Máster Título Propio

Computación y Lenguajes

Aval/Membresía

Association
for Computing
Machinery

tech
universidad

tech universidad

Máster Título Propio
Computación y Lenguajes

- Modalidad: **No escolarizada (100% en línea)**
- Duración: **12 meses**
- Titulación: **TECH Universidad**
- Horario: **a tu ritmo**
- Exámenes: **online**

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

Índice

01

Presentación del programa

La evolución constante de la inteligencia artificial, el *machine learning* y el procesamiento del lenguaje ha transformado el panorama de la computación a nivel global. Estos avances requieren profesionales capaces de comprender tanto los fundamentos teóricos como las aplicaciones prácticas de los lenguajes computacionales. Según la UNESCO, el 90 % de los empleos en el futuro cercano demandarán competencias digitales avanzadas. Esta realidad ha incrementado la necesidad de programas especializados que integren conocimientos en algoritmos, estructuras de datos y semántica de lenguajes. En este contexto, TECH presenta una titulación innovadora y completamente online en Computación y Lenguajes, orientada a proporcionar una preparación académica sólida y actualizada en un entorno digital y flexible.

Un programa exhaustivo y 100% online, exclusivo de TECH y con una perspectiva internacional respaldada por nuestra afiliación con la Association for Computing Machinery"

En un entorno tecnológico cada vez más marcado por la automatización, la inteligencia artificial y el procesamiento del lenguaje natural, comprender cómo se construyen, interpretan y optimizan los lenguajes computacionales se ha convertido en una necesidad clave para avanzar en múltiples áreas del conocimiento y la industria. La interacción entre humanos y máquinas, la eficiencia de los sistemas informáticos y la creación de nuevas herramientas digitales dependen, en gran medida, del dominio de estos lenguajes.

Este programa ofrece una visión integral de los pilares que sustentan la computación y los lenguajes, permitiendo adquirir competencias especializadas que potencian el pensamiento lógico, la resolución de problemas complejos y el diseño de soluciones innovadoras. Se abordan temáticas como la semántica formal, la teoría de lenguajes, los autómatas, la compilación y otros elementos fundamentales para comprender el funcionamiento interno de los sistemas informáticos.

Además, al dominar estos contenidos, se abren oportunidades en sectores altamente dinámicos como la inteligencia artificial, el desarrollo de lenguajes de programación, la computación cuántica o la ingeniería de software. Esta especialización representa un paso estratégico para quienes desean avanzar en su carrera dentro del ámbito académico, técnico o profesional, especialmente en contextos donde se valoran perfiles con una sólida base teórica y capacidad analítica. La profundidad conceptual que se adquiere permite no solo aplicar tecnologías existentes, sino también diseñar las del futuro.

La modalidad online de este programa permite acceder a contenidos actualizados y de alto nivel académico desde cualquier lugar, sin comprometer la calidad ni la exigencia. Esta flexibilidad facilita la compatibilidad con otros proyectos profesionales o académicos y promueve un aprendizaje autónomo, riguroso y orientado al análisis crítico. A través de una plataforma interactiva, materiales multimedia y el acompañamiento de expertos, se garantiza una experiencia enriquecedora que responde a las necesidades actuales del sector digital y académico. Adicionalmente, un reconocido Director Invitado Internacional brindará 10 minuciosas *Masterclasses*.

Asimismo, gracias a que TECH es miembro de la **Association for Computing Machinery (ACM)**, el alumno podrá acceder a recursos exclusivos y actualizados, como publicaciones científicas, cursos especializados y conferencias internacionales. Además, tendrá la oportunidad de ampliar su red de contactos, conectando con expertos en tecnología, inteligencia artificial, ciencia de datos y otras disciplinas clave del sector.

Este **Máster Título Propio en Computación y Lenguajes** contiene el programa universitario más completo y actualizado del mercado. Sus características más destacadas son:

- El desarrollo de casos prácticos presentados por expertos en Computación y Lenguajes
- Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que están concebidos recogen una información científica y práctica sobre aquellas disciplinas indispensables para el ejercicio profesional
- Los ejercicios prácticos donde realizar el proceso de autoevaluación para mejorar el aprendizaje
- Su especial hincapié en metodologías innovadoras
- Las lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet

Un reputado Director Invitado Internacional ofrecerá 10 exclusivas Masterclasses sobre las últimas tendencias en Computación y Lenguajes"

Profundizarás en la teoría de autómatas y lenguajes formales para entender cómo funcionan internamente los sistemas computacionales"

Manejarás estructuras de datos avanzadas para el desarrollo de soluciones eficientes, escalables y orientadas al rendimiento.

Diseñarás algoritmos sofisticados para resolver problemas computacionales con alta demanda en sectores tecnológicos clave.

Incluye en su cuadro docente a profesionales pertenecientes al ámbito de la Computación, que vierten en este programa la experiencia de su trabajo, además de reconocidos especialistas de sociedades de referencia y universidades de prestigio.

Su contenido multimedia, elaborado con la última tecnología educativa, permitirá al profesional un aprendizaje situado y contextualizado, es decir, un entorno simulado que proporcionará un estudio inmersivo programado para entrenarse ante situaciones reales.

El diseño de este programa se centra en el Aprendizaje Basado en Problemas, mediante el cual el alumno deberá tratar de resolver las distintas situaciones de práctica profesional que se le planteen a lo largo del curso académico. Para ello, el profesional contará con la ayuda de un novedoso sistema de vídeo interactivo realizado por reconocidos expertos.

02

¿Por qué estudiar en TECH?

TECH es la mayor Universidad digital del mundo. Con un impresionante catálogo de más de 14.000 programas universitarios, disponibles en 11 idiomas, se posiciona como líder en empleabilidad, con una tasa de inserción laboral del 99%. Además, cuenta con un enorme claustro de más de 6.000 profesores de máximo prestigio internacional.

Estudia en la mayor universidad digital del mundo y asegura tu éxito profesional. El futuro empieza en TECH”

La mejor universidad online del mundo según FORBES

La prestigiosa revista Forbes, especializada en negocios y finanzas, ha destacado a TECH como «la mejor universidad online del mundo». Así lo han hecho constar recientemente en un artículo de su edición digital en el que se hacen eco del caso de éxito de esta institución, «gracias a la oferta académica que ofrece, la selección de su personal docente, y un método de aprendizaje innovador orientado a formar a los profesionales del futuro».

El mejor claustro docente top internacional

El claustro docente de TECH está integrado por más de 6.000 profesores de máximo prestigio internacional. Catedráticos, investigadores y altos ejecutivos de multinacionales, entre los cuales se destacan Isaiah Covington, entrenador de rendimiento de los Boston Celtics; Magda Romanska, investigadora principal de MetaLAB de Harvard; Ignacio Wistuba, presidente del departamento de patología molecular traslacional del MD Anderson Cancer Center; o D.W Pine, director creativo de la revista TIME, entre otros.

La mayor universidad digital del mundo

TECH es la mayor universidad digital del mundo. Somos la mayor institución educativa, con el mejor y más amplio catálogo educativo digital, cien por cien online y abarcando la gran mayoría de áreas de conocimiento. Ofrecemos el mayor número de titulaciones propias, titulaciones oficiales de posgrado y de grado universitario del mundo. En total, más de 14.000 títulos universitarios, en once idiomas distintos, que nos convierten en la mayor institución educativa del mundo.

Los planes de estudio más completos del panorama universitario

TECH ofrece los planes de estudio más completos del panorama universitario, con temarios que abarcan conceptos fundamentales y, al mismo tiempo, los principales avances científicos en sus áreas científicas específicas. Asimismo, estos programas son actualizados continuamente para garantizar al alumnado la vanguardia académica y las competencias profesionales más demandadas. De esta forma, los títulos de la universidad proporcionan a sus egresados una significativa ventaja para impulsar sus carreras hacia el éxito.

Un método de aprendizaje único

TECH es la primera universidad que emplea el *Relearning* en todas sus titulaciones. Se trata de la mejor metodología de aprendizaje online, acreditada con certificaciones internacionales de calidad docente, dispuestas por agencias educativas de prestigio. Además, este disruptivo modelo académico se complementa con el "Método del Caso", configurando así una estrategia de docencia online única. También en ella se implementan recursos didácticos innovadores entre los que destacan vídeos en detalle, infografías y resúmenes interactivos.

La universidad online oficial de la NBA

TECH es la universidad online oficial de la NBA. Gracias a un acuerdo con la mayor liga de baloncesto, ofrece a sus alumnos programas universitarios exclusivos, así como una gran variedad de recursos educativos centrados en el negocio de la liga y otras áreas de la industria del deporte. Cada programa tiene un currículo de diseño único y cuenta con oradores invitados de excepción: profesionales con una distinguida trayectoria deportiva que ofrecerán su experiencia en los temas más relevantes.

Líderes en empleabilidad

TECH ha conseguido convertirse en la universidad líder en empleabilidad. El 99% de sus alumnos obtienen trabajo en el campo académico que ha estudiado, antes de completar un año luego de finalizar cualquiera de los programas de la universidad. Una cifra similar consigue mejorar su carrera profesional de forma inmediata. Todo ello gracias a una metodología de estudio que basa su eficacia en la adquisición de competencias prácticas, totalmente necesarias para el desarrollo profesional.

Google Partner Premier

El gigante tecnológico norteamericano ha otorgado a TECH la insignia Google Partner Premier. Este galardón, solo al alcance del 3% de las empresas del mundo, pone en valor la experiencia eficaz, flexible y adaptada que esta universidad proporciona al alumno. El reconocimiento no solo acredita el máximo rigor, rendimiento e inversión en las infraestructuras digitales de TECH, sino que también sitúa a esta universidad como una de las compañías tecnológicas más punteras del mundo.

La universidad mejor valorada por sus alumnos

Los alumnos han posicionado a TECH como la universidad mejor valorada del mundo en los principales portales de opinión, destacando su calificación más alta de 4,9 sobre 5, obtenida a partir de más de 1.000 reseñas. Estos resultados consolidan a TECH como la institución universitaria de referencia a nivel internacional, reflejando la excelencia y el impacto positivo de su modelo educativo.

03

Plan de estudios

En un contexto donde los avances en lenguajes de programación, teoría de autómatas y semántica formal impulsan el desarrollo de nuevas tecnologías, el estudio profundo de estos fundamentos resulta clave para innovar en áreas como la computación cuántica, la inteligencia artificial y la ciberseguridad. Por ello, este plan de estudios integra contenidos actualizados y estratégicamente estructurados que permiten comprender la lógica interna de los sistemas informáticos. Así, se favorece una visión crítica y avanzada del funcionamiento computacional, en sintonía con las transformaciones digitales que están redefiniendo los estándares de la informática a nivel global.

Diseñarás e implementarás Lenguajes de dominio específico, adaptados a necesidades técnicas concretas”

Módulo 1. Fundamentos de programación

1.1. Introducción a la programación
- 1.1.1. Estructura básica de un ordenador
- 1.1.2. Software
- 1.1.3. Lenguajes de programación
- 1.1.4. Ciclo de vida de una aplicación informática

1.2. Diseño de algoritmos
- 1.2.1. La resolución de problemas
- 1.2.2. Técnicas descriptivas
- 1.2.3. Elementos y estructura de un algoritmo

1.3. Elementos de un programa
- 1.3.1. Origen y características del lenguaje C++
- 1.3.2. El entorno de desarrollo
- 1.3.3. Concepto de programa
- 1.3.4. Tipos de datos fundamentales
- 1.3.5. Operadores
- 1.3.6. Expresiones
- 1.3.7. Sentencias
- 1.3.8. Entrada y salida de datos

1.4. Sentencias de control
- 1.4.1. Sentencias
- 1.4.2. Bifurcaciones
- 1.4.3. Bucles

1.5. Abstracción y modularidad: funciones
- 1.5.1. Diseño modular
- 1.5.2. Concepto de función y utilidad
- 1.5.3. Definición de una función
- 1.5.4. Flujo de ejecución en la llamada de una función
- 1.5.5. Prototipo de una función
- 1.5.6. Devolución de resultados
- 1.5.7. Llamada a una función: parámetros
- 1.5.8. Paso de parámetros por referencia y por valor
- 1.5.9. Ámbito de un identificador

1.6. Estructuras de datos estáticas
- 1.6.1. *Arrays*
- 1.6.2. Matrices. Poliedros
- 1.6.3. Búsqueda y ordenación
- 1.6.4. Cadenas. Funciones de E / S para cadenas
- 1.6.5. Estructuras. Uniones
- 1.6.6. Nuevos tipos de datos

1.7. Estructuras de datos dinámicas: punteros
- 1.7.1. Concepto. Definición de puntero
- 1.7.2. Operadores y operaciones con punteros
- 1.7.3. *Arrays* de punteros
- 1.7.4. Punteros y *arrays*
- 1.7.5. Punteros a cadenas
- 1.7.6. Punteros a estructuras
- 1.7.7. Indirección múltiple
- 1.7.8. Punteros a funciones
- 1.7.9. Paso de funciones, estructuras y *arrays* como parámetros de funciones

1.8. Ficheros
- 1.8.1. Conceptos básicos
- 1.8.2. Operaciones con ficheros
- 1.8.3. Tipos de ficheros
- 1.8.4. Organización de los ficheros
- 1.8.5. Introducción a los ficheros C++
- 1.8.6. Manejo de ficheros

1.9. Recursividad
- 1.9.1. Definición de recursividad
- 1.9.2. Tipos de recursión
- 1.9.3. Ventajas e inconvenientes
- 1.9.4. Consideraciones
- 1.9.5. Conversión recursivo - iterativa
- 1.9.6. La pila de recursión

1.10. Prueba y documentación
- 1.10.1. Pruebas de programas
- 1.10.2. Prueba de la caja blanca
- 1.10.3. Prueba de la caja negra
- 1.10.4. Herramientas para realizar las pruebas
- 1.10.5. Documentación de programas

Módulo 2. Estructura de datos

2.1. Introducción a la programación en C++
- 2.1.1. Clases, constructores, métodos y atributos
- 2.1.2. Variables
- 2.1.3. Expresiones condicionales y bucles
- 2.1.4. Objetos

2.2. Tipos abstractos de datos (TAD)
- 2.2.1. Tipos de datos
- 2.2.2. Estructuras básicas y TAD
- 2.2.3. Vectores y *arrays*

2.3. Estructuras de datos lineales
- 2.3.1. TAD lista: definición
- 2.3.2. Listas enlazadas y doblemente enlazadas
- 2.3.3. Listas ordenadas
- 2.3.4. Listas en C++
- 2.3.5. TAD pila
- 2.3.6. TAD cola
- 2.3.7. Pila y cola en C++

2.4. Estructuras de datos jerárquicas
- 2.4.1. TAD árbol
- 2.4.2. Recorridos
- 2.4.3. Árboles n-arios
- 2.4.4. Árboles binarios
- 2.4.5. Árboles binarios de búsqueda

2.5. Estructuras de datos jerárquicas: árboles complejos
- 2.5.1. Árboles perfectamente equilibrados o de altura mínima
- 2.5.2. Árboles multicamino
- 2.5.3. Referencias bibliográficas

2.6. Montículos y cola de prioridad
- 2.6.1. TAD montículos
- 2.6.2. TAD cola de prioridad

2.7. Tablas *hash*
- 2.7.1. TAD Tabla *hash*
- 2.7.2. Funciones *hash*
- 2.7.3. Función *hash* en tablas *hash*
- 2.7.4. Redispersión
- 2.7.5. Tablas *hash* abiertas

2.8. Grafos
- 2.8.1. TAD grafo
- 2.8.2. Tipos de grafos
- 2.8.3. Representación gráfica y operaciones básicas
- 2.8.4. Diseño de grafos

2.9. Algoritmos y conceptos avanzados sobre grafos
- 2.9.1. Problemas sobre grafos
- 2.9.2. Algoritmos sobre caminos
- 2.9.3. Algoritmos de búsqueda o recorridos
- 2.9.4. Otros algoritmos

2.10. Otras estructuras de datos
- 2.10.1. Conjuntos
- 2.10.2. *Arrays* paralelos
- 2.10.3. Tablas de símbolos
- 2.10.4. *Tries*

Módulo 3. Algoritmia y complejidad

3.1. Introducción a las estrategias de diseño de algoritmos
- 3.1.1. Recursividad
- 3.1.2. Divide y conquista
- 3.1.3. Otras estrategias

3.2. Eficiencia y análisis de los algoritmos
- 3.2.1. Medidas de eficiencia
- 3.2.2. Medir el tamaño de la entrada
- 3.2.3. Medir el tiempo de ejecución
- 3.2.4. Caso peor, mejor y medio
- 3.2.5. Notación asintótica
- 3.2.6. Criterios de análisis matemático de algoritmos no recursivos
- 3.2.7. Análisis matemático de algoritmos recursivos
- 3.2.8. Análisis empírico de algoritmos

3.3. Algoritmos de ordenación
- 3.3.1. Concepto de ordenación
- 3.3.2. Ordenación de la burbuja
- 3.3.3. Ordenación por selección
- 3.3.4. Ordenación por inserción
- 3.3.5. Ordenación por mezcla (*merge sort*)
- 3.3.6. Ordenación rápida (*quicksort*)

3.4. Algoritmos con árboles
- 3.4.1. Concepto de árbol
- 3.4.2. Árboles binarios
- 3.4.3. Recorridos de árbol
- 3.4.4. Representar expresiones
- 3.4.5. Árboles binarios ordenados
- 3.4.6. Árboles binarios balanceados

3.5. Algoritmos con *heaps*
- 3.5.1. Los *heaps*
- 3.5.2. El algoritmo *heap sort*
- 3.5.3. Las colas de prioridad

3.6. Algoritmos con grafos
- 3.6.1. Representación
- 3.6.2. Recorrido en anchura
- 3.6.3. Recorrido en profundidad
- 3.6.4. Ordenación topológica

3.7. Algoritmos *greedy*
- 3.7.1. La estrategia *greedy*
- 3.7.2. Elementos de la estrategia *greedy*
- 3.7.3. Cambio de monedas
- 3.7.4. Problema del viajante
- 3.7.5. Problema de la mochila

3.8. Búsqueda de caminos mínimos
- 3.8.1. El problema del camino mínimo
- 3.8.2. Arcos negativos y ciclos
- 3.8.3. Algoritmo de Dijkstra

3.9. Algoritmos *greedy* sobre grafos
- 3.9.1. El árbol de recubrimiento mínimo
- 3.9.2. El algoritmo de Prim
- 3.9.3. El algoritmo de Kruskal
- 3.9.4. Análisis de complejidad

3.10. *Backtracking*
- 3.10.1. El *backtracking*
- 3.10.2. Técnicas alternativas

Módulo 4. Diseño avanzado de algoritmos

4.1. Análisis de algoritmos recursivos y tipo divide y conquista
- 4.1.1. Planteamiento y resolución de ecuaciones de recurrencia homogéneas y no homogéneas
- 4.1.2. Descripción general de la estrategia divide y conquista

4.2. Análisis amortizado
- 4.2.1. El análisis agregado
- 4.2.2. El método de contabilidad
- 4.2.3. El método del potencial

4.3. Programación dinámica y algoritmos para problemas NP
- 4.3.1. Características de la programación dinámica
- 4.3.2. Vuelta atrás: *backtracking*
- 4.3.3. Ramificación y poda

4.4. Optimización combinatoria
- 4.4.1. Representación de problemas
- 4.4.2. Optimización en 1D

4.5. Algoritmos de aleatorización
- 4.5.1. Ejemplos de algoritmos de aleatorización
- 4.5.2. El teorema de Buffon
- 4.5.3. Algoritmo de Monte Carlo
- 4.5.4. Algoritmo Las Vegas

4.6. Búsqueda local y con candidatos
- 4.6.1. *Gradient Ascent*
- 4.6.2. *Hill Climbing*
- 4.6.3. *Simulated Annealing*
- 4.6.4. *Tabu Search*
- 4.6.5. Búsqueda con candidatos

4.7. Verificación formal de programas
- 4.7.1. Especificación de abstracciones funcionales
- 4.7.2. El lenguaje de la lógica de primer orden
- 4.7.3. El sistema formal de Hoare

4.8. Verificación de programas iterativos
- 4.8.1. Reglas del sistema formal de Hoare
- 4.8.2. Concepto de invariante de iteraciones

4.9. Métodos numéricos
- 4.9.1. El método de la bisección
- 4.9.2. El método de Newton-Raphson
- 4.9.3. El método de la secante

4.10. Algoritmos paralelos
- 4.10.1. Operaciones binarias paralelas
- 4.10.2. Operaciones paralelas con grafos
- 4.10.3. Paralelismo en divide y vencerás
- 4.10.4. Paralelismo en programación dinámica

Módulo 5. Programación avanzada

5.1. Introducción a la programación orientada a objetos
- 5.1.1. Introducción a la programación orientada a objetos
- 5.1.2. Diseño de clases
- 5.1.3. Introducción a UML para el modelado de los problemas

5.2. Relaciones entre clases
- 5.2.1. Abstracción y herencia
- 5.2.2. Conceptos avanzados de herencia
- 5.2.3. Polimorfismo
- 5.2.4. Composición y agregación

5.3. Introducción a los patrones de diseño para problemas orientados a objetos
- 5.3.1. Qué son los patrones de diseño
- 5.3.2. Patrón *factory*
- 5.3.3. Patrón *singleton*
- 5.3.4. Patrón *observer*
- 5.3.5. Patrón *composite*

5.4. Excepciones
- 5.4.1. ¿Qué son las excepciones?
- 5.4.2. Captura y gestión de excepciones
- 5.4.3. Lanzamiento de excepciones
- 5.4.4. Creación de excepciones

5.5. Interfaces de usuario
- 5.5.1. Introducción a Qt
- 5.5.2. Posicionamiento
- 5.5.3. ¿Qué son los eventos?
- 5.5.4. Eventos: definición y captura
- 5.5.5. Desarrollo de interfaces de usuario

