Professional Master's Degree
Parallel and Distributed Computing

tech global university

Professional Master's Degree
Parallel and Distributed Computing

- Modality: **online**
- Duration: **12 months**
- Certificate: **TECH Global University**
- Credits: **60 ECTS**
- Schedule: **at your own pace**
- Exams: **online**

Website: www.techtitute.com/us/information-technology/professional-master-degree/master-parallel-distributed-computing

Index

01

Introduction

Today, most software and electronic systems use Parallel or Distributed Computing. Over the years, smartphones have integrated more powerful multi-core processors. Nowadays, modern computers also incorporate multi-core processors. On the other hand, Distributed Computing has not only played a significant role in boosting other branches of research, such as big data, but it has also become indispensable in various areas, including social networks, corporate environments, and multiplayer online games. All this emphasizes the significance of these two programming approaches, prompting TECH to create a comprehensive professional master´s degree. This program is designed to equip computer scientists with a deep understanding of the advantages and primary applications of Parallel and Distributed Computing.

Give your career and resume a quality boost by incorporating the latest knowledge in Parallel and Distributed Computing into your work"

A strong advanced knowledge of Parallel and Distributed Computing can significantly boost the career path of any computer scientist who wants to be distinguished. Given the complexity and diverse range of applications in Parallel and Distributed Computing, TECH has entrusted a team of experts to prepare all the program's content.

Therefore, computer scientists will encounter topics dedicated to communication and coordination in computer systems, the analysis and programming of parallel algorithms, and distributed systems in computing, among other pertinent subjects. This content is written with a modern and innovative perspective, leveraging the accumulated experience of the teaching staff.

Thus, computer scientists who complete this program have a decisive advantage in projecting their careers towards the development of applications or systems in the fields of: climate, health, big data, cloud computing or blockchain. Furthermore, given the advanced nature of the syllabus, students have the opportunity to pursue a research project in the field of computer science or other related areas.

Moreover, the program is offered entirely online, eliminating the need for physical attendance in classes and removing the constraints of a predetermined schedule. Computer scientists have the freedom to distribute the course load according to their individual interests, enabling them to strike a balance between studying for their Master's degree and managing their personal or professional responsibilities.

This **Professional Master's Degree in Parallel and Distributed Computing** contains the most complete and up-to-date program on the market. The most important features include:

- The program includes the development of practical cases presented by experts in Parallel and Distributed Computing.The contents are created with graphics, schematics, and practical examples, providing relevant information on the disciplines that are essential for professional practice
- The program includes practical exercises that allow the self-assessment and facilitate the learning process
- The program places a special emphasis on innovative methodologies
- The program incorporates a combination of theoretical lessons, expert-led discussions, and individual reflection work
- The program offers content that is accessible from any fixed or portable device with an Internet connection

Enroll now to explore the latest developments in Parallel Computing in cloud environments and Distributed Computing-oriented programming”

You will receive guidance throughout the program from the teaching team, consisting of professionals with extensive experience in Parallel and Distributed Computing"

As a student, you will receive comprehensive support from TECH, the world's largest online academic institution. You will have access to the latest educational technology.

Do not miss the opportunity to be distinguished and able to demonstrate your passion for the present and future of IT.

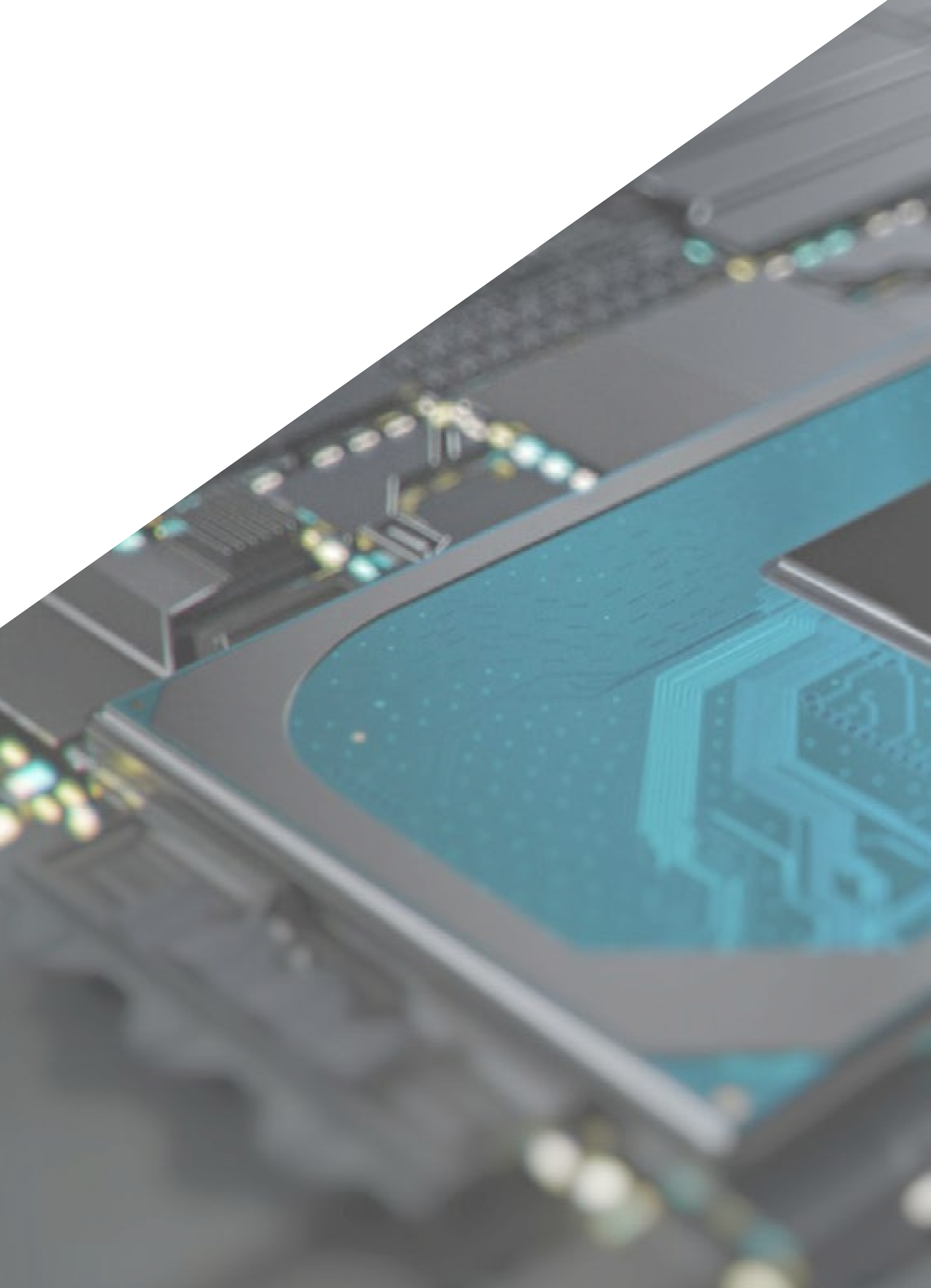
The program's teaching staff comprises professionals from the sector who bring their valuable work experience to the program. In addition, renowned specialists from leading societies and prestigious universities are also part of the teaching team.

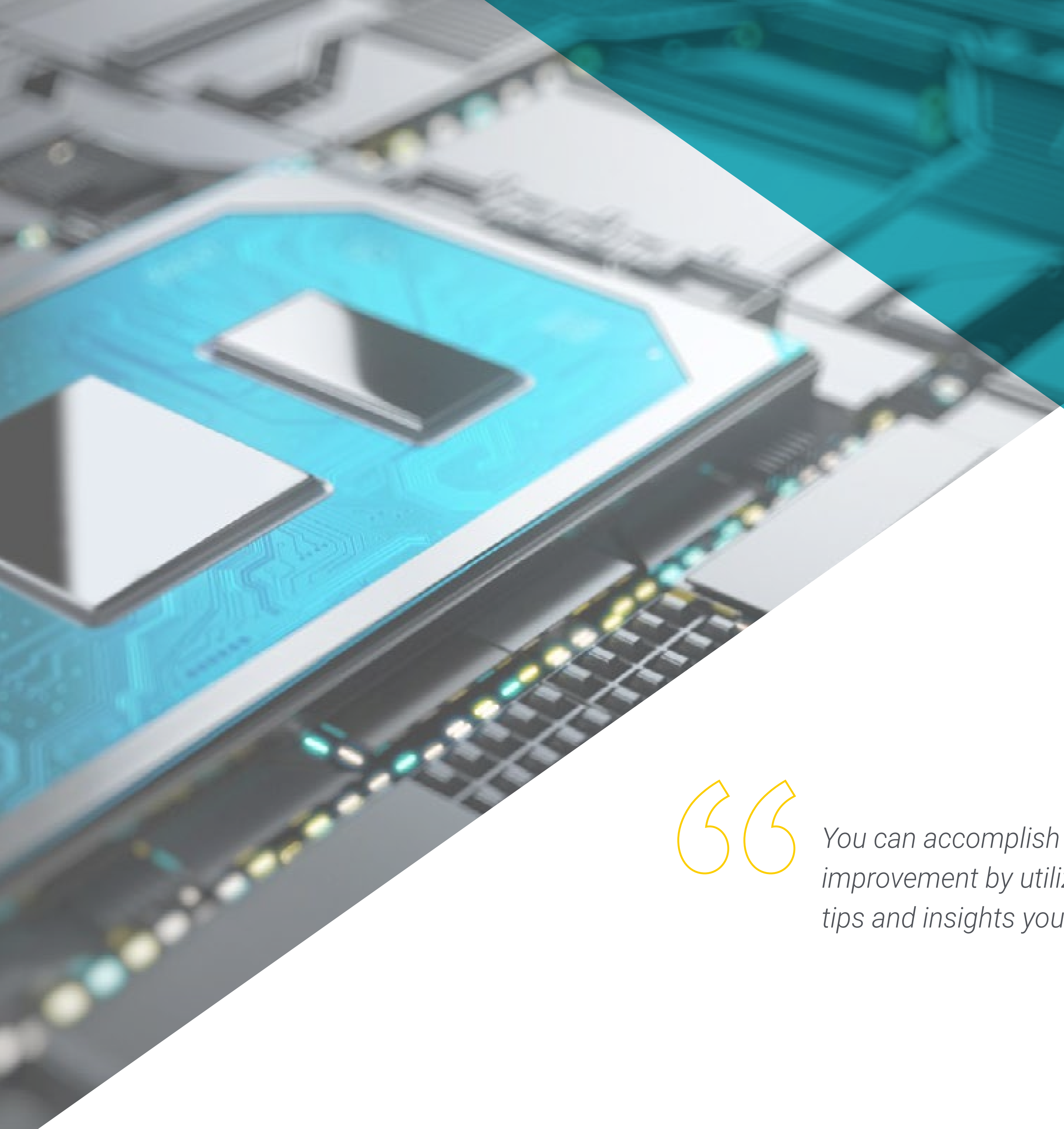
The program offers multimedia content developed using the latest educational technology. This content provides professionals with a contextual and situated learning environment. Through simulated environments, professionals can engage in immersive education that prepares them for real-life situations.

