Financial Technology

Academic Year: (2023 / 2024) **Review date: 27-04-2023**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: GARCIA OLAYA, ANGEL

Type: Compulsory ECTS Credits : 6.0

Year : 1 Semester : 1

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Basic knowledge of Java Programming

OBJECTIVES

- Understanding and mastering the knowledge basis needed to develop and/or apply ideas, often in a research context
- Application of theoretical knowledge and problem solving abilities to practical and new problems in a broader context related to the corresponding study area.
- Ability to integrate knowledge and handle the complexity of decision making with incomplete or limited information, while considering the ethic and social implications of the decisions taken.
- Ability to clearly communicate and support the conclusions both to specialized and general audiences.
- Long-life autonomous learning abilities.
- Ability to apply the different methods and techniques of the Computer Science Area to the financial markets.
- Ability to conceive, design, develop, implement and deploy software for financial markets.
- Ability to write original documents, motivated arguments and work plans or work projects.
- Ability to analyze and evaluate the most common Information Society Technologies currently applied to finances.
- Development of financial software, from the analysis phase to its implementation and integration with other systems.
- Implementation of financial markets techniques following the corresponding standards and procedures.
- Analysis and understanding of the most common tools for big data management, including storage, retrieval and update.

Results of the learning process:
- Understanding the use of Information Systems in Financial Markets
- Knowledge of the most used technological standards.
- Understanding the most used technologies in finances.
- Acquiring a general vision on the most important finalist products.
- Ability to analyze the technological infrastructures for Financial Markets.
- Understanding the main application examples.
- Understanding the infrastructure requirements for the deployment of financial information systems

DESCRIPTION OF CONTENTS: PROGRAMME

1- Market latency and latency metrics.
2- Low latency buses
3- Last-mile messaging
4- Memory grids
5- Messaging and data exchange protocols
6- Serialization of messages
7- Market protocols
8- CEP
9- Cloud
10- Blockchain
11- Trading systems

LEARNING ACTIVITIES AND METHODOLOGY

Lectures: supported by digital materials (100% classroom)
Practical lectures: combination of lectures and exercises (100% classroom)
Mentoring: both at the office or on-line (video-conference) (100% classroom)
E-learning: Forum, videos and other (0% classroom)
Student's individual work: autonomous work performed by the student, including exam preparation (0% classroom)

Teaching methodologies:
- Theoretical lectures, supported with audiovisual material, to develop the main concepts of the subject and provide the students with the required bibliography to guide their self-study.
- Reading of suggested texts (newspaper articles, reports, operation manuals and/or research papers) to be discussed at class or to consolidate/extend the acquired knowledge.
- Solving practical cases, problems, etc. both individually or in groups.
- Oral presentations and discussions in class under teacher moderation.
- Development of reports or written works, both individually or in groups.
- E-learning activities, related to the blended nature of the degree.

ASSESSMENT SYSTEM

Participation during classes and forums: Up to an extra 10%
Practical exercises: 30%
Self-evaluation tests after each topic: 10%
Final exam: 60%

% end-of-term-examination:	60
% of continuous assessment (assigments, laboratory, practicals…):	40

BASIC ELECTRONIC RESOURCES

- . ZeroMQ: http://zeromq.org/
- . Aeron : https://www.youtube.com/watch?v=tM4YskS94b0
- . Solace: http://www.solacesystems.com/
- . UM: https://www.informatica.com/products/data-integration/real-time-integration/ultra-messaging.html
- . Hazelcast: https://www.javacodegeeks.com/2013/11/getting-started-with-hazelcast.html
- . Coherence: https://docs.oracle.com/middleware/1212/coherence/COHTU.pdf
- . Universal Messaging: http://terracotta.org/products/universal-messaging
- . Athmosphere: https://github.com/Atmosphere/atmosphere/wiki
- . Kaazing: https://kaazing.com/
- . Kryo : https://github.com/EsotericSoftware/kryo
- . Protocol Buffers: https://developers.google.com/protocol-buffers/
- . JSON: https://github.com/FasterXML/jackson
- . XML : https://jaxb.java.net/
- . Disruptors: https://lmax-exchange.github.io/disruptor/
- . Latency measures: http://stuff-gil-says.blogspot.com.es
- . Hardware: http://mechanical-sympathy.blogspot.com.es/