



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Software Architecture

Course Presentation

Jeisson Andrés Vergara Vargas

Departamento de Ingeniería de Sistemas e Industrial
<http://colswe.unal.edu.co/~javergarav/>
javergarav@unal.edu.co

2021-I

©

Professor

Jeisson Andrés Vergara Vargas



Profesor Auxiliar

Departamento de Ingeniería de Sistemas e Industrial

Website: <http://colswe.unal.edu.co/~javergarav/>

Email: javergarav@unal.edu.co

Professor

Jeisson Andrés Vergara Vargas



Education

“Ph.D. in Systems Engineering and Computer Science”

Universidad Nacional de Colombia, 2020 - Present

M.Sc. in Systems Engineering and Computer Science

Universidad Nacional de Colombia, 2015 - 2017

Systems and Computing Engineer

Universidad Nacional de Colombia, 2009 - 2014

Professor

Jeisson Andrés Vergara Vargas



Trajectory

Auxiliary Professor

Universidad Nacional de Colombia, 2019 - Present

Occasional Teacher

Universidad Nacional de Colombia, 2016 - 2019

Visiting Professor

EPITECH Paris, France, 2019

Université de Poitiers, France, 2017

Software Architect

2015 - Present

Professor

Jeisson Andrés Vergara Vargas



Research



Principal Investigator
Research Line in **Software Architecture**

<http://colswe.unal.edu.co/>

Course

Software Architecture

Course Name	Arquitectura de Software
Course Code	2016716
Level	Undergraduate
Curricular Program	Ingeniería de Sistemas y Computación
Credits	3
Class Schedule	Tuesday and Thursday Group 1: 07-09, Group 2: 09-11 Virtual Class
Prerequisite	Software Engineering II

Course

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

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.

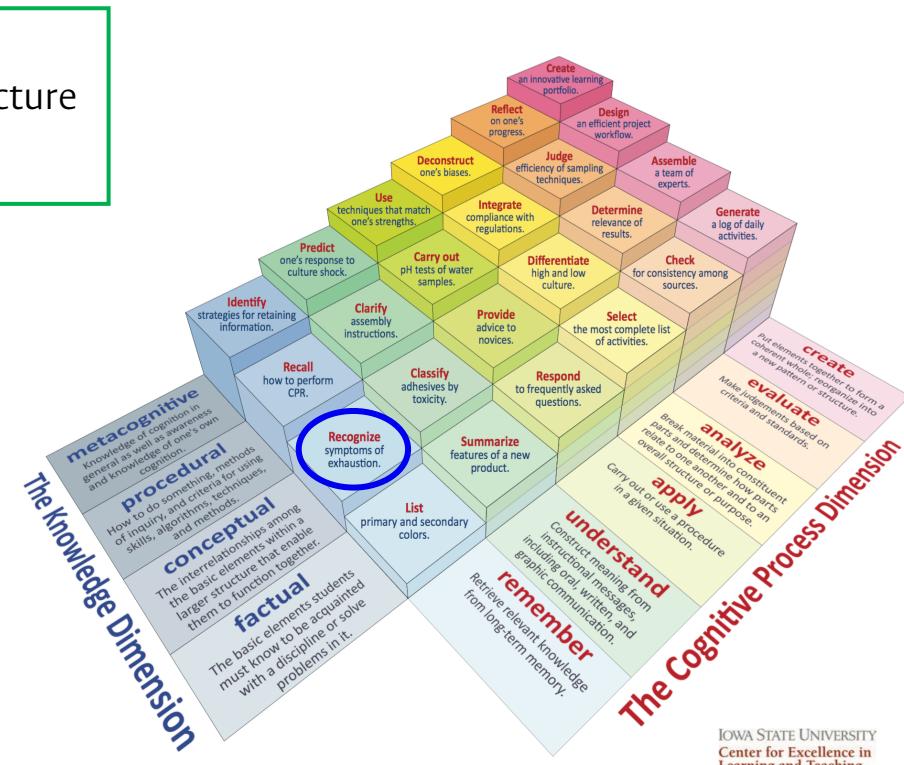
Course

Software Architecture

Learning Objectives

1.

