

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: FUNDAMENTOS DE LA PROGRAMACIÓN

CÓDIGO: CC212

CRÉDITOS: 06

PRE-REQUISITOS:

- BIC01 - INTRODUCCIÓN A LA CIENCIA DE LA COMPUTACIÓN

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Fundamentos de la programación será la asignatura que sentará las bases y establecerá los conceptos básicos y principales de la carrera de Ciencia de la Computación, siendo esta la asignatura base de la que el alumno desarrollará sus destrezas hacia un científico de la computación.

En este sentido, este curso introducirá a los participantes en los conceptos fundamentales de este arte.

Los tópicos incluyen tipos de datos, estructuras de control, funciones, listas, recursividad y la mecánica de la ejecución, prueba y depuración.

### **COMPETENCIAS**

- Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería
- Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería
- Escribir algoritmos básicos para la resolución de problemas científicos
- Conocimiento de la metodología de programación
- Adquisición de competencias específicas de la utilización de los lenguajes de programación
- Desarrollar aplicaciones en base a problemas de lógica computacional

### **PROGRAMA ANALÍTICO**

#### **1. Lenguajes de programación**

- a. Lenguajes naturales y artificiales
- b. Elementos y evolución
- c. Diseño e implementación

- d. Taxonomía, paradigmas y popularidad
- 2. Representación de datos simples en memoria**
  - a. Definición de programa y sus partes
  - b. Conceptos básicos de la programación: variables, tipos de datos, constantes, operadores y sentencias
  - c. Almacenando información e identificadores
- 3. Entrada y salida de datos**
  - a. Escribir y leer información: funciones de entrada y salida de datos
- 4. Sentencias de control**
  - a. Introducción a la programación estructurada
  - b. Estructuras selectivas
  - c. Estructuras iterativas
- 5. Funciones y Recursividad**
  - a. Empleo de funciones: paso por valores
  - b. Alcance de variables: variables locales y globales
  - c. Empleo de la técnica de recursividad
- 6. Estructuras de datos fundamentales 1**
  - a. Arreglos numéricos: vectores y matrices
  - b. Registros
- 7. Punteros**
  - a. Declaraciones
  - b. Operadores
  - c. Paso por referencia
  - d. Memoria dinámica
- 8. Archivos**
  - a. Métodos de acceso
  - b. Tipos de archivos: secuenciales, binarios
  - c. Directorios
  - d. Binarios y ASCII
- 9. Estructuras de datos fundamentales 2**
  - a. Estructuras y uniones
  - b. Listas enlazadas
  - c. Listas doblemente enlazadas

## **BIBLIOGRAFÍA**

1. Programación en C Metodología, Algoritmos y Estructura de Datos. Luis Joyanes Aguilar
2. The C Programming Language. Kernighan & Ritchie
3. Programación en C, Byron Gottfried, Ed: McGrawHill
4. Programación C, Jordi Bataller Mascarell/Rafael Magdalen Ed: Alfaomega
5. Deitel, P. C++ How to program (6<sup>th</sup> edición). Prentice Hall. 2007
6. Forouzan, B., Mosharraf, F., Foundations of Computer Science. Cengage Learning Business Press, 2008
7. Bird, R. J. Introduction to Functional Programming using Haskell. Prentice-Hall Series in Computer Science. Prentice-Hall Europe, London, UK, 2nd edition. 1998
8. Paz-Valderrama, A. Apuntes: Programación funcional en Haskell. Sociedad Peruana

de Computación. 2005

9. Thompson, S. The Craft of Functional Programming, 2E. Addison Wesley, 1999
10. Klemens, Ben. 21st Century C. O'reilly, 2nd edition. 2015
11. Gregoire, Mark. Professional C++. John Wiley & Sons, Inc, 3rd edition. 2014
12. Kalicharan, Noel. Learn to Program with C, Apress. 2015
13. Prinz, Peter & Crawford, Tony. C in Nutshell. O'reilly, 2nd edition. 2010

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: PROGRAMACIÓN ORIENTADA A OBJETOS

CÓDIGO: CC311

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC212 - FUNDAMENTOS DE PROGRAMACIÓN

CONDICIÓN : OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Para el estudiante de Ciencia de Computación, es importante conocer, saber y aplicar los paradigmas más importantes de la programación como son la programación estructurada y la programación orientada objetos, estar preparado y adaptarse a nuevos paradigmas.

En el curso de programación orientada a objetos está diseñado de manera que el final el estudiante haber desarrollado competencias que le permitirán saber, definir, aplicar los principios fundamentales del paradigma orientado a objetos que son las clases, objetos, relaciones entre objetos y clases, herencia, polimorfismo, clases abstractas, clases genéricas, colecciones, persistencia y las interfaces gráficas de usuarios para cual utilizará un lenguaje de programación orientado a objetos como herramienta para soluciones concretas.

### **COMPETENCIAS**

- Argumenta los fundamentos y técnicas orientadas a objetos para diseñar y modelar clases y objetos implementando constructores y destructores
- Argumenta y modela valorando las relaciones entre objetos: de uso, asociación, agregación y composición; y entre clases la herencia para diseñar formulando soluciones a problemas de la realidad
- Identifica y aplica la herencia entre clases, clases abstractas, interfaces para implementar clases polimórficas; también clases genéricas y las colecciones para dar solución a problemas valorando las técnicas orientadas a objetos
- Gestiona las condiciones de error que interrumpen el flujo normal de ejecución de un programa mediante la gestión de excepciones predefinidas y propias
- Explica los fundamentos e implementa aplicaciones que crean y manipulan la serialización de objetos, salvar y recuperar información
- Identifica los hilos valorando la programación concurrente. Describe la arquitectura de tres capas, la implementación en consola y visual. Conoce las Interfaces gráficas de usuario (GUI), valorando los componentes y organizadores para diseñar la interfaz

# PROGRAMA ANALÍTICO

## 1. Introducción a la programación orientada a objetos

- a. Introducción a la programación orientada a objetos
- b. Introducción a la programación estructurada y orientada a objetos
- c. Clases, objetos, diagramas de clases
- d. Constructores y destructores
- e. Objetos y mensajes
- f. Implementa clase valorando el paradigma anterior
- g. Resuelve e implementa programas utilizando una herramienta orientada a objetos

## 2. Relaciones entre clases y objetos

- a. Modela programas utilizando las relaciones entre objetos: de uso, asociación, agregación y composición
- b. Relaciones entre clases, valora la reutilización
- c. Modelamiento utilizando relaciones entre clases
- d. Implementación de estructuras de datos estáticas y dinámicas
- e. Implementación de soluciones con relaciones entre clases y objetos utilizando una herramienta orientada a objetos

## 3. Interfaces, polimorfismo y clases genéricas

- a. Interfaces y Clases Abstractas
- b. Clases polimórficas utilizando la herencia de Clases, Interfaces y Clases abstractas
- c. Clases y métodos genéricos
- d. Colecciones genéricas paramétricas
- e. Métodos de ordenamiento y de búsqueda genéricos
- f. Archivos directos y archivo de índices: Implementa soluciones utilizando herramientas orientadas a objetos

## 4. Excepciones y persistencia

- a. Excepciones en Java
- b. Excepciones predefinidas y Herencia de excepciones creadas por el usuario
- c. Archivos en Java
- d. Persistencia, serialización y deserialización de objetos
- e. Acceso a base de datos desde Java
- f. Implementación de programas valorando el manejo de excepciones
- g. Gestión de la información en memoria principal y secundaria

## 5. Hilos y GUIs:

- a. Programación concurrente, programación de hilos en Java
- b. Arquitectura de las tres capas: datos, lógica e interfaz
- c. Interfaces gráficas de Usuario (GUI) de java
- d. Ventanas, componentes y administradores de distribución de componentes
- e. Implementación de programas utilizando lenguaje de programación orientado a objetos

## 6. Estructuras dinámicas de datos:

- a. Asignación dinámica de memoria
- b. Listas enlazadas simples

## 7. Programación orientada a eventos

### BIBLIOGRAFÍA

1. Deitel, H. y Deitel, P., “Como programar en Java”, 5ta. Edición, Pearson Educación, 2004
2. Lafore R., “Object-oriented programming in C++”, 4ta. Edición, 2002, SAMS
3. Poo, K., Kiong, P., Ashok, S., “Object oriented programming and Java”, 2da. Edición, 2008, Springer
4. Preiss, B., “Data Structures and Algorithms with Object-Oriented Design Patterns in Java”, 1ra. Ed., 2001  
<http://www.brpreiss.com/books/opus5/html/book.html>Thompson, S. The Craft of Functional Programming, 2E. Addison Wesley, 1999

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ARQUITECTURA DE COMPUTADORAS

CÓDIGO: CC321

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC212 - FUNDAMENTOS DE PROGRAMACIÓN

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Este curso tiene por objetivo permitir al alumno gestionar adecuadamente el hardware y el software de un sistema de cómputo, que van desde la estructura básica del computador moderno hasta conocer los conceptos básicos del repertorio de instrucciones.

Garantizar el buen desempeño, la eficiencia de la futura codificación y conocer los fundamentos de la programación de bajo nivel serán tres temáticas fundamentales en el curso. Demostrar que la evolución de la arquitectura de los computadores es una consecuencia del avance y desarrollo tanto del hardware como del software.

