

2016716

Software Architecture



COURSE DESCRIPTION

This course introduces the principal aspects associated with the architectural design of software systems. It studies, evaluates and applies different approaches to solve software design problems and to ensure the quality of the software. Theoretical and practical methodologies are used to illustrate the elements of software architecture.

COURSE GOAL

Drawing upon the experiences in software engineering and the activities to develop throughout this course, at the end of this course students will be able to design complex software architectures, implement software systems aligned with the most demanding functional and non-functional requirements, as well as have a critical view of current challenges in the field of software architecture.

LEARNING OBJECTIVES

By the end of this course, students will be able to:

1. Recognize the importance of software architecture in the field of software engineering.
2. Classify the different approaches of software design.
3. Carry out an adequate architecture definition process.
4. Reflect on the role of the software architect.
5. Provide software architecture representations through the use of architectural views.
6. Integrate architectural styles in order to design different types of software architectures.
7. Use architectural patterns to solve software design problems.
8. Design strategies to ensure the quality of the software from different types of architectural tactics.
9. Create software systems based on good practices of software architecture.

INSTRUCTOR INFO

Jeisson Andrés Vergara Vargas

Profesor Auxiliar

Departamento de Ingeniería de Sistemas e Industrial



<http://colse.unal.edu.co/~javergarav/>
javergarav@unal.edu.co

COURSE INFO

Prerequisite

Software Engineering II

Class Schedule

Tuesday and Thursday

Group 1: 7-9, Group 2: 9-11

453-119

Credits

3

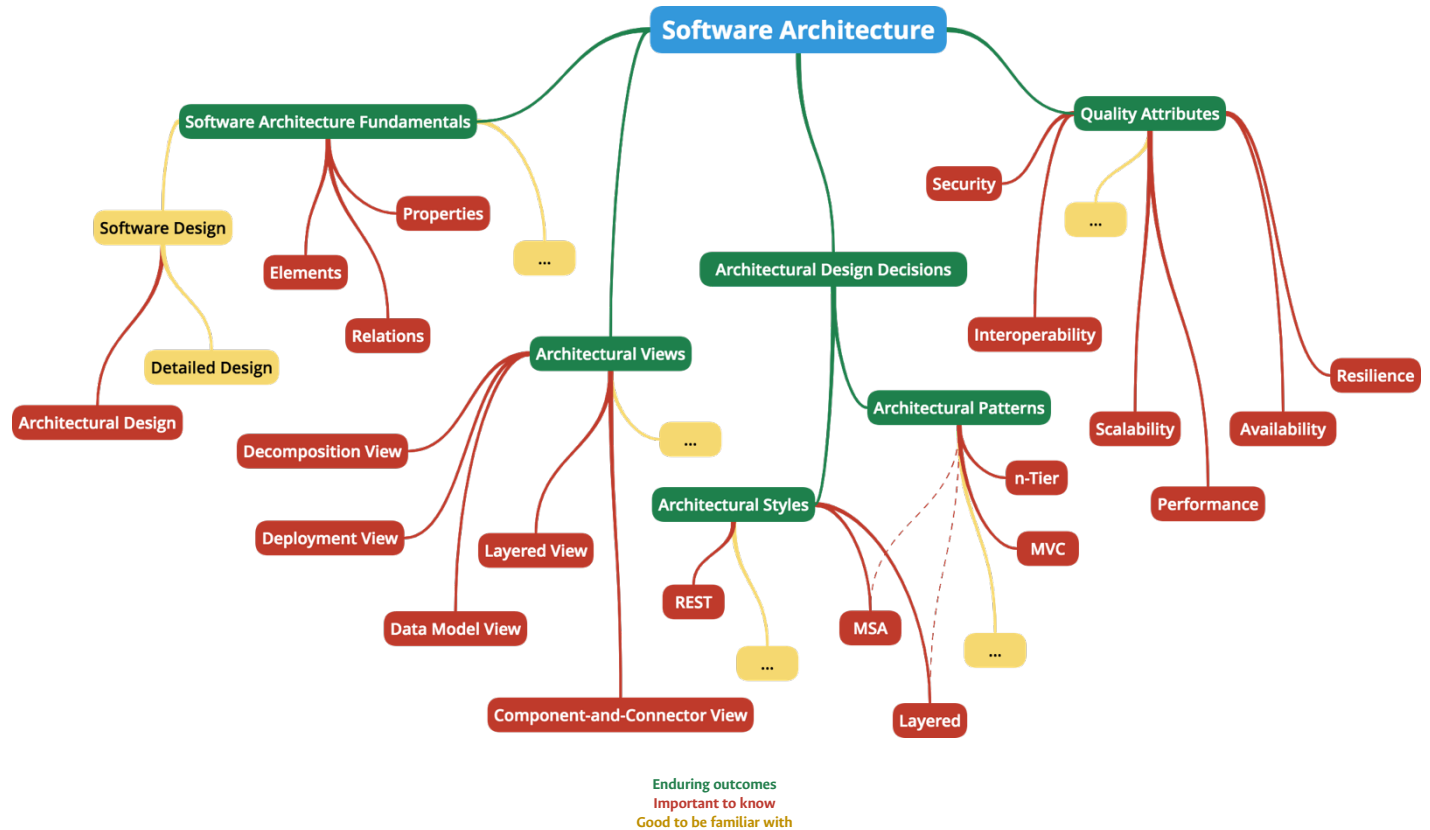
Level

Undergraduate

Curricular Program

