunțat despre data, ora și locația susținerii. Neprezentarea la timpul stabilit poate duce la eliminarea din examen. Prezentarea va dura maximum 20 de minute și va include slide-uri, prezentarea aplicației dezvoltate și a codului. Atât aplicația cât și codul scris trebuie să fie accesibile în cadrul prezentării. Membrii comisiei pot adresa întrebări legate de lucrare, metodologie și resurse folosite, respective întrebări specifice legate de cod.

Criterii de evaluare lucrare de licență/disertație

Comisia de licență/disertație va face evaluarea lucrării, aplicației și prezentării și răspunsului la întrebări și va acorda o **nota** conform următorului barem:

- punct oficiu - 1pct
- raport plagiat
 - 2pct dacă e <10%
 - 1pct dacă e $\geq$ 10% și mai mic decât 20%
 - 0pct dacă e mai mare decât 20%

NU SE ACCEPTA LUCRARI CARE AU RAPORTUL DE
PLAGIAT MAI MARE DE 40%.

- lucrare de licență corespunde cerințelor și e scrisă și redactată corect, are conținut conform unei lucrări de licență/disertație
Punctaj maxim acordat 2.5 pct
- aplicația - complexitate, bine realizată, intuitivă, efort depus, funcțională
Punctaj maxim acordat 3 pct

- raspuns intrebari: se pun 3 intrebari de catre comisie
Punctaj maxim acordat 1.5 pct
- pentru prezentarea lucrarii de licenta/disertatie in cadrul conferintelor studenetesti (de exemplu XGEN) cu publicarea de articole in revista studenteasca aferenta conferintei, se va acorda un punctaj bonus.
Punctaj bonus 1pct

Indrumatorul lucrarii de licenta/disertatie va acorda o nota studentului.

Nota finala se calculeaza ca fiind :

$$\text{Nota_finala} = 10\% * \text{Nota_indrumator} + 90\% * \text{Nota_comisie}$$

Model continut lucrarea de licenta/disertatie INFORMATICA

Cuprins

Capitolul 1. Introducere	8
---------------------------------------	----------

Capitolul 2. Tehnologii folosite	9
Capitolul 3. Considerații generale	9
Capitolul 4. Descrierea aplicației.....	4
Capitolul 5. Implementare, Testare și Validare	11
Capitolul 6. Concluzii.....	12
Bibliografie.....	12

Capitolul 1. Introducere

Capitolul de **Introducere** al unei lucrări de finalizare trebuie să ofere o imagine de ansamblu clară și concisă asupra temei abordate, stabilind cadrul general și direcția studiului. Acesta ar trebui să conțină următoarele elemente esențiale:

- **Motivarea alegerii temei:** Se explică relevanța problemei tratate și contextul în care aceasta apare. Se pot menționa provocările existente, nevoile din domeniu sau impactul pe care soluționarea problemei îl poate avea asupra teoriei sau practicii.
- **Gradul de noutate al temei:** Se evidențiază elementele inovatoare ale cercetării sau soluției propuse, subliniind diferențele față de abordările existente.
- **Obiectivele lucrării:** Se precizează scopul general al lucrării, precum și obiectivele specifice care au fost urmărite pe parcursul cercetării.
- **Relevanța rezultatelor:** Se subliniază contribuțiile aduse de autor și impactul acestora asupra domeniului ales, fie că este vorba despre un Andrei nou model teoretic, o metodă practică sau o aplicație specifică.
- **Metodologia utilizată:** Se oferă o descriere sumară a metodelor și tehnicilor aplicate pentru atingerea obiectivelor (ex.: cercetare teoretică, experimentare, analiză de date, simulări).
- **Descrierea soluției propuse:** Se prezintă pe scurt soluția elaborată, evidențiindu-se modul în care aceasta răspunde problemelor identificate.
- **Structura lucrării:** Se descrie organizarea lucrării, oferind o scurtă privire de ansamblu asupra capitolelor și a modului în care acestea sunt interconectate pentru a sprijini firul logic al cercetării.

Este esențial ca acest capitol să fie scris clar, concis și accesibil, astfel încât să se înțeleagă rapid scopul lucrării, contribuțiile și relevanța.

Capitolul 2. Tehnologii folosite

Capitolul dedicat tehnologiilor într-o lucrare de licență sau disertație în informatică ar trebui să ofere o prezentare clară și detaliată a instrumentelor, limbajelor de programare, bibliotecilor, cadrelor de lucru (frameworks) și altor tehnologii utilizate în realizarea proiectului.

Acesta trebuie să includă o descriere a fiecărei tehnologii, explicând de ce a fost aleasă și evidențiind avantajele sau caracteristicile sale relevante pentru implementarea soluției propuse. În acest context, este important să se sublinieze modul în care fiecare tehnologie sprijină atingerea obiectivelor lucrării.

De exemplu, dacă s-au utilizat baze de date, se poate descrie tipul bazei de date (relațională, NoSQL) și motivul pentru care a fost selectată.

Dacă proiectul implică dezvoltarea web, ar trebui să se menționeze tehnologiile front-end și back-end utilizate, precum și arhitectura aplicată.

Se adauga aici pe langa partea de teorie legata de tehnologia folosita si cateva secvente mai interesante de cod care sa exemplifice folosirea tehnologiei respective. Adaugati si structura bazei de date si descrierea tabelor pentru exemplificarea bazei de date folosite.

Capitolul 3. Considerații generale

Capitolul de **Considerații Generale** ar trebui să arate ce aplicații similare cu cea creată de student există deja și să explice ce fac acestea bine și ce le lipsește. Este important să se compare aplicațiile existente, să se arate ce funcționalități și tehnologii folosesc și să se identifice punctele lor slabe sau lucrurile care pot fi îmbunătățite.