### **COMPETENCIAS**

### **PROGRAMA ANALÍTICO**

#### **1. Computadores de 8 bits**

- Historia (breve) y Tecnología de los computadores
- Bajo los programas y bajo la cubierta
- El Microcontrolador 8051
- Arquitectura del 8051
- Conjunto de Instrucciones y Lenguaje ensamblador
- El TMC51
- Simulador EDSIM51 y programación del 8051
- Timers/Contadores
- Puerto Serie
- Interrupciones
- Display LCD

#### **2. Computadores de 16 y 32 bits**

- Introducción al MSP430 de Texas Instruments

- b. Introducción a MIPS
  - c. Instrucciones del computador
  - d. El procesador MIPS
  - e. Programación de MIPS
- 3. Organización de la Memoria y Arquitectura**
  - a. Codificación, compresión de datos e integridad de datos
  - b. Jerarquía de memoria
  - c. Memorias caché, (mapeo de direcciones, tamaño de bloques, políticas de reemplazo y almacenamiento)
  - d. Memoria Virtual (tablas de paginación, TLB)
- 4. Comunicación e Interfase**
  - a. Fundamentos de entrada y salida: buffering, handshaking, entradas y salidas, entradas y salidas manejadas por interrupciones
  - b. Manejo de interrupción
  - c. Almacenamiento externo, organización física y drivers
  - d. Buses: protocolos de buses, arbitraje, acceso directo a memoria (DMA)
  - e. Arquitecturas RAID
- 5. Multinúcleos, multiprocesadores y clústeres**
  - a. Introducción a SIMD, MIMD, VLIW, EPIC
  - b. Arquitectura sistólica
  - c. Sistemas de memoria compartida
  - d. GPUs para gráficos y cálculos
- 6. Herramientas de Diagnóstico y seguridad**
  - a. Evaluación de los sistemas de cómputo
  - b. Diagnóstico de los sistemas de procesamiento
  - c. Seguridad

## **BIBLIOGRAFÍA**

1. Brey, B. B. The Intel Microprocessors: 8086/8088, 80186, 80286, 80386, 80486, Pentium, Pentium Pro, and Pentium II, Pentium III, Pentium 4, 7th edition. Prentice-Hall. 2005
2. Mano, M. M. Computer System Architecture, 3rd edition. Prentice Hall. 1992
3. D. Tullsen, S. Eggers and H. Levy. Simultaneous Multithreading: Maximizing On-Chip Parallelism, in the Proceedings of the 22rd Annual International Symposium on Computer Architecture, June 1995
4. Carpinelli, J.D. Computer Systems Organization & Architecture. Addison Wesley. 2001
5. Joseph D. Dumas II, Joseph D. Dumas. Computer architecture: fundamentals and principles of computer design. Publicado por CRC Press. 2006
6. John L. Hennessy, David A. Patterson, Andrea C. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau. Computer architecture: a quantitative approach. Publicado por Morgan Kaufmann. 2007
7. John L. Hennessy, David A. Patterson.. Estructura y Diseño de Computadores: La interfaz Hardware/Software. 4th Edition. Editorial Reverté, S.A., 2011



UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: BASES DE DATOS

CÓDIGO: CC412

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC311 - PROGRAMACIÓN ORIENTADA A OBJETOS

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Las bases de datos y los sistemas de bases de datos son un componente básico en la vida cotidiana de la sociedad moderna, la interacción con la administración, los servicios públicos, la empresa ya casi no se puede entender sin la gestión de una base de datos que permita la comunicación entre los distintos actores.

Los sistemas de gestión de bases de datos son por tanto un componente fundamental en las tecnologías de la información y la comunicación sin las que sería imposible imaginar el alcance social que la red y las comunicaciones han alcanzado en la sociedad actual.

### **COMPETENCIAS**

- Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos
- Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los sistemas de información, incluidos los basados en web
- Conocer en profundidad el Modelo Relacional de Datos, el proceso de diseño relacional a través de dependencias funcionales y normalización así como los lenguajes relacionales Cálculo y Álgebra Relacional
- Diseñar una Bases de Datos basada en las fases de diseño conceptual y lógico a partir del modelo conceptual Entidad Relación Extendido

### **PROGRAMA ANALÍTICO**

#### **1. Introducción**

- a. ¿Qué es una base de datos?
  - b. Qué es un Sistema de gestión de base de datos
  - c. Independencia de los datos
  - d. Arquitectura de un sistema de base de datos
- 2. Modelo de Entidad relación (E/R)**
- a. Introducción al modelo de datos ER
  - b. Conjuntos de entidades y de relaciones
  - c. Dominios
  - d. Representaciones equivalentes de una relación
  - e. Cuestiones de diseño
  - f. Especialización
  - g. Agregación
  - h. Reducción a tablas
- 3. El modelo relacional: Conversión de E/R**
- a. Esquemas, Tuplas, Tablas, Dominios
  - b. Conversión a tablas desde un modelo con relaciones
  - c. Conversión a tablas desde un modelo con generalización
  - d. Descubrimiento de claves en las relaciones
- 4. Dependencias Funcionales**
- a. Definición
  - b. Axiomas de Armstrong
  - c. Reglas Adicionales
  - d. Clave de un conjunto de atributos
  - e. No redundancia
  - f. Determinación de las claves de un esquema y cálculo de las claves de un esquema Relacional
- 5. Normalización: Dependencias de múltiple valores**
- a. Redundancia, Anomalías de actualización y Eliminación
  - b. Primera y Segunda forma Normal
  - c. Descomposición sin Pérdida
  - d. Preservación de dependencias
  - e. Forma Normal de Boyce Codd (BCNF)
  - f. Tercera forma Normal, BCNF vs 3NF
  - g. Cuarta y Quinta forma Normal
- 6. Introducción a SQL**
- a. Aplicaciones de las bases de datos
  - b. Sistemas de bases de datos frente a sistemas de archivos
  - c. Visión de los datos
  - d. Modelos de bases de datos
  - e. Lenguajes de bases de datos
  - f. Gestión de transacciones
  - g. Estructura de un sistema de bases de datos
  - h. Usuarios de bases de datos
- 7. Agregaciones, modificaciones, y operaciones avanzadas**
- a. Agregaciones y diseño de agregaciones
  - b. Modificación de datos
  - c. Clasificación

- d. Insertar tuplas en una relación
- e. Eliminar tuplas de una relación
- f. Actualizar el valor de algunos componentes de tuplas existentes

## **8. Vistas y definición de datos**

- a. Concepto de vista
- b. Aplicaciones de vistas
- c. Vistas en SQL
- d. Sintaxis
- e. Motivos por que una lista no es actualizable
- f. Vista sobre una tabla Básica
- g. Vista sobre una concatenación de relaciones
- h. Definir tablas, filas y columnas
- i. Insertar claves de índice
- j. Crear relaciones entre tablas
- k. Asignar tipos de datos

## **9. Restricciones y disparadores (triggers)**

- a. Restricciones en procedimientos almacenados y disparadores
- b. Restricciones en subconsultas
- c. Restricciones en vistas
- d. Definición, uso y sintaxis del trigger
- e. Componentes y nombres de disparadores
- f. Tipos de disparadores
- g. Orden de activación de disparadores

## **10. Módulos de almacenamiento persistentes**

- a. Almacenamiento y estructura de archivos
- b. Indexación, asociación, hashing
- c. Índices
- d. Procesamiento de consultas
- e. Optimización de consultas

## **11. Autorización**

- a. Violaciones de la seguridad
- b. Control de acceso a la base de datos
- c. Tipos de autorización
- d. Autorizaciones y vistas
- e. Concesión de privilegios
- f. Eliminación de privilegios
- g. El concepto de rol o papel
- h. Limitaciones de la autorización SQL

## **12. Transacciones**

- a. Sentencias para una transacción
- b. Transacciones anidadas
- c. Transacciones y procedimientos almacenados

## **BIBLIOGRAFÍA**

1. Hector García-Molina, Jeff Ullman, y Jennifer Widom. Database Systems: The Complete Book (DS:CB). Editorial: Upper Saddle River, N.J. : Pearson Prentice Hall, 2009

2. C. J. Date. Introducción a los Sistemas de Base de Datos. Pearson Prentice Hall, 2001
3. Enrique Rivero Cornelio, Carlos Guardia Rivas, José Carlos Reig Hernández. Bases de datos relacionales: diseño físico (Orientado al DB2 para z/OS de IBM). Editor Universidad Pontificia de Comillas de Madrid, 2004
4. Abiteboul, S; Hull and Vianu, V. Foundations of Databases. Addison-Wesley Publishing Company, 1995
5. Ullman, Jeffrey D. Principles of Database and Knowledge Base Systems, Vol I Computers Science Press, 1988
6. Mendelzon, A. Introducción a las Bases de Datos Relacionales. Ed. Pearson, 2000.
7. Silberschatz, A.; Korth, H.F.; Sudarshan, S. Fundamentos de Bases de Datos. 3ra edición. Madrid, McGraw-Hill, 1998
8. Inmon, W.H.. Building the Data Warehouse. John Wiley, 2002
9. Elmasri, R.; Navathe, S.B. Fundamentals of Database Systems 3rd Edition, Addison-Wesley, 2000
10. Berry, M. Data mining techniques. 1997
11. Bray, T.; Hollander, D.; Layman, A. Name-spaces in XML. World Wide web Consortium. W3C Recommendation, 1999
12. Urman, S. ORACLE 8. Programación en PL/SQL. Osborne McGraw-Hill, Madrid, 1998

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: SISTEMAS OPERATIVOS

CÓDIGO: CC422

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC321 - ARQUITECTURA DE COMPUTADORES

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

El curso se orienta a conocer los fundamentos de la operación de los Programas de Aplicación y de la forma en la que los Sistemas Operativos implementan abstracciones del hardware y llevan a cabo la administración de los recursos del sistema. Por otra parte, se llevarán a cabo sesiones de Laboratorio en la que se presentará la aplicación práctica de las técnicas estudiadas.

### **COMPETENCIAS**

- Comprende los objetivos y funciones de los Sistemas Operativos, su evolución y desarrollo hacia sistemas modernos
- Comprende el funcionamiento interno de un Sistema Operativo, sus procesos y mecanismos de control
- Conoce y comprende el concepto de concurrencia y la gestión de la seguridad en un Sistema Operativo
- Comprender los algoritmos utilizados en la administración de memoria
- Comprende el proceso de administración y optimización del sistema de archivos
- Comprende el modo en que el sistema operativo controla los dispositivos de E/S
- Conoce y comprende los conceptos fundamentales del diseño de un Sistema Operativo

### **PROGRAMA ANALÍTICO**

1. **Funciones del Sistema Operativo**
  - a. Componentes del Sistema Operativo
  - b. Diagrama de casos de uso
  - c. Funciones y componentes como casos de uso
2. **Descripción de procesos**

- a. Ejecución de un proceso
- b. Ejemplo de ejecución Estados de un proceso
- c. Cómo se crean los procesos
- d. Cómo terminan los procesos
- e. El estado bloqueado
- f. El estado suspendido
- g. Tablas de control del sistema operativo
- h. Tabla de procesos
- i. Bloque de control de procesos BCP
- j. Identificación del proceso
- k. Identificación del contexto del CPU
- l. Información de control del proceso
- m. El proceso en el kernel
- n. Modos de ejecución
- o. Creación de procesos
- p. Intercambio de procesos
- q. Cambio en el estado del proceso

### **3. Planificación de procesos**

- a. Cola de procesos
- b. Reloj de tiempo real
- c. Tipos de planificación
- d. Planificación a largo plazo
- e. Planificación a mediano plazo
- f. Planificación a corto plazo
- g. Tipos de procesos
- h. Procesos en tiempo real
- i. Procesos Normales
- j. Procesos no expropiativos
- k. Procesos expropiativos
- l. Políticas de planificación: SCHED\_FIFO, SCHED\_RR y SCHED\_OTHER
- m. Algoritmo de planificación en Linux

### **4. Concurrencia de procesos**

- a. Principios Generales
- b. Exclusión mutua
- c. Sección crítica
- d. Algoritmo de Peterson
- e. Exclusión mutua por Hardware
- f. Por inhabilitación de interrupciones
- g. Por instrucciones en lenguaje de máquina
- h. Semáforos
- i. El estado Espera
- j. Algoritmo para un semáforo binario
- k. Exclusión mutua con semáforos
- l. Ejemplo de semáforo binario
- m. Mensajes
- n. Direccionamiento
- o. Sincronización

- p. Partes de un mensaje
- q. Exclusión Mutua
- 5. Interbloqueo de procesos
  - a. Principios Generales
  - b. Procesos con interbloqueo, procesos sin interbloqueo
  - c. Recursos reutilizables
  - d. Recursos consumibles
  - e. Condiciones para el interbloqueo
  - f. Prevención del interbloqueo
  - g. Exclusión mutua
  - h. Retener y Esperar
  - i. Expropiación
  - j. Círculo de espera
  - k. Predicción del interbloqueo
  - l. Estado seguro
  - m. Estado inseguro
  - n. Detección del interbloqueo
  - o. Recuperación luego del interbloqueo
  - p. Estrategias integradas para evitar el interbloqueo
- 6. **Gestión de Memoria**
  - a. Requisitos de la gestión de memoria
  - b. Reubicación
  - c. Protección
  - d. Compartimiento
  - e. Organización Lógica
  - f. Organización física
  - g. Partición de la memoria
  - h. Particiones estáticas de igual tamaño
  - i. Particiones estáticas de diferente tamaño
  - j. Particiones dinámicas
  - k. Algoritmo de Ubicación
  - l. Sistema de Bloques 2n (Buddy System)
  - m. Reubicación de procesos en la memoria
  - n. Tipos de Direcciones
  - o. Registros usados durante la ejecución
  - p. Paginación
  - q. Asignamiento de marcos libres a páginas de un proceso
  - r. Cálculo de la dirección física de memoria
  - s. Segmentación
  - t. Cálculo de la dirección física de memoria
  - u. Direccionamiento lógico
- 7. **Gestión de Memoria Virtual**
  - a. Estructuras de hardware y control
  - b. ¿Qué ocurre al ejecutar un programa?
  - c. Ventajas de la fragmentación de procesos
  - d. Tipos de memoria
  - e. Hiperpaginación

- f. Principio de Cercanía
- g. Soporte para el uso de memoria virtual
- h. Paginación
- i. La tabla de páginas
- j. Buffer de traducción adelantada
- k. El problema del tamaño de página
- l. Segmentación
- m. Tablas de segmentos
- n. Política de Carga
- o. Política de Reemplazo
- p. Algoritmos de reemplazo
- q. Asignación de páginas para un proceso
- r. Asignación variable
- s. Política de limpieza
- t. Control de carga
- u. Suspensión de procesos

#### **8. Gestión de E/S**

- a. Dispositivos de Entrada/Salida
- b. Tipos de dispositivos E/S
- c. Diferencias entre dispositivos E/S
- d. Organización de las funciones E/S
- e. Evolución de las funciones E/S
- f. Técnicas para ejecutar la E/S
- g. Memoria de Acceso Directo (DMA)
- h. Aspectos del diseño del Sistema Operativo
- i. Características para diseñar un sistema operativo

#### **9. Gestión de Almacenamiento**

- a. Almacenamiento intermedio de la E/S
- b. Buffering de E/S
- c. Buffer simple
- d. Buffer Doble
- e. Buffer Circular
- f. Planificación de Discos
- g. Performance del Disco Duro
- h. Raid, Raid 0, Raid 1, Raid 2, Raid 3, Raid 4, Raid 5 y Raid 6
- i. Caché de disco
- j. Usado menos recientemente
- k. Usado menos frecuentemente

#### **10. Gestión de Archivos**

- a. Conceptos iniciales
- b. Términos más usados
- c. Operaciones típicas con archivos
- d. Objetivos de un sistema de manejo de archivos
- e. Conjunto de requerimientos mínimos
- f. Drivers de dispositivos (Disco y Tape)
- g. Sistema Básico de Archivos
- h. Supervisor Básico de E/S



- i. E/S Lógico
- j. Métodos de acceso
- k. Funciones de la gestión de archivos
- l. Organización y acceso de archivos
- m. Criterios para la Organización de archivos

#### 11. Gestión de Directorios

- a. Organización de directorios
- b. Directorio de archivos
- c. Estructura simple para un directorio
- d. Estructura de dos niveles de archivos
- e. Estructura jerárquica de archivos
- f. Compartiendo archivos
- g. Privilegios de acceso a archivos
- h. Acceso simultáneo
- i. Agrupación de registros
- j. Bloques y Registros
- k. Bloques fijos
- l. Bloques variables
- m. Gestión del almacenamiento secundario
- n. Preasignación
- o. Métodos de ubicación de archivos
- p. Sistema de archivos Linux y Windows: Ext2 y NTFS

## BIBLIOGRAFÍA

1. Mateu, L. Apuntes de Sistemas Operativos. Universidad de Chile. 1999
2. Stallings, W. Operating Systems: Internals and Design Principles, 5/Ed. Prentice Hall. 2005
3. Tanenbaum, A. S. Modern Operating Systems, 2/Ed. Prentice Hall. 2001
4. Tanenbaum, A. S. Operating Systems Design and Implementation, 3/Ed. Prentice Hall. 2006
5. Wale Soyinka. Linux Administration: A Beginner's Guide, 5/Ed. McGraw Hill
6. Evi Nemeth & Garth Snyder & Trent R. Hein. Linux Administration Handbook, 1/Ed. Prentice Hall
7. Ellen Siever, Stephen Figgins, Aaron Weber. Linux in a Nutshell, 4th Ed., A Desktop Quick Reference. O'Reilly Media
8. Steven Pritchard, Bruno Gomes Pessanha, Nicolai Langfeldt, James Stanger, Jeff Dean, et al. LPI Linux Certification in a Nutshell, 2nd Ed. O'Reilly Media

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ALGORITMOS Y ESTRUCTURA DE DATOS

CÓDIGO: CC432

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC311 - PROGRAMACIÓN ORIENTADA A OBJETOS
- CM2H1 - MATEMÁTICA DISCRETA

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Este curso es una continuación del curso de programación orientada a objetos pero centrado en la programación de estructuras de datos simples y complejas. Por tanto, los aspectos que se estudiarán serán principalmente cómo se definen y cómo se implementan las estructuras de datos e incorporarlas a cualquier entorno de programación.

Así mismo, como puede observarse, este curso continúa con la formación de un programador en cualquier ámbito o especialidad, continuando posteriormente en el curso de “Análisis y diseño de algoritmos”, donde se verá algoritmia y optimización en estructuras de datos complejas.

### **COMPETENCIAS**

- Evalúa las propiedades de las estructuras de datos estableciendo su utilidad en las aplicaciones en ciencia e ingeniería comprometiéndose en el uso adecuado que conlleve a resolver problemas de la vida real
- Demuestra su capacidad de análisis diseñando, implementando estructuras que permitan dar solución, trabajando en equipo en los diversos problemas en la ciencia e ingeniería
- Comprende los conocimientos básicos de Algorítmica y ejecuta los principios de esta ciencia de la computación
- Describe los procesos de construcción de Algoritmos identificando las propiedades asociadas que se generan de ellos apreciando su influencia en la ciencia e ingeniería

