



# Systematic Mapping Study of Architectural Trends in Latin America

Diana Alomoto<sup>1(✉)</sup>, Andrés Carrera<sup>1(✉)</sup>, and Gustavo Navas<sup>1,2(✉)</sup>

<sup>1</sup> Universidad Politécnica Salesiana, Quito, Ecuador  
{dalomoto, acarreraj}@est.ups.edu.ec,  
gnavas@ups.edu.ec

<sup>2</sup> IDEIAGEOCA Research Group, Quito, Ecuador

**Abstract.** Software architecture has become an important aspect on systems development; however, despite the fact that there are several architectures that are already popular in the computer world, they do not seem to fully meet the demands of the developer due to the fact that system requirements are becoming increasingly stringent. This paper aims to carry out a systematic mapping study that allows publicizing the different architectures used in research accomplished in Latin America. For the study, numerous articles searches were executed in different bibliographic databases from acknowledged scientific journals, where several articles were obtained and through filters a total of 10 articles were obtained with which this project was worked on. Applying the systematic mapping process, a series of analyses were carried out on the selected scientific articles that allowed a detailed study of the investigations, which led to a varied collection of data that determined the focus of the studies when applying architecture to their systems. The final results of this work show that the scientific articles analyzed are focused on proposing new architectures based on existing ones.

**Keywords:** Software architecture · Systematic mapping study  
Latin America

## 1 Introduction

In recent years software architecture implementation has become an essential point for the construction of systems because it provides a solid foundation on which software can be built. Software architecture “plays a key role as a bridge between needs and execution. By providing an abstract description of a system, architecture exposes certain properties while hiding others” [1].

This work implements a Systematic Mapping, a method that does not aim to answer a specific question, but rather collects, describes and catalogues the available evidence related to a topic of interest [2]. There are several methods for research such as:

- Systematic Mapping Review: Provides an overview of a research area, and identifies the quantity, type of research and results available [3].

- Systematic Literature Review: A form of secondary study that uses a well-defined methodology to identify, analyze and interpret all available evidence related to a specific research question in a way that is unbiased and repeatable [4].
- Systematic Mapping Study: This is based on a broad review of primary studies in a specific topic that aims to identify what evidence is available on the topic [4].

## 2 Related Works

There are several studies that implement Systematic Mapping Study in different areas; however, SMS studies specifically in software architecture are limited. Below are some articles that study previously mentioned topics, but the context is different from that presented in this paper. It is worth stressing that none of these articles focuses on the study of architectural trends in a specific geographical space.

Li applies SMS to collect data from scientific articles related to software architecture and to identify in which types of architecture activities knowledge-based approaches are employed [5].

Liang implements SMS to evaluate articles that implement agile methodologies and software architecture in their studies. This article provides an extensive overview of several aspects such as practices, methods, benefits and costs of combining architecture and agile methodologies [6].

Tofan provides a systematic overview to determine the approach to architectural decisions, this document aims to analyze past work and future plans through studies published between 2002 and 2012 in different virtual libraries [7].

The articles mentioned above were used as a guide to analyze the application of SMS in each of the works, although the implementation of SMS in software architecture is different. Whilst the implementation of software architecture is mentioned in general, no specific type of architecture is named.

In this paper, the main objective of SMS's implementation is to present what types of software architectures are used by authors in articles implemented in Latin America. We could not find papers in which the location is a parameter of study; therefore it is not possible to make a comparison of specific results.

## 3 Research Methodology

This work is based on the systematic mapping process proposed by Paternoster [8] in 2014 as an adaptation of the diagram published by Petersen [9] in 2007.

According to the article published by Kai Petersen [9], Systematic Mapping Study has been recommended for studies related to software engineering. This methodology provides a process to obtain research reports and results through mapping of scientific papers [9]. Systematic Mapping Process presented by Paternoster [8] includes two more stages (rigor and relevance, synthesis) that were considered essential for the development of this research (Fig. 1).