După această analiză, studentul ar trebui să explice ce aduce aplicația lui în plus sau diferit față de celelalte. De exemplu, poate are funcționalități mai utile, este mai ușor de folosit, mai rapidă sau folosește tehnologii mai moderne. Acest capitol ajută se să înțeleagă de ce proiectul este important și ce contribuție nouă aduce în domeniu.

Capitolul 4. Descrierea aplicației

Capitolul de **Descrierea Aplicației** al unei lucrări de licență sau disertație trebuie să ofere o prezentare detaliată și bine structurată a aplicației dezvoltate. Acesta ar trebui să înceapă cu o scurtă introducere despre aplicație, menționând numele acesteia, scopul principal și domeniul în care este utilizată. Este util să fie incluse câteva fraze care să descrie pe scurt funcționalitățile generale și avantajele pe care le oferă utilizatorilor.

Capitolul poate continua cu diagrame UML pentru a oferi o imagine clară asupra arhitecturii și funcționării aplicației:

- **Diagrama Use Case** – pentru a arăta relațiile dintre actori (utilizatori) și funcționalități.
- **Diagrama de Activitate** – pentru a descrie fluxurile principale de procese din aplicație.
- **Diagrama de Stare** – pentru a detalia comportamentul aplicației în diferite scenarii.

După prezentarea diagramelor, se descriu în detaliu funcționalitățile aplicației, structurate pe tipuri de utilizatori (actori). Pentru fiecare actor (de exemplu, **Administrator**, **Utilizator final**, **Vizitator**), se explică ce funcționalități au fost dezvoltate. Descrierile ar trebui să fie clare, să includă exemple practice și să fie însoțite de screenshot-uri relevante din interfața aplicației pentru a susține explicațiile și vizual.

Acest capitol trebuie să fie bine organizat, folosind text, diagrame și imagini pentru a face aplicația ușor de înțeles.

Capitolul 5. Implementare, Testare și Validare

Capitolul Implementare, Testare și Validare ar trebui să ofere detalii despre modul în care a fost realizată dezvoltarea aplicației, precum și despre procesul de testare și validare pentru a asigura funcționarea corectă a acesteia. Acesta trebuie să includă următoarele elemente descrise mai jos.

Prezentarea generală a procesului de implementare: O descriere succintă a etapelor parcurse în dezvoltarea aplicației, explicând cum s-au implementat funcționalitățile principale și cum s-au organizat sarcinile. Se poate menționa dacă s-a utilizat o metodologie specifică, cum ar fi **Agile**, și se pot include screenshot-uri ale instrumentelor utilizate (ex. Jira, Trello).

Strategia de testare: Explicarea abordării utilizate pentru testarea aplicației, cu menționarea tipurilor de teste efectuate, cum ar fi:

- **Testarea funcțională:** Verificarea fiecărei funcționalități conform cerințelor definite.
- **Testarea pe mai multe browsere:** Parcurgerea funcționalităților pe diferite browsere pentru a detecta eventualele disfuncționalități.
- **Testarea validării datelor:** Introducerea de caractere speciale, litere în câmpuri numerice sau alte inputuri invalide pentru a verifica gestionarea erorilor.
- **Testarea proceselor specifice:** De exemplu, încercarea de a adăuga în coș produse care nu sunt în stoc sau testarea modificării stocurilor după ștergerea unei facturi.

Descrierea detaliată a cazurilor de testare: Prezentarea fiecărui planurilor de testare dezvoltate pentru a testa aplicația.

Rezultatele testării și validarea aplicației: Un rezumat al testelor efectuate, incluzând atât situațiile care au funcționat corect, cât și problemele descoperite și cum au fost rezolvate. Aici pot fi incluse și screenshot-uri relevante pentru a demonstra procesul de testare și rezultatele obținute.

Organizarea taskurilor și metodologia de lucru: Dacă s-a utilizat o metodologie precum **Agile**, se poate descrie pe scurt cum a fost aplicată (ex.: împărțirea proiectului în sprinturi, organizarea sarcinilor zilnice). Screenshot-uri ale unor tabele de organizare sau instrumente precum burndown charts pot sprijini această secțiune.

Concluzii privind validarea aplicației: Un scurt rezumat al testelor și al modului în care acestea au confirmat că aplicația este funcțională, fiabilă și pregătită pentru utilizare.

Capitolul 6. Concluzii

Capitolul de Concluzii al unei lucrări de licență sau disertație trebuie să ofere o încheiere clară și bine structurată a proiectului. Acesta ar trebui să includă un rezumat al contribuțiilor aduse, subliniind elementele-cheie ale soluției dezvoltate și modul în care acestea răspund obiectivelor lucrării.

Se poate face o analiză a rezultatelor obținute, evidențiind atât reușitele proiectului, cât și eventualele limitări sau dificultăți întâmpinate pe parcurs. Este important să fie incluse observații personale despre relevanța și impactul soluției în contextul domeniului abordat.

În final, capitolul ar trebui să prezinte posibile direcții viitoare de dezvoltare sau îmbunătățire a aplicației, sugerând cum aceasta ar putea fi extinsă, optimizată sau adaptată pentru noi utilizări sau cerințe. Concluziile trebuie să reflecte o înțelegere clară a rezultatelor și să evidențieze contribuția adusa.

Bibliografie

[1] P. Nume Autori, Titlul, Editura: , anul.

[2] P. Nume, „Titlul,” Available: <https://....> [Accesat zi/luna/ 22].

[3] P. Nume, Curs Pagini Web, Universtatea Tehnica Centrul Universitar
Nord Baia-Mare