



SYSTEM SOFTWARE ENGINEERING

PAST, PRESENT AND FUTURE

A journey through the technological evolution that is transforming our world

Agenda de Presentación

01

The Foundations of the Past

Software crisis and first structured solutions

We will explore how software engineering has evolved from its humble beginnings to become the transforming force of the digital age.

02

The Reality of the Present

Agile methodologies and cloud computing

03

The Vision of the Future

Artificial intelligence and quantum computing

The Beginnings: 1960s–1970s

The Era of Limitations

In the early days, software engineering was completely dominated by hardware limitations. Programmers worked directly with:

- Low-level assembly language
- Machine-specific code
- Extremely limited memory resources
- Basic sequential processing

Software was expensive to develop, difficult to maintain, and tightly bound to specific machines.



The Software Crisis

Delayed Projects

Most projects were delivered months or even years later than planned.

Overrun Budgets

Actual costs dramatically exceeded the initial estimates.

Defective Software

Systems were plagued with errors and critical failures.

This crisis of the late 1960s demonstrated that structured methods and formal disciplines were necessary to create reliable and efficient software.



Present: A Mature Discipline

01 10 Advanced Languages

Python, Java, C++, JavaScript, and modern frameworks that simplify development.

Agile Methodologies

Scrum, Kanban, and DevOps for iterative development and continuous delivery.

Cloud Computing

AWS, Azure, and Google Cloud enable instant global scalability.



Beyond the Code

A Holistic Approach

Modern software engineering goes far beyond just writing code:

- Project management with agile methodologies
- Automated testing and quality assurance
- Cybersecurity integrated from the design stage
- User experience as a top priority
- Distributed systems and microservices



Today's systems support millions of simultaneous users with high availability and optimized performance.

The Future: Exponential Innovation



Artificial Intelligence

AI and Machine Learning will completely transform software design, development, and testing.



Quantum Computing

It will revolutionize algorithms and performance in a fundamental way.



Advanced Cybersecurity

Critical protection for increasingly interconnected systems.



The New Role of the Engineer

Automation

Development of systems that self-optimize and self-repair.

Ethical Responsibility

Consideration of the social and moral impact of technology.

Sustainability

Green technologies and energy efficiency as priorities.

The engineers of the future will create software that is not only powerful, but also intelligent, secure, and responsible toward both the environment and society.



Final Reflections

1 Past

Basic foundations and solving reliability problems.

2 Present

A mature discipline with efficiency, teamwork, and adaptability.

3 Future

Continuous innovation with AI, automation, and responsibility.

Thank you for your attention!

Software engineering will continue to evolve to create a more connected, intelligent, and sustainable world.