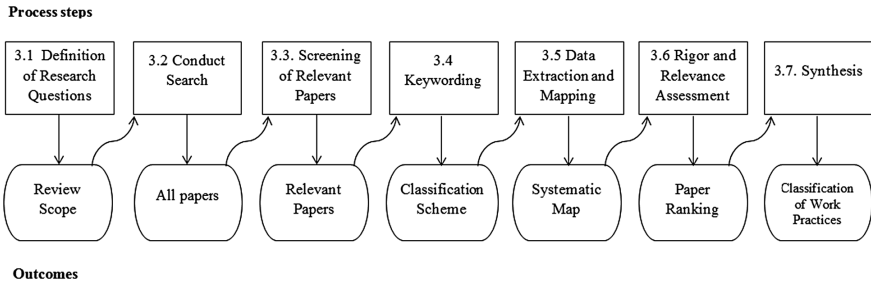


Fig. 1. Systematic Mapping Process [8]

### 3.1 Definition of Research Questions

Systematic Mapping Study (SMS) application aims to analyze the type of architectural methods used in software projects in Latin America. The following questions were asked in order to achieve this objective (Table 1).

Table 1. Definition of search questions

	Research questions	Motivation
RQ1.	What types of software architectures are used in Latin America?	Identify the software architectures that have been proposed, applied or studied
RQ2.	What is the main objective for the implementation of software architecture?	Analyze the main reason why the authors decided to apply software architecture to their systems
RQ3.	What is the most used architectural trend?	Determine which is the architectural trend used in the articles

### 3.2 Conduct Search

Keywords used for article search were based on different virtual libraries results; the search string established in this study was (“software architecture” AND “Latin America”) because it presented more precise results related to the research topic. In addition, the search string was modified according to the established structure of each virtual library.

The search strings (“computing architecture” AND “Latin America”) (“agile architecture”) AND “Latin America” were tested but the results were too large and vague.

In addition, when searching with these strings in IEEE Xplore virtual library, most of the articles found contained topics related only to “Latin America” but without any relation to the main search string “computing architecture” or “agile architecture”. For this reason, these strings were not considered although the results were higher than those of the other virtual libraries, they were inaccurate and inconsistent (Table 2).

This search string selection is due to the fact that there are not many academic articles that expose the topic of software architecture, unlike topics such as agile

architecture and computing architecture that have been analyzed and presented by several authors. In this study, the use of article search tools was not necessary because the number of documents was minimal and the selection of documents could be done manually.

**Table 2.** Keywords for searching in virtual libraries

Virtual library	Search string	Results
ACM Digital Library	("software architecture") AND "Latin America"	0
	("computing architecture") AND "Latin America"	2
	("agile architecture") AND "Latin America"	0
Scopus	("software architecture") AND "Latin America"	3
	("computing architecture" AND "Latin America")	1
	("agile architecture" AND "Latin America")	0
IEEE Xplore	("software architecture") AND "Latin America"	34
	("computing architecture") AND "Latin America"	5827
	("agile architecture") AND "Latin America"	43717
Web of Science	("software architecture") AND "Latin America"	4
	("computing architecture") AND "Latin America"	0
	("agile architecture") AND "Latin America"	0

### 3.3 Screening of Relevant Papers

From 41 articles found (Table 2), 10 articles contained topics related to software architecture in Latin America. Several aspects of inclusion and exclusion (Table 3) were used into account in order to choose the appropriate articles. Once the articles were selected, we made a classification by language: Spanish, English and Portuguese (Table 4). It should be noted that language was not a point of exclusion in this paper as well as publication’s year, however; it is worth mentioning that papers found were published between 2006 and 2017.