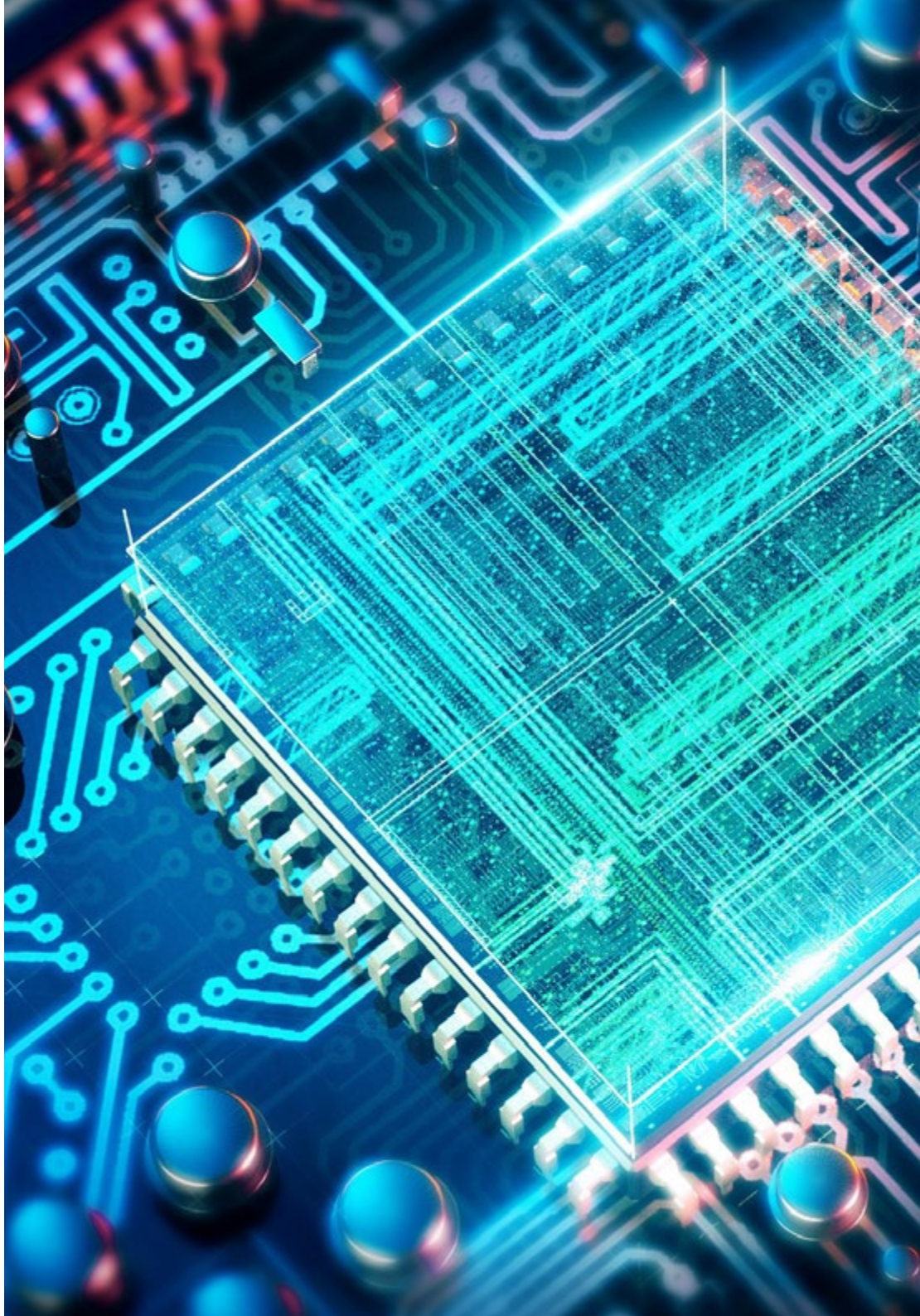
5.6. Introducción a la programación concurrente
- 5.6.1. Introducción a la programación concurrente
- 5.6.2. El concepto de proceso e hilo
- 5.6.3. Interacción entre procesos o hilos
- 5.6.4. Los hilos en C++
- 5.6.5. Ventajas e inconvenientes de la programación concurrente

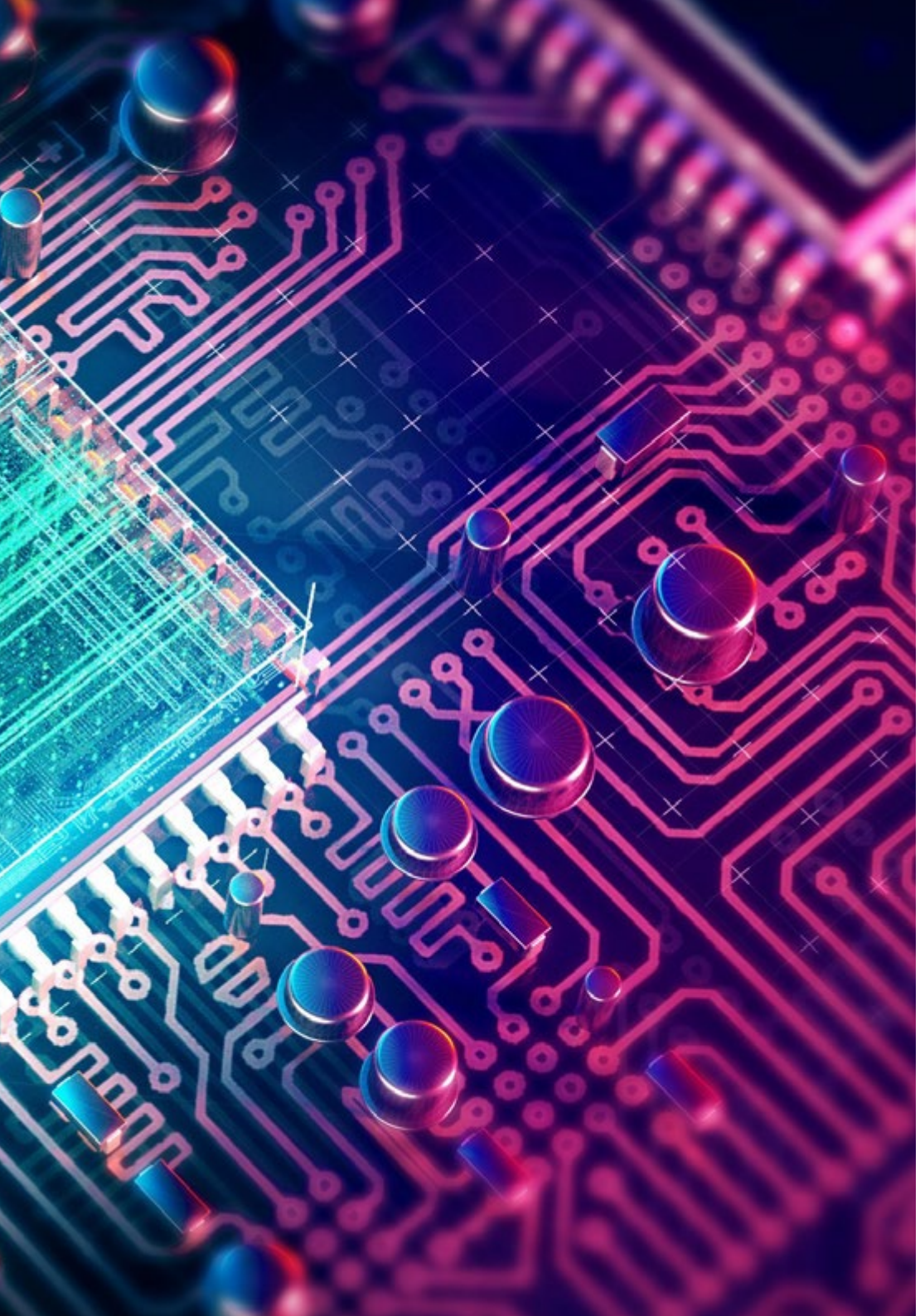
5.7. Gestión de hilos y sincronización
- 5.7.1. Ciclo de vida de un hilo
- 5.7.2. La clase *Thread*
- 5.7.3. Planificación de hilos
- 5.7.4. Grupos de hilos
- 5.7.5. Hilos de tipo *daemon*
- 5.7.6. Sincronización
- 5.7.7. Mecanismos de bloqueo
- 5.7.8. Mecanismos de comunicación
- 5.7.9. Monitores

5.8. Problemas comunes dentro de la programación concurrente
- 5.8.1. El problema de los productores consumidores
- 5.8.2. El problema de los lectores y escritores
- 5.8.3. El problema de la cena de los filósofos

5.9. Documentación y pruebas de software
- 5.9.1. ¿Por qué es importante documentar el software?
- 5.9.2. Documentación de diseño
- 5.9.3. Uso de herramientas para la documentación

5.10. Pruebas de software
- 5.10.1. Introducción a las pruebas del software
- 5.10.2. Tipos de pruebas
- 5.10.3. Prueba de unidad
- 5.10.4. Prueba de integración
- 5.10.5. Prueba de validación
- 5.10.6. Prueba del sistema





Módulo 6. Informática teórica

6.1. Conceptos matemáticos utilizados
- 6.1.1. Introducción a la lógica proposicional
- 6.1.2. Teoría de relaciones
- 6.1.3. Conjuntos numerables y no numerables

6.2. Lenguajes y gramáticas formales e introducción a las máquinas de Turing
- 6.2.1. Lenguajes y gramáticas formales
- 6.2.2. Problema de decisión
- 6.2.3. La máquina de Turing

6.3. Extensiones para las máquinas de Turing, máquinas de Turing restringidas y computadoras
- 6.3.1. Técnicas de programación para las máquinas de Turing
- 6.3.2. Extensiones para las máquinas de Turing
- 6.3.3. Máquinas de Turing restringidas
- 6.3.4. Máquinas de Turing y computadoras

6.4. Indecibilidad
- 6.4.1. Lenguaje no recursivo enumerable
- 6.4.2. Un problema indecidible recursivamente enumerable

6.5. Otros problemas indecibles
- 6.5.1. Problemas indecidibles para las máquinas de Turing
- 6.5.2. Problema de correspondencia de Post (PCP)

6.6. Problemas intratables
- 6.6.1. Las clases P y NP
- 6.6.2. Un problema NP completo
- 6.6.3. Problema de la satisfacibilidad restringido
- 6.6.4. Otros problemas NP completos

6.7. Problemas *co - NP y PSPACE*
- 6.7.1. Complementarios de los lenguajes de NP
- 6.7.2. Problemas resolubles en espacio polinómico
- 6.7.3. Problemas *PSPACE* completos

6.8. Clases de lenguajes basados en la aleatorización
- 6.8.1. Modelo de la MT con aleatoriedad
- 6.8.2. Las clases RP y ZPP
- 6.8.3. Prueba de primalidad
- 6.8.4. Complejidad de la prueba de primalidad

6.9. Otras clases y gramáticas
- 6.9.1. Autómatas finitos probabilísticos
- 6.9.2. Autómatas celulares
- 6.9.3. Células de McCulloch y Pitts
- 6.9.4. Gramáticas de Lindenmayer

6.10. Sistemas avanzados de cómputo
- 6.10.1. Computación con membranas: sistemas P
- 6.10.2. Computación con ADN
- 6.10.3. Computación cuántica

Módulo 7. Teoría de autómatas y lenguajes formales

7.1. Introducción a la teoría de autómatas
- 7.1.1. ¿Por qué estudiar teoría de autómatas?
- 7.1.2. Introducción a las demostraciones formales
- 7.1.3. Otras formas de demostración
- 7.1.4. Inducción matemática
- 7.1.5. Alfabetos, cadenas y lenguajes

7.2. Autómatas finitos deterministas
- 7.2.1. Introducción a los autómatas finitos
- 7.2.2. Autómatas finitos deterministas

7.3. Autómatas finitos no deterministas
- 7.3.1. Autómatas finitos no deterministas
- 7.3.2. Equivalencia entre AFD y AFN
- 7.3.3. Autómatas finitos con transiciones ε

7.4. Lenguajes y expresiones regulares (I)
- 7.4.1. Lenguajes y expresiones regulares
- 7.4.2. Autómatas finitos y expresiones regulares

7.5. Lenguajes y expresiones regulares (II)
- 7.5.1. Conversión de expresiones regulares en autómatas
- 7.5.2. Aplicaciones de las expresiones regulares
- 7.5.3. Álgebra de las expresiones regulares

