

Grado en Ingeniería de la Ciberseguridad Online

Resultados de Aprendizaje y Contenidos de las Materias de Segundo Curso

Primer semestre

Técnicas de programación avanzadas

Resultados de Aprendizaje

- Emplear las principales estructuras de datos jerárquicas (árboles) para la resolución de problemas concretos y como mecanismo para plantear soluciones óptimas.
- Usar las principales estructuras de datos relacionales (grafos) para la resolución de problemas concretos y como mecanismo para plantear soluciones óptimas.
- Realizar cálculos de la complejidad teórica de un algoritmo y su orden de magnitud, para poder argumentar la elección de una solución frente a otra.
- Utilizar las principales estructuras de datos lineales asociativas (tablas hash) para la resolución de problemas concretos y como mecanismo para obtener soluciones óptimas.
- Emplear estrategias algorítmicas clásicas (divide y vencerás, avance rápido y vuelta atrás) a la hora de implementar soluciones a problemas concretos.
- Describir los mecanismos de evaluación de la eficiencia y complejidad de los algoritmos

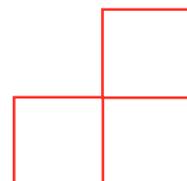
Contenidos

- Estructuras de datos jerárquicas.
- Estructuras de datos relacionales.
- Estructuras de datos funcionales.
- Técnicas avanzadas de diseño de algoritmos: Divide y vencerás, Avance rápido, Vuelta atrás.
- Análisis de algoritmos: eficiencia y complejidad
- Problemas clásicos. (búsquedas, camino corto, mochila, viajante, cambio de moneda)

Programación orientada a objetos

Resultados de Aprendizaje

- Gestionar el control de los programas mediante módulos.



- Aplicar los conceptos básicos de ingeniería del software para la resolución práctica de proyectos de mediana envergadura.
- Utilizar los conceptos básicos inherentes al paradigma de la orientación a objetos: diseño de clases, herencia, dependencias, asociaciones, multiplicidad, modificadores de acceso, sobrecarga, clases abstractas, polimorfismo e interfaces.
- Describir qué es un patrón de diseño software, aplicando algunos a modo de ejemplo.
- Utilizar correctamente herramientas y lenguajes de modelado.
- Implementar programas en un lenguaje de programación orientado a objetos.

Contenidos

- Implementación de Clases.
- Atributos, constructores, métodos.
- Herencia. Colecciones
- Diseño avanzado de Clases. Sobrecarga y reescritura de métodos. Clases abstractas. Polimorfismo. Interfaces
- Aseguramiento de calidad y Patrones de diseño.
- Interfaces gráficas

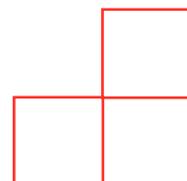
Programación concurrente y distribuida

Resultados de Aprendizaje

- Describir cómo se ejecutan de forma concurrente varios procesos en un sólo procesador mediante multiplexación de tiempo.
- Implementar programas concurrentes utilizando los mecanismos de sincronización y comunicación
- Describir los problemas inherentes a la programación concurrente y la solución al problema de la exclusión mutua.
- Identificar las arquitecturas de los sistemas de n capas cliente servidor y los mecanismos de comunicación inherentes.
- Desarrollar programas informáticos con varias líneas de ejecución basadas en multihilos
- Diferenciar los modelos actuales de computación distribuida y de computación en la nube.

Contenidos

- Conceptos de programación concurrente
- Mecanismos de sincronización y de comunicación (exclusión mutua, cerrojos, semáforos y monitores, paso de mensajes).
- Desarrollo de Aplicaciones en N-capas
- Desarrollo de Aplicaciones C/S utilizando sockets





- Programación multihilo (multithreads)
- Llamadas a procedimientos remotos

Arquitectura de computadores

Resultados de Aprendizaje

- Solucionar sistemas digitales secuenciales sencillos.
- Analizar el almacenamiento de datos, las operaciones aritméticas y las sentencias condicionales en lenguaje ensamblador
- Analizar el manejo de la pila software, el funcionamiento de las "funciones" y paso de mensajes, y el puerto serie y la comunicación con el PC.
- Construir aplicaciones de programación a bajo nivel.
- Describir el funcionamiento interno de una unidad de Control.
- Analizar los problemas de ordenadores multi-procesador

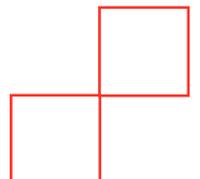
Contenidos

- Sistemas digitales y Microcontroladores
- Manejo de datos
- Ejecución de código
- Arquitecturas y características avanzadas
- Componentes de una computadora digital
- Programación en lenguaje ensamblador
- Microcontroladores

Criptografía

Resultados de Aprendizaje

- Explicar el concepto de criptografía y la teoría de la complejidad
- Aplicar las técnicas de criptografía simétrica.
- Desarrollar un sistema de intercambio y comprobación de hash
- Aplicar las técnicas de criptografía asimétrica
- Desarrollar un sistema de firma digital basado en cifrado asimétrico
- Describir las bases y aplicaciones de las técnicas de criptografía avanzada