The design of this program focuses on Problem-Based Learning, in which professionals must try to solve the different professional practice situations that are introduced to them throughout the academic year.To facilitate the learning process, students will be supported by an innovative interactive video system developed by renowned and experienced experts.

02
Objectives

Acknowledging the rapid advancement of computing, professionals in the field must continuously make efforts to stay updated. In recognition of this need, TECH has developed this program with a specific focus on the latest advancements in Parallel and Distributed Computing. By enrolling in this program, students will not only gain advanced skills in Parallel and Distributed Computing, but they will also explore the diverse applications of today's technologies, including blockchain and cloud computing.





You can accomplish your goal of professional improvement by utilizing the valuable computer tips and insights you will gain from this program"

General Objectives

- Analyze the difference between the different components of Parallel and Distributed Computing
- Measure and compare the efficiency of various components utilized in the system to analyze their performance
- Conduct a comprehensive analysis of Multiplatform Parallel Computing to leverage task-level parallelism among different hardware accelerators
- Analyze in detail current software and architectures
- Delve deeply into the most significant aspects of Parallel and Distributed Computing, expanding your understanding and expertise in the field
- Specialize the students in the application of Parallel and Distributed Computing within diverse industry sectors

> *You will embark on a comprehensive journey covering the most important aspects of Parallel and Distributed Computing, ranging from inherent parallelisms to their diverse applications"*

Specific Objectives

Module 1. Parallelism in Parallel and Distributed Computing

- Analyzing the processing components: processor or memory
- Deepen the architecture of parallelism
- Analyze the different forms of parallelism from the processor's point of view

Module 2. Parallel Decomposition in Parallel and Distributed Computing

- Analyze the significance of parallel process decomposition in addressing computational problems
- Examine various examples that illustrate the application and utilization of computing, along with its parallel decomposition
- Present methodologies and tools that enable the execution of parallel processes, aiming to achieve optimal performance in parallel computing
- Acquire specialized knowledge to identify scenarios for parallel process decomposition and effectively choose and apply appropriate tools for each specific scenario

Module 3. Communication and Coordination in Computing Systems

- Analyze the different architectures and models of distributed systems
- Determine the characteristics of parallel and distributed systems
- Explore the various types of communications that take place at the process level, examining their characteristics and implications
- Explore diverse communication approaches, such as remote, flow-oriented, message-oriented, and multicast communications, examining their unique characteristics and applications

- Identify emerging types of communications along with their strengths and limitations
- Develop the processes to be followed in selecting algorithms for name service, clock synchronization, coordination, and agreement between system elements
- Compile scenarios that utilize various types of communication technologies to enhance performance and scalability

Module 4. Analysis and Programming of Parallel Algorithms

- Analyze the different parallel programming paradigms
- Examine the most advanced tools to carry out parallel programming
- Analyze parallel algorithms designed for fundamental problems, examining their efficiency, scalability, and applicability
- Specify the design and analysis of parallel algorithms
- Develop parallel algorithms and implement them using popular frameworks such as MPI, OpenMP, and OpenCL/CUDA

Module 5. Parallel Architectures

- Analyze the main computer architectures
- Deepen in key aspects such as process, service and execution thread
- Manage running processes in an operating system
- Use classes to launch and manage processes

Module 6. Parallel Performance

- Analyze the aspects of parallel algorithms that impact their performance and scalability
- Establish the primary metrics for evaluating the performance and scalability of parallel algorithms
- Examine the main techniques for comparing parallel algorithms
- Identify the limitations and constraints imposed by hardware resources on the process of parallelization
- Determine the best practices for optimizing performance in shared memory parallel programs, message passing parallel programs, hybrid parallel programs, and parallel programs with heterogeneous computing
- Compile the most advanced tools for analyzing the performance of parallel algorithms
- Introduce the primary patterns of parallel processing
- Specify a robust procedure for defining high-performance parallel programs

Module 7. Distributed Systems in Computing

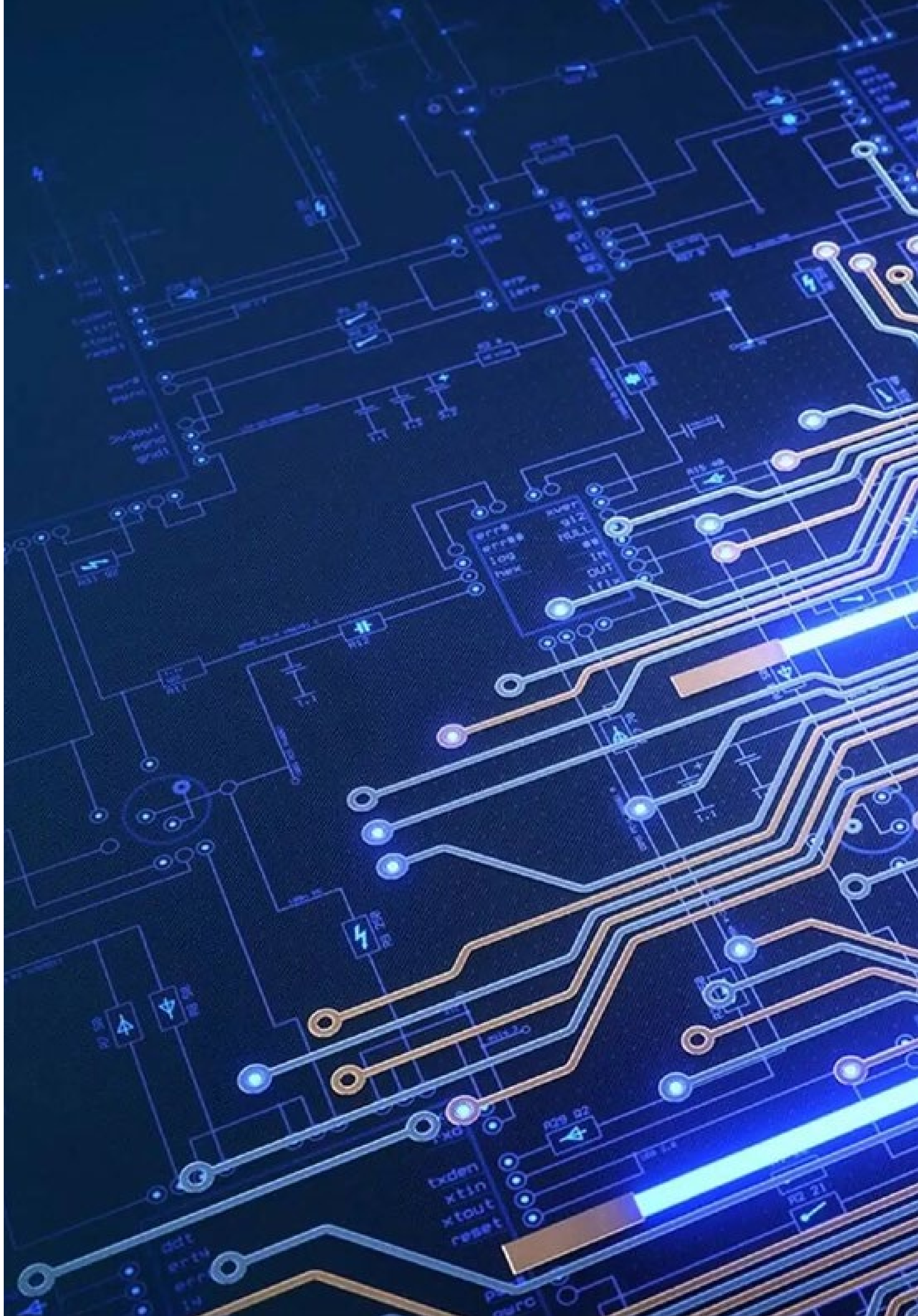
- Develop the key elements of a Distributed System
- Examine the applied security elements in Distributed Systems and their necessity
- Present the most commonly used types of Distributed Systems, including their characteristics, functionalities, and the problems they aim to solve
- Provide a proof of the CAP theorem and its applicability to distributed systems. Consistency Availability Partition Tolerance

