Máster Título Propio

Computación Paralela y Distribuida



tech universidad tecnológica



tech universidad tecnológica

Máster Título Propio

Computación Paralela y Distribuida

Modalidad: **Online**

Duración: **12 meses**

Titulación: **TECH Universidad Tecnológica**

Horas lectivas: **1500 h.**

Acceso web: **www.techtitute.com/informatica/master/master-computacion-paralela-distribuida**

Índice

01 Presentación — pág. 4

02 Objetivos — pág. 8

03 Competencias — pág. 14

04 Dirección del curso — pág. 18

05 Estructura y contenido — pág. 22

06 Metodología — pág. 32

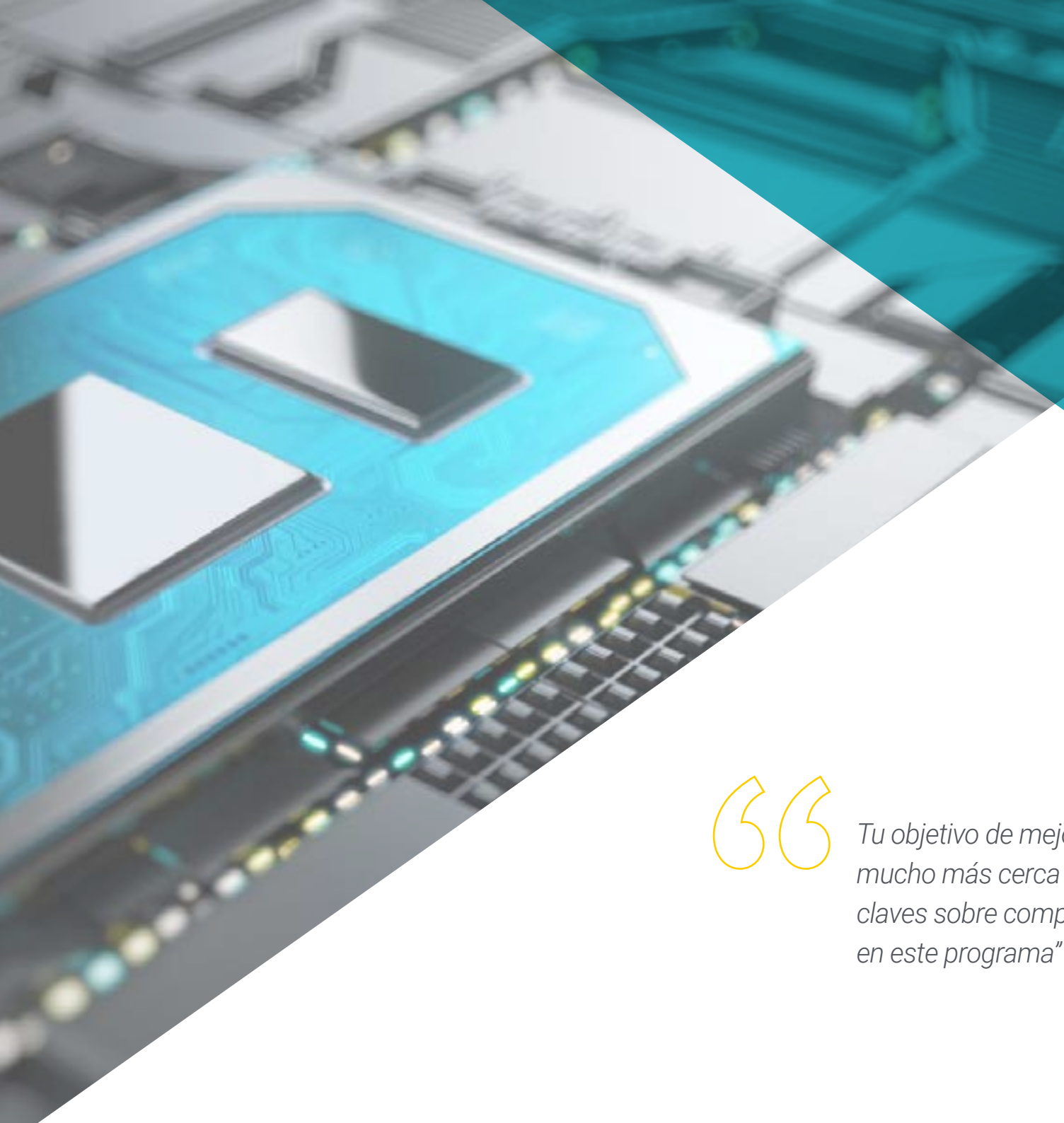
07 Titulación — pág. 40

02

Objetivos

Dado que la informática avanza a pasos agigantados y los profesionales dedicados a este campo deben hacer un esfuerzo de actualización continua, TECH ha enfocado este programa en los desarrollos más novedosos ocurridos en Computación Paralela y Distribuida. De este modo, el alumno no solo adquirirá las competencias más avanzadas al respecto, sino que además se instruirá en las múltiples aplicaciones que tienen hoy en día tecnologías como la del *blockchain* o computación en la nube.





Tu objetivo de mejora profesional estará mucho más cerca gracias a los consejos y claves sobre computación que aprenderás en este programa"

Objetivos generales

- Analizar lo que ocurre entre los diferentes componentes de la Computación Paralela y Distribuida
- Medir y comparar su desempeño para analizar el rendimiento del conjunto de componentes utilizados
- Analizar en profundidad la Computación Paralela Multiplataforma para utilizar paralelismo a nivel de tarea entre distintos aceleradores hardware
- Analizar en detalle el software y arquitecturas actuales
- Desarrollar en profundidad los aspectos relevantes de la Computación Paralela y Distribuida
- Especializar al alumno en el uso de la Computación Paralela y Distribuida en diferentes sectores de aplicación

Harás un recorrido exhaustivo por todas las cuestiones más importantes de la Computación Paralela y Distribuida, desde sus propios paralelismos a las múltiples aplicaciones que tienen"

Objetivos específicos

Módulo 1. Paralelismo en computación paralela y distribuida

- Analizar los Componentes de Procesamiento: Procesador o Memoria
- Profundizar en la Arquitectura del Paralelismo
- Analizar las diferentes formas del Paralelismo desde el punto de vista del Procesador

Módulo 2. Descomposición en paralelo en computación paralela y distribuida

- Analizar la importancia de la Descomposición de Procesos en Paralelo en la Resolución de Problemas de Cómputo
- Examinar distintos ejemplos para demostrar la aplicación y uso de la computación y su Descomposición en Paralelo
- Exponer procedimientos y herramientas que permitan la ejecución de procesos en paralelo, buscando obtener el mejor rendimiento posible
- Desarrollar conocimiento especializado para identificar escenarios de descomposición de procesos en paralelo y elegir y aplicar la herramienta apropiada

Módulo 3. Comunicación y coordinación en sistemas de computación

- Analizar las diferentes arquitecturas y modelos de los sistemas distribuidos
- Determinar las características de los sistemas paralelos y distribuidos
- Profundizar en las diferentes comunicaciones que suceden a nivel de procesos
- Examinar comunicaciones remotas, orientadas a flujos, orientas a mensajes y de multidifusión junto con ejemplos y consideraciones más recientes
- Establecer los tipos de comunicación que están emergiendo, virtudes y limitaciones

- Desarrollar los procesos a seguir en la elección de algoritmos que se aplican para el servicio de nombres, sincronización de relojes, coordinación y acuerdo entre los elementos del sistema
- Compilar los escenarios en los que se usan diferentes tipos de tecnologías de comunicación que mejoran el rendimiento y la escalabilidad

Módulo 4. Análisis y programación de algoritmos paralelos

- Analizar los distintos paradigmas de Programación Paralela
- Examinar las herramientas más avanzadas para llevar a cabo la Programación Paralela
- Analizar algoritmos paralelos para problemas fundamentales
- Concretar el diseño y análisis de algoritmos paralelos
- Desarrollar algoritmos paralelos e implementarlos mediante MPI, OpenMP, OpenCL/CUDA