7.6. Lema de bombeo y clausura de los lenguajes regulares
- 7.6.1. Lema de bombeo
- 7.6.2. Propiedades de clausura de los lenguajes regulares

7.7. Equivalencia y minimización de autómatas
- 7.7.1. Equivalencia de AF
- 7.7.2. Minimización de AF

7.8. Gramáticas independientes de contexto (GIC)
- 7.8.1. Gramáticas independientes de contexto
- 7.8.2. Árboles de derivación
- 7.8.3. Aplicaciones de las GIC
- 7.8.4. Ambigüedad en las gramáticas y lenguajes

7.9. Autómatas a pila y GIC
- 7.9.1. Definición de los autómatas a pila
- 7.9.2. Lenguajes aceptados por un autómata a pila
- 7.9.3. Equivalencia entre autómatas a pila y GIC
- 7.9.4. Autómata a pila determinista

7.10. Formas normales, lema de bombeo de las GIC y propiedades de los LIC
- 7.10.1. Formas normales de las GIC
- 7.10.2. Lema de bombeo
- 7.10.3. Propiedades de clausura de los lenguajes
- 7.10.4. Propiedades de decisión de los LIC

Módulo 8. Procesadores de lenguajes

8.1. Introducción al proceso de compilación
- 8.1.1. Compilación e interpretación
- 8.1.2. Entorno de ejecución de un compilador
- 8.1.3. Proceso de análisis
- 8.1.4. Proceso de síntesis

8.2. Analizador léxico
- 8.2.1. ¿Qué es un analizador léxico?
- 8.2.2. Implementación del analizador léxico
- 8.2.3. Acciones semánticas
- 8.2.4. Recuperación de errores
- 8.2.5. Cuestiones de implementación

8.3. Análisis sintáctico
- 8.3.1. ¿Qué es un analizador sintáctico?
- 8.3.2. Conceptos previos
- 8.3.3. Analizadores descendentes
- 8.3.4. Analizadores ascendentes

8.4. Análisis sintáctico descendente y análisis sintáctico ascendente
- 8.4.1. Analizador LL (1)
- 8.4.2. Analizador LR (0)
- 8.4.3. Ejemplo de analizador

8.5. Análisis sintáctico ascendente avanzado
- 8.5.1. Analizador SLR
- 8.5.2. Analizador LR (1)
- 8.5.3. Analizador LR (k)
- 8.5.4. Analizador LALR

8.6. Análisis semántico (I)
- 8.6.1. Traducción dirigida por la sintaxis
- 8.6.2. Tabla de símbolos

8.7. Análisis semántico (II)
- 8.7.1. Comprobación de tipos
- 8.7.2. El subsistema de tipos
- 8.7.3. Equivalencia de tipos y conversiones

8.8. Generación de código y entorno de ejecución
- 8.8.1. Aspectos de diseño
- 8.8.2. Entorno de ejecución
- 8.8.3. Organización de la memoria
- 8.8.4. Asignación de memoria

8.9. Generación de código intermedio
- 8.9.1. Traducción dirigida por la *sintaxis*
- 8.9.2. Representaciones intermedias
- 8.9.3. Ejemplos de traducciones

8.10. Optimización de código
- 8.10.1. Asignación de registros
- 8.10.2. Eliminación de asignaciones muertas
- 8.10.3. Ejecución en tiempo de compilación
- 8.10.4. Reordenación de expresiones
- 8.10.5. Optimización de bucles

Módulo 9. Informática gráfica y visualización

9.1. Teoría del color
- 9.1.1. Propiedades de la luz
- 9.1.2. Modelos de color
- 9.1.3. El estándar CIE
- 9.1.4. *Profiling*

9.2. Primitivas de salida
- 9.2.1. El controlador de vídeo
- 9.2.2. Algoritmos de dibujo de líneas
- 9.2.3. Algoritmos de dibujo de circunferencias
- 9.2.4. Algoritmos de relleno

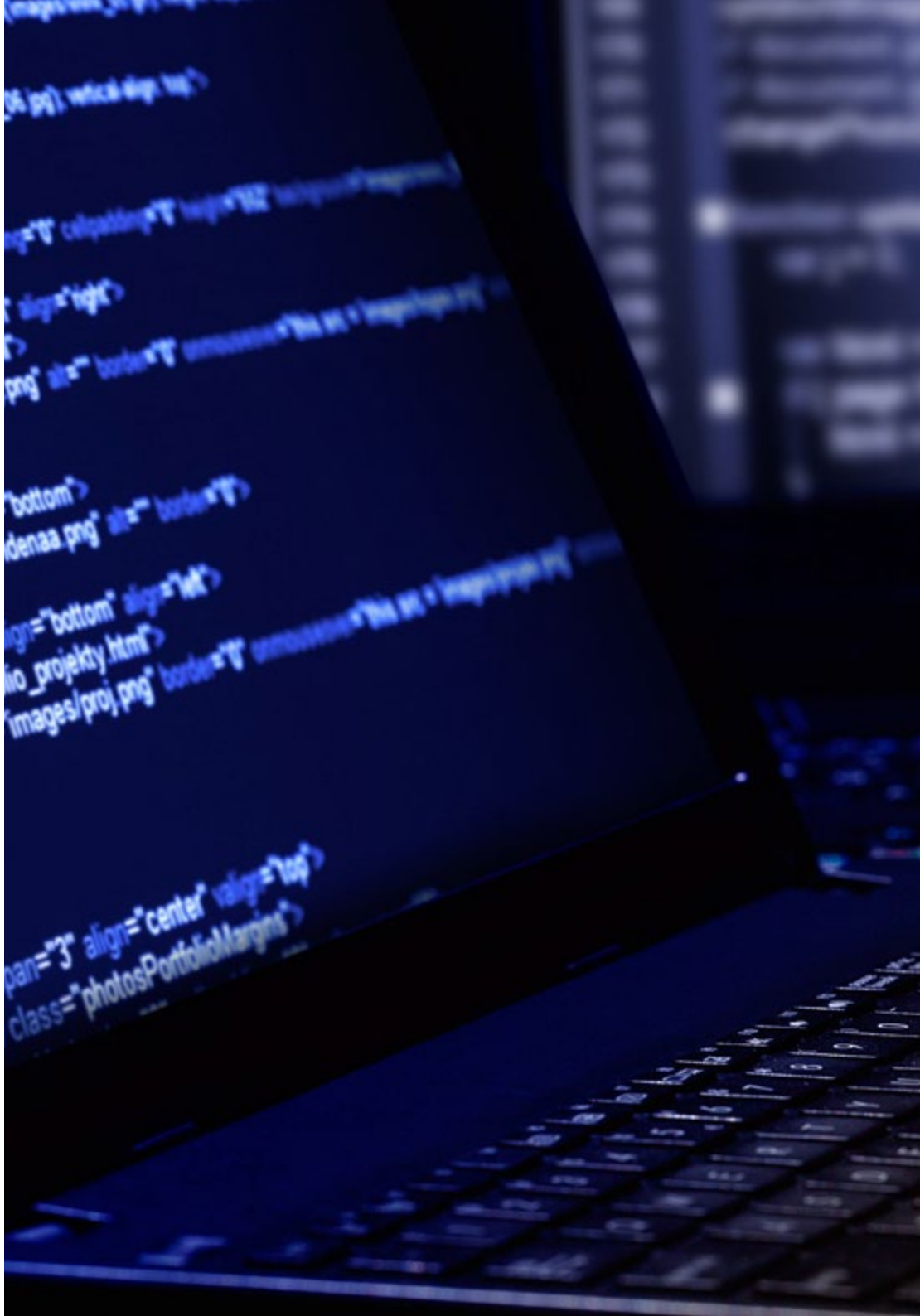
9.3. Transformaciones 2D y sistemas de coordenadas y recorte 2D
- 9.3.1. Transformaciones geométricas básicas
- 9.3.2. Coordenadas homogéneas
- 9.3.3. Transformación inversa
- 9.3.4. Composición de transformaciones
- 9.3.5. Otras transformaciones
- 9.3.6. Cambio de coordenadas
- 9.3.7. Sistemas de coordenadas 2D
- 9.3.8. Cambio de coordenadas
- 9.3.9. Normalización
- 9.3.10. Algoritmos de recorte

9.4. Transformaciones 3D
- 9.4.1. Translación
- 9.4.2. Rotación
- 9.4.3. Escalado
- 9.4.4. Reflexión
- 9.4.5. Cizalla

9.5. Visualización y cambio de coordenadas 3D
- 9.5.1. Sistemas de coordenadas 3D
- 9.5.2. Visualización
- 9.5.3. Cambio de coordenadas
- 9.5.4. Proyección y normalización

9.6. Proyección y recorte 3D
- 9.6.1. Proyección ortogonal
- 9.6.2. Proyección paralela oblicua
- 9.6.3. Proyección perspectiva
- 9.6.4. Algoritmos de recorte 3D

9.7. Eliminación de superficies ocultas
- 9.7.1. *Back - face removal*
- 9.7.2. Z - *buffer*
- 9.7.3. Algoritmo del pintor
- 9.7.4. Algoritmo de Warnock
- 9.7.5. Detección de líneas ocultas



9.8. Interpolación y curvas paramétricas
- 9.8.1. Interpolación y aproximación con polinomios
- 9.8.2. Representación paramétrica
- 9.8.3. Polinomio de Lagrange
- 9.8.4. *Splines* cúbicos naturales
- 9.8.5. Funciones base
- 9.8.6. Representación matricial

9.9. Curvas Bézier
- 9.9.1. Construcción algebraica
- 9.9.2. Forma matricial
- 9.9.3. Composición
- 9.9.4. Construcción geométrica
- 9.9.5. Algoritmo de dibujo

9.10. *B - splines*
- 9.10.1. El problema del control local
- 9.10.2. *B - splines* cúbicos uniformes
- 9.10.3. Funciones base y puntos de control
- 9.10.4. Deriva al origen y multiplicidad
- 9.10.5. Representación matricial
- 9.10.6. *B - splines* no uniformes

Módulo 10. Computación bioinspirada

10.1. Introducción a la Computación bioinspirada
- 10.1.1. Introducción a la Computación bioinspirada

10.2. Algoritmos de adaptación social
- 10.2.1. Computación bioinspirada basada en colonias de hormigas
- 10.2.2. Variantes de los algoritmos de colonias de hormigas
- 10.2.3. Computación basada en nubes de partículas

10.3. Algoritmos genéticos
- 10.3.1. Estructura general
- 10.3.2. Implementaciones de los principales operadores

10.4. Estrategias de exploración y explotación del espacio para algoritmos genéticos
- 10.4.1. Algoritmo CHC
- 10.4.2. Problemas multimodales

10.5. Modelos de Computación evolutiva (I)
- 10.5.1. Estrategias evolutivas
- 10.5.2. Programación evolutiva
- 10.5.3. Algoritmos basados en evolución diferencial

10.6. Modelos de Computación evolutiva (II)
- 10.6.1. Modelos de evolución basados en estimación de distribuciones (EDA)
- 10.6.2. Programación genética

10.7. Programación evolutiva aplicada a problemas de aprendizaje
- 10.7.1. Aprendizaje basado en reglas
- 10.7.2. Métodos evolutivos en problemas de selección de instancias

10.8. Problemas multiobjetivo
- 10.8.1. Concepto de dominancia
- 10.8.2. Aplicación de algoritmos evolutivos a problemas multiobjetivo