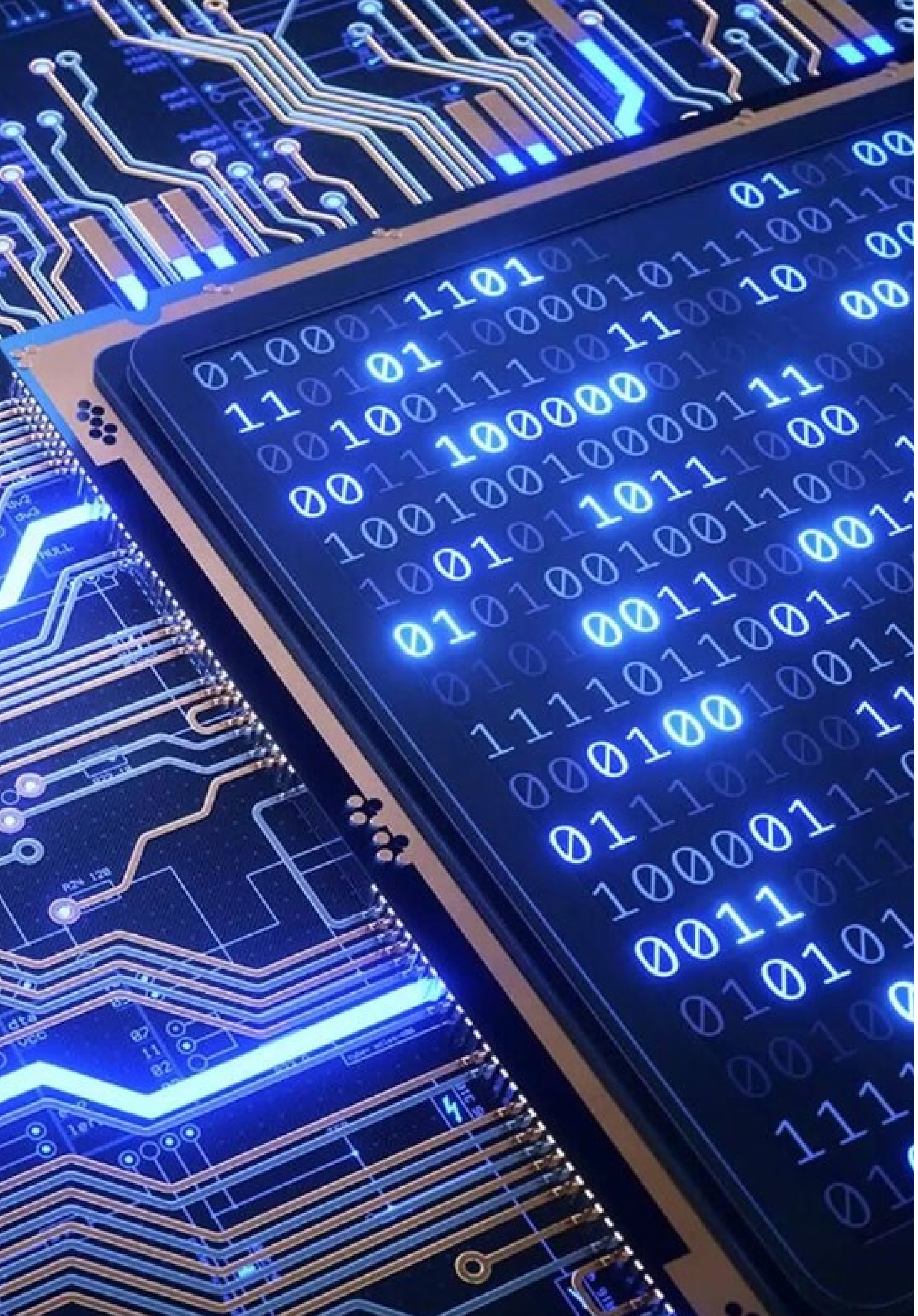
Module 8. Parallel Computing Applied to Cloud Environments

- Develop the Cloud Computing Paradigm
- Identify the different approaches based on the level of automation and service in distributed systems
- Analyze the main pieces of a cloud architecture
- Establish the differences with an architectureOn-Premise
- Analyze the various deployment options of distributed systems.Cloud: Multi-Cloud, Hybrid Cloud
- Explore the inherent benefits of cloud computing
- "Discuss the principles of cloud computing economics, including the transition from CAPEX to OPEX
- Evaluate the commercial offer of the different suppliers Cloud
- Assess the capabilities of cloud computing for supercomputing
- Examine the topic of security in cloud computing

Module 9. Models and Formal Semantics. Examine programming approaches focused on distributed computing

- Identify the benefits of formal semantics
- Explore the ways in which formal semantics can aid in programming for distributed computing
- Specify the potential applications of formal semantics in the context of programming for distributed computing
- Delve deeply into the key tools and techniques that assess the feasibility of projects utilizing distributed computing technology





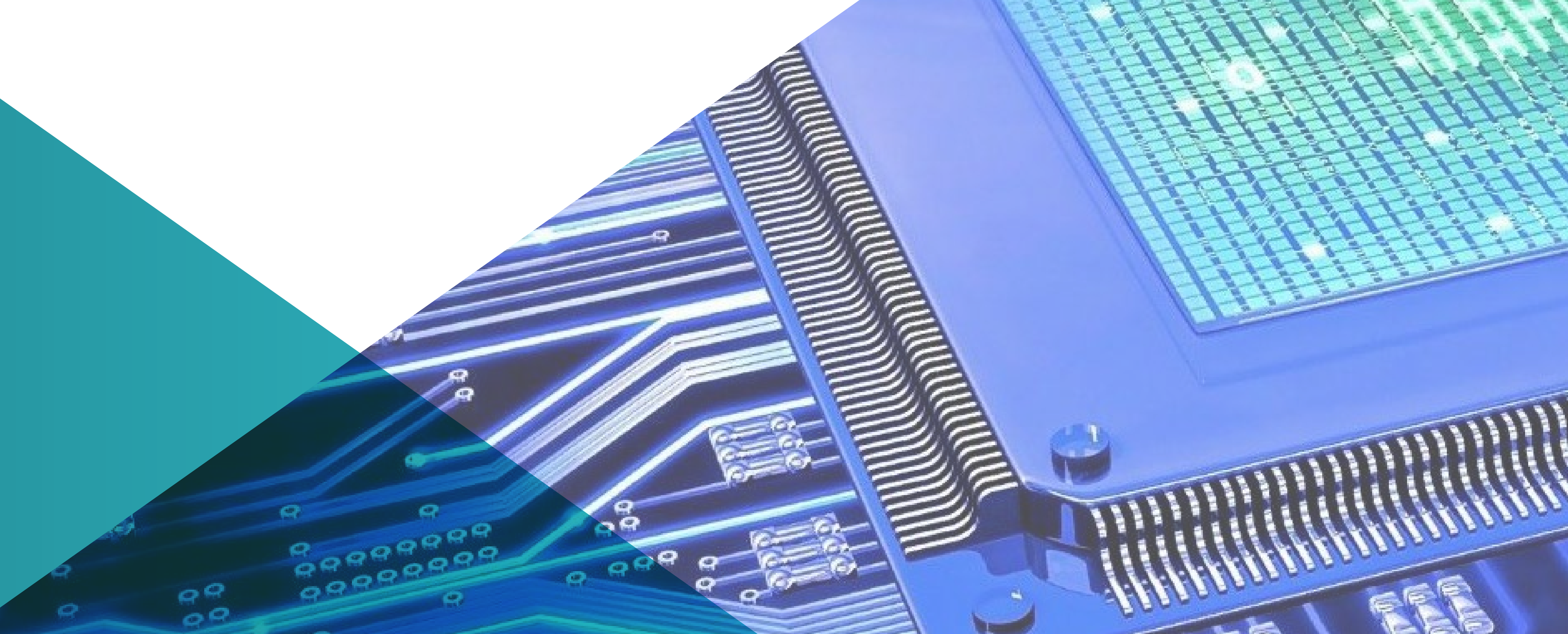
- Identify programming languages that operate within the semantic model
- Determine how this semantic models help us with the programming languages
- Evaluate and compare computing models
- Specify the practical application of distributed models
- Introduce cutting-edge market tools for project implementation

Module 10. Parallel and Distributed Computing Applications

- Demonstrate the great contribution of Parallel and Distributed Computing applications to our environment
- Identify the reference architectures in the market
- Evaluate the advantages of this cases
- Present successful solutions in the market
- Demonstrate why is it important for assessing climate change
- Determine the current importance of GPUs
- Present the impact of this technology on power grids
- Explore distributed engines to serve your potential future customers
- Explore the advantages of distributed computing engines and see how they be beneficial for a company
- Provide examples of in-memory databases and highlight their significance
- Explore how these models contribute to the field of medicine

03 Skills

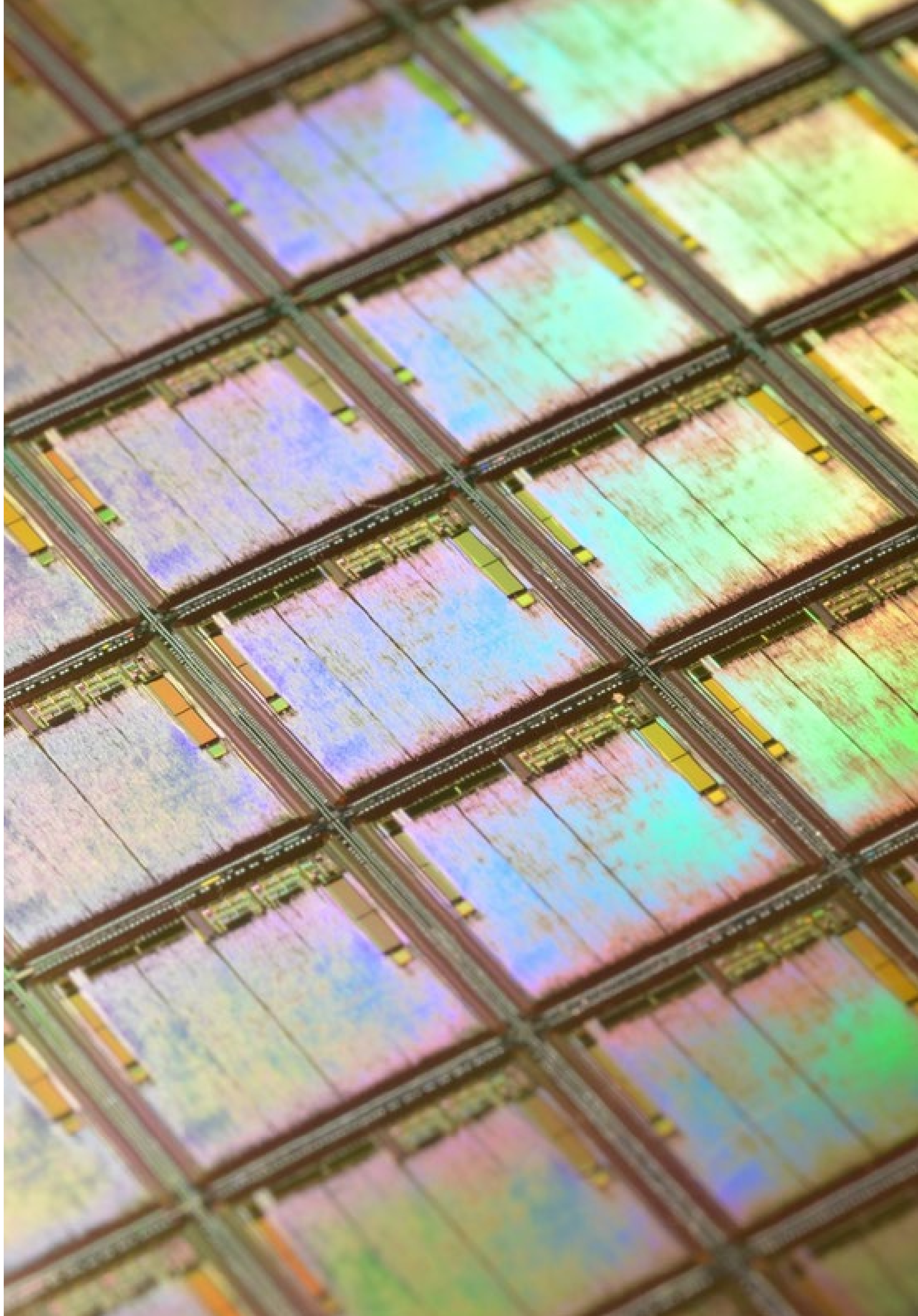
With proficient knowledge in the field of IT, professionals can excel and assume leadership roles within programming teams. This is evident through the teachers themselves, who have gained valuable experience in various IT projects and have held positions of responsibility. For this reason, the Professional Master's Degree in Parallel and Distributed Computing not only covers the fundamental aspects of this field but also equips computer scientists with a set of distinctive and unique skills that will distinguish them in their professional career.

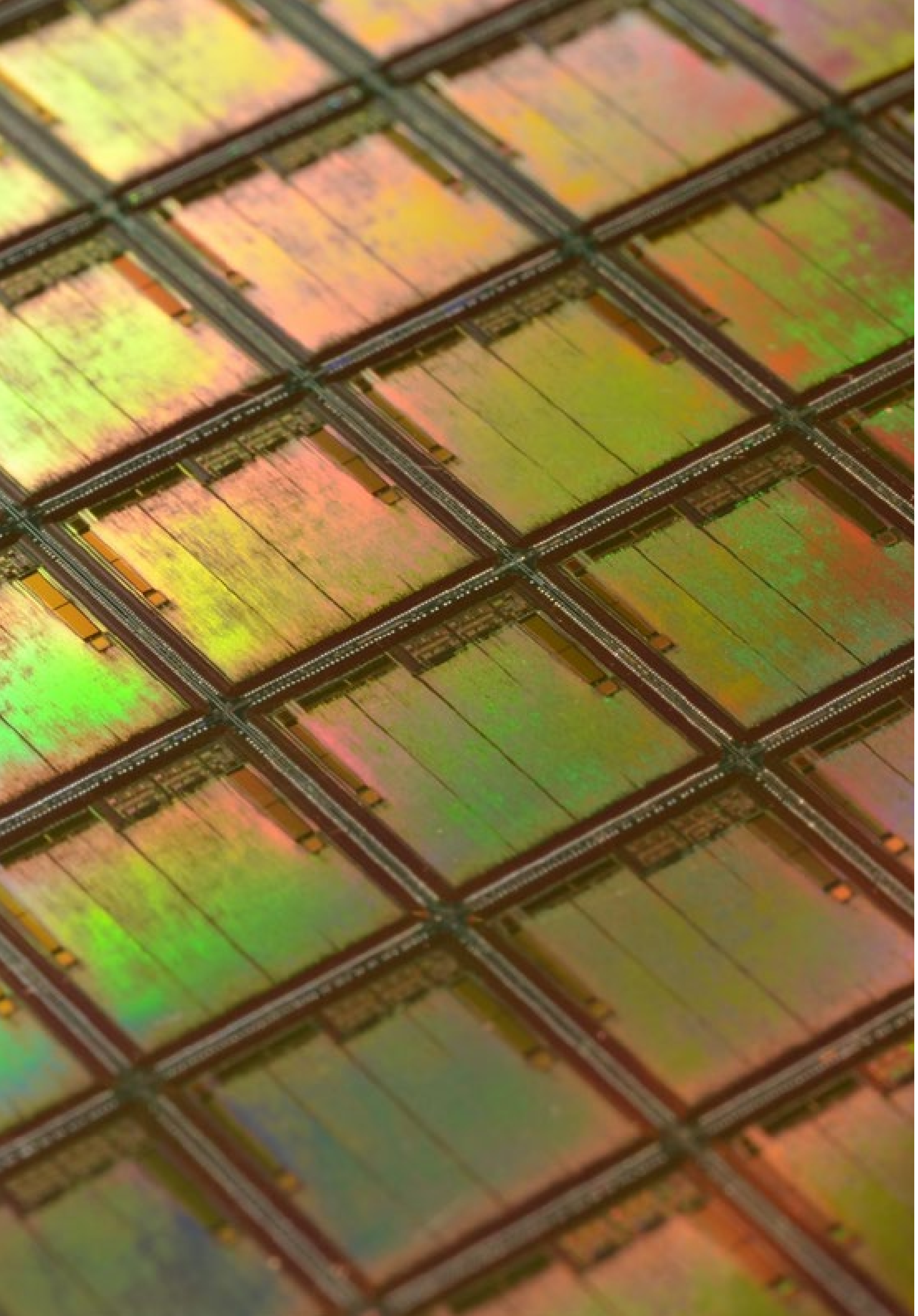


This program will accelerate your career progression, equipping you with the skills and knowledge to take on leadership roles in software programming and development"

General Skills

- Develop a comprehensive understanding of the various levels of parallelism
- Analyze a parallelization strategy by evaluating performance metrics
- Before addressing the communication and coordination that occurs between its components, it is important to determine the main characteristics of parallel and distributed computing
- Demonstrate that inter-process communications, remote calls or indirect communications can occur in this type of systems
- Determine the aspects that negatively impact the performance of parallel applications
- Examine advanced techniques for optimizing parallel code, improving communication in distributed memory systems, managing affinity, balancing workloads, and handling parallel input/output
- Examine hybrid programming models for systems that utilize multiple hardware accelerators, as well as hybrid programming models for systems that have both shared and distributed memory





Specific Skills

- Determine scalability and performance issues that can be addressed through the decomposition of parallel processes
- Analyze the characteristics of shared memory parallelism, message passing parallelism, parallelism on GPUs, and the hybrid scenario
- Establish the need for process resilience and stakeholder modeling in addressing current computing challenges
- Present examples or cases where parallel decomposition has been successfully applied to improve performance and scalability
- Analyze and design parallel algorithms
- Compile the main features or elements of MPI, OpenMP, OpenCL, and CUDA
- Analyze vector and matrix operations
- Analyze parallel and distributed programming, including programming languages, tools, and design patterns
- Identify the elements that allow the interconnection of distributed networks
- Explain the steps for designing a distributed system
- Evaluate the different types of data replication in the systems
- Complile the practical security approaches that may be applicable

04
Course Management

TECH, in its unwavering commitment to delivering exceptional teaching quality, has assembled a team of computer engineers in this program who possess extensive international experience in various projects associated with Parallel and Distributed Computing. Hence, students acquire knowledge that not only fosters their professional development but also originates from the faculty's proficiency in adeptly overseeing interdisciplinary IT teams.

Unlock your academic and professional potential by learning from top-tier educators who are fully dedicated to your success throughout the educational program"

Management

Mr. Olalla Bonal, Martín

- Senior Blockchain Practice Manager at EY
- Blockchain Client Technical Specialist for IBM
- Director of Architecture for Blocknitive
- Non-Relational Distributed Databases Team Coordinator for wedoIT (IBM Subsidiary)
- Infrastructure Architect at Bankia
- Head of Layout Department at T-Systems
- Department Coordinator for Bing Data Spain S.L

Professors

Mr. Villot Guisán, Pablo

- Chief Information Officer, Chief Technical Officer and Founder of New Tech & Talent
- Technology Expert at KPMG Spain
- Blockchain Architect at Everis
- J2EE Developer Commercial Logistics Area in Inditex
- Degree in Computer Engineering from the University of La Coruña
- Microsoft MSCA certification: Cloud Platform

Dr. Blanco, Eduardo

- Computer Science Specialist
- Professor at the Simon Bolivar University
- Doctor in Computer Science, Simon Bolivar University
- Computer Engineer, Simon Bolivar University
- Master's Degree in Computer Science, Simon Bolivar University

Dr. Almendras Aruzamen, Luis Fernando

- Data and Business Intelligence Engineer. Solutio Group, Madrid
- Data engineer at Indizen
- Data and business intelligence engineer in Technology and People
- Database, big data and business intelligence support engineer at Equinix
- Data Engineer Jalasoft
- Product Manager and responsible for the business analytics area at Goja
- Assistant Manager Business Intelligence. VIVA Nuevatel PC's
- Responsible for the datrawarehouse and big data area at Viva
- Software Development Leader at Intersoft
- Degree in Computer Science from Mayor University of San Simon
- Doctorate in Computing Engineering Complutense University of Madrid
- Master's Degree in Computer Engineering from the Complutense University of Madrid
- Master's Degree in Information Systems and Technology Management from the Mayor University of San Simon
- International Instructor Oracle Database. Proydesa- Oracle, Argentina
- Project Management Professional Certification. Scoup Consulting, Chile

Dr. Carratalá Sáez, Rocío

- Researcher specialized in Computer Science
- Teacher in University studies related to computer science
- D. in Computer Science from the Jaume I University
- Graduate in Computational Mathematics from the Jaume I University
- Master in Parallel and Distributed Computing by the Universidad Politécnica de Valencia
- Specialization courses related to Computer Science, mathematics and tools for academic research

Mr. Gozalo Fernández, Juan Luis

- Blockchain-based Product Manager for Open Canarias
- Blockchain DevOps Director at Alastria
- Director of Service Level Technology at Santander Spain
- Tinkerlink Mobile Application Development Manager at Cronos Telecom
- IT Service Management Technology Director at Barclays Bank Spain
- Bachelor's Degree in Computer Engineering from UNED
- Specialization in Deep Learning en DeepLearning.ai

D. Gómez Gómez, Borja

- Business Development Manager for Cloud Innovation at Oracle
- Head of Blockchain and pre-sales architecture solutions at Paradigma Digital
- Senior IT Architect and Consultant at Atmira
- SOA Architect and TCP SI Consultant
- Analyst and Consultant at Everis
- Degree in Computer Engineering from the Complutense University of Madrid
- Master's Degree in Science Computer Engineering at the Complutense University of Madrid

05

Structure and Content

To promote optimal study and skill acquisition, TECH has integrated the most effective pedagogical methodology into this program. Through the utilization of relearning techniques, students experience a significant reduction in the time required to acquire essential knowledge within the program. This accelerated learning is further reinforced by a wealth of audiovisual materials, supplementary readings, and practical exercises that aid in solidifying understanding across all subject matters.

