Degree competences to which the subject contributes

Specific:
- 3. CEFC1. Ability to design, develop, select and value applications and informatic systems affirming its reliability, security and quality corresponding to ethical principals and legislation and current rules.
- 6. CEFC2. Ability to plan, conceive, develop, manage informatic projects, services and systems in all areas, leading their implementation and continuous improvement assassing their economic and social repercussions.
- 2. CEFC6. Basic knowledge and application of algorithmic processes, informatic techniques to design solutions of problems, analyzing if proposed algorisms are apt and complex.
- 8. CEFC8. Ability to analyze, to design, to construct and to maintain applications in a well built, secure and efficient way choosing the most adequated paradigms and languages.
- 10. CETI5. Ability to select, to develop, integrate and manage information systems which satisfy organization necessities with indentified costs and quality criteria.
- 11. CEFC3. Ability to understand the importance of negotiation, effective work habits, leadership and communication skills in all environments for software development.

Transversal:
- 9. SELF-DIRECTED LEARNING - Level 2: Completing set tasks based on the guidelines set by lecturers. Devoting the time needed to complete each task, including personal contributions and expanding on the recommended information sources.
- 12. SELF-DIRECTED LEARNING. Detecting gaps in one's knowledge and overcoming them through critical self-appraisal. Choosing the best path for broadening one's knowledge.
- 16. EFFICIENT ORAL AND WRITTEN COMMUNICATION. Communicating verbally and in writing about learning outcomes, thought-building and decision-making. Taking part in debates about issues related to the own field of specialization.
- 18. TEAMWORK - Level 2. Contributing to the consolidation of a team by planning targets and working efficiently to favor communication, task assignment and cohesion.
- 20. TEAMWORK. Being able to work as a team player, either as a member or as a leader. Contributing to projects pragmatically and responsibly, by reaching commitments in accordance to the resources that are available.
340376 - INEP-I3O23 - Introduction to Software Engineering

**Teaching methodology**

A case study that conveys the course.

Material in the form of notes, summarized in the form of projections, for the study and review of content.

Exercises and problems that students must solve on their own as a means of understanding the contents of the subject.

Done exercises as a mechanism of self-evaluation.

A project in pairs as a fundamental mechanism for learning and for evaluation.

Class sessions where they discussed the hard issues of content, or where it is discussed in detail how they arrived at a certain solution. The basic definitions and concepts of self are related by 80%.

Tutoring lab sessions of work done.

**Learning objectives of the subject**

Understanding the different stages of development, and the different objectives of each stage.

Being able to model a problem and its solutions. Learn to document a development.

Being able to build up a complete specification of small size problem in a environment similar to the professional one: excess of requirements, lack of time, teamwork, telecommuting.

Know how to manage contextual pressure, and therefore know to plan, prioritize and simplify.

**Study load**

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Hours large group: 30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hours medium group: 0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Hours small group: 30h</td>
<td>20.00%</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Self study: 90h</td>
<td>60.00%</td>
</tr>
</tbody>
</table>

A case study that conveys the course.

Material in the form of notes, summarized in the form of projections, for the study and review of content.

Exercises and problems that students must solve on their own as a means of understanding the contents of the subject.

Done exercises as a mechanism of self-evaluation.

A project in pairs as a fundamental mechanism for learning and for evaluation.

Class sessions where they discussed the hard issues of content, or where it is discussed in detail how they arrived at a certain solution. The basic definitions and concepts of self are related by 80%.

Tutoring lab sessions of work done.

Understanding the different stages of development, and the different objectives of each stage.

Being able to model a problem and its solutions. Learn to document a development.

Being able to build up a complete specification of small size problem in a environment similar to the professional one: excess of requirements, lack of time, teamwork, telecommuting.

Know how to manage contextual pressure, and therefore know to plan, prioritize and simplify.

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Hours large group: 30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hours medium group: 0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Hours small group: 30h</td>
<td>20.00%</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Self study: 90h</td>
<td>60.00%</td>
</tr>
</tbody>
</table>
### Phases in the development of a software project

<table>
<thead>
<tr>
<th>Learning time: 7h 30m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 4h</td>
</tr>
<tr>
<td>Self study: 3h 30m</td>
</tr>
</tbody>
</table>

**Description:**
Generic stages should have a development: analysis, specification, design and implementation.

**Related activities:**
Project

**Specific objectives:**
Understanding the various needs and objectives of each stage. Learn to jump from one stage to another. Distinguish when you are talking about a problem and when you are talking about a possible solution to this.

### Modeling problems

<table>
<thead>
<tr>
<th>Learning time: 125h 30m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 20h</td>
</tr>
<tr>
<td>Laboratory classes: 28h</td>
</tr>
<tr>
<td>Guided activities: 66h 40m</td>
</tr>
<tr>
<td>Self study: 10h 50m</td>
</tr>
</tbody>
</table>

**Description:**

**Related activities:**
Project

**Specific objectives:**
Learn modeling problems: namely, to explain in a concise, specific and unambiguous what the problem is to solved.
**Modeling software solutions**

<table>
<thead>
<tr>
<th><strong>Learning time:</strong> 10h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 4h</td>
</tr>
<tr>
<td>Laboratory classes: 2h</td>
</tr>
<tr>
<td>Self study: 4h</td>
</tr>
</tbody>
</table>

**Description:**
Models of the solution: visibility, responsibility.

**Related activities:**
Build static model form a given dynamic model

**Specific objectives:**
Understand the difference between the model of the problem and the model of one possible solutions: similarities and differences.

- Being able to relegate every decision in the correct domain (problem or solution).
- Being able to build static solution model from dynamic solution model

---

**Introduction to persistence and data bases**

<table>
<thead>
<tr>
<th><strong>Learning time:</strong> 7h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 1h</td>
</tr>
<tr>
<td>Self study: 6h</td>
</tr>
</tbody>
</table>

**Description:**
See catalan version

**Related activities:**
See catalan version.

**Specific objectives:**
See catalan version

---

**Qualification system**

See catalan version

**Regulations for carrying out activities**

See castalan version
Bibliography

Basic:


