

TÓPICOS EXAMEN DE TÍTULO

Título Ingeniero Civil de Computación

Estructuras de Datos y Algoritmos

- Estructuras de datos básicas (arreglos, listas ligadas, stacks, colas, tablas de hash; colas priorizadas); Árboles de búsqueda binarios balanceados; y árboles-B.
- Algoritmos de ordenación (Insertionsort, Heapsort, Mergesort, Quicksort, y ordenación en tiempo $O(n)$); técnicas algorítmicas (dividir para conquistar; back-tracking; programación dinámica; algoritmos codiciosos).
- Grafos: representaciones, exploraciones DFS y BFS, y algoritmos fundamentales basados sobre BFS y DFS, árboles de cobertura de costo mínimo, rutas más cortas.

Ingeniería de Software

- Modelos de proceso genéricos (cascada, iterativo, incremental), procesos ágiles, scrum, otros procesos (RUP, Kanban, etc).
- Levantamiento de Requisitos (requisitos funcionales y no funcionales, relatos de usuario, casos de uso).
- Análisis (modelos de dominio, el diagrama de clases UML).
- Diseño (acoplamiento y cohesión, patrones de diseño, diagramas de secuencia y de componentes UML).

Lógica para Ciencia de la Computación

- Lógica proposicional como una herramienta de modelación y razonamiento para poder explicar resultados básicos en este tipo de lógica.
- Demostración y explicación de resultados fundamentales sobre teorías axiomatizables, recursivamente enumerables, decidibles y completas.
- Demostración y explicación de nociones básicas de complejidad computacional, y la importancia de la lógica proposicional en su estudio y, en particular, en la demostración del teorema de Cook.
- Demostración y explicación de nociones básicas sobre teorías, en especial la teoría de primer orden de los números naturales, y, en particular, el teorema de Gödel.
- Demostración mecánica de teoremas.

Teoría de Autómatas y Lenguajes Formales

- Autómatas deterministas, autómatas no-deterministas, equivalencia entre ambos modelos.
- Expresiones regulares, equivalencia con el modelo de autómatas, autómatas no-deterministas con transición en vacío.
- Propiedades de lenguajes regulares: lema de bombeo, propiedades de clausura, problemas sobre lenguajes regulares, algoritmos.
- Teorema de Myhill-Nerode, algoritmo de minimización
- Gramáticas libres de contexto, simplificación de gramáticas, formas normales.

- Autómatas apiladores deterministas, no-deterministas, equivalencia con gramáticas libres de contexto.
- Propiedades de lenguajes libres de contexto: lema de bombeo, propiedades de clausura, algoritmos.
- Máquinas de Turing, lenguajes recursivamente enumerables y decidibles.
- Técnicas para la construcción de máquinas de Turing, extensiones de la máquina de Turing, equivalencias.
- Propiedades de lenguajes decidibles y recursivamente enumerables.
- Problemas indecidibles: máquina de Turing universal, reducciones.

Programación Avanzada

- Programación Orientada a Objetos (objetos, herencia, herencia múltiple, clases abstractas, properties).
- Estructuras de Datos (árboles, diccionarios, colas, stacks, sets).
- Programación Funcional (algunas funciones especiales de Python, comprensión de listas, iterables e iteradores, generadores, funciones lambda, Map, Reduce, Filter, decoradores).
- Manejo de Excepciones (tipos de excepciones, control de excepciones).
- Simulación: introducción a la simulación DES (Discrete Event Simulation).
- Threading (creación y sincronización de threads, concurrencia).
- I/O Manejo de Strings (bytes, serialización en formato binario, serialización en formato JSON).
- Networking (Sockets, cliente, servidor).

Diseño y Análisis de Algoritmos

- Análisis de algoritmos empleando técnicas de ecuaciones de recurrencia, notación asintótica, e inducción; demostración de cotas inferiores para la complejidad de algunos problemas.
- Diseño de algoritmos basados sobre las técnicas de dividir y conquistar, programación dinámica, algoritmos codiciosos, backtracking y branch-and-bound.
- Algoritmos eficientes para resolver problemas de ordenación, búsqueda, secuenciamiento de tareas y problemas en grafos.
- Estructuras de datos avanzadas y los algoritmos útiles para manipularlas.
- Explicación de la noción de complejidad, y las nociones de NP-completitud y reducción.
- Algoritmos para problemas de grafos.

Sistemas Operativos y Redes

- Limitaciones y ventajas de los mecanismos utilizados en un sistema operativo.
- Subsistemas de un sistema operativo, considerando criterios de complejidad, costo, flexibilidad y seguridad.
- Administración de Procesos: creación, planificación (scheduling), sincronización; administración de memoria: direccionamiento, paginación, segmentación, memoria virtual.
- Sistemas de archivos (esquemas de asignación de bloques).

- Redes: modelos de capas, elementos de hardware, sockets.
- Protocolos de enlace: ARP y switching; protocolos de red: direccionamiento IP y routing; protocolos de transporte: TCP y UDP.