Ingeniería de Sistemas y Computación

TREE OF CONTENTS



SCHEDULE

Class		Topic	Contents	
1	Tue. / Mar. 08	Course Presentation		
2	Thu. / Mar. 10	Software Architecture Fundamentals Architectural Views	Software Design: Architectural vs. Detailed	
3	Tue. / Mar. 15		Architectural Elements, Relations and Properties	
4	Thu. / Mar. 17		Decomposition View	Workshop #1: Decomposition View
5	Tue. / Mar. 22		Component-and-Connector (C&C) View	
6	Thu. / Mar. 24		Workshop #2: C&C View	
7	Tue. / Mar. 29	Architectural Design Decisions	Architectural Styles	Laboratory #1: Data
8	Thu. / Mar. 31		Architectural Styles	
9	Tue. / Apr. 05		Laboratory #2: Logic, Part 1	
10	Thu. / Apr. 07	Architectural Views	Data Model View	Workshop #3: Data Model View
Tue. / Apr. 12		Holy Week		
Thu. / Apr. 14				
11	Tue. / Apr. 19	Architectural Design Decisions	Architectural Patterns	
12	Thu. / Apr. 21		Prototype #1: Back-End, Part 1	
13	Tue. / Apr. 26		Laboratory #3: Logic, Part 2	
14	Thu. / Apr. 28	Architectural Views	Layered View	
15	Tue. / May 03		Workshop #4: Layered View	
16	Thu. / May 05	Architectural Design Decisions	Prototype #2: Back-End, Part 2	
17	Tue. / May 10		Laboratory #4: Presentation	
18	Thu. / May 12	Architectural Views	Deployment View	
19	Tue. / May 17		Workshop #5: Deployment View	

20	Thu. / May 19		Prototype #3: Front-End
21	Tue. / May 24	Quality Attributes	The Security Perspective
22	Thu. / May 26		Laboratory #5: Security
23	Tue. / May 31		The Interoperability Perspective
24	Thu. / Jun. 02		Laboratory #6: Interoperability
25	Tue. / Jun. 07		The Performance and Scalability Perspective
26	Thu. / Jun. 09		Prototype #4: Quality Attributes, Part 1
27	Tue. / Jun. 14		Laboratory #7: Performance
28	Thu. / Jun. 16		Laboratory #8: Clustering
29	Tue. / Jun. 21		The Availability and Resilience Perspective
30	Thu. / Jun. 23		Laboratory #9: Horizontal Scaling
31	Tue. / Jun. 28		Exam
32	Thu. / Jun. 30		Prototype #5: Quality Attributes, Part 2
33	Thu. / Jul. 05	Course Completion	