Recognize the importance of software architecture in the field of software engineering.



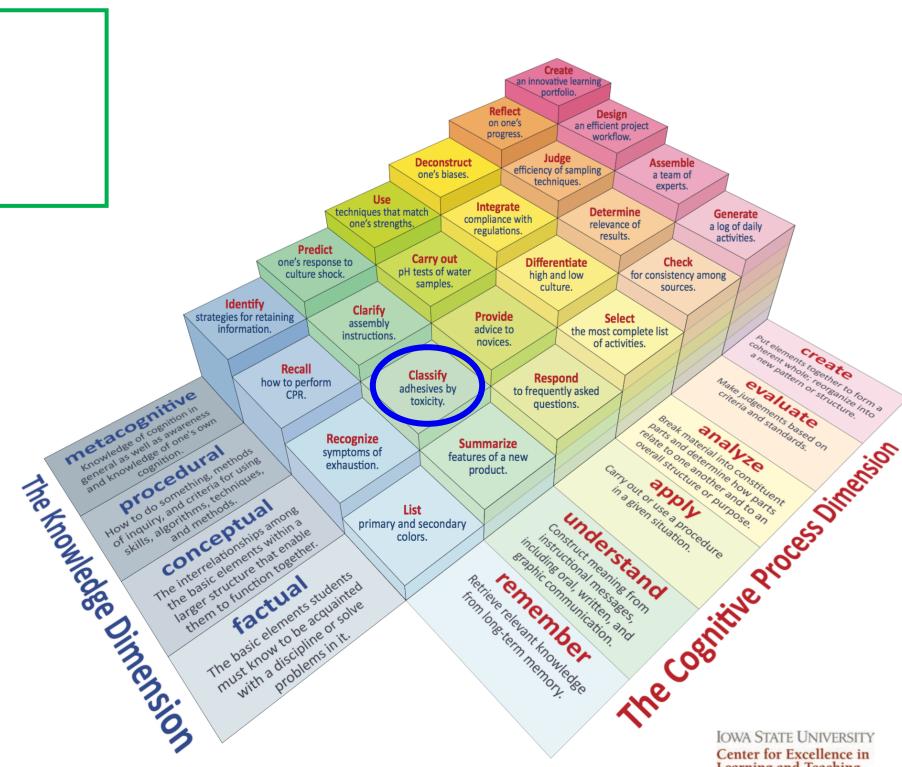
Course

Software Architecture

Learning Objectives

2.

Classify the different approaches of software design.



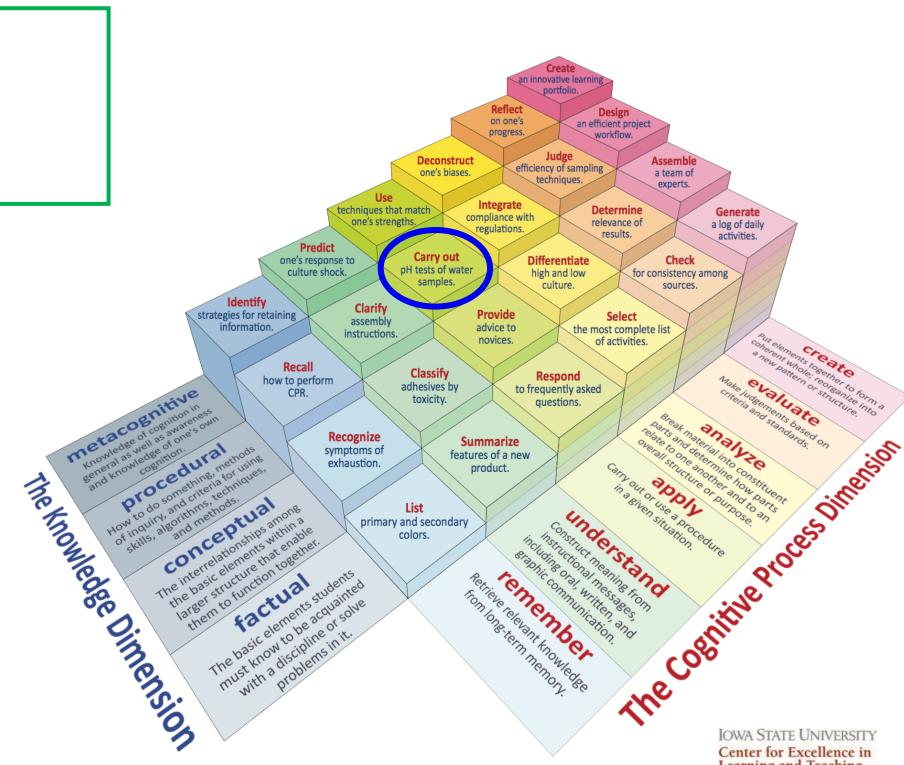
IOWA STATE UNIVERSITY
Center for Excellence in
Learning and Teaching

