SOFTWARE SECURITY

1. Course information

Faculty	Computers, Informatics, and Microelectronics				
Department	Informatics and Systems Engineering				
Study cycle	Cycle II, Master's degree studies				
General field of study	061 Information and Communication Technologies				
Master's study program	**Data Science**				
Year of study	**Semester**	**Evaluation type**	**Formative category**	**Optionality category**	**ECTS credits**
I (full-time education)	I	E	F – fundamental training course	O - obligatory course	5

2. Estimated total time

Total hours in the curriculum	Including				
	Auditory hours		Individual work		
	Lecture	Laboratory/seminar	Term paper	Study of theoretical material	Application development
150	20	20		50	60

3. Prerequisites for access to the course

According to the curriculum plan	Computer Programming, Procedural Programming, Special Mathematics, Data Structures and Algorithms, Object-Oriented Programming, Information Security Technologies, Advanced Programming Techniques
According to competencies	Knowledge and skills in designing and developing algorithms and programs in different programming languages to solve problems on computers, as well as knowledge and skills in designing and using information security technologies.

4. Conditions for conducting the educational process

Lecture	For presenting theoretical material in the classroom, a whiteboard, projector, and computer are required. Student delays and phone conversations during the course will not be tolerated.
Laboratory/seminar	Students will complete reports according to the conditions outlined in the methodological guidelines. The deadline for submitting the laboratory work is one week after its completion. Late submission will result in a penalty of 1 point per week of delay.

5. Specific competencies acquired

Professional competencies	**CPM1 Designing and System Architecture Development.** *Takes on a high level of responsibility in defining the strategy for implementing new technologies in accordance with the company's needs. Considers the current infrastructure, equipment wear, and new technological innovations.* *K1 Architecture models, methodologies, and system design tools* *K2 System architecture requirements: performance, maintainability, extensibility, scalability, availability, security, and accessibility* K3 The costs, benefits, and risks of a system architecture K4 Enterprise architecture and the company's internal standards K5 Emerging new technologies (e.g., distributed systems, virtualization models, data sets, mobile systems) S1 Provides expertise to help solve complex technical problems and ensures the implementation of the best architectural solutions

	S2 Applies his/her technological knowledge from different fields to develop and implement the enterprise architecture. S3 Understands the company's objectives that impact the architecture components (data, applications, security, development, etc.). S4 Assists in communicating the enterprise architecture, along with its standards, principles, and objectives, to the various involved teams. S5 Develops design models and architectural models to assist system analysts in designing coherent applications. **CPM2 Monitoring technological trends. Innovation. Sustainable development.** *Harnesses a wide range of specialized knowledge about new and emerging technologies to formulate future solutions for the enterprise. Provides expert advice to the management team when making strategic decisions. Applies independent thinking and technological knowledge to integrate disparate concepts into original solutions. Defines the objectives and strategy for the sustainable development of IS by the organization's sustainability policy.* K1 Existing and emerging technologies and their relevant applications in the market K2 Business, societal, and research objectives, trends, and needs K3 Relevant sources of information (e.g., journals, conferences and events, opinion leaders, online forums, etc.) K4 Practical approaches to applied research programs K5 Innovation process techniques K6 Criteria and indicators of sustainable development K7 Corporate Social Responsibility (CSR) of stakeholders within the information system infrastructure S1 Monitors information sources and continuously tracks the most promising ones S2 Identifies the vendors and suppliers of the most promising solutions; evaluates, justifies, and proposes the most suitable ones S3 Identifies the advantages and improvements brought by adopting emerging technologies S4 Thinks without preconceived ideas S5 Applies recommendations within projects that support the latest sustainable development strategies **CPM3 Development of applications. Integration of components. Systems engineering.** *Uses specialized knowledge on a large scale to create a process, including establishing internal standards and practices. Mobilizes teams and allocates resources for integration programs. Manages complexity by developing procedures and standard architectures to support the development of consistent products. Establishes and identifies a set of system requirements that must be assigned to the system components. Acts creatively to develop applications and select appropriate technical options. Participates in other development activities. Optimizes the development, maintenance, and performance of applications.* K1 Appropriate programs/modules, DBMS, and suitable programming languages. Cutting-edge technologies. K3 The impact of system integration on the organization or the existing system K4 Interfacing techniques between modules, systems, and components K5 Integration testing techniques K7 Hardware components, tools, and hardware architectures K8 Functional and technical design K9 Fundamentals of information security K10 Prototyping S1 Applies appropriate software and/or hardware architectures. S2 Measures system performance before, during, and after system integration. S3 Identifies and records activities, issues, and corrective actions related to maintenance. S4 Adapts customer needs to existing products. S5 Secures and backs up data to ensure its integrity during data or system integration.

	S6 Explains and communicates with the client regarding design/development. S7 Launches and evaluates test results according to product specifications. S8 Applies data models, processes, to develop efficiently and productively. **CPM5 Process Improvement** *Implements and authorizes innovations and improvements that will enhance competitiveness and/or efficiency. Demonstrates to leadership the benefits of possible changes for the enterprise.* K1. Research, comparison, and measurement methods K2 Evaluation, design, and implementation methods K3 Existing internal processes K4 Relevant developments/evolutions in the ICT field (e.g., virtualization, open data, etc.) and their potential impact on processes K5 Specificity of web, cloud, and mobile technologies S1 Drafts, documents, and catalogs essential processes and procedures S2 Proposes changes to processes to facilitate and streamline improvements

Transversal competencies	**CTM1 Autonomy and responsibility.** *Performs some complex professional tasks under conditions of autonomy and professional independence.*
	CTM2 Social interaction. *Assumes leadership roles in professional activities or organizational structures.*
	CTM3 Professional and personal development. *Exercises self-control over the learning process, anticipates training needs, and critically analyzes their own professional activity.*