**Table 3.** Criteria for inclusion and exclusion of articles

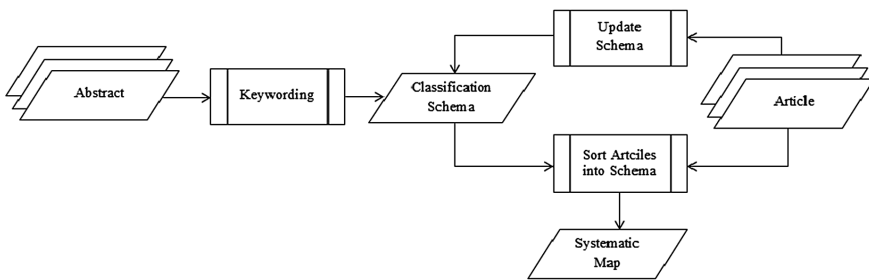
Inclusion	Exclusion
<ul style="list-style-type: none"> <li>- Scientific articles that contain topics related to software architecture and are applied in Latin America</li> <li>- Peer review</li> <li>- Publication’s year is not considered</li> </ul>	<ul style="list-style-type: none"> <li>- Repeated articles</li> <li>- Articles not related to the software architecture</li> <li>- Articles those are related to software architecture but not applicable in Latin America</li> <li>- Dark literature</li> </ul>

**Table 4.** Classification by language

Article title	Language
<b>P1:</b> Declarative and flexible modeling of software product line architectures [10]	Spanish
<b>P2:</b> Evaluation of software development through an MDA tool: a case study [11]	Spanish
<b>P3:</b> Incidence of software quality attributes in the design, construction and deployment of Cloud architectural environments [12]	Spanish
<b>P4:</b> Two-level software architecture for context-aware mobile distributed systems [13]	Spanish
<b>P5:</b> Towards a reference software architecture for improving the accessibility and usability of open course ware [14]	English
<b>P6:</b> Transpiler-based architecture for multi-platform web applications [15]	English
<b>P7:</b> Usage of social and semantic web technologies to design a searching architecture for software requirement artefacts [16]	English
<b>P8:</b> An architectural model for situation awareness in ubiquitous computing [17]	Portuguese
<b>P9:</b> Architectural requirements as basis to quality of software engineering environments [18]	Portuguese
<b>P10:</b> Using expert systems for investigating the impact of architectural anomalies on software reuse [19]	Portuguese

### 3.4 Keywording

This step is based on the classification scheme process (Fig. 2) used in Petersen’s [9] research, this process starts with abstracts reading and a filter to select keywords to be able to structure the classification scheme.



**Fig. 2.** Classification scheme steps [9]

The steps are detailed below:

- **Abstract**

From 10 articles selected a reading of each abstract was made, if it was related to the research it was included in the selected articles list. In some cases abstract was not comprehensible so was read the content of the article and if it was related to the topic of interest then it was included in the list, otherwise it was eliminated from the options. In this filter 10 articles were related to the research topic, for this reason none were discarded.

### – **Keywording**

The aim of this stage is to find words in common between all the articles. Ten scientific articles were classified by language and divided into three groups according to the language of each article. (Table 4).

With ATLAS TI tool was made a ‘word cloud’ of each group of articles previously classified by languages getting a number of words for each group (Table 5).

**Table 5.** Number of words per language

Language	Number of words
Spanish	3519
English	2641
Portuguese	2513

In this step three filters were made and will be explained below:

- *First filter*

From 8673 words found among all the articles, was performed an elimination filter by articles, pronouns and words without relevance to the research topic. In Spanish language group, 4 documents were evaluated, so words with fewer than 4 repetitions in the ‘word cloud’ list were removed from this filter because the word was less likely to have at least one repetition in each article.

Groups of documents with Portuguese and English language, words with less than 3 repetitions were eliminated due to the fact that 3 documents were evaluated in each of the groups. At the end of this filter was obtained 3 lists corresponding to the groups established above.

- *Second filter*

A base list of common words related to software architecture was made with the Spanish language word group, as this group contains more documents; this list was compared with the other documents of the corresponding language groups.

In total, 51 words were left that were considered the most optimal. It should be noted that these words were not consistent in all the articles.

- *Third filter*

This filter consisted on discarding words that were not coincidental in all the articles. The result is a list of 10 words that are repeated at least once in each article, including the number of repetitions of each word in each of the articles classified by language.

When performing these filters the final list of words represents the keywords among all the documents evaluated. Table 6 shows the words that had been sorted according to the order of appearance in all the articles, with the word ‘software’ being the top of the list and the most frequently repeated in all documents.