Course

Software Architecture

Learning Objectives

3.
Carry out an adequate architecture definition process.



IOWA STATE UNIVERSITY
Center for Excellence in
Learning and Teaching

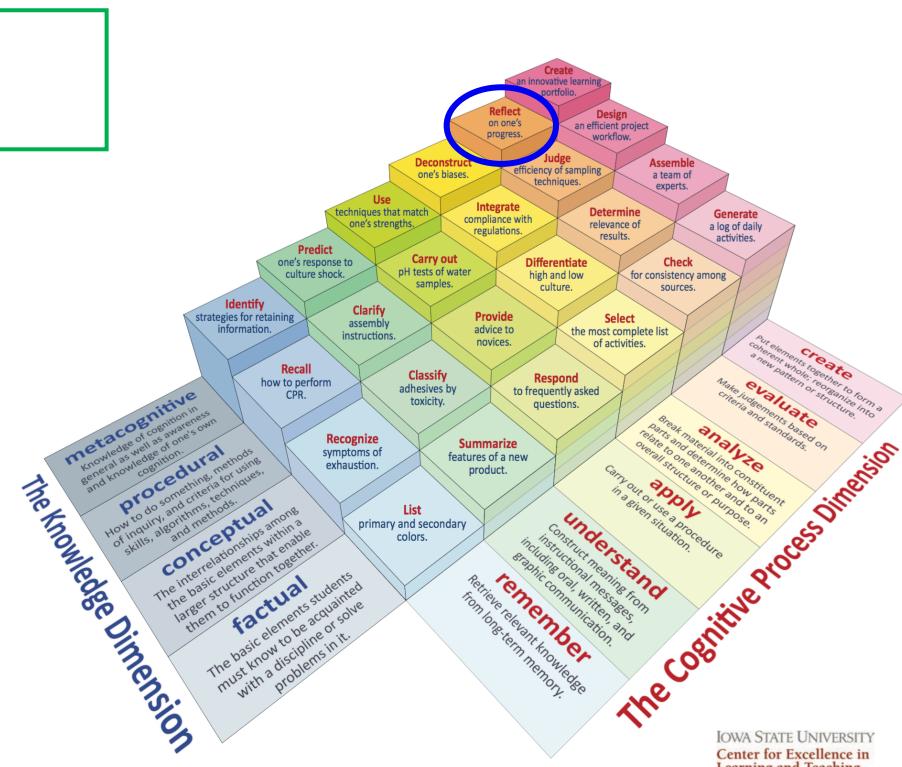
Course

Software Architecture

Learning Objectives

4.

Reflect on the role of the software architect.