Practical exercises that are based on real cases. These exercises are meticulously crafted by our experienced teacher who have developed detailed videos to guide you through the entire process"

Module 1. Parallelism in Parallel and Distributed Computing

1.1. Parallel Processing
    1.1.1. Parallel Processing
    1.1.2. Parallel Processing in Computing. Purpose
    1.1.3. Parallel Processing. Analysis
1.2. Parallel System
    1.2.1. The Parallel System
    1.2.2. Levels of Parallelism
    1.2.3. Parallel System
1.3. Processor Architectures
    1.3.1. Processor Complexity
    1.3.2. Processor Architecture. Mode of Operation
    1.3.3. Processor Architecture. Memory Organization
1.4. Networks in Parallel Processing
    1.4.1. Mode of Operation
    1.4.2. Control Strategy
    1.4.3. Switching Techniques
    1.4.4. Topology
1.5. Parallel Architectures
    1.5.1. Algorithms
    1.5.2. Coupling
    1.5.3. Communication
1.6. Performance of Parallel Computing
    1.6.1. Performance Evolution
    1.6.2. Performance Measures
    1.6.3. Parallel Computing Study Cases
1.7. Flynn's Taxonomy
    1.7.1. MIMD: memoria compartida
    1.7.2. MIMD: memoria distribuida
    1.7.3. MIMD: sistemas híbridos
    1.7.4. Data Flow
1.8. Forms of Parallelism: TLP (Thread Level Paralelism)
    1.8.1. Forms of Parallelism: TLP (Thread Level Paralelism)
    1.8.2. Coarse grain
    1.8.3. Fine grain
    1.8.4. SMT
1.9. Forms of Parallelism: DLP (Data Level Paralelism)
    1.9.1. Forms of Parallelism: DLP (Data Level Paralelism)
    1.9.2. Short Vector Processing
    1.9.3. Vector Processors
1.10. Forms of Parallelism: ILP (Instruction Level Paralelism)
    1.10.1. Forms of Parallelism: ILP (Instruction Level Paralelism)
    1.10.2. Segmented Processors
    1.10.3. Superscalar Processor
    1.10.4. Very Long Instruction Word (VLIW) Processor

Module 2. Parallel Decomposition in Parallel and Distributed Computing

2.1. Parallel Decomposition
    2.1.1. Parallel Processing:
    2.1.2. Architecture
    2.1.3. Supercomputers
2.2. Parallel Hardware and Parallel Software
    2.2.1. Serial Systems
    2.2.2. Parallel Hardware
    2.2.3. Parallel Software
    2.2.4. Input and Output
    2.2.5. Performance
2.3. Parallel Scalability and Recurring Performance Issues
    2.3.1. Parallelism
    2.3.2. Parallel Scalability
    2.3.3. Recurring Performance Issues

2.4. Shared Memory Parallelism
- 2.4.1. Shared Memory Parallelism
- 2.4.2. OpenMP and Pthreads
- 2.4.3. Shared Memory Parallelism Examples:

2.5. Graphics Processing Unit (GPU)
- 2.5.1. Graphics Processing Unit (GPU)
- 2.5.2. Computational Unified Device Architecture (CUDA)
- 2.5.3. Unified Computational Device Architecture (CUDA) 2.5.3. Examples:

2.6. Message Passing Systems
- 2.6.1. Message Passing Systems
- 2.6.2. MPI. Message Passing Interface
- 2.6.3. Message Passing Systems. Examples:

2.7. Hybrid Parallelization with MPI and OpenMP
- 2.7.1. Hybrid Programming
- 2.7.2. MPI/OpenMP Programming Models
- 2.7.3. Hybrid Decomposition and Mapping

2.8. MapReduce Computing
- 2.8.1. Hadoop
- 2.8.2. Other Computing Systems
- 2.8.3. Parallel Computing Examples:

2.9. Model of Actors and Reactive Processes
- 2.9.1. Stakeholder Model
- 2.9.2. Reactive Processes
- 2.9.3. Actors and Reactive Processes. Examples:

2.10. Parallel Computing Scenarios
- 2.10.1. Audio and image processing
- 2.10.2. Statistics/Data Mining
- 2.10.3. Parallel Sorting
- 2.10.4. Parallel Matrix Operations

Module 3. Communication and Coordination in Computing Systems

3.1. Parallel and Distributed Computing Processes
- 3.1.1. Parallel and Distributed Computing Processes
- 3.1.2. Processes and Threads
- 3.1.3. Virtualization
- 3.1.4. Clients and Servers

3.2. Communication in parallel computing
- 3.2.1. Parallel computing
- 3.2.2. Layered Protocols
- 3.2.3. Communication in parallel computing Typology

3.3. Remote Procedure Calling
- 3.3.1. Functioning of RPC (Remote Procedure Call)
- 3.3.2. Parameter Passing
- 3.3.3. Asynchronous RPC
- 3.3.4. Remote Procedure. Examples:

3.4. Message-Oriented Communication
- 3.4.1. Transient Message-Oriented Communication
- 3.4.2. Persistent Message-Oriented Communication
- 3.4.3. Message-Oriented Communication. Examples:

3.5. Flow-Oriented Communication
- 3.5.1. Support for Continuous Media
- 3.5.2. Flows and Quality of Service
- 3.5.3. Flow Synchronization
- 3.5.4. Flow-Oriented Communication. Examples:

3.6. Multicast Communication
- 3.6.1. Multicast at Application Level
- 3.6.2. Rumor-Based Data Broadcasting
- 3.6.3. Multicast Communication. Examples:

3.7. Other Types of Communication
- 3.7.1. Remote Method Invocation
- 3.7.2. Web Services / SOA / REST
- 3.7.3. Event Notification
- 3.7.4. Mobile Agents

3.8. Name Service
- 3.8.1. Name Services in Computing
- 3.8.2. Name Services and Domain Name System
- 3.8.3. Directory Services

3.9. Synchronization
- 3.9.1. Clock Synchronization
- 3.9.2. Logical Clocks, Mutual Exclusion and Global Positioning of Nodes
- 3.9.3. Choice of Algorithms

3.10. Communication Coordination and Agreement
- 3.10.1. Coordination and Agreement
- 3.10.2. Coordination and Agreement Consensus and Problems
- 3.10.3. Communication and Coordination. Currently

Module 4. Analysis and Programming of Parallel Algorithms

4.1. Parallel Algorithms
- 4.1.1. Problem Decomposition
- 4.1.2. Data Dependencies
- 4.1.3. Implicit and Explicit Parallelism

4.2. Parallel programming paradigms
- 4.2.1. Parallel programming with shared memory
- 4.2.2. Parallel programming with distributed memory
- 4.2.3. Hybrid Parallel Programming
- 4.2.4. Heterogeneous Computing - CPU + GPU
- 4.2.5. Quantum Computing New Programming Models with Implicit Parallelism

4.3. Parallel programming with shared memory
- 4.3.1. Parallel programming models with shared memory
- 4.3.2. Parallel Algorithms with Shared Memory
- 4.3.3. Libraries for parallel programming with shared memory

4.4. OpenMP
- 4.4.1. OpenMP
- 4.4.2. Running and Debugging Programs with OpenMP
- 4.4.3. Parallel Algorithms with Shared Memory in OpenMP

4.5. Parallel message-passing programming
- 4.5.1. Fundamental operations of Message Passing
- 4.5.2. Communication and collective computing operations
- 4.5.3. Parallel Message-Passing Algorithms
- 4.5.4. Libraries for parallel programming with message passing

4.6. Message Passing Interface (MPI)
- 4.6.1. Message Passing Interface (MPI)
- 4.6.2. Execution and Debugging of Programs with MPI
- 4.6.3. Parallel Message Passing Algorithms with MPI

4.7. Hybrid Parallel Programming
- 4.7.1. Hybrid Parallel Programming
- 4.7.2. Execution and Debugging of Hybrid Parallel Programs
- 4.7.3. MPI-OpenMP Hybrid Parallel Algorithms

4.8. Parallel Programming with Heterogeneous Computing
- 4.8.1. Parallel Programming with Heterogeneous Computing
- 4.8.2. AIH vs. GPU
- 4.8.3. Parallel Algorithms with Heterogeneous Computing

4.9. OpenCL and CUDA
- 4.9.1. OpenCL vs. CUDA
- 4.9.2. Executing and Debugging Parallel Programs with Heterogeneous Computing
- 4.9.3. Parallel Algorithms with Heterogeneous Computing

4.10. Design of Parallel Algorithms
- 4.10.1. Design of Parallel Algorithms
- 4.10.2. Problem and Context
- 4.10.3. Automatic Parallelization vs. Manual Parallelization
- 4.10.4. Problem Partitioning
- 4.10.5. Computer Communications

Module 5. Parallel Architectures

5.1. Parallel Architectures
- 5.1.1. Parallel Systems. Classification
- 5.1.2. Sources of Parallelism
- 5.1.3. Parallelism and Processors

5.2. Performance of Parallel Systems
- 5.2.1. Performance Metrics and Measurements
- 5.2.2. Speed-up
- 5.2.3. Granularity of Parallel Systems

5.3. Vector Processors
- 5.3.1. Basic Vector Processor
- 5.3.2. Interleaved or Interleaved Memory
- 5.3.3. Performance of Vector Processors