Módulo 5. Arquitecturas paralelas

- Analizar las principales arquitecturas de computadores
- Profundizar en aspectos clave como Proceso, Servicio e Hilo de Ejecución
- Gestionar los procesos en Ejecución en un sistema operativo
- Utilizar clases para lanzar y gestionar procesos

Módulo 6. Desempeño en paralelo

- Analizar los aspectos de algoritmos paralelos que afectan a su desempeño y escalabilidad
- Establecer las principales métricas de desempeño y escalabilidad de algoritmos paralelos
- Examinar las principales técnicas de comparación de algoritmos paralelos
- Identificar las restricciones que los recursos hardware imponen en la Paralelización
- Determinar las mejores prácticas para desempeño de Programas Paralelos de Memoria Compartida, para desempeño de programas paralelos por paso de mensajes, para desempeño de programas paralelos híbridos y para desempeño de programas paralelos con computación heterogénea
- Compilar las herramientas más avanzadas para analizar el rendimiento de algoritmos paralelos
- Presentar los principales patrones de procesamiento en paralelo
- Concretar un procedimiento robusto para la definición de programas paralelos de alto rendimiento

Módulo 7. Sistemas Distribuidos en computación

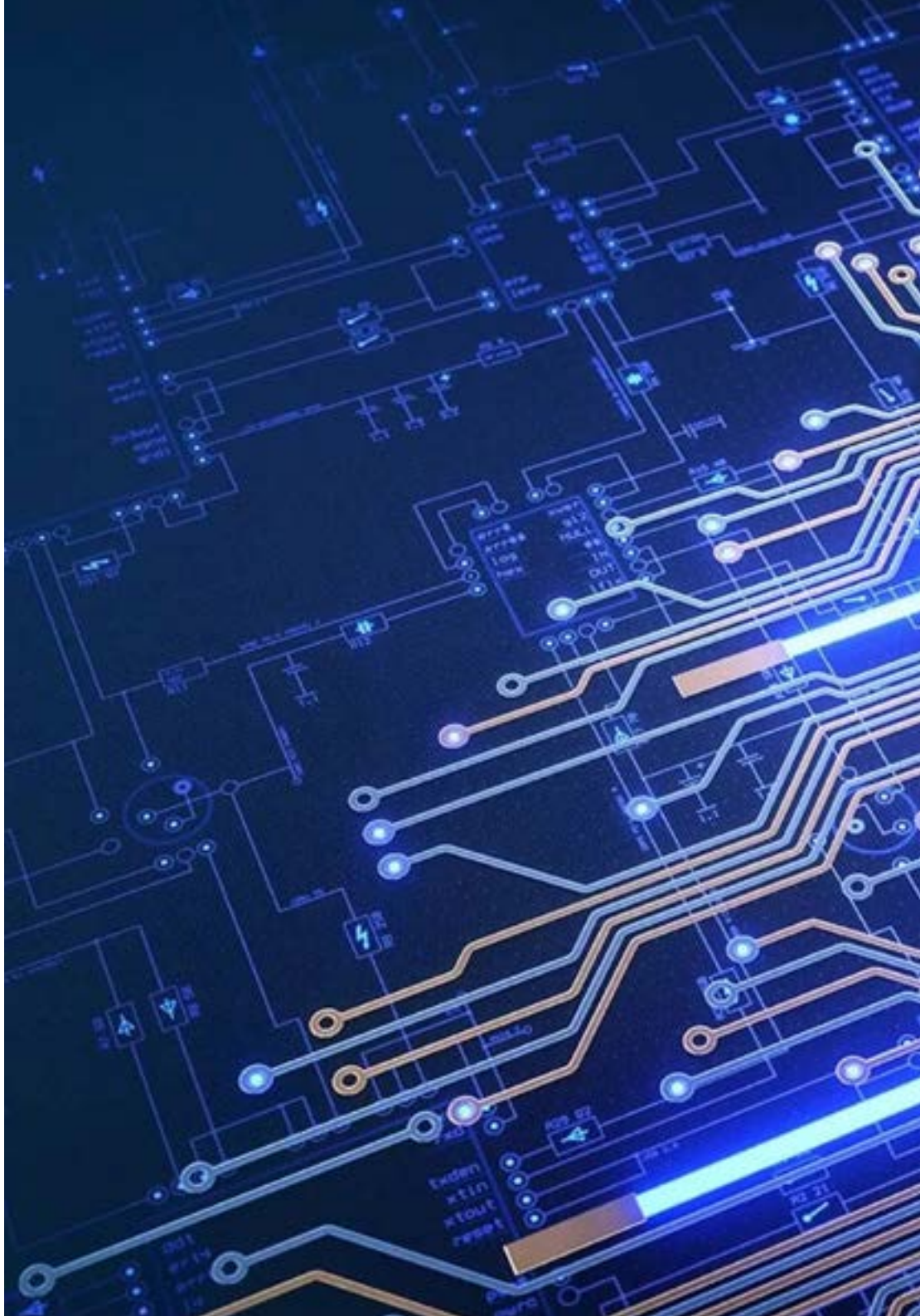
- Desarrollar los elementos clave de un Sistema Distribuido
- Examinar los elementos de seguridad aplicados en los Sistemas Distribuidos y su necesidad
- Presentar los diferentes tipos de Sistemas Distribuidos más comúnmente utilizados, características, funcionalidades y los problemas a resolver
- Demostrar el Teorema CAP aplicable a los Sistemas Distribuidos: Consistency (Consistencia), Availability (Disponibilidad) y Partition Tolerance (Tolerancia a fallos)

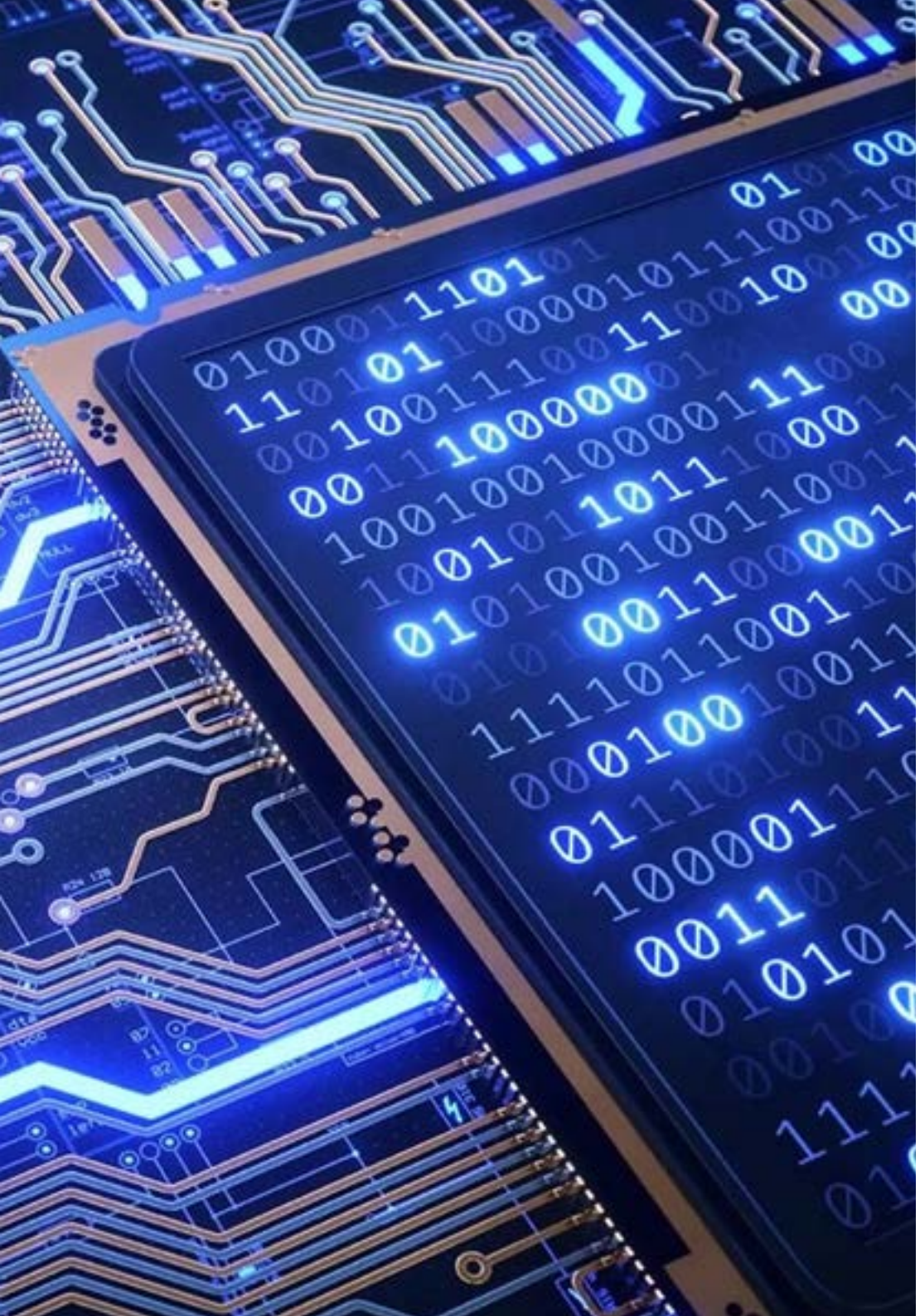
Módulo 8. Computación paralela aplicada a entornos *cloud*

