COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

COURSE SYLLABUS 2025-2026

GENERAL INFORMATION

Course information	
Name	Digital Transformation Workshops
Code	DTC-MIINT-524
Main program	Máster en Industria Inteligente
Offered in	Máster Universitario en Ingeniería Industrial + Máster en Industria Inteligente [1st year] Máster en Industria Inteligente [1st year]
Level	Master's Degree
Semester	2nd (Spring)
Credits	1.5 ECTS
Type	Compulsory
Department	Computer Science and Artificial Intelligence
Coordinator	Marcos Ventosa Pontes

Instructor	
Name	Marcos Ventosa Pontes
Department	Computer Science and Artificial Intelligence
e-mail	mvpontes@comillas.edu
Office hours	Arrange an appointment through email.
Lab instructor	
Name	Francisco Barragán Castro
Department	Computer Science and Artificial Intelligence
e-mail	fbarragan@comillas.edu
Office hours	Arrange an appointment through email.

COURSE SPECIFIC INFORMATION

Contextualization of the course

Contribution to the professional profile of the degree

During the 4th Industrial revolution, companies who can quickly iterate and adapt to the changing market are able to lead innovation and maintain a competitive edge. Agile frameworks help breakdown projects into several dynamic phases where the priority is placed on continuously adapting to customer and market demand. DevOps methodologies provide us with a set of best practices to shorten the development lifecycle of high-quality software. These two frameworks are key in fostering innovation and nimbleness during the continuously changing times.

This course will introduce the two frameworks to help students understand them with a practical, workshop-based approach. Students will start learning the basics of version control systems to understand the key tool for collaborating in software projects. This will set the foundation to understand the DevOps framework and dive deep into CI/CD pipelines that help continuously deliver value. Students will also learn about common practices and software used for leading agile projects and understand the differences with traditional waterfall approaches.

Prerequisites

Students willing to take this course should be familiar with undergraduate-level programming and have previous experience with Python.

Competences[1] – Objectives

Competences

General

CG1. Have acquired advanced knowledge and demonstrated, in a research and technological or highly specialized context, a detailed and well-founded understanding of the theoretical and practical aspects, as well as of the work methodology in one or more fields of study.
Haber adquirido conocimientos avanzados y demostrado, en un contexto de investigación científica y tecnológica o altamente especializado, una comprensión detallada y fundamentada de los aspectos teóricos y prácticos y de la metodología de trabajo en uno o más campos de estudio.

CG2. Know how to apply and integrate their knowledge, understanding, scientific rationale, and problem-solving skills to new and imprecisely defined environments, including highly specialized multidisciplinary research and professional contexts.
Saber aplicar e integrar sus conocimientos, la comprensión de estos, su fundamentación científica y sus capacidades de resolución de problemas en entornos nuevos y definidos de forma imprecisa, incluyendo contextos de carácter multidisciplinar tanto investigadores como profesionales altamente especializados.

CG5. Be able to transmit in a clear and unambiguous manner, to specialist and non-specialist audiences, results from scientific and technological research or state-of-the-art innovation, as well as the most relevant foundations that support them.
Saber transmitir de un modo claro y sin ambigüedades, a un público especializado o no, resultados procedentes de la investigación científica y tecnológica o del ámbito de la innovación más avanzada, así como los fundamentos más relevantes sobre los que se sustentan.

CG6. Have developed sufficient autonomy to participate in research projects and scientific or technological collaborations within their thematic area, in interdisciplinary contexts and, where appropriate, with a high knowledge transfer component.
Haber desarrollado la autonomía suficiente para participar en proyectos de investigación y colaboraciones científicas o tecnológicas dentro de su ámbito temático, en contextos interdisciplinares y, en su caso, con una alta componente de transferencia del conocimiento.

CG7. Being able to take responsibility for their own professional development and their specialization in one or more fields of study.
Ser capaces de asumir la responsabilidad de su propio desarrollo profesional y de su especialización en uno o más campos de estudio.

Specific

CE4. Understand the role of DevOps and Agile frameworks in improving quality and reducing delivery times in projects.
Comprender el papel de los frameworks DevOps y Agile en la mejora de la calidad y la reducción de los plazos de entrega en los proyectos.

Learning outcomes

RA1. Understand the key differences between waterfall and agile methodologies for project management.

RA2. Navigate the Linux command line and develop Bash scripts to automate system tasks.

RA3. Effectively collaborate in software projects using Git.

RA4. Implement end-to-end CI/CD pipelines.