IOWA STATE UNIVERSITY
Center for Excellence in
Learning and Teaching

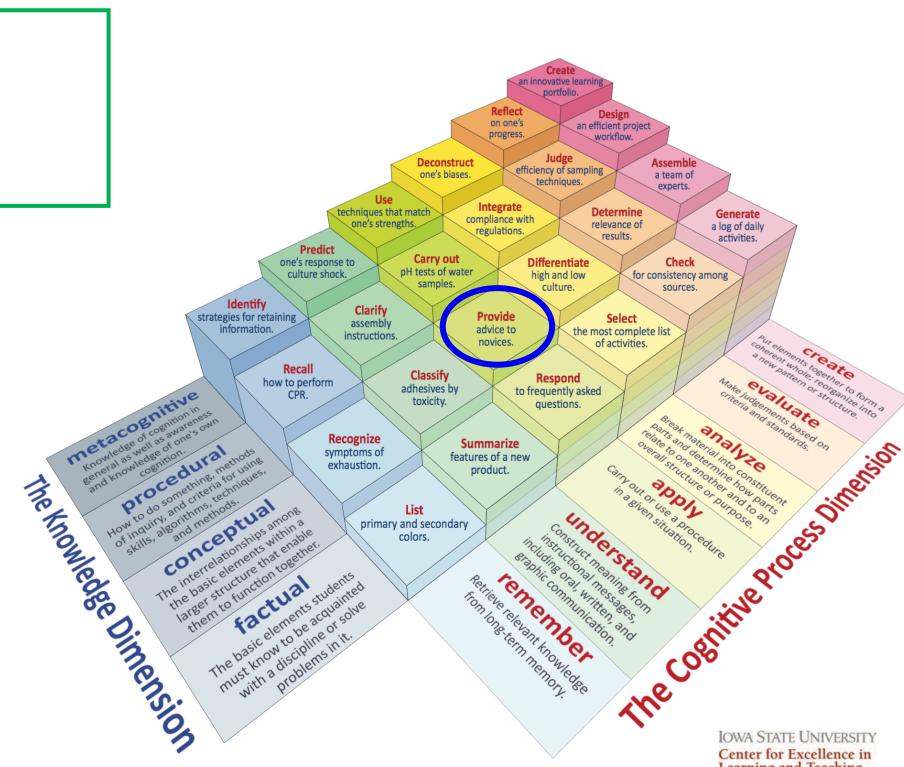
Course

Software Architecture

Learning Objectives

5.

Provide software architecture representations through the use of architectural views.



IOWA STATE UNIVERSITY
Center for Excellence in
Learning and Teaching

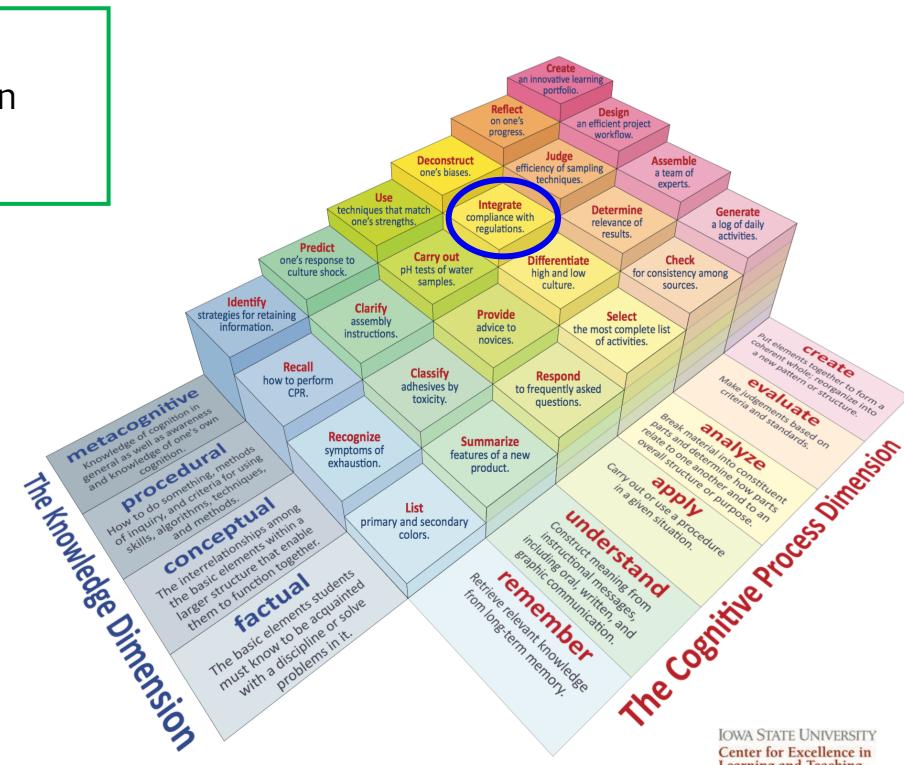
Course

Software Architecture

Learning Objectives

6.