### **PROGRAMA ANALÍTICO**

1. Introducción
  - a. Definición de estructura de datos
  - b. Clasificación de estructura de datos

- c. Operaciones sobre Estructura de Datos
  - d. Estructuras de datos y eficiencia
  - e. Algoritmos para encontrar máximos y mínimos
  - f. Algoritmos para realizar búsquedas secuenciales y binarias
  - g. Algoritmos de ordenamiento, de peor caso cuadrático (selección, inserción)
  - h. Algoritmos de ordenamiento caso promedio (Quicksort, Heapsort, Mergesort)
2. Descripción formal de TAD
- a. Conjuntos
  - b. Implementación de un TAD
  - c. Listas enlazadas simples
  - d. Noción de lista enlazada
  - e. Definición formal de lista enlazada
  - f. Características
  - g. Representación de listas enlazadas mediante arreglos, mediante cursores y mediante apuntadores
  - h. Listas doblemente enlazadas y listas circulares
  - i. Definición formal de lista doblemente enlazada y listas circulares
  - j. Características
  - k. Representación de listas enlazadas
  - l. Pilas y colas. Noción de pila
  - m. Definición formal de pila
  - n. Características
  - o. Acciones primitivas
  - p. Representación de pilas
  - q. Operaciones básicas
  - r. Aplicación y ejemplos
  - s. Noción de cola
  - t. Definición formal de cola
  - u. Características
  - v. Acciones Primitivas
  - w. Representación de colas
  - x. Tipos de colas
3. Árboles
- a. Noción de árbol
  - b. Definición de árbol
  - c. Características
  - d. Acciones primitivas
  - e. Representación de Árboles mediante arreglos y mediante apuntadores
  - f. Tipos de árboles
  - g. Árboles binarios
  - h. Definición formal de árbol binario características
  - i. Acciones primitivas
  - j. Representación de árboles binarios
  - k. Ordenamiento, Búsqueda, Inserción y Eliminación
4. Grafos
- a. Noción de grafo
  - b. Tipos: Grafos no dirigidos, Grafos dirigidos

- c. Definición Formal de grafos no dirigidos
- d. Representación
- e. Operaciones básicas
- f. Aplicaciones y ejemplos
- g. Grafos
- h. Recorrido de grafos
- i. Matriz de adyacencia
- j. Matriz de caminos
- k. Caminos mínimos
- l. Definición Formal de Grafos dirigidos
- m. Búsqueda en Profundidad
- n. Árboles de recubrimiento
- o. Camino Hamiltoniano
- p. Paseo Euleriano
- q. Algoritmo de Dijkstra
- r. Algoritmo de Warshall
- s. Grafos acíclicos
- t. Ordenamiento Topológico

## **BIBLIOGRAFÍA**

1. Weiss, M.A.: Data Structures and Algorithm Analysis in C++, 4th Edition, Pearson/Addison Wesley, 2014
2. Hernández, Z.J. y otros: Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++, Thomson, 2005
3. Shaffer, Clifford A.: Data Structures and Algorithm Analysis in C++, Third Edition, Dover Publications, 2013
4. Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J.A.: Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos. 2ª Edición, Ed. Garceta, 2013
5. Joyanes, L., Zahonero, I., Fernández, M. y Sánchez, L.: Estructura de datos. Libro de problemas, McGraw Hill, 1999
6. Campos Laclaustra, J.: Estructuras de Datos y Algoritmos, Prensas Universitarias de Zaragoza, Colección Textos Docentes, 1995
7. Franch Gutiérrez, X.: Estructuras de Datos. Especificación, Diseño e Implementación, 3ª edición, Ed. Edicions UPC, 2001
8. Mehta, D.P. y Sahni, S.: Handbook of Data Structures and Applications, Chapman & Hall/CRC, 2005

## SÍLABO

### INFORMACIÓN GENERAL

ASIGNATURA: COMPUTACIÓN CENTRADA EN REDES

CÓDIGO: CC511

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC422 - SISTEMAS OPERATIVOS

CONDICIÓN : OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### SUMILLA

Introduce en la estructura, implementación, y fundamentos teóricos de la red de computadoras y las aplicaciones que se han habilitado por esta tecnología. Por tanto, esta asignatura proporciona al estudiante las competencias, conocimientos y habilidades básicas para comprender la problemática de la comunicación de datos y la interconexión de redes (en especial Internet). En este contexto, el paradigma de la comunicación interpersonal será la temática fundamental a desarrollar, teniendo un principal enfoque a las comunicaciones digitales e infraestructuras tecnológicas de las tecnologías de comunicación e información.

### COMPETENCIAS

- Adquirir los conocimientos y desarrollar las aptitudes necesarias en relación al funcionamiento y programación de redes de computadoras y de los mecanismos presentes en su funcionamiento y explotación
- Conocer los conceptos de red de computadores, protocolo, direccionamiento, enrutamiento y demás terminología habitual en este campo
- Conocer la problemática de la interconexión de redes
- Comprender la arquitectura de red y los modelos de referencia más conocidos
- Conocer los servicios básicos de red: DNS, NAT, VPN
- Aplicar los conocimientos en situaciones prácticas, principalmente con la formulación de estrategias para problemas reales

### PROGRAMA ANALÍTICO

1. Introducción
  - a. Uso de las redes de computadoras
  - b. Hardware de Red (PAN, LAN, MAN, WAN)
  - c. Software de Red
  - d. Modelos de Referencia (OSI, TCP/IP)
  - e. Redes de ejemplo (Redes LAN inalámbricas, Redes RFID y de sensores)

- f. Estandarización de Redes
  - g. Una breve historia de las redes
  - h. Calidad de servicio
  - i. Redes TCP/IP y Redes ATM
2. Capa Física
- a. Medios de transmisión guiados (Medios magnéticos, par trenzado, cable coaxial, líneas eléctricas, fibra óptica)
  - b. Transmisión inalámbrica (microondas, infrarroja, ondas de luz)
  - c. Satélites de comunicación
  - d. Modulación digital y multiplexión (transmisión en banda base)
  - e. La Red Telefónica Pública Conmutada (ADSL y Fibra óptica)
  - f. Sistema de telefonía móvil (1G, 2G, 3G y 4G)
  - g. Protocolos y la arquitectura TCP/IP
  - h. El modelo OSI
  - i. Interconexión de redes
3. Capa de Enlace de Datos: Diseño
- a. Detección y Corrección de Errores
  - b. Protocolos elementales de Enlace de Datos
  - c. Protocolos de Ventanas Deslizantes Redes tolerantes
  - d.
  - e. TCP e IP
  - f. El protocolo de control de transmisión (TCP)
  - g. Ipv4 e Ipv6
4. Subcapa de Control de Acceso al Medio: Protocolo de Acceso Múltiple
- a. Ethernet (Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet, Ethernet Conmutada)
  - b. Redes LAN Inalámbricas (802/11 y la pila de protocolos)
  - c. Banda Ancha Inalámbrica (Capa física estándar 802/16)
  - d. Bluetooth (Arquitectura)
  - e. RFID (Formatos de los mensajes de identificación de etiquetas)
  - f. Conmutación de la Capa de Enlace de Datos (Usos de los puentes, repetidores, hubs, enrutadores, y puertas de enlaces o gateways, Redes LAN virtuales)
  - g. Redes y Arquitectura de Frame Relay
5. La Capa de Red: Aspectos del Diseño de la Capa de Red
- a. Algoritmos de Enrutamiento (Enrutamiento por vector distancia, enrutamiento jerárquico, enrutamiento multidifusión, enrutamiento por la ruta más corta, anycast)
  - b. Algoritmos de Control de Congestión (Control de admisión)
  - c. Calidad de Servicio
  - d. Interconexión de Redes
  - e. La Capa de Internet (Protocolo Ip, Ip versión 4, Ip versión 6, Protocolos de control en Internet, multidifusión en Internet, Ip móvil)
  - f. Modo de transferencia asíncrono (ATM)
  - g. Conexiones lógicas, celdas ATM
6. La Capa de Transporte: El servicio del transporte
- a. Elementos de los Protocolos de Transporte (Direccionamiento, establecimiento de una conexión, multiplexación, control de errores)

- b. Control de Congestión
  - c. Los Protocolos de Transporte en Internet (Introducción a UDP, llamada de procedimiento remoto, protocolos en tiempo real)
  - d. Los Protocolos de Transporte de Internet: TCP (Introducción, modelo de servicio, protocolo TCP, encabezado del segmento, establecimiento de una conexión TCP, control de congestión TCP)
  - e. Aspectos del Desempeño (Diseño de los hosts para redes rápidas)
  - f. Redes tolerantes al retardo: TCP, FTP y Telnet
7. La Capa de Aplicación: DNS
- a. Sistema de nombres de dominio (el espacio de los nombres del DNS, registros de los recursos de dominio, servidores de nombres)
  - b. Correo electrónico
  - c. WWW (web estáticas y dinámicas, HTTP, web móvil)
  - d. Audio y Video de Flujo Continuo
  - e. Entrega de Contenido (Contenido y tráfico en Internet, redes de igual a igual)
  - f. Gestión de la congestión y del tráfico
  - g. Control y gestión del tráfico
  - h. Control de la congestión en Frame Relay
8. Introducción a la Seguridad de Redes: Criptografía
- a. Algoritmo de Clave Simétrica
  - b. Algoritmos de Clave Pública (RSA y otros algoritmos de clave pública)
  - c. Firmas Digitales
  - d. Administración de Claves Públicas (Certificados)
  - e. Seguridad en la Comunicación (Ipsec, Firewalls, Redes privadas virtuales, seguridades inalámbricas)
  - f. Protocolos en la Autenticación
  - g. Seguridad de Correo Electrónico
  - h. Seguridad en Web

## **BIBLIOGRAFÍA**

1. ARIGANELLO, ERNESTO - "Redes Cisco". Ra-Ma
2. TANENBAUM/WETHERALL - "Redes de Computadoras". Pearson Quinta edición, 2012
3. STALLINGS, W. - "Redes e Internet de Alta Velocidad. Rendimiento y Calidad de Servicio"; 2a edición
4. STALLINGS, W. - "Comunicaciones y redes de computadores"; 6a/7a edición, Prentice Hall, 2000/2004. Prentice-Hall, 2004
5. HALSALL, F. - "Comunicación de datos, redes de computadores y sistemas abiertos"; 4a edición, Addison- Wesley Longman, 1998.
6. TANENBAUM, A. - "Computer Networks"; 4th edition, Prentice-Hall, 2003
7. FOROUZAN, B.A. - "Transmisión de datos y redes de comunicaciones"; 2a edición, McGraw-Hill, 2001
8. Cisco Systems. - Inc. Guía del primer año. CCNA 1 y 2. Cisco Press, 2003

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: TEORÍA DE AUTÓMATAS, LENGUAJES Y COMPUTACIÓN

CÓDIGO: CC521

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC432 - ESTRUCTURAS DE DATOS

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

En esta materia, donde se fundamenta y se establece la base teórica de los estudios, se encuentran los conceptos que se estudian y se usan en casi todas las demás materias. Qué es un lenguaje de programación, cómo y porqué se define de la forma en que se hace. Cuál es el mecanismo en el que se traduce a un formalismo entendible por una máquina, y porqué es precisamente ese. Y cuáles son las limitaciones que se pueden encontrar, entre muchos otros. En este sentido, este curso demuestra que la Computación es una ciencia, en particular una rama de la matemática que centra su interés en el estudio y definición formal de los computadores.