- Desarrollar el Paradigma de la Computación en la Nube
- Identificar las distintas aproximaciones en base al grado de Automatización y Servicio
- Analizar las principales piezas de una arquitectura en la nube
- Establecer las diferencias con una Arquitectura On-Premise
- Analizar las distintas opciones de Despliegue Cloud: Multi-Cloud, Hybrid Cloud
- Profundizar en los beneficios inherentes a la Computación en la Nube
- Desarrollar los principios de la economía de Computación en la Nube: Paso de CAPEX a OPEX
- Valorar la oferta comercial en los distintos proveedores Cloud
- Evaluar las capacidades de Supercomputación en la Nube
- Examinar la Seguridad en la Computación en la Nube

Módulo 9. Modelos y semántica formal. Programación orientada a computación distribuida

- Identificar las bondades de la semántica formal
- Examinar como la semántica formal ayuda a la programación orientada a computación distribuida
- Concretar las posibilidades de la semántica formal aplicada a la programación orientada a la computación distribuida
- Desarrollar en profundidad las principales herramientas en cuanto a la viabilidad de los proyectos en el uso de esta tecnología
- Identificar lenguajes de programación en el modelo semántico





- Determinar cómo estos modelos semánticos nos ayudan con los lenguajes de programación
- Evaluar y comparar los modelos de computación
- Concretar el uso de los modelos distribuidos
- Presentar las herramientas de mercado más avanzadas para los proyectos

Módulo 10. Aplicaciones de la computación paralela y distribuida

- Demostrar el gran aporte de las aplicaciones en Computación Paralela y Distribuida a nuestro entorno
- Determinar las Arquitecturas de referencia en el mercado
- Evaluar los beneficios de estos casos de uso
- Presentar soluciones de éxito en el mercado
- Demostrar por qué es importante para evaluar el cambio climático
- Determinar la importancia actual de las GPU
- Presentar el impacto de esta tecnología en las redes eléctricas
- Explorar motores distribuidos para dar servicio a nuestros clientes
- Conocer las bondades de los motores distribuidos para reportar beneficios a nuestras empresas
- Presentar ejemplos de Base de datos en memoria y su importancia
- Examinar cómo ayudan estos modelos a la Medicina

03

Competencias

Con el adecuado conocimiento en este campo de la informática, el profesional puede destacarse y colocarse al frente de equipos de programación enteros. Muestra de ello son los propios docentes, que acumulan una distinguida experiencia en cargos de responsabilidad y al frente de numerosos proyectos informáticos diversos. Por este motivo, el Máster Título Propio en Computación Paralela y Distribuida no solo se centra en las claves esenciales de esta materia, sino que impulsa al informático con una serie de competencias distintivas y únicas que le hagan destacar en su trabajo diario.

