Universitat de Lleida

DEGREE CURRICULUM
SOFTWARE ARCHITECTURES

Coordination: GARCIA GONZALEZ, ROBERTO

Academic year 2023-24

SOFTWARE ARCHITECTURES 2023-24

Subject's general information

<table>
<tr><td><b>Subject name</b></td><td>SOFTWARE ARCHITECTURES</td></tr>
<tr><td><b>Code</b></td><td>102055</td></tr>
<tr><td><b>Semester</b></td><td>1st Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td><b>Typology</b></td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>4</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>4</td><td>OPTIONAL</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td><b>Course number of credits (ECTS)</b></td><td>9</td></tr>
<tr><td><b>Type of activity, credits, and groups</b></td><td>
<table>
<tr><th>Activity type</th><td>PRALAB</td><td>TEORIA</td></tr>
<tr><th>Number of credits</th><td>3</td><td>6</td></tr>
<tr><th>Number of groups</th><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td><b>Coordination</b></td><td>GARCIA GONZALEZ, ROBERTO</td></tr>
<tr><td><b>Department</b></td><td>COMPUTER ENGINEERING AND DIGITAL DESIGN</td></tr>
<tr><td><b>Teaching load distribution between lectures and independent student work</b></td><td>Total load: 225h
- Lectures (40%) = 90h
- Independent student work (60%) = 135h</td></tr>
<tr><td><b>Important information on data processing</b></td><td>Consult this link for more information.</td></tr>
<tr><td><b>Language</b></td><td>English</td></tr>
</table>

SOFTWARE ARCHITECTURES 2023-24

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
CHAMORRO PADIAL, JORGE	jorge.chamorro@udl.cat	2	
CONTRERAS PÉREZ, FRANCESC	francesc.contreras@udl.cat	3,6	
GARCIA GONZALEZ, ROBERTO	roberto.garcia@udl.cat	5,2	Please, make your appointment via e-mail

Subject's extra information

To properly follow this subject, it is recommended to have consolidated software engineering and Web applications development skills like those presented in the Software Engineering and Web Systems and Technologies courses.

Learning objectives

- To present Web engineering and its fundamental applications patterns and architectures in the context of an agile methodology and Behaviour Driven Development.
- To have a global vision of the existent technologies to implement enterprise applications using the previous patterns and architectures, especially Java, Javascript and the Web.
- To put into practice the previous concepts and technologies through the development of a Web application project using Java (Spring) and TypeScript (Angular), following an agile methodology and behaviour-driven development (BDD).

Competences

Strategic Competences

CT2. Mastering a foreign language, especially English.

CT3. Training Experience in the use of the new technologies and the information and communication technologies.

Cross-disciplinary Competences

EPS11. Capacity to understand the needs of the user expressed in a no technical language.

Specific Competences

GII-IS1. Capacity to develop, maintain and evaluate services and software systems that satisfy all the requirements of the user and behave in a reliable and efficient way, they can develop, keep and fulfil quality requirements, applying the theories, principles, methods and uses of the software engineering.

GII-IS3. Capacity to give solution to problems of integration taking into account the strategies, standards and available technologies.

GII-IS4. Capacity to identify and analyse problems and design, develop, implement, verify and find software solutions on the base of a suitable knowledge of the theories, models and current techniques.

Subject contents

1. Agile Web Applications Development Process
    1. Behaviour Driven Development (BDD) with Cucumber Java
    2. Scrum using GitHub
    3. Teamwork coordination using Version Control
    4. Continuous Integration using GitHub Actions
    5. Automated deployment on the cloud
2. Application Architecture Patterns
    1. Introduction and layered architecture
    2. Patterns in the domain layer
    3. Patterns in the integration layer
    4. Patterns in the presentation layer
    5. Technologies for pattern application
3. Web Application Implementation
    1. Server-side: Java (Spring)
    2. Client-side: TypeScript (Angular)

Methodology

The methodology is based on a Project-Based Learning approach where an enterprise software application is developed, focusing on Web-based applications. The course starts with a review of Web development frameworks from an industrial point of view analysing different reports about the state of the art of software development, etc. Based on these dimensions, Spring is selected for the server-side and Angular for the client one.

The first weeks of the course focus on fundamental aspects of software development, including project management ideas and development process concepts. Software architecture, specification, analysis and design are presented from a client/server and Web perspective, guided by enterprise application patterns. These concepts are the starting point for the students' project development.

The students select a project, under professors guidance, and complete a first deliverable where they apply these fundamental concepts to specify, analyse and design their project.

The rest of the course focuses on the implementation of the defined project and the required concepts related to Agile software development, Spring or Angular are introduced as needed to complete the development of the project.