10.9. Redes neuronales (I)
- 10.9.1. Introducción a las redes neuronales
- 10.9.2. Ejemplo práctico con redes neuronales

10.10. Redes neuronales (II)
- 10.10.1. Casos de uso de las redes neuronales en la investigación médica
- 10.10.2. Casos de uso de las redes neuronales en la economía
- 10.10.3. Casos de uso de las redes neuronales en la visión artificial

Desarrollarás habilidades para el análisis formal y semántico de programas, mejorando la eficiencia del software significativamente”

04

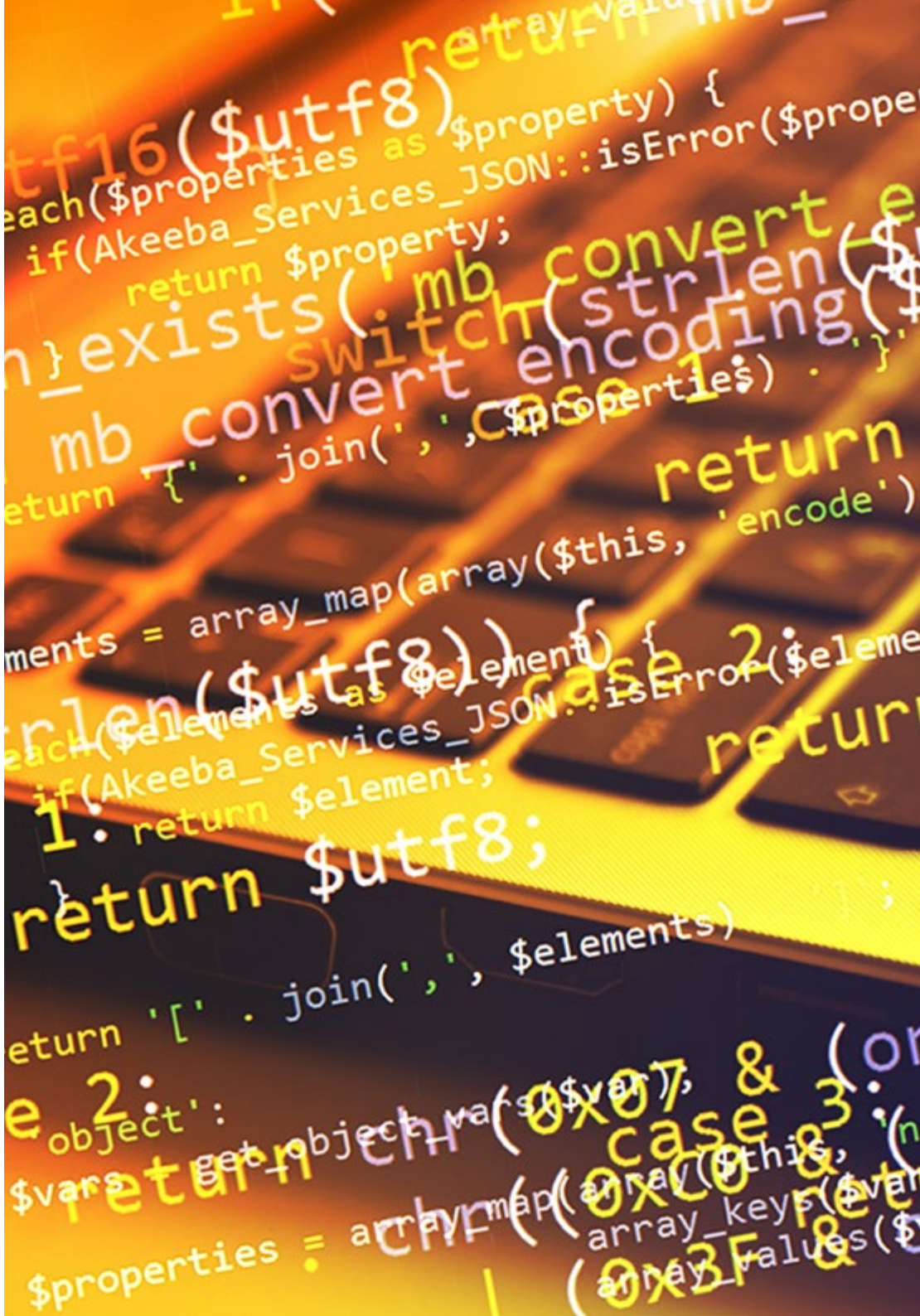
Objetivos docentes

Este Máster Título Propio tiene como objetivo profundizar en los fundamentos teóricos y prácticos que sustentan el diseño, análisis y optimización de lenguajes computacionales. Para ello, combina el estudio de estructuras gramaticales, semánticas y modelos de cómputo con herramientas aplicadas al desarrollo de sistemas inteligentes. Asimismo, promueve el pensamiento abstracto y la resolución rigurosa de problemas complejos, claves en entornos de alta demanda tecnológica. Gracias a esta perspectiva integral, se favorece la capacidad de innovar en áreas emergentes como el procesamiento del lenguaje natural, los lenguajes formales y la arquitectura de compiladores.

Integrarás el pensamiento computacional en proyectos de innovación tecnológica, lo que te permitirá ofrecer soluciones avanzadas a diferentes entidades"

Objetivos generales

- Comprender en profundidad los fundamentos teóricos de los lenguajes de programación y su aplicación en entornos computacionales avanzados y especializados
- Analizar de manera crítica las estructuras formales que sustentan el diseño y evolución de lenguajes computacionales en distintos contextos
- Aplicar con solvencia modelos de cómputo para la resolución eficiente de problemas complejos en el ámbito de la informática actual
- Dominar los principios esenciales de la semántica formal y su relación directa con la ejecución precisa de programas informáticos
- Estudiar con rigurosidad los autómatas, gramáticas y lenguajes formales desde una perspectiva computacional y científica integral
- Integrar conocimientos fundamentales en teoría de la computación con desarrollos tecnológicos actuales y futuros del sector
- Evaluar sistemáticamente el rendimiento y la eficiencia de diferentes lenguajes, compiladores y entornos de desarrollo avanzados
- Diseñar soluciones computacionales innovadoras a partir del análisis detallado de estructuras lógicas, sintácticas y semánticas
- Desarrollar competencias especializadas para la investigación aplicada en el campo de la computación y los lenguajes formales
- Interpretar con criterio el impacto de los lenguajes computacionales en la evolución constante de los sistemas inteligentes modernos



Objetivos específicos

Módulo 1. Fundamentos de programación

- Identificar los elementos básicos de un lenguaje de programación
- Aplicar estructuras de control en la resolución de problemas
- Utilizar variables, operadores y funciones en programas simples
- Desarrollar programas estructurados siguiendo buenas prácticas

Módulo 2. Estructura de datos

- Implementar estructuras lineales como listas y pilas
- Utilizar árboles y grafos en representaciones computacionales
- Evaluar la eficiencia de diferentes estructuras de datos
- Seleccionar estructuras adecuadas según el problema planteado

Módulo 3. Algoritmia y complejidad

- Analizar la eficiencia temporal y espacial de algoritmos
- Clasificar algoritmos según su complejidad computacional
- Resolver problemas utilizando paradigmas algorítmicos básicos
- Comparar alternativas de solución según su rendimiento

Módulo 4. Diseño avanzado de algoritmos

- Aplicar técnicas de programación dinámica y *greedy*
- Diseñar algoritmos para problemas de optimización compleja
- Utilizar estructuras avanzadas en la construcción de algoritmos
- Evaluar el rendimiento de soluciones algorítmicas avanzadas

Módulo 5. Programación avanzada

- Desarrollar programas usando conceptos de programación orientada a objetos
- Integrar manejo de errores y depuración en proyectos complejos
- Implementar patrones de diseño en aplicaciones escalables
- Utilizar librerías y *frameworks* para acelerar el desarrollo

Módulo 6. Informática teórica

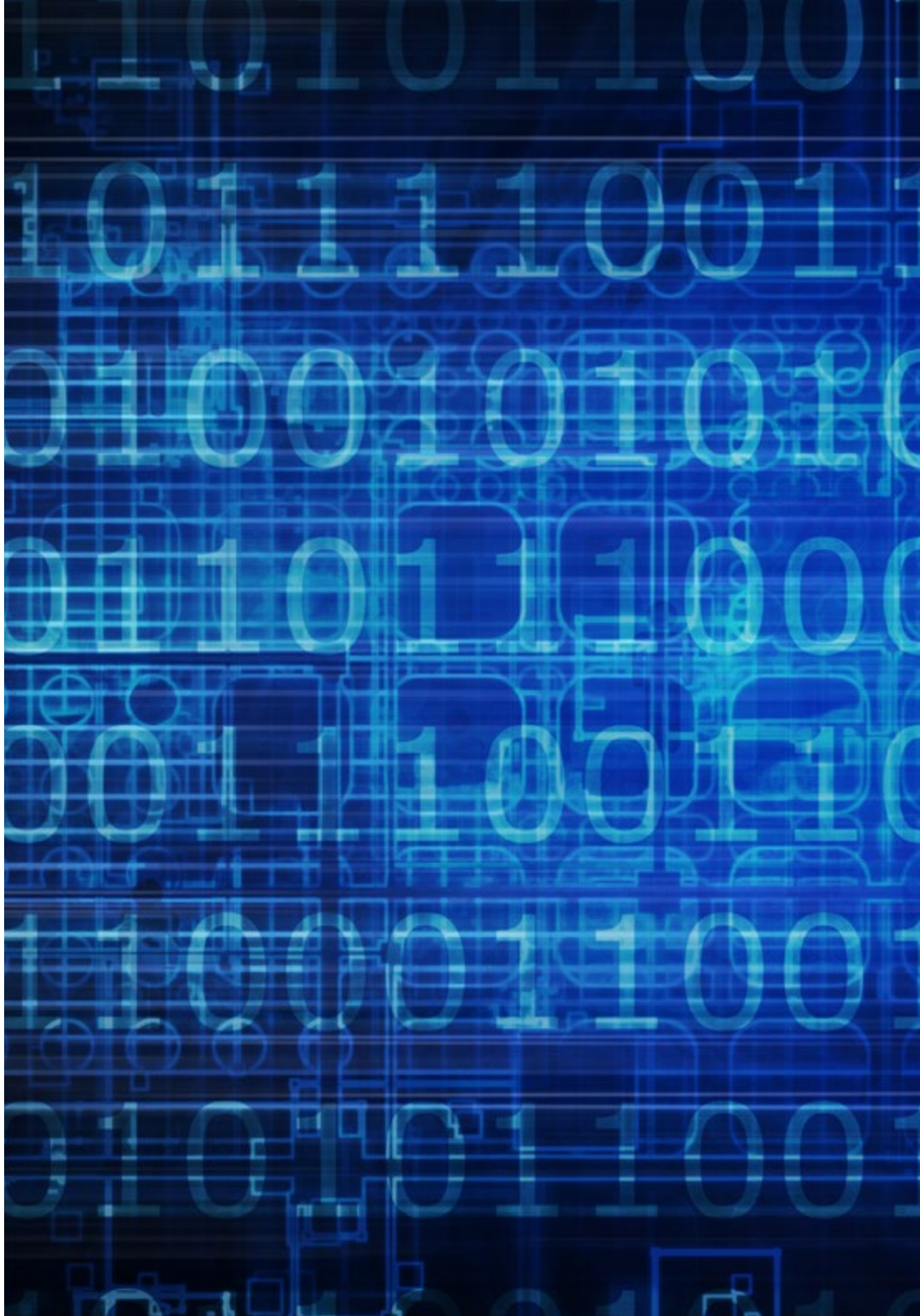
- Comprender los límites del cómputo a través de modelos teóricos
- Estudiar problemas no computables y su clasificación
- Analizar las clases P y NP en teoría de la computación
- Aplicar conceptos teóricos en la resolución de problemas abstractos

Módulo 7. Teoría de autómatas y lenguajes formales

- Diseñar autómatas finitos deterministas y no deterministas
- Generar gramáticas regulares y libres de contexto
- Analizar lenguajes formales mediante autómatas y expresiones regulares
- Relacionar tipos de lenguajes con jerarquías formales

Módulo 8. Procesadores de lenguajes

- Implementar analizadores léxicos y sintácticos básicos
- Comprender el proceso de traducción en compiladores
- Diseñar gramáticas para lenguajes de programación específicos
- Integrar fases del compilador en un entorno funcional



Módulo 9. Informática gráfica y visualización

- Utilizar librerías gráficas para representar datos visualmente
- Comprender los fundamentos del modelado gráfico computacional
- Implementar técnicas de renderizado en dos y tres dimensiones
- Diseñar sistemas interactivos de visualización de información

Módulo 10. Computación bioinspirada

- Aplicar algoritmos evolutivos en la resolución de problemas complejos
- Comprender los fundamentos de redes neuronales y su entrenamiento
- Emplear estrategias bioinspiradas en contextos de inteligencia artificial
- Comparar métodos bioinspirados con algoritmos tradicionales

Proyéctate hacia sectores emergentes como la computación cuántica, el PLN y la ciberseguridad avanzada. ¡Hazte indispensable!"