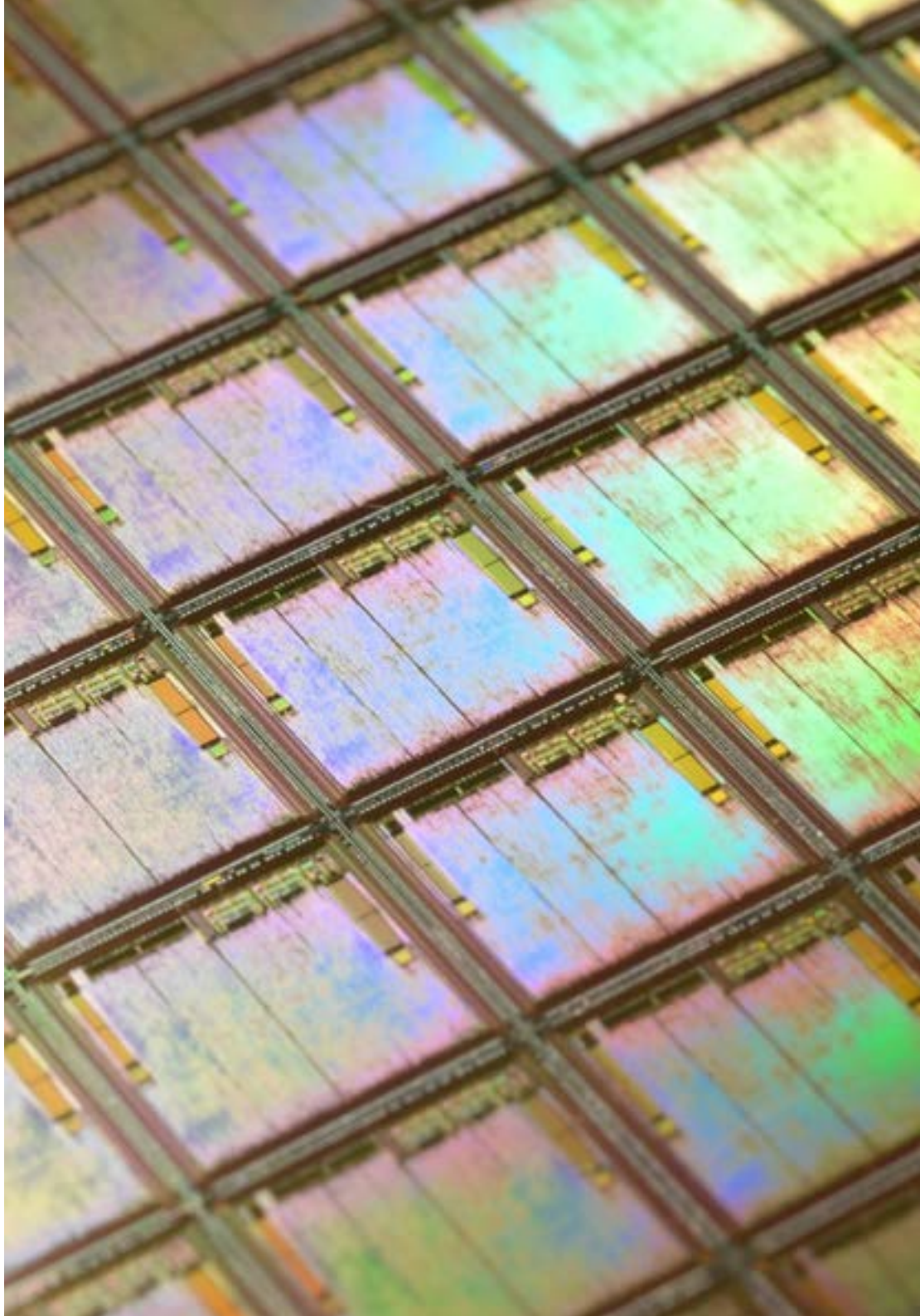


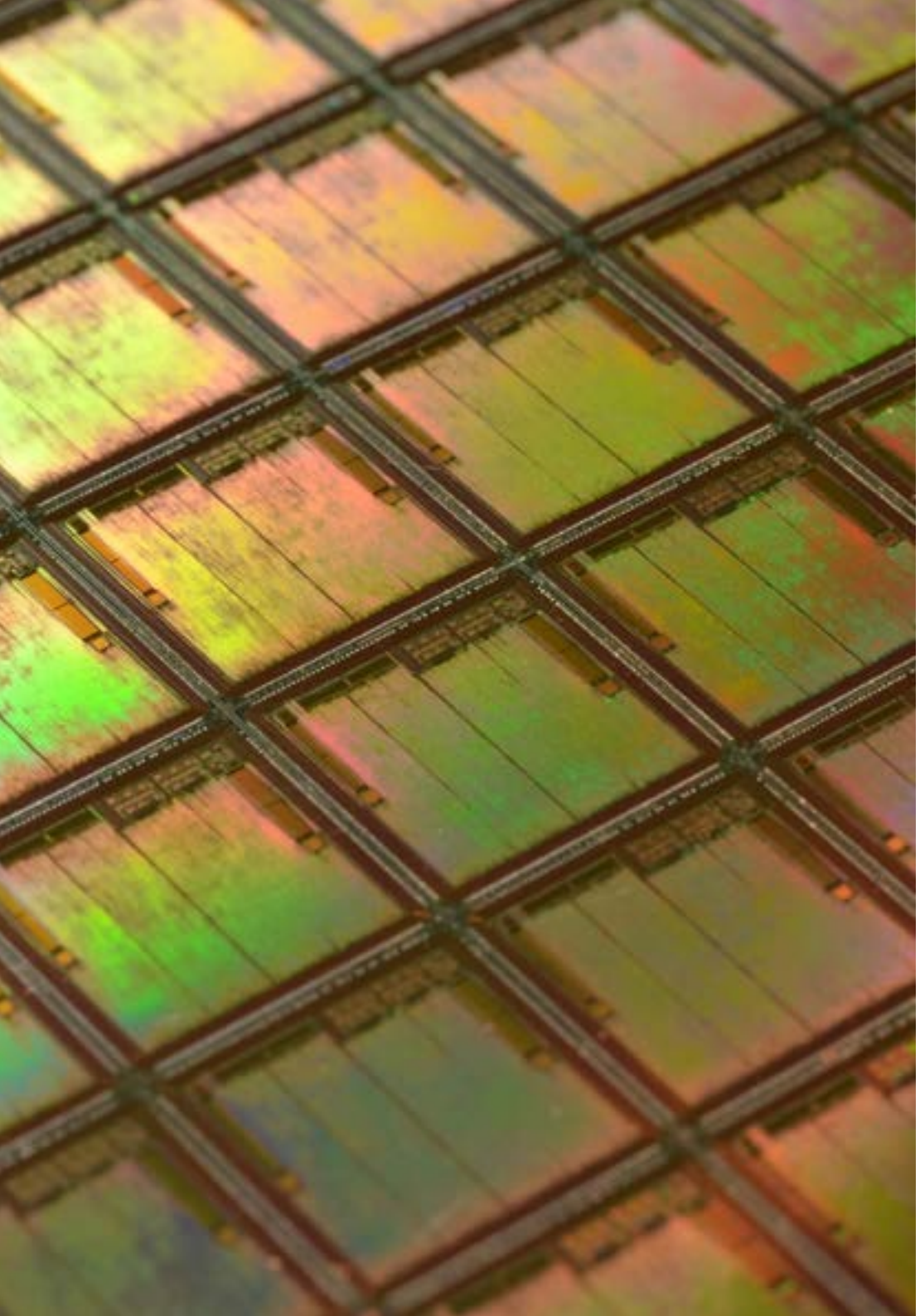


Esta titulación será la clave de tu ascenso meteórico hacia los puestos de mayor responsabilidad en programación y desarrollo de todo tipo de software”

Competencias generales

- Desarrollar conocimiento especializado sobre los diferentes niveles de Paralelismo
- Analizar una estrategia de Paralelización basándose en las Métricas de Rendimiento
- Determinar las características principales de la computación paralela y distribuida antes de abordar la comunicación y coordinación que ocurre entre sus componentes
- Demostrar que en este tipo de sistemas pueden suceder comunicaciones entre procesos, llamadas remotas o comunicaciones indirectas
- Determinar los aspectos que penalizan al desempeño de aplicaciones paralelas
- Analizar técnicas avanzadas de optimización de código en paralelo, de optimización de comunicaciones en sistemas de memoria distribuida, control de afinidad, balanceo de carga y gestión de entrada/salida paralela
- Examinar modelos de Programación Híbrida para sistemas con varios aceleradores hardware y modelos de programación híbrida para sistemas con memoria compartida/distribuida





Competencias específicas

- Determinar los problemas de escalabilidad y rendimiento que pueden ser solucionados mediante la descomposición de procesos en paralelo
- Analizar las características de una propuesta de paralelismo de memoria compartida, paralelismo de paso de mensajes y paralelismo en GPUs, así como el escenario hibrido
- Establecer la necesidad de resiliencia de procesos y el modelo de actores en la solución de problemas actuales de computo
- Exponer ejemplos o casos en donde la descomposición paralela ha tenido una correcta aplicación en rendimiento o escalabilidad
- Analizar y diseñar Algoritmos Paralelos
- Compilar las Primitivas principales de MPI, OpenMP, OpenCL/CUDA
- Examinar los Procesos Vectoriales y Matriciales
- Analizar la programación paralela y distribuida, lenguajes de programación, herramientas y patrones de diseño
- Identificar los elementos que permiten la interconexión de las redes distribuidas
- Fundamentar los pasos del diseño de un Sistema Distribuido
- Evaluar los diferentes tipos de replicación de datos en los sistemas existentes
- Compilar los enfoques prácticos de seguridad aplicables

04

Dirección del curso

En su búsqueda incesante de la mayor calidad docente posible, TECH ha reunido en este programa a un compendio de ingenieros informáticos con experiencia internacional en multitud de proyectos relacionados con la Computación Paralela y Distribuida. Así, el alumno adquiere unos conocimientos que no solo están centrados en su mejora profesional, sino que además provienen del propio conocimiento experto del cuerpo docente a la hora de gestionar equipos informáticos multidisciplinares.





Desarrolla tu potencial académico y profesional con los mejores docentes, que estarán comprometidos contigo durante toda la titulación”

Dirección



D. Olalla Bonal, Martín

- Client Technical Specialist Blockchain en IBM
- Director de arquitectura blockchain Hyperledger y Ethereum en Blocknitive
- Director del área blockchain en PSS Tecnologías de la Información
- Chief Information Officer en ePETID – Global Animal Health
- IT arquitecto de infraestructura en Bankia – wdoIT (IBM – Bankia Join Venture)
- Director de proyectos y gerente en Daynet servicios integrales
- Director de tecnología en Wiron Construcciones Modulares
- Jefe del departamento informático en Dayfisa
- Responsable del departamento informático en Dell Computer, Majsa e Hippo Viajes
- Técnico electrónico en IPFP Juan de la Cierva

Profesores

D. Villot Guisán, Pablo

- Arquitecto responsable de numerosos proyectos tecnológicos en KPMG, trabajando con Global Cars SCF, Zurich Santander, Flash Allfunds, Asentify SCF y Plataforma Next SCF
- Arquitecto responsable de diversos proyectos en Everis, con trabajos en Open Banking Platform, Proyecto Optimum: Centro de Excelencia Cloud SCIB y H2020: Provenance
- Desarrollo de soluciones blockchain para el Ministerio de la Presidencia, Repsol e Iberia
- Graduado en Ingeniería Informática en la Universidad de La Coruña
- Certificado Microsoft en MSCA: Cloud Platform

D. Gozalo Fernández, Juan Luis

- Ingeniero Informático
- Director Blockchain DevOps en Alastria
- Director Desarrollo Aplicación Móvil Tinkerlink en Cronos Telecom
- Director Informática en Banco Santander
- Director Tecnología Gestión de Servicio IT en Barclays Bank España
- Licenciado en Ingeniería Superior Informática por la Universidad Nacional Educación a Distancia (UNED)