Development plan

Week	Description	Face-to-Face Activities	Autonomous Student Work
1	Frameworks and Methodologies	Study Frameworks JVM Servidor Study Frameworks JavaScript Client	Define project idea
2	Frameworks and Methodologies	SCRUM and BDD	Define project using methodology terms
3	Sprint 1	Tutorials presentations Joint project development	Project development
4	Sprint 1	Tutorials presentations Joint project development	Project development
5	Sprint 1	Tutorials presentations Joint project development	Project development

Week	Description	Face-to-Face Activities	Autonomous Student Work
6	Sprint 1	Tutorials presentations Joint project development	Project development
7	Sprint 1	Enterprise application patterns Joint project development	Study patterns application to project
8	Sprint 1	Enterprise application patterns Sprint Retrospective	Study patterns application to project
9		1st Midterm Exam	Study
10	Sprint 2	Tutorials presentations Joint project development	Project development
11	Sprint 2	Tutorials presentations Joint project development	Project development
12	Sprint 2	Tutorials presentations Joint project development	Project development
13	Sprint 2	Joint project development Sprint Retrospective	Project development
14	Sprint 3	Tutorials presentations Joint project development	Project development
15	Sprint 3	Tutorials presentations Joint project development	Project development
16	Sprint 3	Joint project development Sprint Retrospective	Project development
17-18		2nd Midterm Exam	Study

Evaluation

Acr.	Evaluation Activity	Rate	Minimum Qualification	Group Activity	Mandatory	Remedia Exam
P1	1st Midterm Exam	20%	NO	NO	NO	NO
P2	2nd Midterm Exam	20%	NO	NO	NO	NO
E1	1st Project Deliverable	20%	NO	2 o 3	NO	NO
E2	2ª Project Deliverable	20%	NO	2 o 3	NO	NO
E3	3ª Project Deliverable	20%	NO	2 o 3	NO	NO
Final Qualification = 0,2*P1 + 0,2*P2 + 0,2*E1 + 0,2*E2 + 0,2*E3						

The evaluation is fundamentally based on the development of a project following an agile methodology following 3 sprints and 3 deliverables:

- Sprints 1, 1st Deliverable: 20% grade
- Sprints 2, 2nd Deliverable: 20% grade

- Sprints 3, 3rd Deliverable: 20% grade

For each sprint (review), the performance in developing the assigned tasks will be evaluated, which will be contextualised for each corresponding deliverable that takes into account also the developed product. The grade will combine the outcomes for each deliverable together with its associated sprints.

The evaluation is complemented by two exams performed individually:

- 1st Midterm Exam: 20% grade, about fundamental aspects of the frameworks and methodologies used so far.
- 2nd Midterm Exam: 20% grade, about the development experience of the project as a whole.

Alternative Evaluation

Students who have the approval to be evaluated through alternative evaluation (see requirements and procedure in the evaluation regulations) must carry out the following activities.

Acr.	Evaluation Activity	Rate	Minimum Qualification	Group Activity	Mandatory	Remedia Exam
PF	Final Exam	40%	>3	NO	YES	YES
E1	1st Project Deliverable	20%	NO	NO	NO	NO
E2	2ª Project Deliverable	20%	NO	NO	NO	NO
E3	3ª Project Deliverable	20%	NO	NO	NO	NO
Final Qualification = 0,4*PF + 0,2*E1 + 0,2*E2 + 0,2*E3						

The evaluation is fundamentally based on the development of a project following an agile methodology following 3 sprints and 3 deliverables:

- Sprints 1, 1st Deliverable: 20% grade
- Sprints 2, 2nd Deliverable: 20% grade
- Sprints 3, 3rd Deliverable: 20% grade

For each sprint (review), the performance in developing the assigned tasks will be evaluated, which will be contextualised for each corresponding deliverable that takes into account also the developed product. The grade will combine the outcomes for each deliverable together with its associated sprints.

The evaluation is complemented by a final exam:

- Final exam: 40% grade, about fundamental aspects of the frameworks and methodologies used so far, plus the development experience of the project as a whole.

Bibliography

- Fowler, M.; Rice, D. (2003). Patterns of Enterprise Application Architecture. Addison-Wesley.
  - https://discovery.udl.cat/iii/encore/record/C__Rb1199416
- Walls, C. (2015). Spring in Action, 4th Edition. Manning.
  - https://discovery.udl.cat/iii/encore/record/C__Rb1326886
- Carnell, J. (2017) Spring microservices in action. Manning.
  - https://discovery.udl.cat/iii/encore/record/C__Rb1354387
- Wilken, J. (2018). Angular in action. Manning.
  - https://discovery.udl.cat/iii/encore/record/C__Rb1354388
- Amuthan, G. (2014). Spring MVC: Beginner's guide Birmingham. Packt Publishing.
  - https://www.dawsonera.com/abstract/9781783284887