

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://colawe.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

Virtual Class

Credits

3

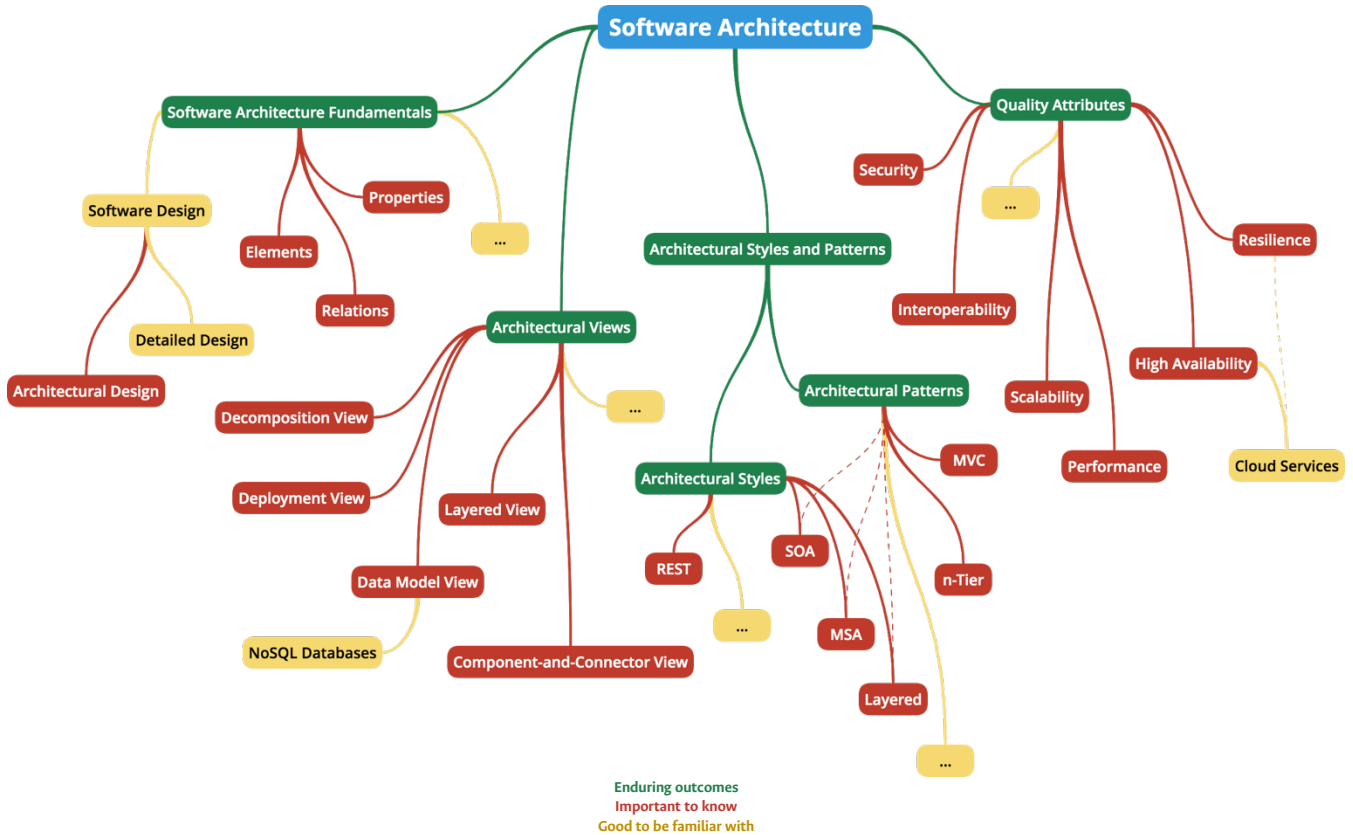
Level

Undergraduate

Curricular Program

Ingeniería de Sistemas y Computación

TREE OF CONTENTS



SCHEDULE

Session		Topic	Content		Project
1	Tue. / Feb. 23	Course Presentation			
2	Thu. / Feb. 25	[Fundamentals]	Software Design: Detailed vs. Architectural		
3	Tue. / Mar. 2		Elements, Relations and Properties		
4	Thu. / Mar. 4		The Architecture Definition Process		
5	Tue. / Mar. 9	[Arch. Views]	Decomposition View	W1	P1
6	Thu. / Mar. 11		Component-and-Connector (C&C) View	W2	
7	Tue. / Mar. 16	[Arch. Styles and Patterns]	Architectural Styles	L1	
Wed. / Mar. 17 / 23:59					Delivery #1
8	Thu. / Mar. 18	[Arch. Styles and Patterns]	Architectural Styles	L2	P2
9	Tue. / Mar. 23	[Arch. Views]	Data Model View	W3	
10	Thu. / Mar. 25	[Arch. Styles and Patterns]	Architectural Patterns		
Mon. / Apr. 5 / 23:59					Delivery #2
11	Tue. / Apr. 6	[Arch. Styles and Patterns]	Architectural Patterns	L3	P3
12	Thu. / Apr. 8				
13	Tue. / Apr. 13			L4	
Wed. / Apr. 14 / 23:59					Delivery #3
14	Thu. / Apr. 15	[Arch. Views]	Layered View	W4	P4
15	Tue. / Apr. 20	[Arch. Views]	Deployment View	W5	
16	Thu. / Apr. 22	[Arch. Styles and Patterns]			
17	Tue. / Apr. 27	Exam #1			
Wed. / Apr. 28 / 23:59					Delivery #4

18	Thu. / Apr. 29	Project: Partial Presentation			
19	Tue. / May 4	[Quality Attributes]	The Security Perspective		P5
20	Thu. / May 6		L5		
21	Tue. / May 11				
22	Thu. / May 13		The Interoperability Perspective		
Mon. / May 17 / 23:59					Delivery #5
23	Tue. / May 18	[Quality Attributes]	The Interoperability Perspective	L6	P6
24	Thu. / May 20				