**Table 6.** Word Filtering

WORDS	Spanish					Portuguese				English				TOTAL
	P 1	P 2	P 3	P 4	T.C	P 1	P 2	P 3	T.C	P 1	P 2	P 3	T.C	
Software	50	30	87	14	181	2	116	49	167	37	32	62	131	479
Architecture	3	15	15	33	66	13	55	7	75	22	25	22	69	210
Development	4	59	11	6	80	1	34	1	36	2	13	23	38	154
Systems	1	21	5	14	41	1	40	32	73	2	9	5	16	130
System	17	36	16	16	85	2	7	10	19	2	1	12	15	119
Engineering	15	1	7	3	26	4	27	8	39	5	2	37	44	109
Components	14	15	8	5	42	2	5	39	46	2	5	6	13	101
User	1	12	4	12	29	13	6	11	30	3	1	30	34	93
Design	3	9	18	2	32	6	16	8	30	7	13	4	24	86
Results	2	8	2	1	13	5	4	5	14	2	11	16	29	56

**– Classification schema**

Because of the matching words in all the articles were only 10. Classification scheme consisted in some cases on joining words to create sentences and in others only words that were considered important were chosen and searched in all the articles to find matches.

**Table 7.** Classification scheme

SPANISH	P1	Abstraction level	Internal variability	Reduction of development time	Software Product Line	Visual declarative Language FVS	
	P2	MDA	Interoperability	Reduction of development time	Portability	Usability	
	P3	ISO,IEC 12207	Product Quality	Software Life Cycle	Cloud Computing	Reusability	Reference Architecture
	P4	Layered Architecture	Abstraction level	Mobile Devices	Distributed Mobile Systems	Architecture CC (Composite Capability)	
ENGLISH	P5	Reference Architecture	Usability	Accessibility	Software Product Line	Reusability	Interoperability
	P6	Reduction of development time	Layered Architecture	MVC Architecture	Cloud Instances	Web environments	
	P7	Semantic Web	End user	Layered Architecture	Web environments	Re-use of products	
PORTUGUESE	P8	Usability	Ubiquitous Computing	Interoperability	Mobile Devices	Architecture for 'situation awareness'	
	P9	ISO,IEC 12207	Client/server architecture	Reference Architecture	Product quality	MVC Architecture	
	P10	Reuse of Architectural components	End user	Software Life Cycle	Architectural evolutionary systems	Detection of architectural anomalies	

Table 7 shows the phrases with matches that were found, as well as key phrases for each of the items that do not necessarily have repetitions.

In some cases the written word was not found explicitly but the content referred to one of the phrases, so they were included in the table.

### 3.5 Data Extraction and Mapping

Table 7 was considered as the basis for various classifications and data extraction. Three types of classifications were made, which will be detailed as follows:

- **Classification by type**

- *Platforms*: OCW, Web Environments.
- *Architectures*: MDA, Reference Architecture, Layered Architecture, CC Architecture, MVC Architecture, Client/Server Architecture and Architecture for ‘Situation Awareness’.
- *Characteristics*: Product quality, Architectural anomalies, Reuse of products, Reuse of architectural components, Abstraction level and internal variability.
- *Tools*: Mobile device.
- *Attributes*: Development time reduction, Reusability, Interoperability, Usability, Portability and Accessibility.
- *Systems*: Software product line, Architectural evolutionary systems, Mobile distributed systems and Ubiquitous computing.
- *Standard*: ISO/IEC/IEEE 12207.
- *Processes*: Software lifecycle, Semantic Web.
- *Languages*: Visual declarative language FVS.
- *Technology*: Cloud Computing, Cloud Instances.