05

Salidas profesionales

El avance constante de la inteligencia artificial, los lenguajes formales y la ingeniería del *software* ha ampliado significativamente las oportunidades laborales en el ámbito de la computación. En consecuencia, contar con una especialización sólida en estas áreas permite acceder a perfiles altamente demandados, como el de diseñador de lenguajes, desarrollador de compiladores, investigador en teoría de la computación o experto en análisis algorítmico. Además, este tipo de preparación resulta clave para incorporarse en sectores estratégicos como la ciberseguridad, el procesamiento del lenguaje natural o la computación cuántica, donde se valora cada vez más el dominio profundo de los fundamentos computacionales.

Modelarás procesos computacionales utilizando sistemas formales y matemáticos de vanguardia"

Perfil del egresado

Este programa impulsa el desarrollo de un perfil técnico y analítico, capaz de comprender y aplicar los principios que sustentan la lógica computacional moderna. Gracias a una visión integradora entre teoría y práctica, el egresado se caracteriza por su habilidad para diseñar soluciones eficientes, modelar sistemas complejos y desarrollar herramientas innovadoras en diversos entornos tecnológicos. Además, su dominio de Lenguajes formales, algoritmos avanzados y estructuras computacionales lo posiciona como un profesional altamente competente en entornos académicos, científicos o industriales donde se requiere un conocimiento profundo del comportamiento interno de los sistemas digitales.

Incorporarás técnicas de computación bioinspirada en proyectos de innovación que emulan procesos naturales y evolutivos.

- **Pensamiento crítico y resolución de problemas complejos:** Capacidad para analizar situaciones desde una perspectiva lógica y proponer soluciones fundamentadas
- **Comunicación técnica eficaz:** Habilidad para expresar conceptos computacionales de manera clara y precisa, tanto oralmente como por escrito
- **Trabajo colaborativo en entornos multidisciplinares:** Disposición para integrarse y contribuir en equipos con perfiles diversos del ámbito tecnológico
- **Adaptación al cambio y aprendizaje continuo:** Disposición para actualizar conocimientos en un sector marcado por la innovación constante

Después de realizar el programa universitario, podrás desempeñar tus conocimientos y habilidades en los siguientes cargos:

1. **Ingeniero de *software* especializado en compiladores:** Diseña e implementa traductores de lenguajes de programación, optimizando su rendimiento y funcionalidad.
2. **Arquitecto de Lenguajes de programación:** Crea nuevos lenguajes o adapta los existentes según necesidades específicas de sectores tecnológicos.
3. **Investigador en computación teórica:** Estudia modelos de cómputo, complejidad algorítmica y lenguajes formales para resolver problemas abstractos aplicables a sistemas reales.
4. **Desarrollador de sistemas inteligentes:** Construye soluciones basadas en lógica computacional y procesamiento del lenguaje natural para aplicaciones avanzadas.
5. **Analista de algoritmos:** Evalúa y optimiza algoritmos para mejorar su eficiencia en contextos de alto rendimiento computacional.
6. **Especialista en procesamiento del lenguaje natural:** Diseña sistemas capaces de interpretar, generar o traducir lenguaje humano en entornos digitales.
7. **Ingeniero en informática gráfica:** Desarrolla soluciones visuales interactivas, simulaciones y entornos gráficos de alto nivel para distintas industrias.
8. **Consultor en estructuras computacionales:** Asesora en la selección e implementación de modelos y estructuras para el diseño de software eficiente.

> *Implementarás soluciones gráficas interactivas y visualizaciones útiles en el desarrollo de software"*

06

Licencias de software incluidas

TECH es referencia en el mundo universitario por combinar la última tecnología con las metodologías docentes para potenciar el proceso de enseñanza-aprendizaje. Para ello, ha establecido una red de alianzas que le permite tener acceso a las herramientas de software más avanzadas del mundo profesional.

Al matricularte recibirás, de forma completamente gratuita, las credenciales de uso académico de las siguientes aplicaciones de software profesional”

TECH ha establecido una red de alianzas profesionales en la que se encuentran los principales proveedores de software aplicado a las diferentes áreas profesionales. Estas alianzas permiten a TECH tener acceso al uso de centenares de aplicaciones informáticas y licencias de software para acercarlas a sus estudiantes.

Las licencias de software para uso académico permitirán a los estudiantes utilizar las aplicaciones informáticas más avanzadas en su área profesional, de modo que podrán conocerlas y aprender su dominio sin tener que incurrir en costes. TECH se hará cargo del procedimiento de contratación para que los alumnos puedan utilizarlas de modo ilimitado durante el tiempo que estén estudiando el programa de Máster Título Propio en Computación y Lenguajes, y además lo podrán hacer de forma completamente gratuita.

TECH te dará acceso gratuito al uso de las siguientes aplicaciones de software:

Google Career Launchpad

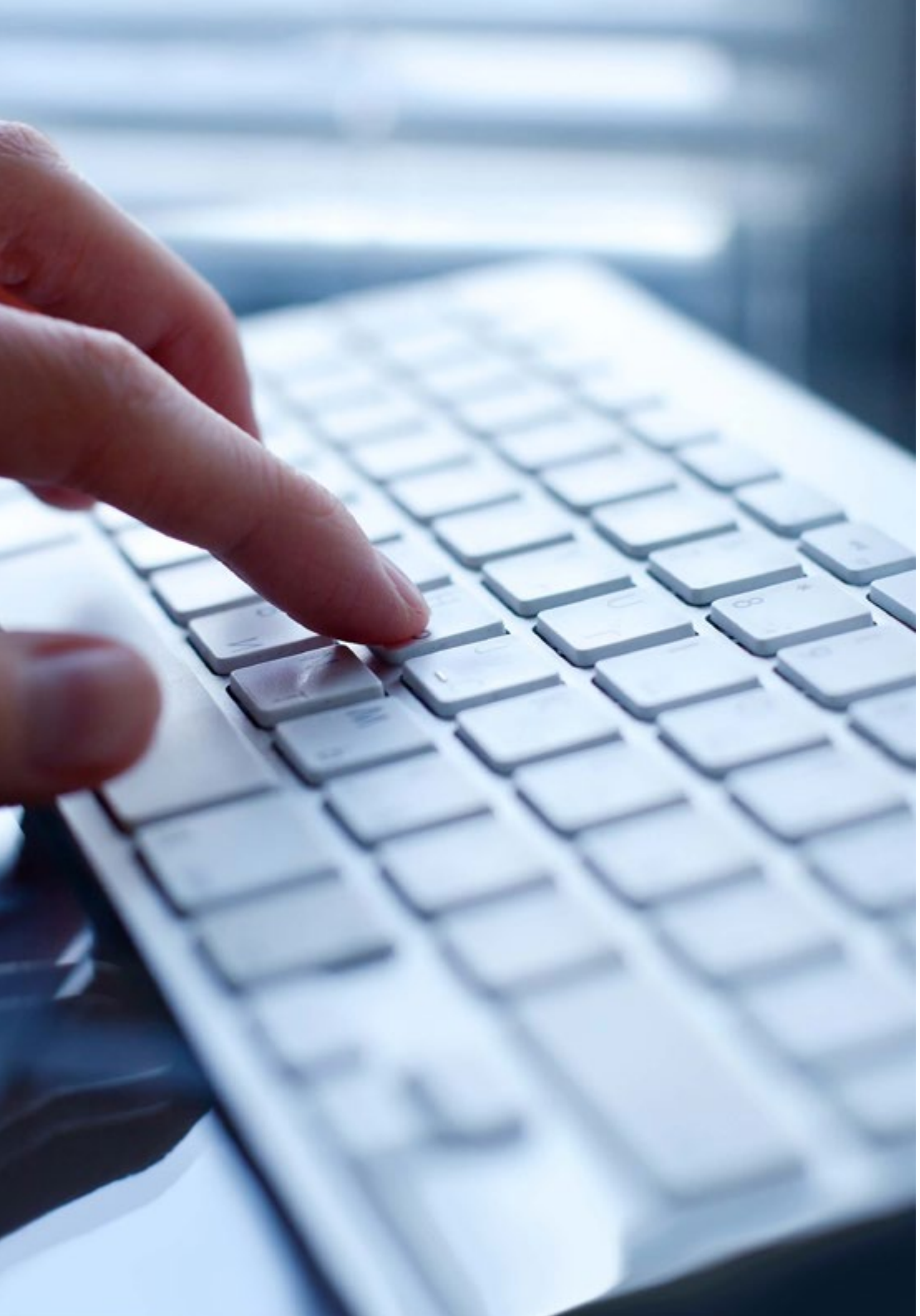
Google Career Launchpad es una solución para desarrollar habilidades digitales en tecnología y análisis de datos. Con un valor estimado de **5.000 dólares**, se incluye de forma **gratuita** en el programa universitario de TECH, brindando acceso a laboratorios interactivos y certificaciones reconocidas en el sector.

Esta plataforma combina capacitación técnica con casos prácticos, usando tecnologías como BigQuery y Google AI. Ofrece entornos simulados para experimentar con datos reales, junto a una red de expertos para orientación personalizada.

Funciones destacadas:

- **Cursos especializados:** contenido actualizado en cloud computing, machine learning y análisis de datos
- **Laboratorios en vivo:** prácticas con herramientas reales de Google Cloud sin configuración adicional
- **Certificaciones integradas:** preparación para exámenes oficiales con validez internacional
- **Mentorías profesionales:** sesiones con expertos de Google y partners tecnológicos
- **Proyectos colaborativos:** retos basados en problemas reales de empresas líderes

En conclusión, **Google Career Launchpad** conecta a los usuarios con las últimas tecnologías del mercado, facilitando su inserción en áreas como inteligencia artificial y ciencia de datos con credenciales respaldadas por la industria.



DBeaver Enterprise Edition

DBeaver Enterprise Edition es la versión profesional del reconocido gestor de bases de datos DBeaver, con un precio comercial aproximado de **250 euros** anuales. Durante el programa universitario en TECH se ofrece **gratis**, permitiendo a los egresados administrar, desarrollar y analizar datos en entornos complejos de manera profesional y segura.