6. Course objectives

General objective	Students' acquisition of the concepts, notions, and examples of secure software development. Familiarizing students with basic techniques specific to secure programming.
Specific objectives	Upon successful completion of this course, students will be able to: • Implement security-based programming techniques • Use security standards • Develop security-based applications

7. Course content

Syllabus of teaching activities	Number of hours	
	Full-time education	Part-time education
Course topics		
T1. **Introduction to Software Product Security.** Importance of security in software development. Types of threats and vulnerabilities.	2	
T2 **Basic Principles of Software Product Security.** Confidentiality, integrity, availability. The software development life cycle (SDLC) and security.	2	
T3. **Securing Requirements and Design.** Analysis and validation of security requirements. Secure design and architectural models.	2	
T4. **Secure Programming Techniques.** Programming practices to prevent common vulnerabilities. Concrete examples of programming languages.	2	
T5. **Security Auditing and Testing.** Methods for evaluating software security. Penetration testing and code review.	2	
T6. **Vulnerability Management.** Identifying, classifying, and addressing vulnerabilities. Patch management tools and techniques.	2	

T7. **Security in Cloud and Mobile Environments.** Specific challenges for cloud applications. Mobile application security.	2	
T8. **Software Lifecycle Security (DevSecOps).** Integrating security into DevOps processes. Automating secure testing and deployment.	2	
T9. **Legal and Ethical Aspects of Software Security.** Relevant legal regulations (GDPR, CCPA, etc.). Ethics in software development.	2	
T10. **Case Studies and Discussions.** Analyzing notable security incidents. Lessons learned and best practices.		
Total lectures:	20	
Topics for Practical and Laboratory Work:		
1. **Vulnerability Assessment.** Using scanning tools for vulnerability evaluation.	2	
2. **Writing Secure Code.** Programming exercises for identifying and correcting vulnerabilities.	2	
3. **Penetration Testing.** Simulating attacks and evaluating the security of applications.	2	
4. **Patch Management.** Implementing a patch management system.	2	
5. **Securing Web Applications.** Implementing protective measures for web applications.	2	
6. **Mobile Application Security.** Creating a prototype of a secure mobile application.	2	
7. **DevSecOps in Practice.** Setting up a DevOps environment that integrates security.	2	
8. **Security Incident Simulations.** Incident response and crisis management exercises.	2	
9. **Case Study Analysis.** Learning from others' mistakes through case studies.	2	
10. **Project and Work Presentations.** Presenting and analyzing the security projects completed during the course.	2	
Total practical and laboratory work:	20	

8. Using generative AI

Permission to use	The use of generative AI in assignments and projects is permitted, provided that students adhere to the following rules: • Generative AI may be used to generate ideas, text structures, or code, but all generated materials must be reviewed and adjusted by the student to ensure that they meet academic requirements. • Any use of generative AI must be declared in the appendix section of each paper, using the phrase: "During the preparation of this paper, the author used [NAME OF TOOL / SERVICE] for the purpose of [REASON]. After using this tool / service, the author reviewed and edited the content as necessary and assumes full responsibility for the content of the paper."
Restrictions to use	Students ***MUSTN'T consider generative AI as a reliable source of information***, as it does not provide clear references or documented sources. • ***Direct citation of AI-generated content*** in academic papers as if it were a primary source ***isn't permitted.*** • Activities in which the use of **generative AI is prohibited** are specified by the teacher and are usually ***intermediate and final assessments*** or that don't involve professional competence development activities.

9. Bibliographic references

Obligatory	1. ***Security-Driven Software Development: Learn to analyze and mitigate risks in your software projects***, by Aspen Olmsted (Author), Publisher: Packt Publishing - ebooks Account (March 15, 2024), ISBN-13 978-1835462836 2. ***Secure Coding in C and C++ (SEI Series in Software Engineering) 2nd Edition***, by Robert Seacord (Author), Addison-Wesley Professional (April 12, 2013), ISBN-13 978-0321822130 3. ***Secure Programming Cookbook for C and C++: Recipes for Cryptography, Authentication, Input Validation & More***, by John Viega, Matt Messier (Authors), Publisher: O'Reilly Media (July 24, 2003, ISBN-13 978-0596003944 (3 exemplare) 4. ***Application Security Program Guide: Building a Comprehensive Application and Product Security Program***, by Ahmed Abdul-Rahman (Author), (December 21, 2023), ISBN-13 979-8988840909
Supplementary	1. ***Software building security in, Gary McGraw (Author), Addison-Wesley, Boston, 2011, ISBN: 9780321356703, 0321356705***

10. Evaluation

Periodic		Current	Individual study	Project/thesis	Exam
Mid term 1	Mid term 2				
15%	15%	15%	15%	-	40%

Minimum performance standard:
Attendance and participation in lectures, practical lessons, and laboratory work.
Achieving a minimum grade of "5" in each evaluation and laboratory work.
Achieving a minimum grade of "5" in the exam.

11. Evaluation criteria

Activity	Evaluation components	Evaluation method, evaluation criteria	Weight in final grade for the Activity	Weight in course evaluation
Full-time education				
Mid term I	Theoretical content, topics 1-3	Test	100%	**15%**
Mid term II	Theoretical content, topics 4-5	Activities during practical work/seminar	100%	**15%**
Current evaluation	Practical activity	Attendance and participation in classes	50%	**15%**
Individual study	Classification of research by activity type	Presentation/Discussion on the topic	100%	**15%**
Final examination	Theoretical and practical content	Oral exam. Grading according to grading scale	100%	**40%**