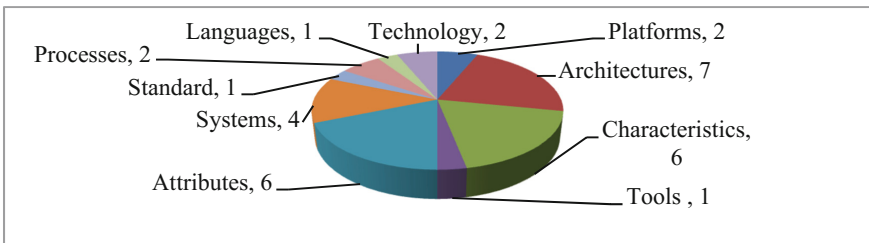


Fig. 3. Classification by type

The types chosen group certain general characteristics that in some way encompass their uses and operations, as can be seen in the illustration (Fig. 3) each of the types has specified the number of elements it contains.

The *platforms* group together web environments and Open Course Ware that are mentioned in certain articles.



There are *architectures* that are used and some articles may even use several architectures at once, but for the case of the classification scheme the most important ones have been taken from each article and listed.

*Characteristics and attributes* refer to the expectations that the research aims to achieve, final products or improvements that are managed and tried to obtain, in addition to correcting the inconveniences that appear, different elements that contribute to the objectives of the articles were grouped in tools and systems.

One *standard* that may have been found relevant to research is ISO/IEC/IEEE 12207 which refers to the life cycle processes of the ISO organization's software.

In the *language* grouping we could find the declarative visual language FVS on which one of the scientific articles studied is based.

The *technology* that could be observed with more relevance is Cloud Computing which is mentioned in some articles.

Finally, a general grouping of the stages, periods and organized works are the *processes* where software and semantic web life cycles are found.

- **Classification by content**

We made a classification of the terms found by sections of the scientific articles. It was divided into 8 sections explained below:

- Abstract/Objectives
- Introduction/Development: Two subcategories are derived from this section.
  - Applies
  - Research/Testing/Assessments
- Conclusions/Results
  - Based on
  - Proposes
- Related works/Future works

The details observed reveal that categories chosen are based on the parts of a scientific article with the exception of the subcategories listed which were put together to give greater meaning to the classification.

The objective of this classification is search each of the phrases for every article and locate according to the category, if the phrase or word was repeated then it was placed in the category with more repetitions, meaning that a word or phrase could be repeated in several sections of the article however it was placed in the category considered most relevant.

It is necessary to mention that not all articles have the same writing formats with exception of (Abstract, Introduction, and Conclusions) which are parts of an article. For this reason we have unified sections of the articles that we consider to be similar. As can be seen from the graph (Fig. 4), the category with the highest number of words or phrases is "Abstract/Objectives" followed by "Based on".

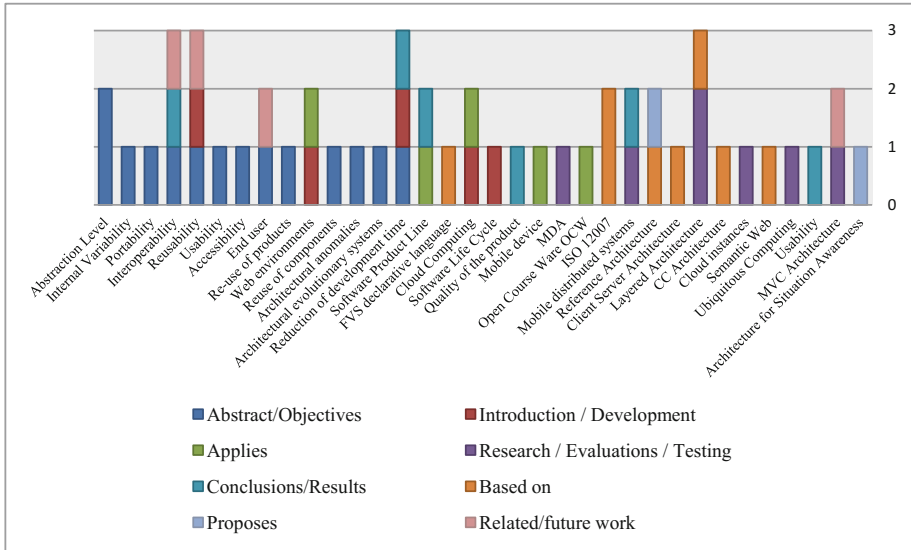


Fig. 4. Classification by content

• **Classification by architectures**

In this classification the main architectures used in each scientific article were searched, although several articles used different architectures. This classification is only based on the most relevant architectures of each article (Table 8). It was divided into 3 groups (Use, Study and Propose).