Esta plataforma capacita al egresado TECH para optimizar la gestión de bases de datos relacionales y no relacionales, generar consultas SQL inteligentes, diseñar esquemas avanzados y visualizar información con gráficos interactivos. Además, integra funciones de análisis empresarial conectando con herramientas de *Business Intelligence*, convirtiendo datos en conocimiento estratégico para decisiones.

Funciones destacadas:

- **Compatibilidad amplia:** soporta Oracle, SQL Server, PostgreSQL, MongoDB, Cassandra y más
- **Editor SQL avanzado:** autocompletado, depuración y asistentes inteligentes
- **Visualización de datos:** paneles interactivos y gráficos integrados
- **Integración con Tableau:** conexión directa con herramientas de *Business Intelligence*
- **Diseño de esquemas:** edición de ERD e ingeniería inversa
- **Administración completa:** *backup*, restauración, comparación y gestión de usuarios

En conclusión, **DBeaver Enterprise Edition** impulsa al egresado TECH a dominar la gestión de datos con precisión, eficiencia e innovación.

07

Metodología de estudio

TECH es la primera universidad en el mundo que combina la metodología de los ***case studies*** con el ***Relearning***, un sistema de aprendizaje 100% online basado en la reiteración dirigida.

Esta disruptiva estrategia pedagógica ha sido concebida para ofrecer a los profesionales la oportunidad de actualizar conocimientos y desarrollar competencias de un modo intensivo y riguroso. Un modelo de aprendizaje que coloca al estudiante en el centro del proceso académico y le otorga todo el protagonismo, adaptándose a sus necesidades y dejando de lado las metodologías más convencionales.

TECH te prepara para afrontar nuevos retos en entornos inciertos y lograr el éxito en tu carrera"

El alumno: la prioridad de todos los programas de TECH

En la metodología de estudios de TECH el alumno es el protagonista absoluto. Las herramientas pedagógicas de cada programa han sido seleccionadas teniendo en cuenta las demandas de tiempo, disponibilidad y rigor académico que, a día de hoy, no solo exigen los estudiantes sino los puestos más competitivos del mercado.

Con el modelo educativo asincrónico de TECH, es el alumno quien elige el tiempo que destina al estudio, cómo decide establecer sus rutinas y todo ello desde la comodidad del dispositivo electrónico de su preferencia. El alumno no tendrá que asistir a clases en vivo, a las que muchas veces no podrá acudir. Las actividades de aprendizaje las realizará cuando le venga bien. Siempre podrá decidir cuándo y desde dónde estudiar.

“En TECH NO tendrás clases en directo (a las que luego nunca puedes asistir)”

Los planes de estudios más exhaustivos a nivel internacional

TECH se caracteriza por ofrecer los itinerarios académicos más completos del entorno universitario. Esta exhaustividad se logra a través de la creación de temarios que no solo abarcan los conocimientos esenciales, sino también las innovaciones más recientes en cada área.

Al estar en constante actualización, estos programas permiten que los estudiantes se mantengan al día con los cambios del mercado y adquieran las habilidades más valoradas por los empleadores. De esta manera, quienes finalizan sus estudios en TECH reciben una preparación integral que les proporciona una ventaja competitiva notable para avanzar en sus carreras.

Y además, podrán hacerlo desde cualquier dispositivo, pc, tableta o smartphone.

El modelo de TECH es asincrónico, de modo que te permite estudiar con tu pc, tableta o tu smartphone donde quieras, cuando quieras y durante el tiempo que quieras”

Case studies o Método del caso

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de negocios del mundo. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, su función era también presentarles situaciones complejas reales. Así, podían tomar decisiones y emitir juicios de valor fundamentados sobre cómo resolverlas. En 1924 se estableció como método estándar de enseñanza en Harvard.

Con este modelo de enseñanza es el propio alumno quien va construyendo su competencia profesional a través de estrategias como el *Learning by doing* o el *Design Thinking*, utilizadas por otras instituciones de renombre como Yale o Stanford.

Este método, orientado a la acción, será aplicado a lo largo de todo el itinerario académico que el alumno emprenda junto a TECH. De ese modo se enfrentará a múltiples situaciones reales y deberá integrar conocimientos, investigar, argumentar y defender sus ideas y decisiones. Todo ello con la premisa de responder al cuestionamiento de cómo actuaría al posicionarse frente a eventos específicos de complejidad en su labor cotidiana.

Método *Relearning*

En TECH los *case studies* son potenciados con el mejor método de enseñanza 100% online: el *Relearning*.

Este método rompe con las técnicas tradicionales de enseñanza para poner al alumno en el centro de la ecuación, proveyéndole del mejor contenido en diferentes formatos. De esta forma, consigue repasar y reiterar los conceptos clave de cada materia y aprender a aplicarlos en un entorno real.

En esta misma línea, y de acuerdo a múltiples investigaciones científicas, la reiteración es la mejor manera de aprender. Por eso, TECH ofrece entre 8 y 16 repeticiones de cada concepto clave dentro de una misma lección, presentada de una manera diferente, con el objetivo de asegurar que el conocimiento sea completamente afianzado durante el proceso de estudio.

El Relearning te permitirá aprender con menos esfuerzo y más rendimiento, implicándote más en tu especialización, desarrollando el espíritu crítico, la defensa de argumentos y el contraste de opiniones: una ecuación directa al éxito.

Un Campus Virtual 100% online con los mejores recursos didácticos

Para aplicar su metodología de forma eficaz, TECH se centra en proveer a los egresados de materiales didácticos en diferentes formatos: textos, vídeos interactivos, ilustraciones y mapas de conocimiento, entre otros. Todos ellos, diseñados por profesores cualificados que centran el trabajo en combinar casos reales con la resolución de situaciones complejas mediante simulación, el estudio de contextos aplicados a cada carrera profesional y el aprendizaje basado en la reiteración, a través de audios, presentaciones, animaciones, imágenes, etc.

Y es que las últimas evidencias científicas en el ámbito de las Neurociencias apuntan a la importancia de tener en cuenta el lugar y el contexto donde se accede a los contenidos antes de iniciar un nuevo aprendizaje. Poder ajustar esas variables de una manera personalizada favorece que las personas puedan recordar y almacenar en el hipocampo los conocimientos para retenerlos a largo plazo. Se trata de un modelo denominado *Neurocognitive context-dependent e-learning* que es aplicado de manera consciente en esta titulación universitaria.

Por otro lado, también en aras de favorecer al máximo el contacto mentor-alumno, se proporciona un amplio abanico de posibilidades de comunicación, tanto en tiempo real como en diferido (mensajería interna, foros de discusión, servicio de atención telefónica, email de contacto con secretaría técnica, chat y videoconferencia).

Asimismo, este completísimo Campus Virtual permitirá que el alumnado de TECH organice sus horarios de estudio de acuerdo con su disponibilidad personal o sus obligaciones laborales. De esa manera tendrá un control global de los contenidos académicos y sus herramientas didácticas, puestas en función de su acelerada actualización profesional.

La modalidad de estudios online de este programa te permitirá organizar tu tiempo y tu ritmo de aprendizaje, adaptándolo a tus horarios"

La eficacia del método se justifica con cuatro logros fundamentales:

1. Los alumnos que siguen este método no solo consiguen la asimilación de conceptos, sino un desarrollo de su capacidad mental, mediante ejercicios de evaluación de situaciones reales y aplicación de conocimientos.
2. El aprendizaje se concreta de una manera sólida en capacidades prácticas que permiten al alumno una mejor integración en el mundo real.
3. Se consigue una asimilación más sencilla y eficiente de las ideas y conceptos, gracias al planteamiento de situaciones que han surgido de la realidad.
4. La sensación de eficiencia del esfuerzo invertido se convierte en un estímulo muy importante para el alumnado, que se traduce en un interés mayor en los aprendizajes y un incremento del tiempo dedicado a trabajar en el curso.

La metodología universitaria mejor valorada por sus alumnos

Los resultados de este innovador modelo académico son constatables en los niveles de satisfacción global de los egresados de TECH.

La valoración de los estudiantes sobre la calidad docente, calidad de los materiales, estructura del curso y sus objetivos es excelente. No en valde, la institución se convirtió en la universidad mejor valorada por sus alumnos según el índice global score, obteniendo un 4,9 de 5.

Accede a los contenidos de estudio desde cualquier dispositivo con conexión a Internet (ordenador, tablet, smartphone) gracias a que TECH está al día de la vanguardia tecnológica y pedagógica.

Podrás aprender con las ventajas del acceso a entornos simulados de aprendizaje y el planteamiento de aprendizaje por observación, esto es, Learning from an expert.

Así, en este programa estarán disponibles los mejores materiales educativos, preparados a conciencia:

Material de estudio

Todos los contenidos didácticos son creados por los especialistas que van a impartir el curso, específicamente para él, de manera que el desarrollo didáctico sea realmente específico y concreto.

Estos contenidos son aplicados después al formato audiovisual que creará nuestra manera de trabajo online, con las técnicas más novedosas que nos permiten ofrecerte una gran calidad, en cada una de las piezas que pondremos a tu servicio.

Prácticas de habilidades y competencias

Realizarás actividades de desarrollo de competencias y habilidades específicas en cada área temática. Prácticas y dinámicas para adquirir y desarrollar las destrezas y habilidades que un especialista precisa desarrollar en el marco de la globalización que vivimos.