Integrate architectural styles in order to design different types of software architectures.



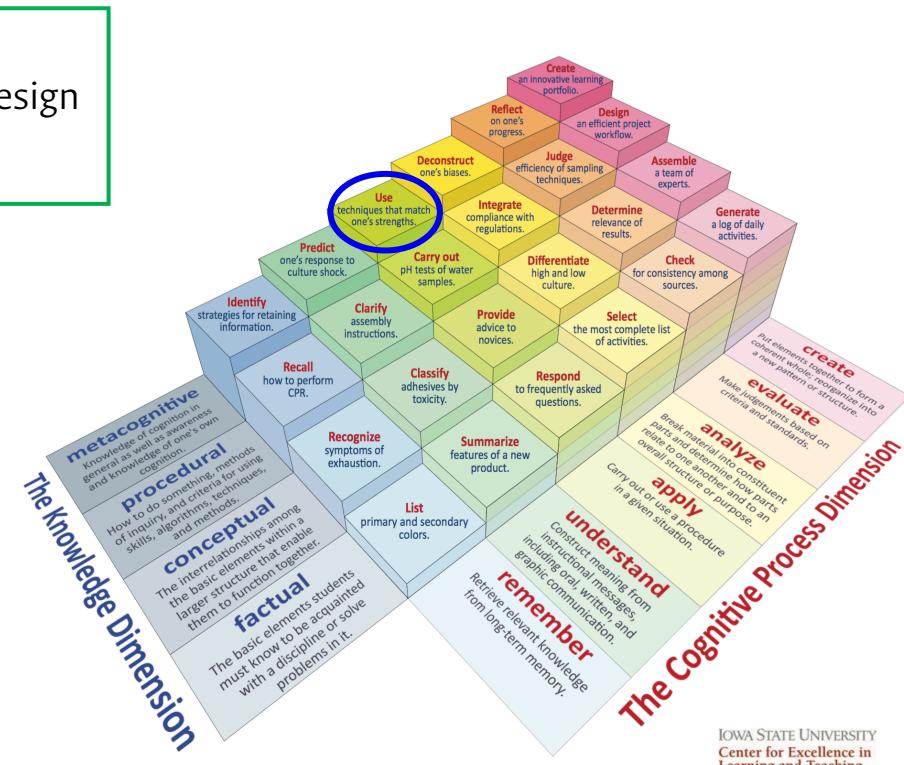
IOWA STATE UNIVERSITY
Center for Excellence in
Learning and Teaching