### **COMPETENCIAS**

- Entender y tener destreza en la definición de lenguajes regulares y libres de contexto así como de las máquinas que lo reconocen
- Entender los fundamentos teóricos de la computabilidad y decidibilidad

### **PROGRAMA ANALÍTICO**

1. **Introducción a la TC**
  - a. Conjuntos en TC
  - b. Relaciones y Funciones en TC
  - c. Teoría de Grafos
  - d. Semigrupos y Grupos
2. **Cadenas y Lenguajes**
  - a. Símbolo
  - b. Alfabeto
  - c. Cadena
  - d. Operaciones con cadenas
  - e. Lenguajes



- f. Operaciones con Lenguajes
- g. Técnicas Básicas de Demostración
- 3. **Lenguajes Regulares**
  - a. Definición recursiva de LR
  - b. Expresiones Regulares(ER)
  - c. Equivalencia de ER
  - d. Propiedades de ER
  - e. Derivadas
  - f. Sistemas de Ecuaciones
- 4. **Máquinas de Estado Finito**
  - a. Definición MEF
  - b. Representación Gráfica
  - c. Sumador Binario
  - d. Clasificación de MEF: Moore y Mealy
- 5. **Autómatas de Estados Finitos**
  - a. Autómatas Finitos Determinísticos (AFD)
  - b. Minimización de AFD
  - c. AFD Equivalentes
  - d. Autómatas Finitos no Determinísticos (AFND)
  - e. Representación de AFND
  - f. Equivalencia entre AFD y AFND
  - g. Autómatas con transiciones épsilon: representación, función de transición extendida
- 6. **Gramáticas**
  - a. Regla
  - b. Derivaciones
  - c. Definición de Gramática Formal
  - d. Gramáticas Regulares
  - e. Conversión de GR a AFD
  - f. Conversión de GR a AFND
- 7. **Lenguajes libres de contexto**
  - a. Gramáticas Libres de Contexto
  - b. Árboles de Derivación
  - c. Simplificación de Gramáticas
  - d. Eliminación de Símbolos Inútiles
  - e. Eliminación de Producciones épsilon
  - f. Eliminación de Producciones Unitarias
  - g. Forma Normal de Chomsky
  - h. Eliminación de Factores Comunes izquierdos
  - i. Eliminación Recursividad Izquierda
  - j. Eliminación de ambigüedad
  - k. Forma Normal de Greibach
  - l. Autómata de Pila Determinístico
  - m. Autómata de Pila No Determinístico
- 8. **Máquina de Turing**
  - a. Definición formal de máquina de Turing
  - b. Restricciones a la máquina de Turing

- c. Construcción Modular máquina de Turing

## 9. **Decibilidad**

- a. Lenguajes Decibles
- b. Los problemas de Halting
- c. Teorías Lógicas de decibilidad

## 10. **Reducibilidad**

- a. Problemas Insolubles en la teoría de lenguajes
- b. Un problema simple que es insoluble
- c. Funciones Computables
- d. Reducibilidad de Turing

## **BIBLIOGRAFÍA**

1. Brookshear J. Teoría de la Computación. Addison Wesley. 1993
2. Dexter C. Kozen. "Automata and computability". Springer-Verlag. 1997
3. Efim Kinber, Carl Smith. "Theory of computing: a gentle Introduction". Prentice-Hall. 2001
4. Harry R. Lewis, Christos H. Papadimitriou. "Elements of the theory of computation". Prentice-Hall. 1998
5. Hopcroft, J.E.; Motwani, R. i Ullman, J.D. "Introduction to Automata Theory, Languages and Computation". 2da ed. Addison-Wesley, 2001
6. J. Glenn Brookshear. "Teoría de la Computación: lenguajes formales, autómatas y complejidad". Addison-Wesley Iberoamericana. 1993.
7. Joaquim Gabarró Valles. "Informatica classica: autómatas, gramatiques, indecidibilidad, parallelismo masivo". 1995
8. Jozef Gruska. "Foundations of computing". International Thomson Computer Press. 1997.
9. Sipser. Introduction to the Theory of Computation. PWS. 1997

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ANÁLISIS Y DISEÑO DE ALGORITMOS

CÓDIGO: CC531

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC432 - ALGORITMOS Y ESTRUCTURAS DE DATOS

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Este curso es una continuación de Estructuras de datos y, que por tanto, desarrollará como optimizar dichas estructuras y mejorar la capacidad de abstracción. Introducir y desarrollar estructuras y algoritmos. El curso también ofrecerá una introducción al contexto histórico y social de la informática y una revisión del ámbito de esta disciplina.

Por tanto, en esta asignatura se abordan aspectos relativos a la resolución de problemas mediante técnicas fundamentales de computación, tanto exactas como aproximadas. En particular, se profundiza en aspectos relativos a complejidad computacional y a técnicas algorítmicas.

### **COMPETENCIAS**

- Implementar programas utilizando las estrategias de programación fuerza bruta, algoritmos voraces, divide y vencerás, backtracking, utilizando el lenguaje C++, analizando su complejidad algorítmica
- Implementar estructuras de datos avanzadas, como árboles y grafos analizando la complejidad algorítmica y presentando su proyecto final con aplicación de las estrategias adecuadas

### **PROGRAMA ANALÍTICO**

1. **Análisis de la eficiencia de los algoritmos**
  - a. Complejidad de los algoritmos
  - b. Notaciones asintóticas
  - c. Análisis de Tiempos de Ejecución para cada caso
  - d. Ecuaciones de Recurrencia
  - e. Medir eficiencia a través de metaheurísticas
2. **Introducción a la optimización simple**

- a. Análisis de algoritmos iterativos y recursivos
  - b. Análisis de algoritmos de ordenación y búsqueda
  - c. Dividir y Conquistar
  - d. Análisis Probabilístico y Algoritmos aleatorios
3. **Clasificación y estadísticas de orden**
- a. Heapsort
  - b. Quicksort
  - c. Ordenamiento en tiempo lineal
  - d. Medianas y estadísticas de orden
4. **Diseño Avanzado y Técnicas de Análisis**
- a. Programación dinámica
  - b. Algoritmos golosos
  - c. Análisis amortizado
5. **Algoritmos de grafos**
- a. Algoritmos de grafo elementales
  - b. Árboles de expansión de mínimos
  - c. Caminos más cortos de un solo proveedor
  - d. Todos los pares de rutas más cortas
  - e. Flujo máximo

## **BIBLIOGRAFÍA**

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. Introduction to algorithms. The MIT Press, 2009
2. R. Peña. Diseño de programas: Formalismo y abstracción. Tercera edición, Pearson/Prentice Hall 2005
3. Steven R. Skiena. The Algorithm Design. Ed.: Springer
4. G. Brassard, P. Bradley. Fundamentos de algoritmia. Prentice Hall, 1997
5. R. Neapolitan, K. Naimipour. Foundations of algorithms, 3a edición. Jones and Bartlett Publishers, 2003
6. Donald E. Knuth. The Art of Computer Programming
7. M. Rodríguez Artalejo, P. A. González Calero, M. A. Gómez Martín. Estructuras de datos: un enfoque moderno. Editorial Complutense 2011
8. N. Martí Oliet, Y. Ortega Mallén, J. A. Verdejo López. Estructuras de datos y métodos algorítmicos: ejercicios resueltos. Colección Prentice Práctica, Pearson/Prentice Hall 2003

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: INGENIERÍA DE SOFTWARE

CÓDIGO: CC541

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC412 - BASES DE DATOS

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

En este curso el alumno comienza una etapa profesional de desarrollador a través del conocimiento de los procesos software que se presentan en el ciclo de vida del software. Presentar a los alumnos los diferentes modelos de evaluación de procesos y las métricas del proceso de software, además de identificar los requerimientos funcionales y no funcionales de la construcción o de un software en el modelamiento de análisis de requerimientos y prototipos de un software.

En este sentido los alumnos deben ser capaces de seleccionar y aplicar patrones de diseño apropiados en la construcción de una aplicación de software y aplicar el diseño de componentes y el diseño de reuso en las aplicaciones presentadas por los alumnos.