5.4. Matrix Processors
- 5.4.1. Basic Organization
- 5.4.2. Programming in Matrix Processors
- 5.4.3. Programming in Matrix Processors. Practical Example

5.5. Interconnection Networks
- 5.5.1. Interconnection Networks
- 5.5.2. Topology, Flow Control and Routing
- 5.5.3. Interconnection Networks. Classification According to Topology

5.6. Multiprocessors
- 5.6.1. Multiprocessor Interconnection Networks
- 5.6.2. Memory and Cache Consistency
- 5.6.3. Probe Protocols

5.7. Synchronization
- 5.7.1. Bolts (Mutual exclusion)
- 5.7.2. P2P Synchronization Events
- 5.7.3. Global Synchronization Events

5.8. Multicomputers
- 5.8.1. Multicomputer Interconnection Networks
- 5.8.2. Switching Layer
- 5.8.3. Routing Layer

5.9. Advanced Architectures
- 5.9.1. Data Stream Machines
- 5.9.2. Other Architectures

5.10. Parallel and Distributed Programming
- 5.10.1. Parallel Programming Languages
- 5.10.2. Parallel Programming Tools
- 5.10.3. Design Patterns
- 5.10.4. Concurrency of Parallel and Distributed Programming Languages

Module 6. Parallel Performance

6.1. Performance of Parallel Algorithms
- 6.1.1. Ahmdal's Law
- 6.1.2. Gustarfson's Law
- 6.1.3. Performance Metrics and Scalability of Parallel Algorithms

6.2. Comparison of Parallel Algorithms
- 6.2.1. Benchmarking
- 6.2.2. Mathematical Analysis of Parallel Algorithms
- 6.2.3. Asymptotic Analysis of Parallel Algorithms

6.3. Hardware Resource Constraints
- 6.3.1. Memory
- 6.3.2. Processing
- 6.3.3. Communication
- 6.3.4. Dynamic Resource Partitioning

6.4. Parallel Program Performance with Shared Memory
- 6.4.1. Optimal Task Partitioning
- 6.4.2. Thread Affinity
- 6.4.3. SIMD Parallelism
- 6.4.4. Parallel Programs with Shared Memory. Examples:

6.5. Performance of Message-Passing Parallel Programs
- 6.5.1. Performance of Message-Passing Parallel Programs
- 6.5.2. Optimization of MPI Communications
- 6.5.3. Affinity Control and Load Balancing
- 6.5.4. Parallel I/O
- 6.5.5. Parallel programs by message passing. Examples:

6.6. Performance of Hybrid Parallel Programs
- 6.6.1. Performance of Hybrid Parallel Programs
- 6.6.2. Hybrid Programming for Shared/Distributed Memory Systems
- 6.6.3. Hybrid Parallel Programs. Examples:

6.7. Performance of Programs with Heterogeneous Computation
- 6.7.1. Performance of Programs with Heterogeneous Computation
- 6.7.2. Hybrid Programming for Systems with Multiple Hardware Accelerators
- 6.7.3. Programs with Heterogeneous Computing. Examples:

6.8. Performance Analysis of Parallel Algorithms
- 6.8.1. Performance Analysis of Parallel Algorithms
- 6.8.2. Performance Analysis of Parallel Algorithms. Data Science
- 6.8.3. Performance Analysis of Parallel Algorithms. Recommendations

6.9. Parallel Patterns
- 6.9.1. Parallel Patterns
- 6.9.2. Main Parallel Patterns
- 6.9.3. Parallel Patterns Comparison

6.10. High Performance Parallel Programs
- 6.10.1. Process
- 6.10.2. High Performance Parallel Programs
- 6.10.3. High Performance Parallel Programs Real Uses

Module 7. Distributed Computing Systems

7.1. Distributed Systems
- 7.1.1. Sistemas Distribuidos (SD)
- 7.1.2. Proof of the CAP Theorem (or Brewer's Conjecture)
- 7.1.3. Fallacies of Distributed Systems Programming
- 7.1.4. Ubiquitous Computing

7.2. Distributed Systems Features
- 7.2.1. Heterogeneity
- 7.2.2. Security/Safety
- 7.2.3. Scales
- 7.2.4. Fault Tolerance
- 7.2.5. Concurrency
- 7.2.6. Transparency

7.3. Networks and Interconnection of Distributed Networks
- 7.3.1. Networks and Distributed Systems. Network Performance
- 7.3.2. Networks Available to Create a Distributed System. Typology
- 7.3.3. Distributed network protocols vs. Centralized
- 7.3.4. Interconnection of Networks. Internet

7.4. Communication Between Distributed Processes
- 7.4.1. Communication Between SD Nodes. Problems and Failures
- 7.4.2. Mechanisms to Implement Over RPC and RDMA to Avoid Failures
- 7.4.3. Mechanisms to Implement in the Software to Avoid Failures

7.5. Distributed Systems Design
- 7.5.1. Efficient Design of Distributed Systems (DS)
- 7.5.2. Programming Patterns in Distributed Systems (DS)
- 7.5.3. Service Oriented Architecture (SOA)
- 7.5.4. Service Orchestration and Microservices Data Management

7.6. Distributed Systems Operation
- 7.6.1. Systems Monitoring
- 7.6.2. Implementing an Efficient Logging System in a DS
- 7.6.3. Monitoring in Distributed Networks
- 7.6.4. Use of a Monitoring Tool for an SD Prometheus and Grafana

7.7. System Replication
- 7.7.1. System Replication Typology
- 7.7.2. Immutable Architecture
- 7.7.3. Container Systems and Virtualizing Systems as Distributed Systems
- 7.7.4. Blockchain Networks as Distributed Systems

7.8. Distributed Multimedia Systems
- 7.8.1. Distributed Exchange of Images and Videos. Problems
- 7.8.2. Multimedia Object Servers
- 7.8.3. Network Topology for a Multimedia System
- 7.8.4. Analysis of Distributed Multimedia Systems: Netflix, Amazon, Spotify, etc
- 7.8.5. Distributed Multimedia Systems in Education

7.9. Distributed File Systems
- 7.9.1. Distributed File Sharing. Problems
- 7.9.2. Applicability of the CAP Theory to Databases
- 7.9.3. Distributed Web File Systems: Akamai
- 7.9.4. IPFS Distributed Document File Systems
- 7.9.5. Sistemas de bases de datos distribuidas

7.10. Enfoques de seguridad en Sistemas Distribuidos
- 7.10.1. Security in Distributed Systems
- 7.10.2. Known Attacks on Distributed Systems
- 7.10.3. Tools for Testing the Security of a DS

Module 8. Parallel Computing Applied to Cloud Environments

8.1. Cloud Computing
- 8.1.1. State of the Art of the IT Landscape
- 8.1.2. The "Cloud"
- 8.1.3. Cloud Computing

8.2. Security and Resilience in the Cloud
- 8.2.1. Regions, Availability and Failure Zones
- 8.2.2. Tenant or Cloud account management
- 8.2.3. Cloud Identity and Access Control

8.3. Cloud Networking
- 8.3.1. Software-Defined Virtual Networks
- 8.3.2. Network Components of a Software-Defined Network
- 8.3.3. Connection with other Systems

8.4. Cloud Services
- 8.4.1. Infrastructure as a Service
- 8.4.2. Platform as a Service
- 8.4.3. Serverless Computing
- 8.4.4. Software as a Service

8.5. Cloud Storage
- 8.5.1. Block Storage in the Cloud
- 8.5.2. Block Storage in the Cloud
- 8.5.3. Block Storage in the Cloud

8.6. Block Storage in the Cloud
- 8.6.1. Cloud Monitoring and Management
- 8.6.2. Interaction with the Cloud: Administration Console
- 8.6.3. Interaction with Command Line Interface
- 8.6.4. API-Based Interaction

8.7. Cloud-Native Development
- 8.7.1. Cloud Native Development
- 8.7.2. Containers and Container Orchestration Platforms
- 8.7.3. Continuous Cloud Integration
- 8.7.4. Use of Events in the Cloud

8.8. Infrastructure as Code in the Cloud
- 8.8.1. Management and Provisioning Automation in the Cloud
- 8.8.2. Terraform
- 8.8.3. Scripting Integration

8.9. Creation of a Hybrid Infrastructure
- 8.9.1. Interconnection
- 8.9.2. Interconnection with Datacenter
- 8.9.3. Interconnection with other Clouds

8.10. High-Performance Computing
- 8.10.1. High-Performance Computing
- 8.10.2. Creation of a High-Performance Cluster
- 8.10.3. Application of High-Performance Computing

Module 9. Models and Formal Semantics. Examine programming approaches focused on distributed computing

9.1. Semantics Data Model
- 9.1.1. Semantics Data Model
- 9.1.2. Semantics Data Model. Purposes
- 9.1.3. Semantics Data Model. Applications

9.2. Semantic Model of Programming Languages
- 9.2.1. Language Processing
- 9.2.2. Translation and Interpretation
- 9.2.3. Hybrid Languages

9.3. Models of Computation
- 9.3.1. Monolithic Computing
- 9.3.2. Parallel Computing
- 9.3.3. Distributed Computing
- 9.3.4. Cooperative Computing (P2P)

9.4. Parallel Computing
- 9.4.1. Parallel Architecture
- 9.4.2. Hardware
- 9.4.3. Software

9.5. Distribution Models Grid Computing
- 9.5.1. Grid Computing Architecture
- 9.5.2. Grid Computing Architecture Analysis
- 9.5.3. Grid Computing Architecture Applications

9.6. Distributed Model. Cluster Computing
- 9.6.1. Cluster Computing Architecture
- 9.6.2. Grid Computing Architecture Analysis
- 9.6.3. Grid Computing Architecture Applications

9.7. Cluster Computing. Current Tools to Implement Cluster Computing. Hypervisors
- 9.7.1. Market Competitors
- 9.7.2. VMware Hypervisor
- 9.7.3. Hyper-V

9.8. Distribution Models Cloud Computing
- 9.8.1. Architecture Cloud Computing
- 9.8.2. Cloud Computing Architecture. Analysis
- 9.8.3. Cloud Computing Architecture. Applications