Course

Software Architecture

Learning Objectives

7.
Use architectural patterns to solve software design problems.



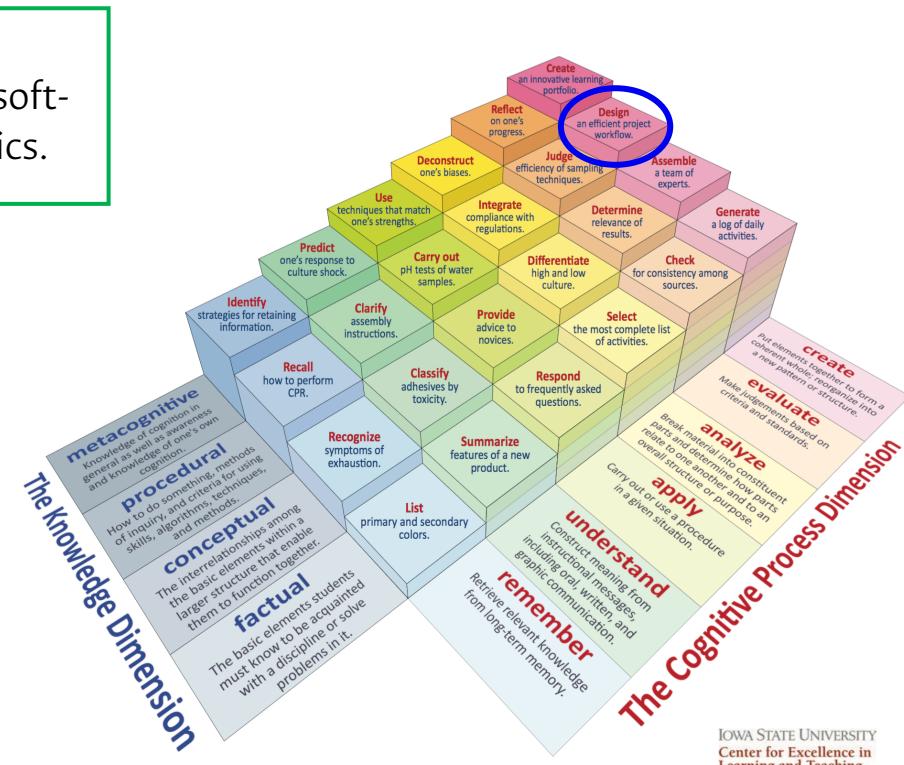
Course

Software Architecture

Learning Objectives

8.

Design strategies to ensure the quality of the software from different types of architectural tactics.



IOWA STATE UNIVERSITY
Center for Excellence in
Learning and Teaching

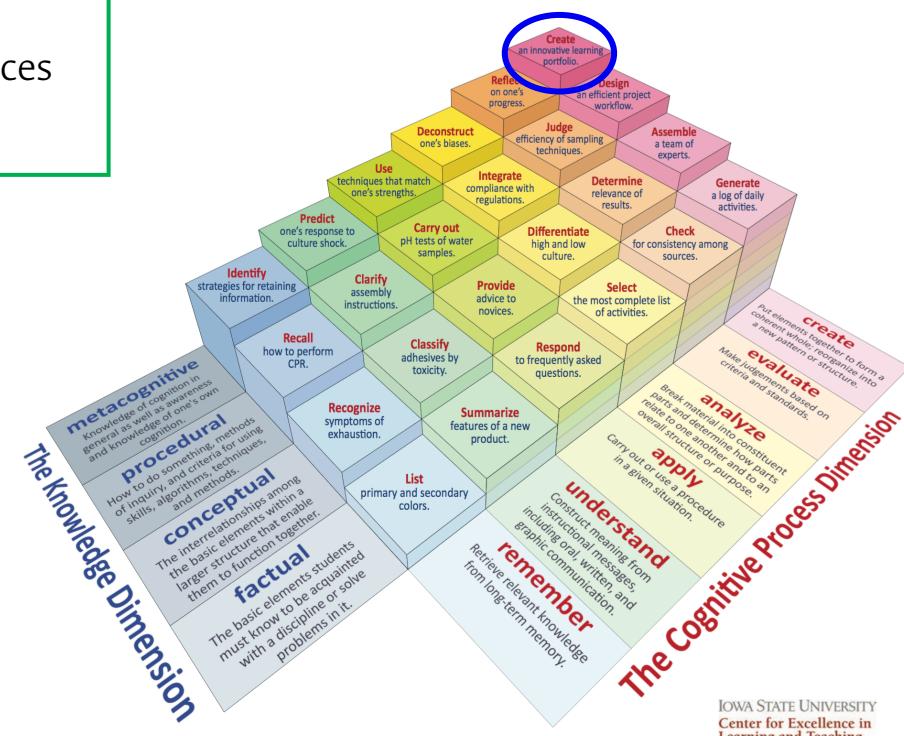
Course

Software Architecture

Learning Objectives

9.