### **COMPETENCIAS**

- Identificar, modelar y especificar requisitos software y de negocio, para la construcción de sistemas software que los implementen
- Construir los modelos de diseño, tanto de alto nivel como detallados, para la construcción de sistemas software
- Conocer herramientas que dan soporte a la construcción de sistemas software, almacenamiento y procesamiento de datos

### **PROGRAMA ANALÍTICO**

- 1. Introducción a la Ingeniería del Software**
  - a. Origen y definiciones básicas
  - b. Conceptos fundamentales
- 2. Ingeniería de Requisitos de Software**
  - a. Definición de Requisito, Tipos y Propiedades
  - b. Modelos de Proceso de IR
  - c. Propuesta metodológica

- d. Etapa de Adquisición de Requisitos
- e. Etapa de Análisis de Requisitos
- f. Etapa de Validación y Verificación de Requisitos
- 3. **Modelado de Requisitos con UML 2,0 - Diagramas de Casos de Uso**
  - a. Definición y especificación de Casos de Uso
  - b. Notación gráfica
  - c. Tipos de Relaciones en un Diagrama de Casos de Uso
  - d. Ejemplos y casos prácticos
- 4. **Análisis Orientado a Objetos con UML 2,0**
  - a. Introducción al modelado de SW con UML 2,0
  - b. Diagramas de Clases de Dominio
  - c. Diagramas de Clases de Análisis
  - d. Diagramas de Actividad
- 5. **Diseño Orientado a Objetos con UML 2,0**
  - a. Diagramas de Estados
  - b. Diagramas de Secuencia
  - c. Diagramas de Comunicación
  - d. Diagramas de Tiempos
  - e. Diagramas de Componentes
  - f. Diagramas de Despliegue
- 6. **Desarrollo de Sistemas Orientados a Objetos: Proceso Unificado de Desarrollo (RUP)**
  - a. Introducción
  - b. Principios básicos de RUP
  - c. Fases y Etapas de RUP
- 7. **Gestión y Pruebas de software**
  - a. Gestión de Configuración Software
  - b. Pruebas de Software: Introducción y Principios
  - c. Pruebas de Software: Estrategias y Procesos
- 8. **Calidad de Software y Mantenimiento Software**

## **BIBLIOGRAFÍA**

1. Blum, B. I. Software Engineering: A Holistic View. Oxford University Press US, 7th edition., 1992
2. Pressman, R. S. Software Engineering: A Practitioner's Approach. McGrawHill, 6th edition. 2004
3. Schach, S. R. Object-Oriented and Classical Software Engineering. McGrawHill., 2004
4. BROWN, W.J., MCCORMICK, H.W., THOMAS, S.W. AntiPatterns: refactoring software, architectures and projects in crisis. John Wiley and Sons
5. COLLARD, J.F. BURNSTEIN, I. Practical Software Testing: A Process-Oriented Approach. Springer
6. GALIN, D. Software Quality Assurance: From theory to implementation. Addison-Wesley
7. KAN, S.H. Metrics and Models in Software Quality Engineering. Addison-Wesley
8. KANER, C., NGUYEN, H.Q., FALK, J. Testing Computer Software. John Wiley & Sons

9. LEON, A. Software Configuration Management Handbook. Artech House.
10. PIGOSKI, T.M. Practical Software Maintenance. John Wiley & Sons.

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ADMINISTRACIÓN DE REDES

CÓDIGO: CC612

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC511 - COMPUTACIÓN CENTRADA EN REDES

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

En la actualidad podemos observar en cualquier ambiente de trabajo, e incluso cualquier ambiente de nuestro alrededor con un sistema informático. Como nos podemos imaginar todas estas máquinas se encuentran interconectadas entre sí las unas con las otras de manera que existe una comunicación entre todas ellas, pudiendo ser a nivel local, mediante una red LAN, a nivel global, con una conexión WAN o Internet.

Estas infraestructuras implican a varios usuarios comunicándose entre ellos e intercambiando información, compartiendo recursos como impresoras, programas y muchas más cosas que requieren de un elemento imprescindible para manejar tanto el Software como el Hardware, como es el Administrador de Red.

Toda esta malla de comunicaciones e intercambio de información debe realizarse de forma segura y con la menor tolerancia a fallos posible, ya que la información es el punto central y crítico de las comunicaciones entre los nodos.

### **COMPETENCIAS**

- Conocer el trabajo del administrador de sistemas informáticos y cómo llevarlo a cabo de forma adecuada
- Comprender el trabajo de un Sistema Operativo como elemento central del administrador de red
- Conocer el proceso de comunicación e intercambio de información entre los distintos Sistemas Operativos
- Conocer los conceptos y las herramientas que tiene a su disposición un Administrador de Red
- Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas
- Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores de Internet y diseñar e implementar aplicaciones basadas en ellas



- Capacidad para gestionar adecuadamente la información disponible integrando creativamente conocimientos y aplicándolos a la resolución de problemas informáticos utilizando el método científico

## **PROGRAMA ANALÍTICO**

1. Administración básica a nivel de sistema operativo
  - a. Comandos básicos e instalación de SO
  - b. Gestión de aplicaciones
  - c. Gestión de usuarios
  - d. Administración de Ficheros
2. Protocolos de la capa de red
  - a. Funcionalidad y datagrama
  - b. Encaminamiento IP
  - c. Protocolos de encaminamiento dinámico
  - d. ICMP
3. Protocolos de la capa de transporte
  - a. UDP
  - b. TCP
4. Administración avanzada a nivel de sistema operativo
  - a. Discos redundantes
  - b. Monitorización del sistema
  - c. Administración de red y encaminamiento
5. Administración básica de servidores
  - a. CIDR, VLSM y SSH
  - b. HTTP
  - c. DHCP
  - d. NAT
6. Administración avanzada de servidores
  - a. DNS
  - b. DDNS
  - c. Servidor de correo
7. Control de acceso a servicios
  - a. Proxy
  - b. Firewall

## **BIBLIOGRAFÍA**

1. "UNIX. Interconexión de redes". U.O. Pabrai. RA-MA. 1997
2. "TCP/IP". S. Feit. Osborne McGraw-Hill. 1998
3. "Linux. Administración del sistema y de la red". I. Alegría Loinaz, et al. Pearson/Prentice-Hall. 2005
4. "Red Hat Enterprise Linux and Fedora Core 4". R. Petersen. McGraw-Hill. 2005
5. Villalón Huerta, A. (2002). Seguridad en UNIX y redes. Versión 2.1
6. "Seguridad Práctica en UNIX e Internet (2ª edición)". S. Garfinkel, G. Spafford: Ed. McGraw-Hill. 2000
7. Dhanjani, Nitesh (2008). Claves hackers en Linux y UNIX. Mc Graw-Hill
8. Dwivedi (2007). Hacking Exposed Web 2.0: Web 2.0 Security Secrets and Solutions. Estats Units: McGraw-Hill

9. "Fundamentos de Seguridad en Redes. Aplicaciones y Estándares (2ª edición)". W. Stallings. Pearson-Prentice Hall.2004
10. "Seguridad en Redes Telemáticas". J. Carracedo Gallardo. McGraw-Hill. 2004
11. "Red Hat Linux Firewalls". B. McCarty. Anaya Multimedia. 2003
12. Craig Hunt. TCP/IP network administration, Ed. 3, Publicado por O'Reilly, 2002
13. Tony Bautts, Terry Dawson, Gregor N. Purdy. Linux network administrator's guide. Ed. 3 Publicado por O'Reilly, 2005
14. L. Parziale. "TCP/IP Tutorial and Technical Overview". 8th edition. IBM RedBooks. 2006
15. Q. Li. "IPv6 Core Protocols Implementation". 1st edition. Morgan Kaufmann Publishers. 2005
16. F. Halsall. "Redes de Computadores e Internet". 5ª edición. Addison-Wesley. 2006
17. B. Sosinsky. "Networking Bible". 1st edition. Wiley Publishing. 2009
18. E. Cole. "Network Security Bible". 2nd edition. John Wiley & Sons. 2009

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: MATEMÁTICA COMPUTACIONAL

CÓDIGO: CC622

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC531 - ANÁLISIS Y DISEÑO DE ALGORITMOS
- CM4F1 - ANÁLISIS Y MODELAMIENTO NUMÉRICO I

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

El objetivo de las matemáticas computacionales, en pocas palabras, es encontrar o desarrollar algoritmos que resuelvan problemas matemáticos computacionalmente (es decir, usando computadoras). En particular, deseamos que cualquier algoritmo que desarrollemos cumpla cuatro propiedades: Exactitud (un algoritmo preciso es capaz de devolver un resultado numérico muy cerca del resultado correcto o analítico), Eficiencia (un algoritmo eficiente es capaz de resolver rápidamente el problema matemático con recursos computacionales razonables), Robustez (un algoritmo robusto funciona para una amplia variedad de entradas  $x$ ), y Estabilidad (un algoritmo estable no es sensible a pequeños cambios en el entrada  $x$ ).

### **COMPETENCIAS**

- Desarrollar las matemáticas para la implementación de algoritmos específicos para los sistemas gráficos disponibles
- Utilizar herramientas algorítmicas y matemáticas para crear todo tipo de imágenes, como las imágenes sintéticas de objetos y escenas tridimensionales
- Emplear los algoritmos de optimización comúnmente utilizados en la práctica, y elegir un algoritmo para un problema dado
- Desarrollar estructuras de datos y algoritmos para automatizar la obtención de resultados existentes y nuevos en probabilidad, estadística y métodos numéricos

### **PROGRAMA ANALÍTICO**

1. Matemáticas para la Computación Gráfica
  - a. Transformaciones geométricas
  - b. Interpolación
  - c. Curvas y superficies
  - d. Geometría analítica

- e. Álgebra geométrica
- 2. Optimización Computacional
  - a. Preliminares: ejemplos de problemas de optimización concretos
  - b. Programación lineal: formato de los problemas, soluciones admisibles y optimalidad, Método Simplex y Método del Punto-Interior
  - c. Programación no lineal: tipos de problemas, condiciones de optimalidad, direcciones de búsqueda y búsqueda lineal
  - d. Algoritmos de programación diferenciable y no diferenciable
- 3. Probabilidad Computacional
  - a. Preliminares
  - b. Algoritmos para variables aleatorias continuas
  - c. Algoritmos para variables aleatorias discretas
  - d. Simulación estocástica
- 4. Métodos Numéricos
  - a. Análisis lineal
  - b. El Método de Galerkin: formulación y solución de PDEs
  - c. Métodos para la solución de ecuaciones diferenciales que dependen del tiempo

## **BIBLIOGRAFÍA**

1. Mathematical Structures for Computer Graphics. Janke, Steven. John Wiley & Sons. 2015
2. Numerical Algorithms. Solomon, Justin. CRC Press. 2015
3. Fundamental of Computer Graphics. 4th edition. Marschner, Steve and Shirley, Peter. CRC Press. 2016
4. Mathematics for Computer Graphics. 5th edition. Vince, John. Springer. Undergraduate Topics in Computer Science (UTiCS). 2017
5. Optimization in Engineering: Models and Algorithms. Sioshansi, Ramteen and Conejo, Antonio. Springer. Springer Optimization and Its Applications, Volume 120. 2017
6. Numerical Optimization with Computational Errors. Zaslavski, Alexander. Springer Optimization and Its Applications, Volume 108. 2016
7. Computational Optimization, Methods and Algorithms. Koziel, Slawomir and Yang, Xin She. Springer. Studies in Computational Intelligence, Volume 356. 2011
8. Computational Probability: Algorithms and Applications in the Mathematical Science. 2nd edition. Drew, John et al. Springer. International Series in Operations Research & Management Science, Volume 246. 2017
9. An introduction to Computational Stochastic PDEs. Lord, Gabriel et al. Cambridge University Press. Cambridge Texts in Applied Mathematics. 2014

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: PROGRAMACIÓN PARALELA

CÓDIGO: CC632

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC531 - ANÁLISIS Y DISEÑO DE ALGORITMOS

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

El fundamento de la programación secuencial orientada a bloques es que una reordenación en la (re)utilización de los datos de entrada de un problema permite obtener un gran beneficio computacional gracias a la utilización eficiente de la jerarquía de memoria. Una adecuada división de nuestro problema en bloques permite obtener programas más eficientes para una misma arquitectura.