9.9. Distribution Models Amazon Cloud Computing
- 9.9.1. Amazon Cloud Computing Functional Criteria
- 9.9.2. Amazon Cloud Computing Licensing
- 9.9.3. Amazon Cloud Computing Reference Architectures

9.10. Distribution Models Microsoft Cloud Computing
- 9.10.1. Cloud Computing Microsoft. Functional Criteria
- 9.10.2. Cloud Computing Microsoft. Licensing
- 9.10.3. Cloud Computing Microsoft. Reference Architectures

Module 10. Parallel and Distributed Computing Applications

10.1. Parallel and Distributed Computing in Today's Applications
- 10.1.1. Hardware
- 10.1.2. Software
- 10.1.3. Importance of Timing

10.2. Climate. Climate Change
- 10.2.1. Climate Applications. Data Sources
- 10.2.2. Climate Applications. Data Volumes
- 10.2.3. Climate Applications. Real Time

10.3. GPU computación paralela
- 10.3.1. GPU computación paralela
- 10.3.2. GPUs vs. CPU. GPU Usage
- 10.3.3. GPU. Examples:

10.4. Smart Grid. Computing in Power Grids
- 10.4.1. Smart Grid
- 10.4.2. Conceptual Models. Examples:
- 10.4.3. Smart Grid. Example

10.5. Distributed Engine. ElasticSearch
- 10.5.1. Distributed Engine. ElasticSearch
- 10.5.2. Architecture with ElasticSearch. Examples:
- 10.5.3. Distributed Engine. Case Uses

10.6. Big Data Framework
- 10.6.1. Big Data Framework
- 10.6.2. Architecture of Advanced Tools
- 10.6.3. Big Data in Distributed Computing

10.7. Memory Database
- 10.7.1. Memory Database
- 10.7.2. Redis Solution. Case Study
- 10.7.3. Deployment of Solutions With In-Memory Database

10.8. Blockchain
- 10.8.1. Blockchain Architecture. Components
- 10.8.2. Collaboration Between Nodes and Consensus
- 10.8.3. Blockchain Solutions. Implementations

10.9. Distributed Systems in Medicine
- 10.9.1. Architecture Components
- 10.9.2. Distributed Systems in Medicine. Operation
- 10.9.3. Distributed Systems in Medicine. Applications

10.10. Distributed Systems in the Aviation Sector
- 10.10.1. Architecture Design
- 10.10.2. Distributed Systems in the Aviation Sector. Component Functionalities
- 10.10.3. Distributed Systems in the Aviation Sector. Applications

“This program acts as the catalyst you need to achieve the well-deserved professional advancement you have diligently worked towards over a significant period of time”

06

Methodology

This academic program offers students a different way of learning. Our methodology uses a cyclical learning approach: **Relearning.**

This teaching system is used, for example, in the most prestigious medical schools in the world, and major publications such as the **New England Journal of Medicine** have considered it to be one of the most effective.

Discover Relearning, a system that abandons conventional linear learning, to take you through cyclical teaching systems: a way of learning that has proven to be extremely effective, especially in subjects that require memorization"

Case Study to contextualize all content

Our program offers a revolutionary approach to developing skills and knowledge. Our goal is to strengthen skills in a changing, competitive, and highly demanding environment.

At TECH, you will experience a learning methodology that is shaking the foundations of traditional universities around the world"

You will have access to a learning system based on repetition, with natural and progressive teaching throughout the entire syllabus.

The student will learn to solve complex situations in real business environments through collaborative activities and real cases.

A learning method that is different and innovative

This TECH program is an intensive educational program, created from scratch, which presents the most demanding challenges and decisions in this field, both nationally and internationally. This methodology promotes personal and professional growth, representing a significant step towards success. The case method, a technique that lays the foundation for this content, ensures that the most current economic, social and professional reality is taken into account.

Our program prepares you to face new challenges in uncertain environments and achieve success in your career"

The case method has been the most widely used learning system among the world's leading Information Technology schools for as long as they have existed. The case method was developed in 1912 so that law students would not only learn the law based on theoretical content. It consisted of presenting students with real-life, complex situations for them to make informed decisions and value judgments on how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

What should a professional do in a given situation? This is the question that you are presented with in the case method, an action-oriented learning method. Throughout the course, students will be presented with multiple real cases. They will have to combine all their knowledge and research, and argue and defend their ideas and decisions.

Relearning Methodology

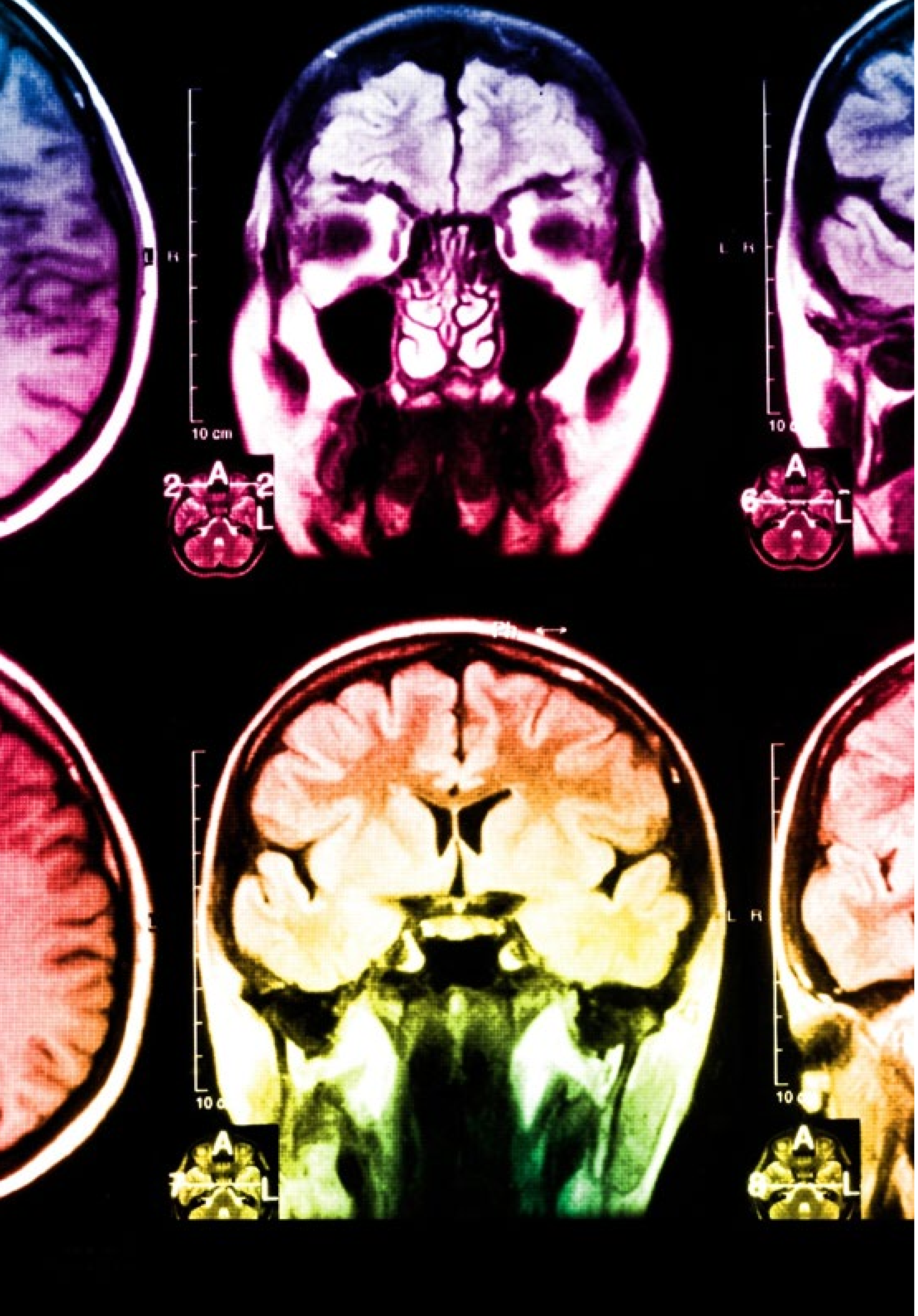
TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

We enhance the Case Study with the best 100% online teaching method: Relearning.

In 2019, we obtained the best learning results of all online universities in the world.

At TECH you will learn using a cutting-edge methodology designed to train the executives of the future. This method, at the forefront of international teaching, is called Relearning.

Our university is the only one in the world authorized to employ this successful method. In 2019, we managed to improve our students' overall satisfaction levels (teaching quality, quality of materials, course structure, objectives...) based on the best online university indicators.



In our program, learning is not a linear process, but rather a spiral (learn, unlearn, forget, and re-learn). Therefore, we combine each of these elements concentrically. This methodology has trained more than 650,000 university graduates with unprecedented success in fields as diverse as biochemistry, genetics, surgery, international law, management skills, sports science, philosophy, law, engineering, journalism, history, and financial markets and instruments. All this in a highly demanding environment, where the students have a strong socio-economic profile and an average age of 43.5 years.

Relearning will allow you to learn with less effort and better performance, involving you more in your training, developing a critical mindset, defending arguments, and contrasting opinions: a direct equation for success.

From the latest scientific evidence in the field of neuroscience, not only do we know how to organize information, ideas, images and memories, but we know that the place and context where we have learned something is fundamental for us to be able to remember it and store it in the hippocampus, to retain it in our long-term memory.

In this way, and in what is called neurocognitive context-dependent e-learning, the different elements in our program are connected to the context where the individual carries out their professional activity.

This program offers the best educational material, prepared with professionals in mind:

Study Material

All teaching material is produced by the specialists who teach the course, specifically for the course, so that the teaching content is highly specific and precise.

These contents are then applied to the audiovisual format, to create the TECH online working method. All this, with the latest techniques that offer high quality pieces in each and every one of the materials that are made available to the student.

Classes

There is scientific evidence suggesting that observing third-party experts can be useful.