D. Gómez Gómez, Borja

- Business Development Manager for Cloud Innovation en Oracle
- Jefe de Blockchain y soluciones de arquitectura para preventas en Paradigma Digital
- Arquitecto senior IT en Atmira
- Arquitecto SOA y consultor en TCP SI
- Analista y consultor en Everis
- Licenciado en Ingeniería Informática en la Universidad Complutense de Madrid
- Máster en Science Computer Engineering en la Universidad Complutense de Madrid

D. Almendras Aruzamen, Luis Fernando

- Líder de desarrollo de software en Intersoft
- Ingeniero de datos en Indizen
- Ingeniero de datos y *business intelligence* en Tecnología y Personas
- Ingeniero de soporte de bases de datos, *big data* y *business intelligence* en *Equinix*
- Product Manager y responsable del área de business analytics en Goja
- Responsable del área de datrawarehouse y big data en Viva
- Licenciado en Informática por la Universidad Mayor de San Simón
- Doctorando en Ingeniería Informática por la Universidad Complutense de Madrid
- Máster en Ingeniería Informática por la Universidad Complutense de Madrid
- Máster en Sistemas de Información y Gestión de Tecnologías por la Universidad Mayor de San Simón

Dr. Blanco, Eduardo

- Docente de la Universidad Simón Bolívar
- Doctor en Computación por la Universidad Simón Bolívar
- Magister en Ciencias de la Computación por la Universidad Simón Bolívar
- Ingeniero en Computación por la Universidad Simón Bolívar

Dra. Carratalá Sáez, Rocío

- Doctora en Informática por la Universitat Jaume I
- Graduada en Matemática Computacional por la Universitat Jaume I
- Máster en Computación Paralela y Distribuida por la Universitat Politècnica de València
- 10 cursos de especialización relacionados con ciencias de la computación, matemáticas y herramientas para la investigación académica

05

Estructura y contenido

Para facilitar una mejor labor de estudio y adquisición de competencias, TECH ha incorporado en este programa universitario la metodología pedagógica más efectiva. Gracias al *relearning,* el alumno ve considerablemente reducido el tiempo que tiene que invertir en adquirir los conocimientos más importantes del programa, lo que además se ve reforzado por la gran cantidad de material audiovisual, lecturas complementarias y ejercicios prácticos que ayudan a afianzar toda la materia.





Ejercicios prácticos basados en casos reales y vídeos en detalle elaborados por los propios profesores serán la clave de tu éxito en este programa universitario"

Módulo 1. Paralelismo en computación paralela y distribuida

1.1. Procesamiento paralelo
- 1.1.1. Procesamiento paralelo
- 1.1.2. Procesamiento paralelo en computación. Finalidad
- 1.1.3. Procesamiento paralelo. Análisis

1.2. Sistema paralelo
- 1.2.1. El sistema paralelo
- 1.2.2. Niveles de paralelismo
- 1.2.3. Composición del sistema paralelo

1.3. Arquitecturas de procesadores
- 1.3.1. Complejidad del procesador
- 1.3.2. Arquitectura de procesadores. Modo de operación
- 1.3.3. Arquitectura de procesadores. Organización de la memoria

1.4. Redes en el procesamiento paralelo
- 1.4.1. Modo de operación
- 1.4.2. Estrategia de control
- 1.4.3. Técnicas de conmutación
- 1.4.4. Topología

1.5. Arquitecturas paralelas
- 1.5.1. Algoritmos
- 1.5.2. Acoplamiento
- 1.5.3. Comunicación

1.6. Rendimiento de la computación paralela
- 1.6.1. Evolución del rendimiento
- 1.6.2. Medidas de *performance*
- 1.6.3. Computación Paralela. Casos de estudio

1.7. Taxonomía de Flynn
- 1.7.1. MIMD: Memoria compartida
- 1.7.2. MIMD: Memoria distribuida
- 1.7.3. MIMD: Sistemas híbridos
- 1.7.4. Flujo de datos

1.8. Formas de paralelismo: TLP (*Thread Level Paralelism*)
- 1.8.1. Formas de paralelismo: TLP (*Thread Level Paralelism*)
- 1.8.2. *Coarse grain*
- 1.8.3. *Fine grain*
- 1.8.4. SMT

1.9. Formas de paralelismo: DLP (*Data Level Paralelism*)
- 1.9.1. Formas de paralelismo: DLP (*Data Level Paralelism*)
- 1.9.2. *Short vector processing*
- 1.9.3. *Vector processors*

1.10. Formas de Paralelismo: ILP (*Instruction Level Paralelism*)
- 1.10.1. Formas de Paralelismo: ILP (*Instruction Level Paralelism)*
- 1.10.2. Procesador segmentado
- 1.10.3. Procesador superescalar
- 1.10.4. Procesador *Very Long Instruction Word* (VLIW)

Módulo 2. Descomposición en paralelo en computación paralela y distribuida

2.1. Descomposición en paralelo
- 2.1.1. Procesamiento paralelo
- 2.1.2. Arquitecturas
- 2.1.3. Supercomputadoras

2.2. Hardware paralelo y software paralelo
- 2.2.1. Sistemas en serie
- 2.2.2. Hardware paralelo
- 2.2.3. Software paralelo
- 2.2.4. Entrada y salida
- 2.2.5. Rendimiento

2.3. Escalabilidad paralela y problemas de rendimiento recurrentes
- 2.3.1. Paralelismo
- 2.3.2. Escalabilidad en paralelo
- 2.3.3. Problemas recurrentes de rendimiento

2.4. Paralelismo de memoria compartida
- 2.4.1. Paralelismo de memoria compartida
- 2.4.2. OpenMP y Pthreads
- 2.4.3. Paralelismo de memoria compartida. Ejemplos

2.5. Unidad de Procesamiento Gráfico (GPU)
- 2.5.1. Unidad de Procesamiento Gráfico (GPU)
- 2.5.2. Arquitectura Unificada de Dispositivos Computacionales (CUDA)
- 2.5.3. Arquitectura Unificada de Dispositivos Computacionales. Ejemplos

2.6. Sistemas de paso de mensajes
- 2.6.1. Sistemas de paso de mensajes
- 2.6.2. MPI. Interfaz de paso de mensajes
- 2.6.3. Sistemas de paso de mensajes. Ejemplos

2.7. Paralelización híbrida con MPI y OpenMP
- 2.7.1. La programación híbrida
- 2.7.2. Modelos de programación MPI/OpenMP
- 2.7.3. Descomposición y mapeo híbrido

2.8. Computación MapReduce
- 2.8.1. Hadoop
- 2.8.2. Otros sistemas de cómputo
- 2.8.3. Computación Paralela. Ejemplos

2.9. Modelo de actores y procesos reactivos
- 2.9.1. Modelo de actores
- 2.9.2. Procesos reactivos
- 2.9.3. Actores y procesos reactivos. Ejemplos

2.10. Escenarios de Computación Paralela
- 2.10.1. Procesamiento de audio e imágenes
- 2.10.2. Estadística/minería de datos
- 2.10.3. Ordenación paralela
- 2.10.4. Operaciones matriciales paralelas

Módulo 3. Comunicación y coordinación en sistemas de computación

3.1. Procesos de Computación Paralela y Distribuida
- 3.1.1. Procesos de Computación Paralela y Distribuida
- 3.1.2. Procesos e hilos
- 3.1.3. Virtualización
- 3.1.4. Clientes y servidores

3.2. Comunicación en Computación Paralela
- 3.2.1. Computación en Computación Paralela
- 3.2.2. Protocolos por capas
- 3.2.3. Comunicación en Computación Paralela. Tipología

3.3. Llamada a procedimiento remoto
- 3.3.1. Funcionamiento de RPC (*Remote Procedure Call*)
- 3.3.2. Paso de parámetros
- 3.3.3. RPC asíncrono
- 3.3.4. Procedimiento remoto. Ejemplos

3.4. Comunicación orientada a mensajes
- 3.4.1. Comunicación transitoria orientada a mensajes
- 3.4.2. Comunicación persistente orientada a mensajes
- 3.4.3. Comunicación orientada a mensajes. Ejemplos

3.5. Comunicación orientada a flujos
- 3.5.1. Soporte para medios continuos
- 3.5.2. Flujos y calidad de servicio
- 3.5.3. Sincronización de flujos
- 3.5.4. Comunicación orientada a flujos. Ejemplos