Contenidos

- Introducción a la criptografía. Teoría de Complejidad
- Criptografía clásica. Cifrado simétrico.
- Funciones Hash.
- Cifrado asimétrico. Intercambio de clave. Firma digital.
- Esquemas de compromiso.
- Compartición de secretos

Segundo semestre

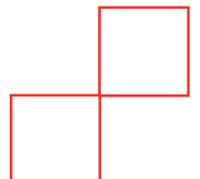
Sistemas operativos

Resultados de Aprendizaje

- Explicar cómo se organiza la ejecución de diferentes programas en máquinas con uno o varios procesadores.
- Interpretar la organización y funcionamiento de algunos sistemas operativos actuales, en especial los de la familia Microsoft y Unix
- Planificar la utilización de un determinado sistema operativo en relación con el contexto de trabajo.
- Diseñar programas y utilidades de apoyo a los sistemas operativos.
- Describir cómo los sistemas operativos realizan la gestión de memoria, E/S y ficheros
- Describir los principales retos de la securización de los sistemas operativos

Contenidos

- Introducción a las funciones y estructura de un sistema operativo moderno, sus componentes software fundamentales.
- Procesos y nociones de concurrencia.
- La interfaz de programación de aplicaciones (API).
- Planificación de procesos y rendimiento,
- Gestión de memoria principal, gestión de dispositivos de entrada/salida y gestión del sistema de ficheros.
- Seguridad en sistemas operativos





Bases de datos

Resultados de Aprendizaje

- Explicar el concepto de modelo entidad-relación, junto al modelo relacional
- Describir los principales problemas a tener en cuenta en el diseño de bases de datos
- Reconocer un sistema gestor de base de datos comercial.
- Generar consultas de acceso y/o modificación de una Base de Datos.
- Desarrollar una aplicación de consulta y/o modificación de datos de una Base de Datos.
- Diseñar Bases de Datos.

Contenidos

- Análisis de bases de datos y el modelo entidad-relación
- Teoría de Bases de datos y el modelo relacional
- Diseño de Bases de Datos. Problemas fundamentales en las bases de datos
- Sistemas de Gestión de Bases de Datos
- Lenguajes de consulta relacionales
- Bases de datos avanzadas.

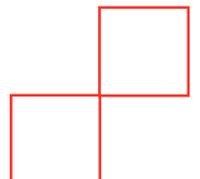
Técnicas de Hacking

Resultados de Aprendizaje

- Diferenciar las técnicas de recogida de información basadas en footprinting, OSINT, y otras técnicas no invasivas
- Discriminar las técnicas de recogida de información basadas en fingerprinting y otras técnicas invasivas
- Diferenciar las técnicas de hacking sobre sistemas procesadores
- Aplicar las técnicas de hacking sobre aplicaciones tradicionales y entornos web
- Aplicar las técnicas de hacking sobre redes de comunicaciones
- Aplicar las técnicas de hacking mediante mecanismos de denegación de servicio

Contenidos

- Técnicas de recogida de información
- Hacking de procesadores y firmware
- Técnicas básicas de explotación en Windows y Linux
- Ingeniería inversa y desbordamientos. Inyecciones y forgeries
- Envenenamientos, MitM, suplantaciones y secuestros





- Denegaciones de servicio

Metodologías de desarrollo seguro

Resultados de Aprendizaje

- Describir los ciclos de desarrollo seguro, modelos de madurez, gestión del cambio y entornos DevOps dentro del esquema de seguridad en el desarrollo de software
- Describir los diferentes métodos de desarrollo de software que la industria actualmente utiliza, así como los mecanismos de gestión del conocimiento empleados
- Diferenciar las best practice de desarrollo de código más habituales
- Describir las amenazas existentes en los entornos de desarrollo de software
- Aplicar los mecanismos de securización de software
- Diferenciar las diferentes alternativas para evaluar la eficacia de un software en cuanto a su securización

Contenidos

- Esquema de seguridad en el desarrollo de software
- Controles de entorno y seguridad
- Seguridad del entorno de software
- Problemas de seguridad en el código fuente
- Mecanismos de protección del software
- Evaluación de la eficacia de la seguridad del software

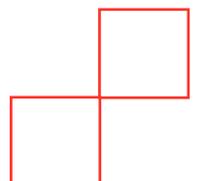
Metodologías de gestión de proyectos

Resultados de Aprendizaje

- Diferenciar entre metodologías waterfall y agile, y los proyectos tipo donde aplicarlas
- Describir la organización de un proyecto con metodología waterfall
- Aplicar la metodología waterfall de PMI
- Describir la organización de un proyecto con metodología SCRUM
- Diferenciar los roles de un Product Owner y un Scrum Master dentro de la gestión de un proyecto agile
- Aplicar la metodología agile SCRUM

Contenidos

- Introducción: Los proyectos y su entorno
- Áreas de conocimiento, fases y grupos de procesos.





- Fases en la Gestión de Proyectos según el Project Management Institute
- Metodologías Agile - SCRUM
- Product Owner vs Scrum Master
- Diseño de producto para construcción iterativa-incremental. Gestión del Product Backlog