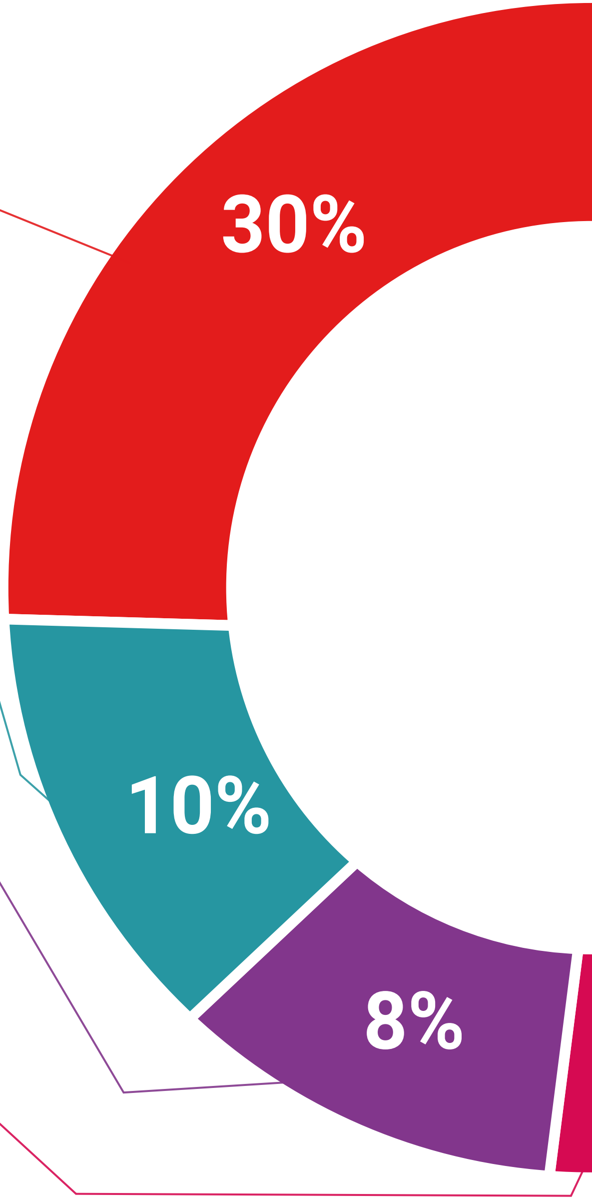
Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.

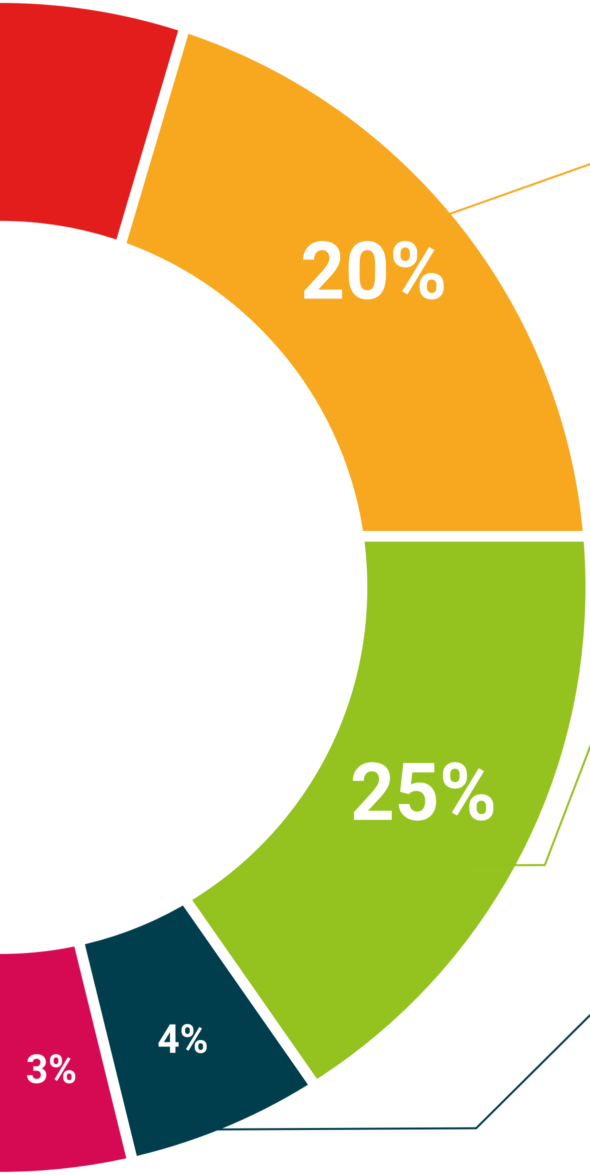
Practising Skills and Abilities

They will carry out activities to develop specific skills and abilities in each subject area. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop in the context of the globalization that we are experiencing.

Additional Reading

Recent articles, consensus documents and international guidelines, among others. In TECH's virtual library, students will have access to everything they need to complete their course.





Case Studies

Students will complete a selection of the best case studies chosen specifically for this program. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Interactive Summaries

The TECH team presents the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

This exclusive educational system for presenting multimedia content was awarded by Microsoft as a "European Success Story".

Testing & Retesting

We periodically evaluate and re-evaluate students' knowledge throughout the program, through assessment and self-assessment activities and exercises, so that they can see how they are achieving their goals.

07
Certificate

The Professional Master's Degree in Parallel and Distributed Computing guarantees students, in addition to the most rigorous and up-to-date education, access to a Professional Master's Degree diploma issued by TECH Global University.

Successfully complete this program and receive your university qualification without having to travel or fill out laborious paperwork"

This program will allow you to obtain your **Professional Master's Degree diploma in Parallel and Distributed Computing** endorsed by **TECH Global University**, the world's largest online university.

TECH Global University is an official European University publicly recognized by the Government of Andorra (***official bulletin***). Andorra is part of the European Higher Education Area (EHEA) since 2003. The EHEA is an initiative promoted by the European Union that aims to organize the international training framework and harmonize the higher education systems of the member countries of this space. The project promotes common values, the implementation of collaborative tools and strengthening its quality assurance mechanisms to enhance collaboration and mobility among students, researchers and academics.

This **TECH Global University** title is a European program of continuing education and professional updating that guarantees the acquisition of competencies in its area of knowledge, providing a high curricular value to the student who completes the program.

Title: **Professional Master's Degree in Parallel and Distributed Computing**

Modality: **online**

Duration: **12 months**

Accreditation: **60 ECTS**


tech global university

Mr./Ms. ________________, with identification document ____________
has successfully passed and obtained the title of:

Professional Master's Degree in Parallel and Distributed Computing

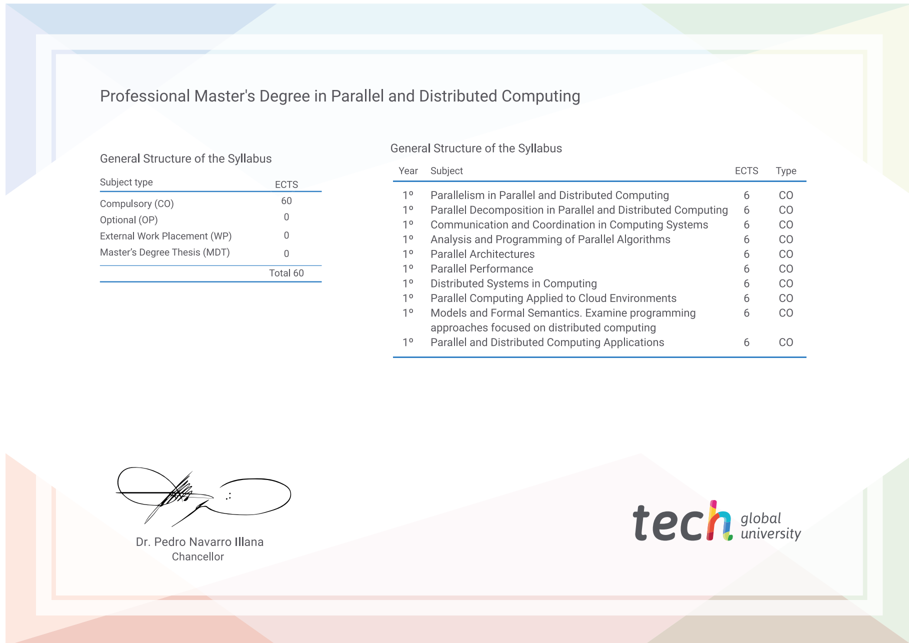
This is a program of 1,500 hours of duration equivalent to 60 ECTS, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

TECH Global University is a university officially recognized by the Government of Andorra on the 31st of January of 2024, which belongs to the European Higher Education Area (EHEA).

In Andorra la Vella, on the 28th of February of 2024

Dr. Pedro Navarro Illana
Rector

Unique TECH Code: AFWORD23S techtitute.com/certificates


Professional Master's Degree in Parallel and Distributed Computing

General Structure of the Syllabus

Subject type	ECTS
Compulsory (CO)	60
Optional (OP)	0
External Work Placement (WP)	0
Master's Degree Thesis (MDT)	0
	Total 60

General Structure of the Syllabus

Year	Subject	ECTS	Type
1º	Parallelism in Parallel and Distributed Computing	6	CO
1º	Parallel Decomposition in Parallel and Distributed Computing	6	CO
1º	Communication and Coordination in Computing Systems	6	CO
1º	Analysis and Programming of Parallel Algorithms	6	CO
1º	Parallel Architectures	6	CO
1º	Parallel Performance	6	CO
1º	Distributed Systems in Computing	6	CO
1º	Parallel Computing Applied to Cloud Environments	6	CO
1º	Models and Formal Semantics. Examine programming approaches focused on distributed computing	6	CO
1º	Parallel and Distributed Computing Applications	6	CO

Dr. Pedro Navarro Illana
Chancellor

tech global university

*Apostille Convention. In the event that the student wishes to have their paper diploma issued with an apostille, TECH Global University will make the necessary arrangements to obtain it, at an additional cost.

Professional Master's Degree
Parallel and Distributed Computing

» Modality: **online**
» Duration: **12 months**
» Certificate: **TECH Global University**
» Credits: **60 ECTS**
» Schedule: **at your own pace**
» Exams: **online**

Professional Master's Degree
Parallel and Distributed Computing