Table 8. Classification by architectures

P1	Software architecture product line	Use
P2	MDA (Model Driven Architecture)	Study
P3	Reference architecture	Study
P4	Two-Tier architecture in layers	Propose
P5	Reference architecture	Use
P6	Transpiler-based architecture	Propose
P7	Search architecture	Propose
P8	Architecture for ‘situation awareness’	Propose
P9	Reference architecture	Propose
P10	ARCHIDES tool architecture	Use

As illustrated in the graph (Fig. 5), 50% of the articles seek to propose their own architecture to provide solutions to various problems, followed by 30% that use architectures that are already known in software development and finally 20% study architectures to evaluate either to develop quality models or to test the selected architecture.

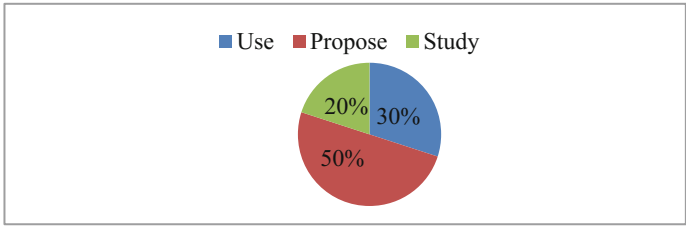


Fig. 5. Classification by architectures

### 3.6 Rigor and Relevance

Scientific rigor refers to the accuracy of the research approach used, whilst industrial relevance refers to the realism of the research used in the article and if it is representative for professionals.

Table 9 shows a list of criteria with their respective rating headings, these criteria were proposed and described by Ivarsson and Gorschek [20], which were extracted for implementation in this paper.

Table 9. Evaluation criteria for rigor and relevance [20]

Rigor	Relevance
<ul style="list-style-type: none"> <li>• Context described</li> <li>• Study design described</li> <li>• Validity discussed</li> </ul>	<ul style="list-style-type: none"> <li>• Subjects</li> <li>• Contexts</li> <li>• Scale</li> <li>• Research method</li> </ul>
<i>Qualification</i> - Strong description (1) - Medium description (0.5) - Weak description (0)	<i>Qualification</i> - Contribute to relevance (1) - Do not contribute to relevance (0)

After the criteria of rigor and relevance were established, Tables 10 and 11 shows an individual evaluation of each of the scientific articles studied, according to the weighting described previously.

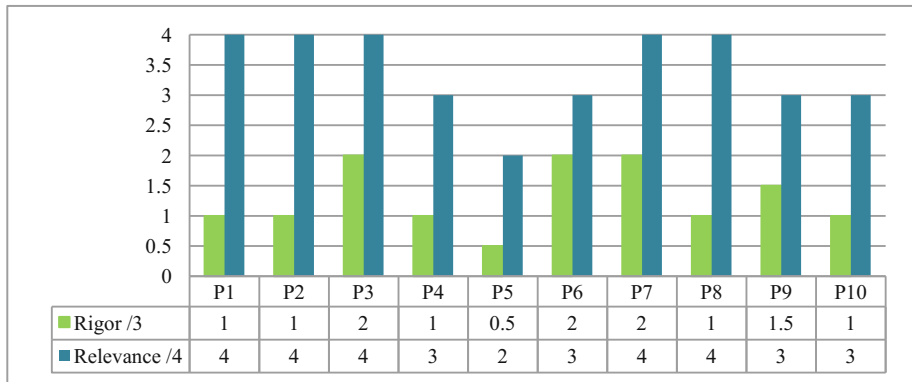
Table 10. Rigor assessment

RIGOR										
Criteria	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Context described	1	1	1	0.5	0.5	1	1	0.5	0.5	1
Study design described	0	0	1	0.5	0	0.5	1	0.5	1	0
Validity discussed	0	0	0	0	0	0.5	0	0	0	0

**Table 11.** Relevance assessment

RELEVANCE										
Criteria	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Subjects	1	1	1	1	1	1	1	1	1	1
Context	1	1	1	1	0	1	1	1	1	1
Scale	1	1	1	0	0	0	1	1	0	0
Research method	1	1	1	1	1	1	1	1	1	1

Illustration (Fig. 6) compares the final results acquired by the evaluation, making a total sum of partial results for each article. Rigor results are lower compared to those of relevance. According to these results it can be determined that the articles studied have a greater focus on the realism of the research method; however, there is a lack of precision in the applied research approach.