Arquitectura de Computadores

- Fundamentos: sistemas numéricos, representaciones posicionales y de punto flotante, operaciones aritméticas, representación de números negativos (complementos); implementación de operaciones y almacenamiento: lógica booleana, compuertas lógicas, circuitos de control, unidades de cómputo, elementos secuenciales (flip-flop); unidades de memoria endianness.
- Diseño y control de un computador: programabilidad: construcción de un computador básico, lenguaje de máquina (señales de control); unidad de control, ciclos y subrutinas, microarquitecturas (Harvard/Von Neumann); ISAs (RISC/CISC); Programación de bajo nivel: assembly y assembler, tipos de dato, tipos de direccionamiento, convenciones de llamada.
- Comunicación entre partes: interacción con dispositivos de I/O, conexión de dispositivos con CPU y memoria, mapeo de memoria y comunicación por puertos, interrupciones, DMA.
- Optimización de rendimiento: memoria caché (principios de localidad, jerarquía de memoria, métricas de rendimiento de una jerarquía de memoria, funciones del controlador de caché); multiprogramación (memoria virtual, soporte del hardware para memoria virtual, scheduling de procesos); paralelismo a nivel de instrucción (ILP), taxonomía de Flynn, GPUs.

Bases de Datos

- Modelos de datos de alto nivel e implementación bajo el paradigma relacional.
- Consultas de mediana complejidad utilizando el lenguaje de consulta SQL.
- Procedimientos almacenados y *triggers* sencillos usando el lenguaje nativo.
- Nociones básicas sobre la arquitectura y la implementación de sistemas de bases de datos.
- Modelos para almacenar información semiestructurada y desarrollo de consultas de complejidad leve en estos paradigmas.
- Teorema CAP y las distintas arquitecturas de bases de datos distribuidas.

Tecnologías y Aplicaciones Web

- Protocolos y estándares de la web
- Arquitectura de una aplicación web
- Lenguaje Ruby, framework Rails
- Controladores; modelos; vistas; HTML y CSS
- Manejo de sesión
- JavaScript y el DOM; APIs y REST
- Protocolos seguros
- Usabilidad

Sistemas Distribuidos

- Arquitecturas distribuidas: modelo cliente-servidor y P2P.
- Redes P2P y tablas de hash distribuidas (DHT).
- Algoritmos para resolver problemas de sincronización, consenso, elección de líderes y detección de deadlocks en un sistema distribuido.
- Sistemas basados sobre procesos/threads distribuidos comunicados de acuerdo a un modelo apropiado para el problema abordado, tomando en consideración las problemáticas inherentes a este tipo de sistemas.
- Modelos de consistencia y transacciones distribuidas.
- Tolerancia a fallas.

Inteligencia Artificial

- Agentes inteligentes, dotados de capacidad de percibir, razonar y actuar.
- Inferencia deductiva (programación en lógica, algoritmos de búsqueda heurística para un solo agente, construcción de heurísticas admisibles, búsqueda con adversarios).

Sistemas de Información

- Qué es información, tipos de información y cómo se modela la información de un sistema.
- Componentes de un sistema de información; aspectos y dificultades involucrados en el desarrollo de un proyecto de sistema de información.
- Ciclo de vida y tipos específicos de sistemas de información.
- Casos de empresas y sus necesidades relacionadas con sistemas de información; soluciones basadas sobre sistemas de información para los problemas detectados.
- Consideraciones éticas involucradas a lo largo del ciclo de vida de un sistema de información.
- Características y perfil profesional del gestor de sistemas de información.
- Importancia de la seguridad en los sistemas de información.
- Rol de sistemas de información en las organizaciones y en la sociedad.

Criptografía y Seguridad Computacional

- Criptografía Simétrica: cifradores perfectamente secretos y el One-Time Pad; indistinguibilidad computacional; generadores pseudo-aleatorios (PRG) y funciones pseudo-aleatorias (PRF); esquemas con cifrados indistinguibles basados sobre PRGs; esquemas seguros bajo ataques de texto plano escogido (CPA) basados sobre PRF; modos de operación para cifradores de bloque (o PRFs) (ECB, CBC, OFB, CTR); autenticación de mensajes y funciones de hash (CBC-MAC, HMAC); ataques de texto cifrado escogido y cifrados autenticados.

- Criptografía asimétrica: teoría de números para criptografía asimétrica (divisibilidad, primalidad, grupos, teorema del resto chino); supuestos criptográficos (factorización, RSA, logaritmo discreto, Diffie-Hellman Computacional y Diffie-Hellman Decisional); criptadores de llave pública (RSA, El-Gamal); esquemas de firmas digitales (RSA, Schnorr).
- Criptografía en la práctica: certificados, TLS.
- Tópicos avanzados en criptografía moderna: sistemas de demostración de nula divulgación, computación multipartita segura.

Reconocimiento de Patrones

- Extracción de características donde el reconocimiento de patrones no se realiza a partir de información visual.
- Selección de características (análisis de componentes principales, discriminante de Fisher, búsqueda exhaustiva, búsqueda secuencial, branch & bound).
- Clasificadores automáticos capaces de resolver problemas reales basados sobre técnicas de clasificador lineal, árbol binario de decisión, vecino más cercano, Mahalanobis, Bayes, SVM, redes neuronales y Deep Learning.
- Evaluación y comparación del desempeño de la clasificación (validación cruzada, bootstrap, e intervalos de confianza basados sobre distribuciones estadísticas).
- Aplicaciones.