3.6. Comunicación de multidifusión
- 3.6.1. Multidifusión a nivel de aplicación
- 3.6.2. Difusión de datos basada en rumores
- 3.6.3. Comunicación de multidifusión. Ejemplos

3.7. Otros tipos de comunicación
- 3.7.1. Invocación de métodos remotos
- 3.7.2. Servicios web / SOA / REST
- 3.7.3. Notificación de eventos
- 3.7.4. Agentes móviles

3.8. Servicio de nombres
- 3.8.1. Servicios de nombres en computación
- 3.8.2. Servicios de nombres y sistema de dominio de nombres
- 3.8.3. Servicios de directorio

3.9. Sincronización
- 3.9.1. Sincronización de relojes
- 3.9.2. Relojes lógicos, exclusión mutua y posicionamiento global de los nodos
- 3.9.3. Elección de algoritmos

3.10. Comunicación. Coordinación y acuerdo
- 3.10.1. Coordinación y acuerdo
- 3.10.2. Coordinación y acuerdo. Consenso y problemas
- 3.10.3. Comunicación y coordinación. Actualidad

Módulo 4. Análisis y programación de algoritmos paralelos

4.1. Algoritmos paralelos
- 4.1.1. Descomposición de problemas
- 4.1.2. Dependencias de datos
- 4.1.3. Paralelismo implícito y explícito

4.2. Paradigmas de Programación Paralela
- 4.2.1. Programación Paralela con memoria compartida
- 4.2.2. Programación Paralela con memoria distribuida
- 4.2.3. Programación Paralela híbrida
- 4.2.4. Computación heterogénea – CPU + GPU
- 4.2.5. Computación cuántica. Nuevos modelos de programación con paralelismo implícito

4.3. Programación Paralela con memoria compartida
- 4.3.1. Modelos de Programación Paralela con memoria compartida
- 4.3.2. Algoritmos paralelos con memoria compartida
- 4.3.3. Librerías para Programación Paralela con memoria compartida

4.4. OpenMP
- 4.4.1. OpenMP
- 4.4.2. Ejecución y depuración de programas con OpenMP
- 4.4.3. Algoritmos paralelos con memoria compartida en OpenMP

4.5. Programación Paralela por paso de mensajes
- 4.5.1. Primitivas de paso de mensajes
- 4.5.2. Operaciones de comunicación y computación colectiva
- 4.5.3. Algoritmos paralelos por paso de mensajes
- 4.5.4. Librerías para Programación Paralela con paso de mensajes

4.6. *Message Passing Interface* (MPI)
- 4.6.1. *Message Passing Interface* (MPI)
- 4.6.2. Ejecución y depuración de programas con MPI
- 4.6.3. Algoritmos paralelos por paso de mensajes con MPI

4.7. Programación paralela híbrida
- 4.7.1. Programación paralela híbrida
- 4.7.2. Ejecución y depuración de programas paralelos híbridos
- 4.7.3. Algoritmos paralelos híbridos MPI-OpenMP

4.8. Programación paralela con computación heterogénea
- 4.8.1. Programación paralela con computación heterogénea
- 4.8.2. CPU vs. GPU
- 4.8.3. Algoritmos paralelos con computación heterogénea

4.9. OpenCL y CUDA
- 4.9.1. OpenCL vs. CUDA
- 4.9.2. Ejecución y depuración de programas paralelos con computación heterogénea
- 4.9.3. Algoritmos paralelos con computación heterogénea

4.10. Diseño de algoritmos paralelos
- 4.10.1. Diseño de algoritmos paralelos
- 4.10.2. Problema y contexto
- 4.10.3. Paralelización automática vs. Paralelización manual
- 4.10.4. Particionamiento del problema
- 4.10.5. Comunicaciones en computación

Módulo 5. Arquitecturas paralelas

5.1. Arquitecturas paralelas
- 5.1.1. Sistemas paralelos. Clasificación
- 5.1.2. Fuentes de paralelismo
- 5.1.3. Paralelismo y procesadores

5.2. Rendimiento de los sistemas paralelos
- 5.2.1. Magnitudes y medidas de rendimiento
- 5.2.2. *Speed-up*
- 5.2.3. Granularidad de los sistemas paralelos

5.3. Procesadores vectoriales
- 5.3.1. Procesador vectorial básico
- 5.3.2. Memoria entrelazada o intercalada
- 5.3.3. Rendimiento de los procesadores vectoriales

5.4. Procesadores matriciales
- 5.4.1. Organización básica
- 5.4.2. Programación en procesadores matriciales
- 5.4.3. Programación en procesadores matriciales. Ejemplo práctico

5.5. Redes de interconexión
- 5.5.1. Redes de interconexión
- 5.5.2. Topología, control de flujo y encaminamiento
- 5.5.3. Redes de interconexión. Clasificación según topología

5.6. Multiprocesadores
- 5.6.1. Redes de interconexión para multiprocesadores
- 5.6.2. Consistencia de memoria y cachés
- 5.6.3. Protocolos de sondeo

5.7. Sincronización
- 5.7.1. Cerrojos (exclusión mutua)
- 5.7.2. Eventos de sincronización P2P
- 5.7.3. Eventos de sincronización globales

5.8. Multicomputadores
- 5.8.1. Redes de interconexión para multicomputadores
- 5.8.2. Capa de conmutación
- 5.8.3. Capa de encaminamiento

5.9. Arquitecturas avanzadas
- 5.9.1. Máquinas de flujo de datos
- 5.9.2. Otras arquitecturas

5.10. Programación paralela y distribuida
- 5.10.1. Lenguajes para programación paralela
- 5.10.2. Herramientas de programación paralela
- 5.10.3. Patrones de diseño
- 5.10.4. Concurrencia de lenguajes de programación paralela y distribuida

Módulo 6. Desempeño en paralelo

6.1. Desempeño de algoritmos paralelos
- 6.1.1. Ley de Ahmdal
- 6.1.2. Ley de Gustarfson
- 6.1.3. Métricas de desempeño y escalabilidad de algoritmos paralelos

6.2. Comparativa de algoritmos paralelos
- 6.2.1. *Benchmarking*
- 6.2.2. Análisis matemático de algoritmos paralelos
- 6.2.3. Análisis asintótico de algoritmos paralelos

6.3 Restricciones de los recursos hardware
- 6.3.1. Memoria
- 6.3.2. Procesamiento
- 6.3.3. Comunicaciones
- 6.3.4. Particionamiento dinámico de recursos

6.4 Desempeño de programas paralelos con memoria compartida
- 6.4.1. División óptima en tareas
- 6.4.2. Afinidad de *Threads*
- 6.4.3. Paralelismo SIMD
- 6.4.4. Programas paralelos con memoria compartida. Ejemplos

6.5. Desempeño de programas paralelos por paso de mensajes
- 6.5.1. Desempeño de programas paralelos por paso de mensajes
- 6.5.2. Optimización de comunicaciones en MPI
- 6.5.3. Control de afinidad y balanceo de carga
- 6.5.4. I/O Paralela
- 6.5.5. Programas Paralelos por Paso de Mensajes. Ejemplos

6.6. Desempeño de programas paralelos híbridos
- 6.6.1. Desempeño de programas paralelos híbridos
- 6.6.2. Programación híbrida para sistemas de memoria compartida/distribuida
- 6.6.3. Programas paralelos híbridos. Ejemplos

6.7. Desempeño de programas con computación heterogénea
- 6.7.1. Desempeño de programas con computación heterogénea
- 6.7.2. Programación híbrida para sistemas con varios aceleradores hardware
- 6.7.3. Programas con computación heterogénea. Ejemplos

6.8. Análisis de rendimiento de algoritmos paralelos
- 6.8.1. Análisis de rendimiento de algoritmos paralelos
- 6.8.2. Análisis de Rendimiento de algoritmos paralelos. Herramientas
- 6.8.3. Análisis de Rendimiento de algoritmos paralelos. Recomendaciones

6.9. Patrones paralelos
- 6.9.1. Patrones paralelos
- 6.9.2. Principales patrones paralelos
- 6.9.3. Patrones paralelos. Comparativa