[1] Competences in English are a free translation of the official Spanish version.

CONTENTS

Contents
Workshops
Workshop 1. Agile methodologies (2 hours)
1.1 The Agile philosophy 1.2 Scrum vs. waterfall 1.3 Agile practices and tools
Workshop 2. Bash (2 hours)
2.1 Introduction to Bash 2.2 Lab: Tinkering with Bash
Workshop 3. Version control systems (4 hours)
3.1 Introduction to version control systems: Git 3.2 Overview of version control development platforms 3.3 Lab: Collaborating on a project
Workshop 4. DevOps methodologies (6 hours)
4.1 The DevOps philosophy 4.2 Understanding a basic CI/CD pipeline 4.3 CI/CD platforms 4.4 Lab: Creating a CI/CD pipeline

TEACHING METHODOLOGY

General methodological aspects

Inspired by the “learn by doing” paradigm, this course is designed to provide students with the basic understanding of tools they require to integrate in modern software teams. Every workshop will start with a theoretical introduction, and we will break out into groups for lab work.

In-class activities	Competences
▪ **Lectures:** The lecturer will introduce the fundamental concepts of each unit, along with some practical recommendations, and will go through worked examples to support the explanation. Active participation will be encouraged by raising open questions to foster discussion.	CG1, CG7, CE4
▪ **Lab sessions:** Under the instructor’s supervision, students, divided into small groups, will apply the concepts and techniques covered in the lectures to create simple CI/CD pipelines using Git.	CG1, CG2, CG5, CG6, CG7, CE4
▪ **Tutoring** for groups or individual students will be organized upon request.	–
Out-of-class activities	**Competences**
▪ Personal study of the course material and resolution of the proposed exercises.	CG1, CG7, CE4
▪ Lab session preparation to make the most of in-class time.	CG1
▪ Lab results analysis and report writing.	CG2, CG5, CE4

STUDENT WORK-TIME SUMMARY

IN-CLASS HOURS			
Lectures		Lab sessions	
7.5		7.5	
OUT-OF-CLASS HOURS			
Self-study	Lab preparation	Lab completion	Lab report writing
12	4	8	6
		ECTS credits:	1.5 (45 hours)

EVALUATION AND GRADING CRITERIA

The use of AI to produce entire assignments or significant parts of them, without citing the source or tool used, or without explicit permission in the assignment description, will be considered plagiarism and will be subject to the University's General Regulations.

Evaluation activities	Grading criteria	Weight
Final exam	▪ Understanding of the theoretical concepts. ▪ Application of these concepts to problem-solving.	30%
Lab assignments	▪ Application of theoretical concepts to real problem-solving. ▪ Ability to understand and execute Bash commands. ▪ Ability to work on shared projects using Git. ▪ Ability to create CI/CD pipelines. ▪ Written communication skills.	70%

Grading

Regular assessment

- **Theory:**
  - Final exam: 30%
- **Lab assignments:**
  - Bash: 20%
  - Git: 20%
  - CI/CD: 30%

In order to pass the course, the weighted average mark must be greater or equal to 5 out of 10 points, the mark of the final exam must be greater or equal to 4 out of 10 points, and the laboratory mark (the weighted average of the assignments) must be at least 5 out of 10 points. Otherwise, the final grade will be the lower of the three marks.

Retake

Only those activities with a failing grade will be repeated, let them be the final exam or any of the lab assignments. The rest of the marks will be preserved. The final grade will be computed as in the regular assessment period and according to the same restrictions.

Course rules

- Class attendance is mandatory according to Article 93 of the General Regulations (Reglamento General) of Comillas Pontifical University and Article 6 of the Academic Rules (Normas Académicas) of the ICAI School of Engineering. Not complying with this requirement may have the following consequences:
  - Students who fail to attend more than 15% of the lectures may be denied the right to take the final exam during the regular assessment period.

- Regarding laboratory, absence to more than 15% of the sessions can result in losing the right to take the final exam of the regular assessment period and the retake. Missed sessions must be made up for credit.

- According to Article 168, section 2.e) of the General Regulations of Universidad Pontificia Comillas, a serious offense is defined as *"any action aimed at falsifying or defrauding the systems used to assess academic performance."* If any irregularity is detected in an academic activity, the work will be graded with a zero (0.0), and disciplinary proceedings may be initiated. In laboratory assignments and the final project, the following will be considered irregularities: the total or partial copying of source code or answers from other students, whether from the current academic year or previous years. The literal or paraphrased reproduction of content from external sources without proper citation will also be considered an attempt of plagiarism. This includes content generated using generative artificial intelligence models, which must comply with the guidelines outlined in the following section.