**Fig. 6.** Comparison of results between rigor and relevance

### 3.7 Synthesis

According to the data collected and the results analyzed, we answer to the first research questions that motivated this work.

#### – RQ1: What types of software architectures are used in Latin America?

According to Table 7 presented as our classification scheme; the architectures that were most used in the scientific articles are: MVC Architecture, Reference Architecture, Layered Architecture, MDA Architecture, and Client/Server Architecture.

Architectures previously mentioned were used in the articles either to use, study, or as a basis for proposing another type of architecture.

It is worth mentioning that the architectures presented were not the only ones mentioned in the articles as shown in Table 8, however, architectures mentioned

previously have acquired more relevance in their respective studies, besides, are already known in software development.

– **RQ2: What is the main objective for the implementation of software architecture?**

Illustration (Fig. 3) shows a classification by type in which there are two groups that present the greatest number of elements, these groups are:

- *Attributes*: in this group there are elements that stand out as quality attributes, among which are evident: reduction of development time, reusability, interoperability, usability, portability, accessibility.
- *Characteristics*: this group refers to elements that were mentioned in the articles such as: product quality, architectural anomalies, product reuse, reuse of architectural components, level of abstraction, internal variability.

A study of the two previously presented groups determined that the scientific articles focus on the quality attributes because the elements that appear in this group are the most mentioned in the articles and the ones that have more repetitions in the studies as shown in Table 7.

Therefore, according to the results obtained, the main reason why authors implement software architecture in their studies is to develop, implement or study a system that complies with one of these qualities. The attributes are mentioned in 6 of the 10 articles studied.

– **RQ3: What is the most used architectural trend?**

‘Layered Architecture’ is the most commonly used in scientific articles according to Illustration 4 to be researched, evaluated, and tested or as a basis for proposing a new architecture.

‘Reference Architecture’ is also in the higher range of architectures as show in illustration (Fig. 4), however, in one of the articles it is only briefly mentioned for this reason it was not considered as one of the most relevant architectures used.

Therefore, it is evident that the architectural trend most used in the scientific articles developed is the ‘Layered architecture’, being considered as the preferred architecture for researches in Latin America.

## 4 Conclusions

Software architecture has become a focal point for carrying out work concerning the development or study of computer systems that has had a growing impact in recent years in Latin America. As it was demonstrated in this paper there are a great variety of software architectures that have been applied, studied or proposed for different projects, however, according to the data collection 50% of the articles propose software architectures while 30% use and 20% study architectures already known in the technological field.

'Layered Architecture' was determined as a trend from the results obtained, in the articles that study or use this architecture, abstraction level and reduction in development time stand out as common characteristics.

Systematic study carried out served as a starting point for a more specific understanding of the software architectures used in Latin America, the different preferences, the most used and the most stable ones for users, so that we can statistically agree on the trends that could be related to future work.

The results of the study carried out in this work reflect the scarcity of articles referring to software architecture that have been published in Latin America, as can be seen only 10 articles were found which are divided into three different languages this shows that the number of articles that were studied is reduced compared to other studies that implement SMS.