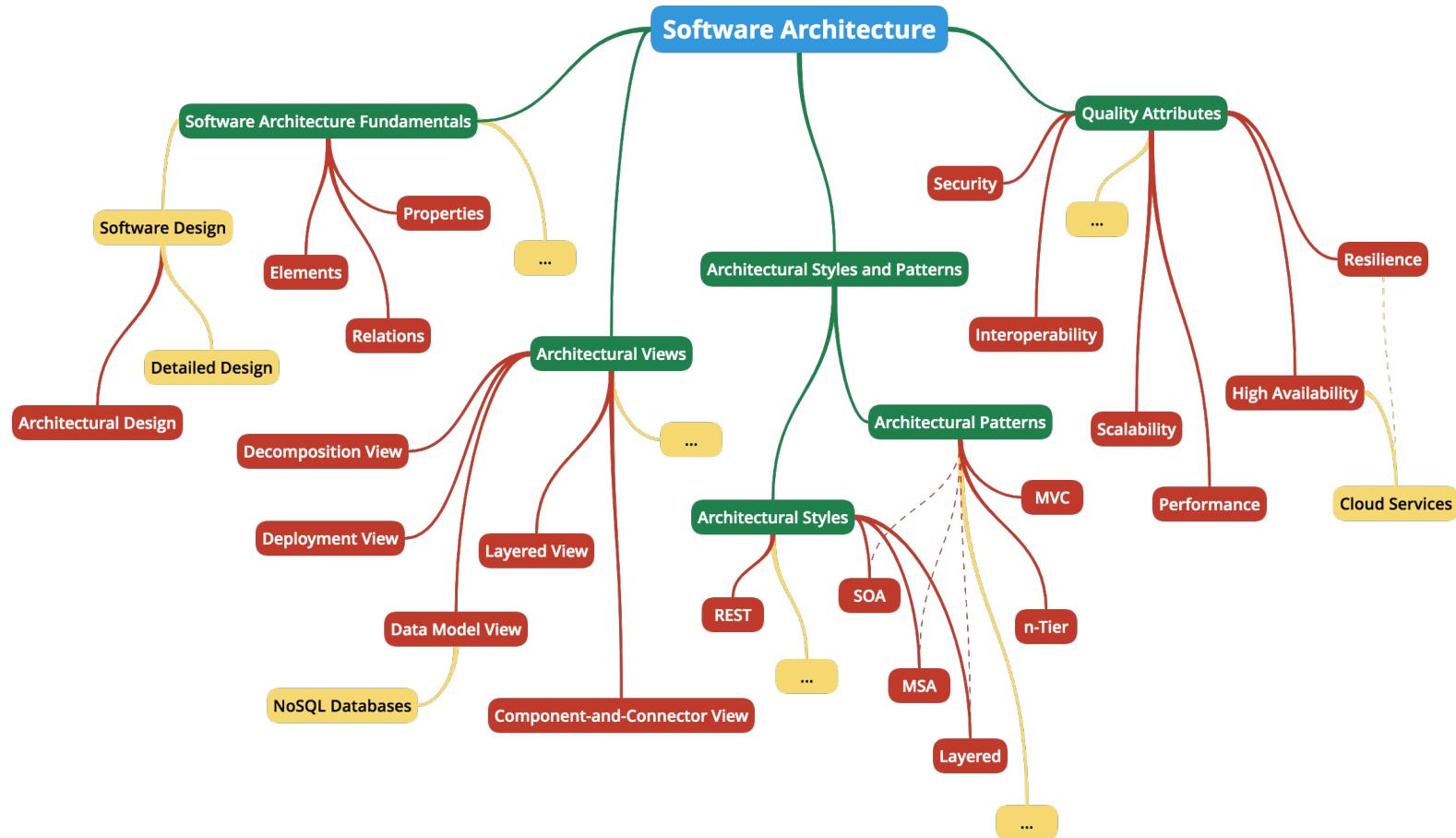
Create software systems based on good practices of software architecture.



Course

Software Architecture

Tree of Contents



Course

Software Architecture

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%

Course

Software Architecture

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.

Course

Software Architecture

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.