

Course	Systems Analysis
Coordinator	Vitor Santos
ECTS	6
Objectives:	<p>This course aims to provide students with knowledge of cutting edge technologies for the specification, analysis, design, implementation and maintenance of complex software systems. The following aspects are considered fundamental:</p> <ul style="list-style-type: none"> -Provide students with knowledge about Software Engineering methods -Provide knowledge about the software design methods 'state of the art' - Evaluation and quality assurance in software systems approaches - Support training for process and business environments analysis <p>At the end of this unit the student should be able to:</p> <ol style="list-style-type: none"> 1- Understand the key technologies and methodological issues 2-Knowing the data and information management methods 3-Mastering the methods and issues related to the design of systems 4-Handle tools and techniques for modeling and development of information systems.
Curricular Unit Contents:	<p>The curricular unit is organized in six Learning Units (LU):</p> <p>LU1 - General concepts Software product Characterization Software engineering basic elements</p> <p>LU2 - Models of software development processes Linear sequential, RAD , prototyping, evolutionary models</p> <p>LU3 - Systems development Methods Structured Analysis, SSADM, Information Engineering, SSM, OO Methods</p> <p>LU4 - Object Oriented Analysis and Design (OOA/OOD) Concepts. Unified Modeling Language Analysis and specification of software requirements</p> <p>LU5 - Data Modelling Entity Relationship Model Extensions to the ER model Defining the content data structure</p>

	<p>LU6 - Software Engineering and the software development problematic</p> <p>Software development cycle in the object-oriented perspective: Rational Unified Process, Iconix.</p> <p>Agile methods: XP, Scrum and Crystal.</p> <p>Process improvement: Levels of maturity (Capability Maturity Model).</p> <p>Metrics in the software development process.</p> <p>Software Patterns</p>
Teaching methods:	<p>Teaching based on lectures and practical classes. The lectures are, in essence, for expository sessions, which serve to introduce the fundamental concepts of databases associated with each of the topics. The practical classes are based on the software design in UML, using the computers and installed software.</p> <p>Teaching Methods</p> <ul style="list-style-type: none"> • Expository and interrogative teaching:lectures and discussions. • Declarative:tutorials tools • Active and participative:case studies, participation in project teams, software design with UML.
Grading methods:	<p>Evaluation:</p> <p>1st round:Two Theoretical tests (50%) + Practicals Works (50%)</p> <p>2nd round:final exam (100%).</p>
Bibliography:	<ul style="list-style-type: none"> • Software Engineering:A Practitioner's Approach Roger S. Pressman McGraw-Hill Higher Education 7 edition (1 April 2009), ISBN-10:0071267824 • Requirements Analysis:From Business Views to Architecture David C. Hay Prentice Hall (2002) ISBN:978- 0130282286 • Effective Methods for Software Testing, 3rd Edition William E. Perry Wiley (2006) ISBN:978-0764598371 • UML Metodologias e ferramentas CASE, by Alberto Silva e Carlos Videira, Centro Atlântico, ISBN:989-615-009