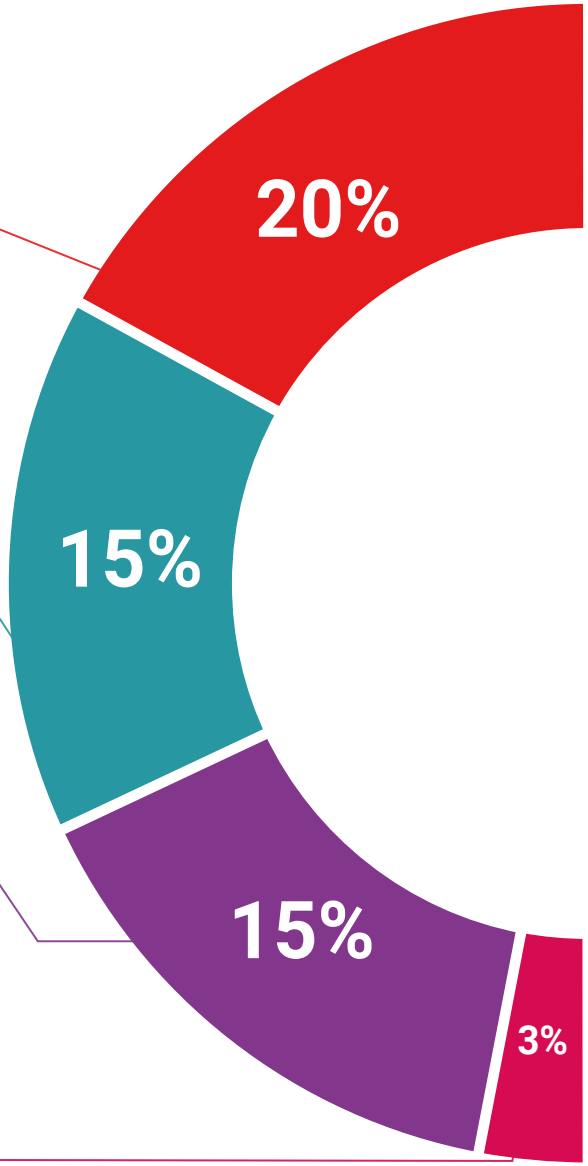
Resúmenes interactivos

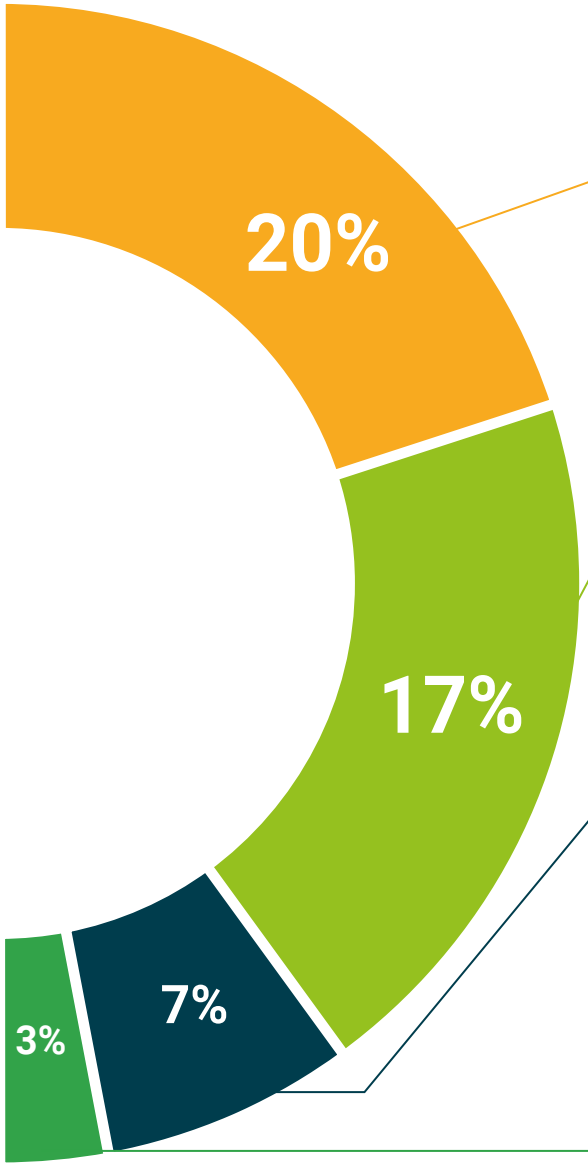
Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

Este sistema exclusivo educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".

Lecturas complementarias

Artículos recientes, documentos de consenso, guías internacionales... En nuestra biblioteca virtual tendrás acceso a todo lo que necesitas para completar tu capacitación.





Case Studies

Completarás una selección de los mejores *case studies* de la materia. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.

Testing & Retesting

Evaluamos y reevaluamos periódicamente tu conocimiento a lo largo del programa. Lo hacemos sobre 3 de los 4 niveles de la Pirámide de Miller.

Clases magistrales

Existe evidencia científica sobre la utilidad de la observación de terceros expertos.

El denominado *Learning from an expert* afianza el conocimiento y el recuerdo, y genera seguridad en nuestras futuras decisiones difíciles.

Guías rápidas de actuación

TECH ofrece los contenidos más relevantes del curso en forma de fichas o guías rápidas de actuación. Una manera sintética, práctica y eficaz de ayudar al estudiante a progresar en su aprendizaje.

08

Cuadro docente

El cuadro docente de este programa está conformado por expertos con amplia trayectoria académica y profesional en áreas clave de la computación y los lenguajes formales. Gracias a su experiencia en investigación aplicada y en el desarrollo de soluciones tecnológicas avanzadas, ofrecen una visión actualizada y rigurosa del sector. Además, su participación en proyectos internacionales permite trasladar al aula casos reales y metodologías innovadoras, lo que enriquece notablemente el proceso de aprendizaje. Esta combinación de conocimiento profundo y enfoque práctico garantiza una experiencia académica de alto nivel, alineada con las exigencias del entorno científico y tecnológico actual.

Estarás asesorado en todo momento por el equipo docente, integrado por especialistas de renombre en Computación y Lenguajes informáticos"

Director Invitado Internacional

Dr. Jeremy Gibbons es considerado una **eminencia internacional** por sus contribuciones en el campo de la **Metodología de la Programación** y sus aplicaciones en **Ingeniería de Software**. Por más de dos décadas, este experto asociado al Departamento de Ciencias de la Computación de la Universidad de Oxford ha impulsado **diferentes proyectos de desarrollo** cuyos resultados más palpables son aplicados por informáticos de diversas partes del mundo.

Su trabajo abarca áreas como la **programación genérica,** los métodos formales, la biología computacional, la bioinformática y el diseño de algoritmos con Haskell. Este último tema lo desarrolló ampliamente de conjunto con su mentor, el Doctor Richard Bird.

Desde su rol como **Director** del **Grupo de Investigación en Álgebra de Programación**, Gibbons ha propiciado avances con relación a los **Lenguajes de Programación Funcional** y la **Teoría de Patrones en Programación**. Al mismo tiempo, las aplicaciones de sus innovaciones han estado ligadas al marco sanitario, como lo demuestra su colaboración con **CancerGrid** y **Datatype-Generic Programming**. A su vez, estas y otras iniciativas reflejan su interés por resolver problemas prácticos en la **investigación** del **Cáncer** y la **Informática Clínica**.

Asimismo, Gibbons también ha dejado una huella significativa como **Editor en Jefe** de **publicaciones académicas** en The Journal of Functional Programming y The Programming Journal: The Art, Science, and Engineering of Programming. A través de esas responsabilidades ha llevado a cabo una intensa labor de **divulgación** y **diseminación del conocimiento**. Además, ha liderado varias cátedras de estudio vinculadas a instituciones de renombre como la Universidad Oxford Brookes y en la Universidad de Auckland, Nueva Zelanda.

Por otro lado, este especialista es miembro del Grupo de Trabajo 2.1 sobre Lenguajes Algorítmicos y Cálculos de la **Federación Internacional para el Procesamiento de la Información (IFIP)**. Con esta organización ofrece mantenimiento a los lenguajes de programación ALGOL 60 y ALGOL 68.

Dr. Gibbons, Jeremy

- Director del Programa de Ingeniería de Software de la Universidad de Oxford, Reino Unido
- Subdirector del Laboratorio de Informática y Departamento de Ciencias de la Computación de la Universidad de Oxford
- Catedrático en Kellogg College, la Universidad Oxford Brookes y en la Universidad de Auckland de Nueva Zelanda
- Director del Grupo de Investigación Álgebra de la Programación
- Redactor Jefe de las revistas The Art, Science, and Engineering of Programming y Journal of Functional Programming
- Doctor en Ciencias Informáticas por la Universidad de Oxford
- Licenciado en Informática por la Universidad de Edimburgo
- Miembro de: Grupo de Trabajo 2.1 sobre Lenguajes Algorítmicos y Cálculos de la Federación Internacional para el Procesamiento de la Información (IFIP)

Gracias a TECH podrás aprender con los mejores profesionales del mundo"

09

Titulación

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

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

Este **Máster Título Propio en Computación y Lenguajes** contiene el programa universitario 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.**

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

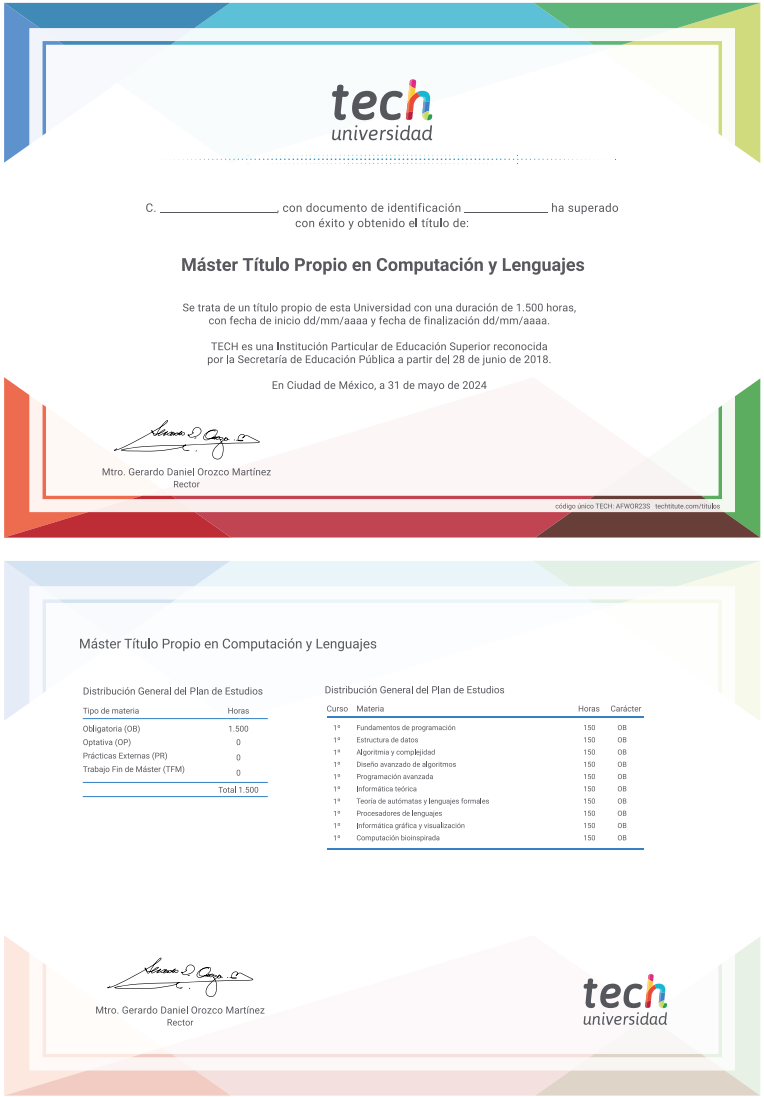
TECH es miembro de la **Association for Computing Machinery (ACM)**, la red internacional que agrupa a los principales referentes en computación y ciencias de la información. Esta distinción refuerza su compromiso con la excelencia académica, la innovación tecnológica y la capacitación de profesionales en el ámbito digital.

Aval/Membresía

Título: **Máster Título Propio en Computación y Lenguajes**

Modalidad: **No escolarizada (100% en línea)**

Duración: **12 meses**



tech universidad

C. _____, con documento de identificación _____ ha superado con éxito y obtenido el título de:

Máster Título Propio en Computación y Lenguajes

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.

En Ciudad de México, a 31 de mayo de 2024

Mtro. Gerardo Daniel Orozco Martínez
Rector

código único TECH: AFWOR23S techtitute.com/titulos

Máster Título Propio en Computación y Lenguajes

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	Fundamentos de programación	150	OB
1º	Estructura de datos	150	OB
1º	Algoritmia y complejidad	150	OB
1º	Diseño avanzado de algoritmos	150	OB
1º	Programación avanzada	150	OB
1º	Informática teórica	150	OB
1º	Teoría de autómatas y lenguajes formales	150	OB
1º	Procesadores de lenguajes	150	OB
1º	Informática gráfica y visualización	150	OB
1º	Computación bioinspirada	150	OB

Mtro. Gerardo Daniel Orozco Martínez
Rector

tech universidad

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

tech universidad

Máster Título Propio
Computación y Lenguajes

- Modalidad: **No escolarizada (100% en línea)**
- Duración: **12 meses**
- Titulación: **TECH Universidad**
- Horario: **a tu ritmo**
- Exámenes: **online**

Máster Título Propio

Computación y Lenguajes

Aval/Membresía

Association
for Computing
Machinery

tech universidad