Por otra parte, los sistemas informáticos actuales son inherentemente paralelos: plataformas de memoria compartida, plataformas de memoria distribuida, multi-núcleo, computación heterogénea, ... Sin embargo, la mayoría de las aplicaciones actuales no explotan dichas arquitecturas paralelas, produciéndose como consecuencia una mala utilización de los recursos y un bajo rendimiento. Es necesario, por tanto, conocer las técnicas que nos permitan obtener programas más rápidos y eficientes que utilicen toda la potencia de la arquitectura subyacente.

### **COMPETENCIAS**

- Coordinar las tareas de todos los elementos involucrados en el funcionamiento de un sistema de procesamiento de datos distribuidos y de altas prestaciones
- Diseñar y dimensionar equipos de procesamiento de datos de altas prestaciones y alta disponibilidad
- Conocer las principales arquitecturas de los sistemas de alta disponibilidad
- Resolver problemas mediante algoritmos paralelos basados en memoria o procesamiento distribuido

### **PROGRAMA ANALÍTICO**

1. Introducción a la programación de altas prestaciones

2. Programación orientada a bloques
3. Introducción a las arquitecturas paralelas
4. Paradigmas de computación paralela
5. Diseño de programas en arquitecturas paralelas
6. Software de programación de arquitecturas de altas prestaciones
7. **Tipo de datos abstractos**
  - a. Stacks
  - b. Colas
  - c. Colas con prioridad
  - d. Conjuntos
  - e. Tablas
  - f. Árboles
  - g. Búsqueda en árboles binarios, Heaps y árboles AVL
8. **Algoritmos de grafos**
  - a. Búsqueda siguiendo depth-first and breath-first
  - b. Mínimo spanning tree
  - c. Búsqueda en árboles
9. **Programación dinámica en paralelo**

## **BIBLIOGRAFÍA**

1. Kumar, V; Grama, A; Gupta, A; Karypis, G.. Introduction to Parallel Computing, Second Edition. Addison Wesley, 2003
2. Geist, A.; Beguelin, A; Dongarra, J. et al. PVM Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing. The MIT Press, 1994
3. Pacheco, P.S. Parallel Programming with MPI. Morgan Kaufmann Publishers 1997
4. Enrique Arias; Algoritmos de altas prestaciones para la simulación, estimación y control de sistemas no lineales.; Universidad Politécnica de Valencia; 2003
5. G. H. Golub, C. F. Van Loan; Matrix computations; The Johns Hopkins University Press; 1996
6. J. L. Hennessy, D. A. Patterson; Computer architecture: a quantitative approach; Morgan Kaufman Publishers 2003
7. Julio Ortega, Mancia Anguita y Alberto Prieto; Arquitectura de computadores; Thomson Paraninfo 2005
8. P. Pacheco; Parallel Programming with MPI ; MIT Press 1994
9. R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, J. McDonald; Parallel Programming in OpenMP; Morgan Kaufman Publishers 2000
10. W. Gropp, E. Lusk, A. Skjellum; Using MPI: Portable Parallel Programming with the Message- Passing Interface; MIT Press 1994

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: DESARROLLO DE SOFTWARE

CÓDIGO: CC642

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC541 - INGENIERÍA DEL SOFTWARE

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

El futuro licenciado en Ciencia de la Computación tendrá una formación variada en todos los campos de producción e investigación tanto en sus capacidades técnicas como habilidades de emprendimiento. Este curso le capacitará como usuario de una gran cantidad de áreas y le pondrá en contacto con creativos de las diferentes áreas de producción.

En el actual panorama, el soporte en que llegan al usuario la mayoría de productos multimedia es la web. Por lo tanto, el futuro licenciado tiene que dominar los diferentes aspectos de desarrollo web, a través de los diferentes lenguajes de programación de servidor y gestores de bases de datos para la creación de aplicaciones.

### **COMPETENCIAS**

- Diseñar y programar aplicaciones y servicios Con acceso a datos, usando las arquitecturas, los lenguajes y las herramientas más apropiadas
- Conocer cuáles son los estándares para contenidos digitales
- Gestionar la seguridad en sistemas informáticos

### **PROGRAMA ANALÍTICO**

1. Introducción y conceptos básicos
  - a. Introducción al currículo de estándares web/contenidos, La historia de Internet y la web y la evolución de los estándares web
  - b. Arquitectura OSI
  - c. Protocolos de internet
2. **Texto etiquetado y hojas de estilo**
  - a. Conceptos básicos de HTML
  - b. El elemento del HTML
  - c. Elegir el doctype correcto para los documentos HTML
  - d. Etiquetar contenido textual en HTML

- e. CSS y su importancia en un buen diseño
  - f. Herencia y cascada
  - g. Creación de estilos de texto con CSS
  - h. Construcción del esqueleto de una página web
- 3. **Lenguajes de programación interpretado y sus recursos en desarrollo web**
- 4. **Lenguajes para entornos de programación multiplataforma**
- 5. **Programación del lado servidor**
  - a. Arquitectura y funcionamiento de un desarrollo basado en cliente-servidor
  - b. Sintaxis
  - c. Variables y operadores
  - d. Uso de para aplicaciones web
  - e. Programación Orientada a Objetos
  - f. Programación de aplicaciones y proyectos
- 6. **Acceso a base de datos**
  - a. Controladores y direcciones
  - b. Sintaxis
  - c. Acceso básico a los datos
  - d. Transacciones
  - e. Metadatos
  - f. Integración de aplicaciones con PHP
- 7. **Servicios Web**
  - a. Introducción a los servicios Web
  - b. XML-RPC
  - c. SOAP
  - d. WSDL y UDDI
- 8. **Seguridad en internet**
  - a. Cifrado
  - b. Firma digital
  - c. Vulnerabilidad y ataques



UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: SEGURIDAD EN SISTEMAS INFORMÁTICOS

CÓDIGO: CC711

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC612 - ADMINISTRACIÓN DE REDES

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

La asignatura inicialmente dará al alumno los conocimientos previos para realizar una implementación de sistemas de gestión de seguridad de información, también proporcionará los conocimientos para realizar un buen control de las actividades de seguridad en sistemas informáticos, que permite afianzar los conceptos, técnicos y métodos que permitan administrar la seguridad de la información de la empresa en forma eficiente y minimizando riesgos que coloquen en peligro la continuidad operativa de la organización. Su temática comprende: Conceptos de seguridad informática, niveles de seguridad en las corporaciones, gestión de riesgos y en redes de computadores y, más concretamente, en las redes TCP/IP.

### **COMPETENCIAS**

- Protege la información de las organizaciones de los diferentes riesgos informáticos que puedan alterar o dañar los recursos informáticos, por medio de diversos mecanismos de seguridad siguiendo las técnicas de seguridad y las mejores prácticas de la industria relacionadas con seguridad de la información
- Entender los distintos tipos de vulnerabilidades que presentan las redes TCP/IP
- Ver qué técnicas de prevención existen contra los ataques más frecuentes
- Conocer los diferentes sistemas de detección de intrusos
- Alcanzar unos conocimientos básicos del funcionamiento de las herramientas criptográficas más utilizadas
- Conocer los sistemas de autenticación más importantes, identificando sus características

### **PROGRAMA ANALÍTICO**

1. **Seguridad de la Información y de la Informática**
  - a. Enfoque integral de la seguridad de la información
  - b. Gestión de riesgos

- c. Análisis de riesgos de tecnología de información
  - d. Seguridad informática vs seguridad de la información
  - e. Objetivos de la seguridad informática
  - f. Amenaza, vulnerabilidad y ataques
  - g. Estándares y pilares de la seguridad
2. **Técnicas de Identificación y Autenticación**
- a. Control de acceso, sujeto y objeto
  - b. Confidencialidad, integridad, disponibilidad, categorías de control de acceso
  - c. Control de acceso preventivo, detectivo, correctivo, implementación de control de accesos: administración, lógico y físico
  - d. Identificación, autenticación, autorización, técnicas de identificación, autenticación: tipo 1: Algo que conoces, tipo 2: algo que tienes, tipo 3: algo que eres, otros tipos de autenticación
3. **Ataques contra las redes TCP/IP**
- a. Seguridad en redes TCP/IP
  - b. Actividades previas a la realización de un ataque
  - c. Escuchas de red
  - d. Fragmentación IP
  - e. Ataques de denegación de servicio
  - f. Deficiencias de programación
4. **Mecanismos de prevención**
- a. Sistemas cortafuegos
  - b. Construcción de sistemas cortafuegos
  - c. Zonas desmilitarizadas
  - d. Características adicionales de los sistemas cortafuegos
5. **Mecanismos para la detección de ataques e intrusiones**
- a. Necesidad de mecanismos adicionales en la prevención y protección
  - b. Sistemas de detección de intrusos
  - c. Escáneres de vulnerabilidades
  - d. Sistemas de decepción
  - e. Prevención de intrusos
  - f. Detección de ataques distribuidos
6. **Mecanismos de protección**
- a. Conceptos básicos de criptografía
  - b. Sistemas de autenticación
  - c. Protección a nivel de enlace: redes inalámbricas
  - d. Protección a nivel de red: IPsec
  - e. Protección a nivel de transporte: SSL/TLS/WTLS
  - f. Redes privadas virtuales (VPN)
7. **Aplicaciones seguras**
- a. Autenticación mediante cliente web: portales cautivos
  - b. El protocolo SSH
  - c. Correo electrónico seguro
8. **Criptografía**
- a. Introducción a la Criptografía y Algoritmos clásicos de Criptografía
  - b. Algoritmos de clave simétrica y funciones unidireccionales
  - c. Algoritmos de clave simétrica