25	Tue. / May 25		The Performance and Scalability Perspective		
Wed. / May 26 / 23:59					Delivery #6
26	Thu. / May 27	[Quality Attributes]	The Performance and Scalability Perspective	L7	P7
27	Tue. / Jun. 1		L8		
28	Thu. / Jun. 3				
29	Tue. / Jun. 8		The Availability and Resilience Perspective		
30	Thu. / Jun. 10				
31	Tue. / Jun. 15	Exam #2			Delivery #7
Wed. / Jun. 16 / 23:59					
32	Thu. / Jun. 17	Project: Final Presentation			
33	Tue. / Jun. 22	Final Class			

Notation

Workshops	Laboratories	Project
W1: Decomposition View	L1: Infrastructure (homework)	P1: Prototype #1 – First Design
W2: Component-and-Connector (C&C) View	L2: Back-End Architecture I	P2: Prototype #2 – Back-End I
W3: Data Model View	L3: Back-End Architecture II	P3: Prototype #3 – Back-End II
W4: Layered View	L4: Front-End Architecture	P4: Prototype #4 – Front-End
W5: Deployment View	L5: Security	P5: Prototype #5 – Security
	L6: Interoperability	P6: Prototype #6 – Interoperability
	L7: Clustering	P7: Prototype #7 – Scalability and Reliability
	L8: Performance Testing and Scaling	

ASSESSMENT

Component	%
Project	40%
Prototypes (P)	
$(P1 + P2 + P3 + P4 + P5 + P6 + P7_1 + P7_2) / 8$	
Laboratories	15%
Laboratories (L)	
$(L1 + L2 + L3 + L4 + L5 + L6 + L7 + L8) / 8$	
Workshops	15%
Workshops (L)	
$(W1 + W2 + W3 + W4 + W5) / 5$	
Class Activities	15%
Class Activities (CA)	
$(CA_1 + CA_2 + \dots + CA_n) / n$	
Exams	15%
Exams (E)	
$(E1 + E2) / 2$	
TOTAL	100%

POLICIES

- Class start time is at 7:10 for group 1 and at 9:10 for group 2.
- All classes will be developed synchronously, using virtual platforms.
- All classes will be recorded and will be available for consultation until the Sunday immediately following.
- 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.