Guidelines for the use of generative artificial intelligence (AI)

- **Exams.** The use of generative artificial intelligence models or programming assistants is strictly prohibited in the exams. These activities must reflect exclusively the student's own knowledge and individual work.
- **Laboratory.** The use of AI-based programming assistants and generative language models is permitted under the following conditions:
  - These tools may be used as support for understanding technical concepts, obtaining suggestions on how to approach the proposed exercises, and generating code snippets or initial drafts of reports.
  - Their use must always be complementary and must not replace the student's individual work. Submitting automatically generated content as one's own, without proper understanding, review, and adaptation, is not allowed.
  - Any relevant content generated wholly or partially using these tools must be explicitly cited, clearly indicating which parts were generated with AI and which tools were used. The sequence of prompts must be included as an annex at the end of the report.
  - Instructors reserve the right to ask oral questions regarding content generated with AI assistance to assess the student's understanding. Failure to explain or justify such content may negatively impact the grade for the activity.

  The responsible use of these tools is encouraged as a means of supporting individual study (e.g., to clarify concepts, generate additional exercises, or receive feedback). However, students should be aware that responses generated by AI models may contain errors, and it is their responsibility to critically assess and verify the information provided.

WORK PLAN AND SCHEDULE[2]

Activities	Date/Periodicity	Deadline
Final exam	Two weeks after the last session	–
Lab sessions	Weeks 2, 4, 6 and 7	–
Review and self-study of the concepts covered in the lectures	After each lesson	–
Lab preparation	Before every lab session	–
Lab report writing	–	Two weeks after the end of each session

BIBLIOGRAPHY AND RESOURCES

Basic references

- Slides prepared by the lecturer (available in Moodle).

[2] A detailed work plan of the subject can be found in the course summary sheet (see following page). Nevertheless, this schedule is tentative and may vary to accommodate the rhythm of the class.

Complementary references

- K. Beck, et al., *Principles behind the Agile manifesto*, 2001, [Online]. Available: https://agilemanifesto.org/principles.html
- G. Kim, J. Humble, P. Debois, and J. Willis, *The DevOps Handbook: How to Create World-Class Agility, Reliability and Security in Technology Organizations,* 2nd Ed., IT Revolution Press, 2021. ISBN-13: 978-1-950-50840-2
- A. Wiggins, *The Twelve-Factor App*, 2017, [Online]. Available: https://12factor.net/
- A. Sinha, *Introduction to DevOps on AWS*, 2023, [Online]. Available: https://docs.aws.amazon.com/whitepapers/latest/introduction-devops-aws/introduction-to-devops.html

In compliance with current legislation on the **protection of personal data**, we inform and remind you that you can check the privacy and data protection terms you accepted at registration by entering this website and clicking "download".

https://servicios.upcomillas.es/sedeelectronica/inicio.aspx?csv=02E4557CAA66F4A81663AD10CED66792

	In-class activities				Out-of-class activities				Learning outcomes
Week	Time [h]	Lecture	Laboratory	Assessment	Time [h]	Self-study	Lab preparation and report writing	Other activities	Code
1	2	Course overview (0.5h) 1.1. The Agile philosophy (0.5h) 1.2. Scrum vs. waterfall (0.5h) 1.3 Agile practices and tools (0.5h)			1	Review and self-study (1h)			RA1
2	2	2.1. Introduction to Bash (0.5h)	Lab 1. Tinkering with Bash (1.5h)		5	Review and self-study (1h)	Lab completion (2h) Report writing (2h)		RA2
3	2	3.1. Introduction to version control systems: Git (1.5h) 3.2. Overview of version control development platforms (0.5h)			4	Review and self-study (2h)	Lab preparation (2h)		RA3
4	2		Lab 2. Collaborating on a project (2h)		4		Lab completion (2h) Report writing (2h)		RA3
5	2	4.1. The DevOps philosophy (0.5h) 4.2. Understanding a basic CI/CD pipeline (1h) 4.3. CI/CD platforms (0.5h)			4	Review and self-study (2h)	Lab preparation (2h)		RA4
6	2		Lab 3. Creating a CI/CD pipeline (2h)		2		Lab completion (2h)		RA4
7	2		Lab 3. Creating a CI/CD pipeline (1h)		4		Lab completion (2h) Report writing (2h)		RA4
8	1			Final exam	6	Final exam preparation (6h)			RA1 – RA4