6.10. Programas paralelos de alto rendimiento
- 6.10.1. Proceso
- 6.10.2. Programas paralelos de alto rendimiento
- 6.10.3. Programas paralelos de alto rendimiento. Usos reales

Módulo 7. Sistemas distribuidos en computación

7.1. Sistemas Distribuidos
- 7.1.1. Sistemas Distribuidos (S.D)
- 7.1.2. Demostración del teorema de CAP (o Conjetura de Brewer)
- 7.1.3. Falacias de la programación sobre Sistemas Distribuidos
- 7.1.4. Computación ubicua

7.2. Sistemas Distribuidos. Características
- 7.2.1. Heterogeneidad
- 7.2.2. Extensibilidad
- 7.2.3. Seguridad
- 7.2.4. Escalabilidad
- 7.2.5. Tolerancia a fallos
- 7.2.6. Concurrencia
- 7.2.7. Transparencia

7.3. Redes e interconexión de redes distribuidas
- 7.3.1. Redes y los Sistemas Distribuidos. Prestaciones de las redes.
- 7.3.2. Redes disponibles para crear un sistema distribuido. Tipología
- 7.3.3. Protocolos de red Distribuidos vs. Centralizados
- 7.3.4. Interconexión de redes. Internet

7.4. Comunicación entre procesos distribuidos
- 7.4.1. Comunicación entre nodos de un SD. Problemas y fallas
- 7.4.2. Mecanismos que implementar sobre RPC y RDMA para evitar fallas
- 7.4.3. Mecanismos que implementar en el software para evitar fallas

7.5. Diseño de Sistemas Distribuidos
- 7.5.1. Diseño eficiente de Sistemas Distribuidos (S.D)
- 7.5.2. Patrones para la programación en Sistemas Distribuidos (S.D)
- 7.5.3. Arquitectura Orientada a Servicios (*Service Oriented Architecture* (SOA))
- 7.5.4. *Service Orchestration y Microservices Data Management*

7.6. Operación de Sistemas Distribuidos
- 7.6.1. Monitorización de los sistemas
- 7.6.2. Implantación de un sistema de trazas (*logging*) eficiente en un SD
- 7.6.3. Monitorización en redes distribuidas
- 7.6.4. Uso de una herramienta de monitorización para un SD Prometheus y Grafana

7.7. Replicación de sistemas
- 7.7.1. Replicación de sistemas. Tipologías
- 7.7.2. Arquitecturas inmutables
- 7.7.3. Los sistemas contenedores y sistemas virtualizadores como Sistemas Distribuidos
- 7.7.4. Las redes *blockchain* como Sistemas Distribuidos

7.8. Sistemas multimedia distribuidos
- 7.8.1. Intercambio distribuido de imágenes y videos. Problemática
- 7.8.2. Servidores de objetos multimedia
- 7.8.3. Topología de red para un sistema multimedia
- 7.8.4. Análisis de los sistemas multimedia distribuidos: Netflix, Amazon, Spotify, etc.
- 7.8.5. Los sistemas distribuidos multimedia en educación

7.9. Sistemas de ficheros distribuidos
- 7.9.1. Intercambio distribuido de ficheros. Problemática
- 7.9.2. Aplicabilidad del teorema de CAP a las bases de datos
- 7.9.3. Sistemas de ficheros web distribuidos: Akamai
- 7.9.4. Sistemas de ficheros documentales distribuidos IPFS
- 7.9.5. Sistemas de Bases de datos distribuidas

7.10. Enfoques de Seguridad en Sistemas Distribuidos
- 7.10.1. Seguridad en Sistemas Distribuidos
- 7.10.2. Ataques conocidos a Sistemas Distribuidos
- 7.10.3. Herramientas para probar la seguridad de un SD

Módulo 8. Computación paralela aplicada a Entornos Cloud

8.1. Computación en la nube
- 8.1.1. Estado del arte del panorama IT
- 8.1.2. La "nube"
- 8.1.3. Computación en la nube

8.2. Seguridad y resiliencia en la nube
- 8.2.1. Regiones, zonas de disponibilidad y fallo
- 8.2.2. Administración de los *tenant* o cuentas de *cloud*
- 8.2.3. Identidad y control de acceso en la nube

8.3. *Networking* en la nube
- 8.3.1. Redes virtuales definidas por software
- 8.3.2. Componentes de red de una red definida por software
- 8.3.3. Conexión con otros sistemas

8.4. Servicios en la nube
- 8.4.1. Infraestructura como servicio
- 8.4.2. Plataforma como servicio
- 8.4.3. Computación *serverless*
- 8.4.4. Software como servicio

8.5. Almacenamiento en la nube
- 8.5.1. Almacenamiento de bloques en la nube
- 8.5.2. Almacenamiento de ficheros en la nube
- 8.5.3. Almacenamiento de objetos en la nube

8.6. Interacción y monitorización de la nube
- 8.6.1. Monitorización y gestión de la nube
- 8.6.2. Interacción con la nube: consola de administración
- 8.6.3. Interacción con *Command Line Interface*
- 8.6.4. Interacción basada en APIs

8.7. Desarrollo *cloud-native*
- 8.7.1. Desarrollo nativo en *cloud*
- 8.7.2. Contenedores y plataformas de orquestación de contenedores
- 8.7.3. Integración continua en la nube
- 8.7.4. Uso de eventos en la nube

8.8. Infraestructura como código en la nube
- 8.8.1. Automatización de la gestión y el aprovisionamiento en la nube
- 8.8.2. Terraform
- 8.8.3. Integración con *scripting*

8.9. Creación de una infraestructura híbrida
- 8.9.1. Interconexión
- 8.9.2. Interconexión con *datacenter*
- 8.9.3. Interconexión con otras nubes

8.10. Computación de alto rendimiento
- 8.10.1. Computación de alto rendimiento
- 8.10.2. Creación de un clúster de alto rendimiento
- 8.10.3. Aplicación de la computación de alto rendimiento

Módulo 9. Modelos y semántica formal. Programación orientada a computación distribuida

9.1. Modelo semántico de datos
- 9.1.1. Modelos semánticos de datos
- 9.1.2. Modelos semánticos de datos. Propósitos
- 9.1.3. Modelos semánticos de datos. Aplicaciones

9.2. Modelo semántico de lenguajes de programación
- 9.2.1. Procesamiento de lenguajes
- 9.2.2. Traducción e interpretación
- 9.2.3. Lenguajes híbridos

9.3. Modelos de computación
- 9.3.1. Computación monolítica
- 9.3.2. Computación paralela
- 9.3.3. Computación distribuida
- 9.3.4. Computación cooperativa (P2P)

9.4. Computación paralela
- 9.4.1. Arquitectura paralela
- 9.4.2. Hardware
- 9.4.3. Software

9.5. Modelo distribuido. *Grid Computing* o computación en malla
- 9.5.1. Arquitectura *Grid Computing*
- 9.5.2. Arquitectura *Grid Computing*. Análisis
- 9.5.3. Arquitectura *Grid Computing*. Aplicaciones

9.6. Modelo Distribuido. *Cluster Computing* o computación en clúster
- 9.6.1. Arquitectura *Cluster Computing*
- 9.6.2. Arquitectura *Cluster Computing*. Análisis
- 9.6.3. Arquitectura *Cluster Computing*. Aplicaciones

9.7. *Cluster Computing*. Herramientas actuales para implementarlo. Hipervisores
- 9.7.1. Competidores del mercado
- 9.7.2. VMware Hipervisor
- 9.7.3. Hyper-V

9.8. Modelo distribuido. *Cloud Computing* o computación en *cloud*
- 9.8.1. Arquitectura *Cloud Computing*
- 9.8.2. Arquitectura *Cloud Computing*. Análisis
- 9.8.3. Arquitectura *Cloud Computing*. Aplicaciones

9.9. Modelo distribuido. *Cloud Computing* Amazon
- 9.9.1. *Cloud Computing* Amazon. Funcionalidades
- 9.9.2. *Cloud Computing* Amazon. Licenciamientos
- 9.9.3. *Cloud Computing* Amazon. Arquitecturas de referencia