ASSESSMENT

Component	%
Project	45%
Prototypes (P)	
$(P1 + P2 + P3 + P4 + P5) / 5$	
Laboratories	15%
Laboratories (L)	
$(L1 + L2 + L3 + L4 + L5 + L6 + L7 + L8 + L9) / 9$	
Workshops	15%
Workshops (L)	
$(W1 + W2 + W3 + W4 + W5) / 5$	
Class Activities	15%
Class Activities (CA)	
$(CA_1 + CA_2 + \dots + CA_n) / n$	
Exam	10%
TOTAL	100%

POLICIES

- Class start time is at 7:10 for group 1 and at 9:10 for group 2.
- Project deliveries will be made through Moodle, the day before the presentation (deadline: 23h59).
- For project presentations, the attendance of all team members is mandatory.
- The minimum grade to approve the course is three point zero (3.0).
- Missing classes could leave students to get a grade of zero point zero (0.0), according to: “Artículo 32 del Acuerdo 008 de 2008 – Estatuto Estudiantil de la Universidad Nacional de Colombia”.
- Plagiarism cases will be informed to the academic office according to the academic regulations.
- This course, like any other course at the Universidad Nacional de Colombia, should be a safe environment for each person to express their ideas. This means that there must be respect for everyone's opinion and empathy to see the world from another person's perspective.

REFERENCES

- **[BASS]** L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. 2013.
- **[BROWN]** S. Brown, Software Architecture for Developers: A Practical and Pragmatic Guide to 21st Century Software Architecture. 2012.
- **[CERVANTES]** H. Cervantes and R. Kazman, Designing Software Architectures - A Practical Approach. 2016.
- **[CLEMENTS]** P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford, Documenting Software Architectures: Views and Beyond. 2011.
- **[ERL]** T. Erl, Cloud Computing - Concepts, Technology & Architecture. 2013.
- **[HENDRICKSEN]** D. Hendricksen, 12 Essential Skills for Software Architects. 2012.
- **[IEEE]** IEEE, ISO/IEC/IEEE: Systems and Software Engineering - Architecture Description. 2011.
- **[JOSUTTIS]** N. M. Josuttis, SOA in Practice: The Art of Distributed System Design. 2007.
- **[KRUCHTEN]** P. Kruchten, "Architectural Blueprints - The "4+1" View Model of Software Architecture," IEEE Softw., vol. 12, no. November, pp. 42-50, 1995.
- **[MASSÉ]** M. Massé, REST API - Design Rulebook. 2012.
- **[NEWMAN]** S. Newman, Building Microservices: Designing Fine-Grained Systems. 2015.
- **[OUSSALAH-1]** M. C. Oussalah, Software Architecture 1. 2014.
- **[OUSSALAH-2]** M. C. Oussalah, Software Architecture 2. 2014.
- **[RICHARDS-1]** M. Richards, Software Architecture Patterns. 2014.
- **[RICHARDS-2]** M. Richards and N. Ford, Fundamentals of Software Architecture – An Engineering Approach. 2020.
- **[RICHARDSON-1]** C. Richardson and F. Smith, Microservices: From Design to Deployment. 2016.
- **[RICHARDSON-2]** C. Richardson, Microservices Patterns. 2019.
- **[ROZANSKI]** N. Rozanski and E. Woods, Software Systems Architecture, 2nd ed. 2011.
- **[STRAUCH]** C. Strauch, "NoSQL Databases," Lect. Notes Stuttgart Media, pp. 1-8, 2010.
- **[TAYLOR]** R. N. Taylor, N. Medvidovic, and E. M. Dashofy, Software Architecture - *Foundations*, Theory, and Practice. 2010.
- **[WEBBER]** J. Webber, S. Parastatidis, and I. Robinson, REST in Practice. 2010.