## References

1. Garlan, D.: Software architecture: a travelogue. In: Proceedings of the on Future of Software Engineering - FOSE 2014 (2014). <https://doi.org/10.1145/2593882.2593886>
2. James, K., Randall, N., Haddaway, N.: A methodology for systematic mapping in environmental sciences. *Environ. Evidence*, **5**(1) (2016). <https://doi.org/10.1186/s13750-016-0059-6>
3. Marques Samary, M.: Charla Systematic Review. <https://users.dcc.uchile.cl/>
4. Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in Software Engineering. (2007)
5. Li, Z., Liang, P., Avgeriou, P.: Application of knowledge-based approaches in software architecture: a systematic mapping study. *Inf. Software Technol.* (2013). <https://doi.org/10.1016/j.infsof.2012.11.005>
6. Tofan, D., Galster, M., Avgeriou, P., Schuitema, W.: Past and future of software architectural decisions – a systematic mapping study. *Inf. Software Technol.* **56** (2014). <https://doi.org/10.1016/j.infsof.2014.03.009>
7. Yang, C., Liang, P., Avgeriou, P.: A systematic mapping study on the combination of software architecture and agile development. *J. Syst. Software* **111** (2016). <https://doi.org/10.1016/j.jss.2015.09.028>
8. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T. and Abrahamsson, P.: Software development in startup companies: a systematic mapping study. *Inf. Software Technol.* **56**(10) (2014). <https://doi.org/10.1016/j.infsof.2014.04.014>
9. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic Mapping Studies in Software Engineering (2007)
10. Asteasuain, F., Perez Vultaggio, L.: Declarative and flexible modeling of software product line architectures. *IEEE Latin Am. Trans.* **14** (2016). <https://doi.org/10.1109/tla.2016.7437236>
11. Duarte, J., Tonanez, M., Cernuzzi, L., Lopez, O.: Evaluation of software development through an MDA tool: a case study. *IEEE Latin Am. Trans.* **6** (2008). <https://doi.org/10.1109/tla.2008.4653855>
12. Cabrera, S., Abad, E., Danilo Jaramillo, H., Poma, G., Verdum, J.: Incidence of software quality attributes in the design, construction and deployment of Cloud architectural environments. In: 2015 10th Iberian Conference on Information Systems and Technologies (CISTI) (2015). <https://doi.org/10.1109/cisti.2015.7170460>

13. Medina Acosta, M., Lopez Dominguez, E., Gomez Castro, G., Pomares Hernandez, S., Medina Nieto, M.: Two-level software architecture for context-aware mobile distributed systems. *IEEE Latin Am. Trans.* **13**(4) (2015). <https://doi.org/10.1109/tla.2015.7106376>
14. Morales, G., Benedí, J.: Towards a reference software architecture for improving the accessibility and usability of open course ware. In: Proceedings of the 11th European Conference on Software Architecture Companion Proceedings - ECSA 2017 (2017). <https://doi.org/10.1145/3129790.3129796>
15. Andres, B., Perez, M.: Transpiler-based architecture for multi-platform web applications. In: 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM) (2017). <https://doi.org/10.1109/etcm.2017.8247456>
16. Chicaiza, J., López, J., Piedra, N., Martínez, O., Tovar, E.: Usage of social and semantic web technologies to design a searching architecture for software requirement artefacts. *IET Software* **4**(6) (2010). <https://doi.org/10.1049/iet-sen.2010.0046>
17. Lopes, J., Souza, R., Gadotti, G., Pernas, A., Yamin, A., Geyer, C.: An architectural model for situation awareness in ubiquitous computing. *IEEE Latin Am. Trans.* **12**(6) (2014). <https://doi.org/10.1109/tla.2014.6894008>
18. Nakagawa, E., Maldonado, J.: Architectural requirements as basis to quality of software engineering environments. *IEEE Latin Am. Trans.* **6**(3) (2008). <https://doi.org/10.1109/tla.2008.4653856>
19. Jullian Oliveira do Nascimento, R., Andre Guerra Fonseca, C., Dantas de Medeiros Neto, F.: Using Expert Systems for Investigating The Impact of Architectural Anomalies on Software Reuse. *IEEE Latin Am. Trans.* **15**(2) (2017). <https://doi.org/10.1109/tla.2017.7854635>
20. Ivarsson, M., Gorschek, T.: A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Eng.* **16**(3) (2010). <https://doi.org/10.1007/s10664-010-9146-4>