9.10. Modelo distribuido. *Cloud Computing* Microsoft
- 9.10.1. *Cloud Computing* Microsoft. Funcionalidades
- 9.10.2. *Cloud Computing* Microsoft. Licenciamientos
- 9.10.3. *Cloud Computing* Microsoft. Arquitecturas de referencia

Módulo 10. Aplicaciones de la Computación Paralela y Distribuida

10.1. La Computación Paralela y Distribuida en las aplicaciones actuales
- 10.1.1. Hardware
- 10.1.2. Software
- 10.1.3. Importancia de los tiempos

10.2. Clima. Cambio climático
- 10.2.1. Aplicaciones de clima. Fuentes de datos
- 10.2.2. Aplicaciones de clima. Volúmenes de datos
- 10.2.3. Aplicaciones de clima. Tiempo real

10.3. GPU Computación Paralela
- 10.3.1. GPU Computación Paralela
- 10.3.2. GPUs vs. CPU. Uso de GPU
- 10.3.3. GPU. Ejemplos

10.4. Smart *Grid*. Computación en las redes eléctricas
- 10.4.1. Smart *Grid*
- 10.4.2. Modelos conceptuales. Ejemplos
- 10.4.3. Smart *Grid*. Ejemplo

10.5. Motor distribuido. ElasticSearch
- 10.5.1. Motor distribuido. ElasticSearch
- 10.5.2. Arquitectura con ElasticSearch. Ejemplos
- 10.5.3. Motor distribuido. Casos de Uso

10.6. *Big Data Framework*
- 10.6.1. *Big Data Framework*
- 10.6.2. Arquitectura de herramientas avanzadas
- 10.6.3. *Big Data* en Computación Distribuida

10.7. Base de datos en memoria
- 10.7.1. Base de datos en memoria
- 10.7.2. Solución de Redis. Caso de éxito
- 10.7.3. Despliegue de soluciones con base de datos en memoria

10.8. *Blockchain*
- 10.8.1. Arquitectura *Blockchain*. Componentes
- 10.8.2. Colaboración entre nodos y consensos
- 10.8.3. Soluciones *Blockchain*. Implementaciones

10.9. Sistemas Distribuidos en medicina
- 10.9.1. Componentes de arquitectura
- 10.9.2. Sistemas Distribuidos en medicina. Funcionamiento
- 10.9.3. Sistemas Distribuido en medicina. Aplicaciones

10.10. Sistemas Distribuidos en el sector aéreo
- 10.10.1. Diseño de arquitectura
- 10.10.2. Sistemas Distribuidos en el sector aéreo. Funcionalidades de los componentes
- 10.10.3. Sistemas Distribuidos en el sector aéreo. Aplicaciones

07

Titulación

El Máster Título Propio en Computación Paralela y Distribuida garantiza, además de la capacitación más rigurosa y actualizada, el acceso a un título de Máster Tìtulo Propio expedido por TECH Universidad Tecnológica.





Supera con éxito este programa y recibe tu titulación universitaria sin desplazamientos ni farragosos trámites”

Este **Máster Título Propio en Computación Paralela y Distribuida** contiene el programa más completo y actualizado del mercado.

Tras la superación de la evaluación, el alumno recibirá por correo postal* con acuse de recibo su correspondiente título de **Máster Propio** emitido por **TECH Universidad Tecnológica.**

El título expedido por **TECH Universidad Tecnológica** expresará la calificación que haya obtenido en el Máster Título Propio, y reunirá los requisitos comúnmente exigidos por las bolsas de trabajo, oposiciones y comités evaluadores de carreras profesionales.

Título: **Máster Título Propio en Computación Paralela y Distribuida**

Nº Horas Oficiales: **1.500 h.**


tech universidad tecnológica

Otorga la presente

CONSTANCIA

a

C. _____, con documento de identificación nº_____

Por haber superado con éxito y acreditado el programa de

MÁSTER TÍTULO PROPIO

en

Computación Paralela y Distribuida

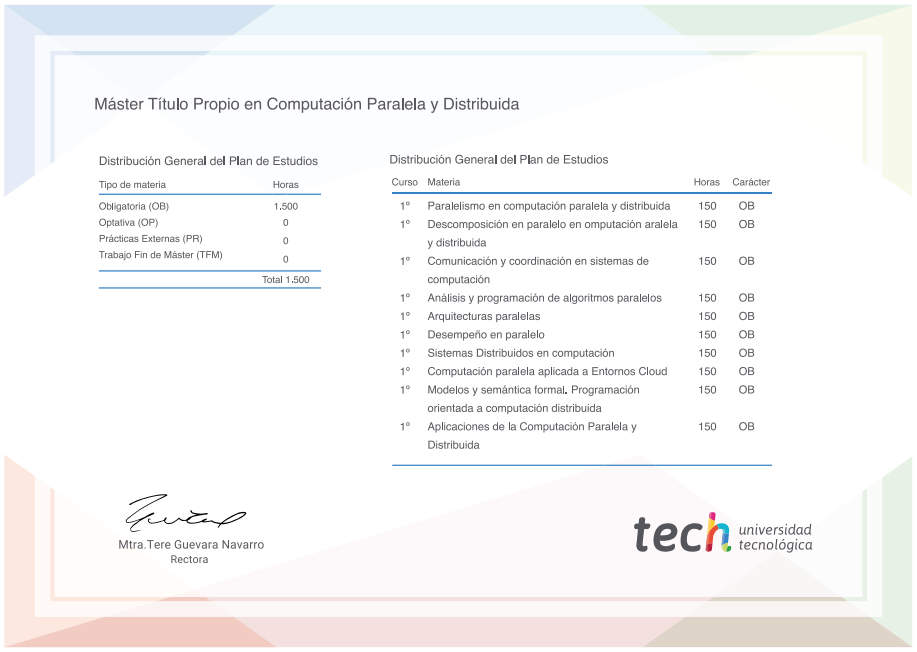
Se trata de un título propio de esta Universidad con una duración de 1.500 horas, con fecha de inicio dd/mm/aaaa y fecha de finalización dd/mm/aaaa.

TECH es una Institución Particular de Educación Superior reconocida por la Secretaría de Educación Pública a partir del 28 de junio de 2018.

A 17 de junio de 2020

Mtra.Tere Guevara Navarro
Rectora

Este título propio se deberá acompañar siempre del título universitario habilitante expedido por la autoridad competente para ejercer profesionalmente en cada país. código único TECH: AFWOR23S techtitute.com/titulos


Máster Título Propio en Computación Paralela y Distribuida

Distribución General del Plan de Estudios

Tipo de materia	Horas
Obligatoria (OB)	1.500
Optativa (OP)	0
Prácticas Externas (PR)	0
Trabajo Fin de Máster (TFM)	0
	Total 1.500

Distribución General del Plan de Estudios

Curso	Materia	Horas	Carácter
1º	Paralelismo en computación paralela y distribuida	150	OB
1º	Descomposición en paralelo en omputación aralela y distribuida	150	OB
1º	Comunicación y coordinación en sistemas de computación	150	OB
1º	Análisis y programación de algoritmos paralelos	150	OB
1º	Arquitecturas paralelas	150	OB
1º	Desempeño en paralelo	150	OB
1º	Sistemas Distribuidos en computación	150	OB
1º	Computación paralela aplicada a Entornos Cloud	150	OB
1º	Modelos y semántica formal. Programación orientada a computación distribuida	150	OB
1º	Aplicaciones de la Computación Paralela y Distribuida	150	OB

Mtra.Tere Guevara Navarro
Rectora

tech universidad tecnológica

*Apostilla de La Haya. En caso de que el alumno solicite que su título en papel recabe la Apostilla de La Haya, TECH EDUCATION realizará las gestiones oportunas para su obtención, con un coste adicional.

tech universidad tecnológica

Máster Título Propio
Computación Paralela y Distribuida

Modalidad: **Online**
Duración: **12 meses**
Titulación: **TECH Universidad Tecnológica**
Horas lectivas: **1500 h.**

Máster Título Propio

Computación Paralela y Distribuida



tech universidad tecnológica