- d. Certificados digitales y PKI
  - e. Aplicaciones Criptográficas
  - f. Esteganografía
9. **Análisis forense**
- a. Introducción al Análisis Forense y Tipos
  - b. Adquisición y recopilación de evidencias
  - c. Análisis de imágenes
  - d. Análisis Forense en sistemas y redes

## **BIBLIOGRAFÍA**

1. Cheswick, W.R.; Bellovin, S.M.; Rubin, A.D. (2003). Firewalls and Internet Security: Repelling the Wily Hacker. (5a ed.). Addison-Wesley Professional Computing
2. Oppliger, R. (2001). Security technologies for the Word Wide Web. (1a ed.). Artech House
3. Menezes, J.; van Oorschot, P.C.; Vanstone, S.A. (2000). Handbook of Applied Cryptography. (5a ed.). CRC Press
4. GREENWALD G. (2014). Snowden: Sin un lugar para esconderse. Barcelona: Ediciones B
5. INFORMATION SYSTEMS AUDIT AND CONTROL ASSOCIATION. (2016). CISA Review Manual. Chicago: ISACA
6. INFORMATION SYSTEMS AUDIT AND CONTROL ASSOCIATION. (2012) COBIT 5 Un marco de negocio para el Gobierno y la Gestión de la TI en la Empresa. USA:ISACA
7. INFORMATION SYSTEMS AUDIT AND CONTROL ASSOCIATION. (2012). COBIT 5 Procesos catalizadores. USA: ISACA
8. MITNICK, K., WOZNIAK S. (2012). Ghost in the Wires: My Adventures as the Worlds Most Wanted Hacker. USA: Little, Brown and Company
9. MITNICK, K., SIMON W. (2008) El Arte de la Intrusion - Como Ser un Hacker o Evitarlos, España: Ra-MA
10. MITNICK, K., SIMON W. (2005). The Art of Intrusion: The Real Stories Behind the Exploits of Hackers, Intruders and Deceivers. USA: Wiley Publishing
11. MITNICK, K., SIMON W. (2003). The Art of Deception: Controlling the Human Element of Security. USA: Wiley Publishing
12. SCHNEIER B. (2013) Carry On: Sound advice from Schneier on Security. USA: Wiley Publishing
13. DOUGLAS COMER. Redes globales de información con Internet y TCP/IP. 1ra Edición. Prentice-Hall Hispanoamericana. México
14. RAUL SILES PELAEZ. Análisis de Seguridad de la familia de protocolos TCP/IP y sus servicios asociados. 1ra Edición. Junio del 2002
15. FÚSTER, A.; DE LA GUÍA, D.; HERNÁNDEZ, L.; MONTOYA, F.; MUÑOZ, Técnicas Criptográficas de Protección de Datos. J. Ra-Ma, 1997
16. JEIMY J. CANO, Auditoria de Seguridad, Evaluación de Seguridad y Pruebas de Penetración: tres paradigmas de la Seguridad Informática. Universidad de Los Andes, Colombia, 1998
17. STALLINGS, WILLIAM Cryptography and Network Security. Principles and Practice. 2nd ed. Prentice Hall International Editions, 1999

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: INTELIGENCIA ARTIFICIAL

CÓDIGO: CC721

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC632 - PROGRAMACIÓN PARALELA

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Este curso representa la puerta de entrada o presentación a las técnicas de Inteligencia Artificial y la ingeniería del conocimiento, es decir, los tópicos relacionados a los sistemas inteligentes. Estas técnicas se incluyen hoy en día entre las más requeridas para la resolución de problemas complejos en cualquier ámbito del desarrollo científico o profesional relacionado a la Ciencia de la Computación. Desarrollar programas que tengan comportamiento inteligente con diferentes temas como búsqueda, representación de conocimiento, agentes inteligentes, procesamiento de lenguaje natural, aprendizaje, redes neuronales, árboles de decisión y aprendizaje de máquina.

### **COMPETENCIAS**

- Representación del conocimiento y razonamiento, búsqueda avanzada del conocimiento y razonamiento
- Evaluar las posibilidades de simulación de la inteligencia, para lo cual se estudiarán las técnicas de modelización del conocimiento
- Construir una noción de inteligencia que soporte después las tareas de su simulación

### **PROGRAMA ANALÍTICO**

#### **1. Tópicos Fundamentales en Sistemas Inteligentes**

- a. Historia de la inteligencia artificial
- b. Cuestiones filosóficas
- c. La prueba de Turing
- d. Experimento de pensamiento del “Cuarto Chino” de Searle

- e. Temas típicos en IA
- f. Definiciones fundamentales
- g. Comportamiento óptimo vs comportamiento actuando como humano
- h. Razonamiento óptimo vs razonamiento actuando como humano
- i. Preguntas filosóficas
- j. Modelando el mundo
- k. El rol de la heurística

## **2. Búsqueda y Satisfacción de la Restricción**

- a. Problemas de espacio
- b. Búsqueda de fuerza bruta (respiro primero, profundidad primero, profundidad primero con profundización iterativa)
- c. Búsqueda del mejor primero (mejor primero genérico, algoritmo de Dijkstra, A\*, admisibilidad de A\*)
- d. Juegos de dos jugadores (búsqueda mínima, poda alfa-beta)
- e. Satisfacción de la restricción (backtracking o métodos de búsqueda local y seguimiento)

## **3. Representación del Conocimiento y Razonamiento**

- a. Inferencia no monotónica
- b. Razonamiento probabilístico
- c. Teorema de Bayes

## **4. Búsqueda Avanzada**

- a. Algoritmos genéticos
- b. Simulated annealing
- c. Búsqueda local

## **5. Representación Avanzada del Conocimiento y Razonamiento**

- a. Incerteza
- b. Razonamiento probabilístico
- c. Redes Bayesianas
- d. Conjuntos difusos y teoría de la posibilidad
- e. Teoría de la decisión

## **6. Agentes**

- a. Definición de agentes
- b. Arquitectura de agentes
- c. Agentes reactivos simples
- d. Planeadores reactivos
- e. Arquitecturas de capas
- f. Ejemplos de arquitecturas y aplicaciones
- g. Teoría de agentes
- h. Acuerdos

- i. Intenciones
  - j. Agentes de decisión teórica
  - k. Procesos de decisión de Markov (PDM)
  - l. Agentes que aprenden
  - m. Sistemas multiagente
  - n. Sistemas multiagente inspirados económicamente
  - o. Agentes colaborativos
  - p. Equipos de agentes
  - q. Modelando agentes
  - r. Aprendizaje multiagente
7. **Procesamiento de Lenguaje Natural**
- a. Gramáticas determinísticas y estocásticas
  - b. Algoritmos de parsing
  - c. Métodos basados en corpus
  - d. Recuperación de información
  - e. Traslación de lenguaje
  - f. Reconocimiento del habla
8. **Aprendizaje de Máquina y Redes Neuronales**
- a. Definición y ejemplos de aprendizaje de máquina
  - b. Aprendizaje supervisado
  - c. Árboles de aprendizaje por decisión
  - d. Redes neuronales de aprendizaje
  - e. Redes de aprendizaje por creencia
  - f. Algoritmo del vecino más cercano
  - g. Teoría de aprendizaje
  - h. El problema del sobreajuste
  - i. Aprendizaje no supervisado
  - j. Aprendizaje por refuerzo

## **BIBLIOGRAFÍA**

1. Goldberg, D. Genetic Algoritmos in Search, Optimization and Machine learning. Addison Wesley. 1989
2. Haykin, S. Neural networks: A comprehensive Foundation. Prentice Hall. 1999
3. Nilsson, N. Inteligencia Artificial: Una nueva visión. McGraw-Hill 2001
4. Russell, S. and Norvig, P. Inteligencia Artificial: Un enfoque moderno. Prentice Hall. 2003
5. Winston, P.H. y Horn, B.K. *LISP 3a. ed.*. Addison--Wesley, 1991
6. Guy L. Steele Common Lisp: The Language, 2nd edición. Editorial: Burlington, MA : Digital Press, ©1984
7. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach: Editorial: Upper Saddle River, N.J. : Prentice Hall/Pearson Education, ©2003

8. T. Mitchell. Machine Learning. McGraw Hill, 1997
9. Nilsson, Nils J. Inteligencia Artificial. Una nueva síntesis. 1ra Edición. McGraw Hill Interamericana de España, S.A.U. 2001

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: COMPUTACIÓN GRÁFICA

CÓDIGO: CC731

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC622 - MATEMÁTICA COMPUTACIONAL

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

El presente curso está concebido para estudiantes del séptimo semestre. Esta asignatura ofrece una introducción general de los principios, técnicas y herramientas necesarias para el desarrollo de aplicaciones y sistemas gráficos en 2D y 3D. Además, ofrece una experiencia práctica en profundidad en el desarrollo de aplicaciones y sistemas gráficos modernos.

### **COMPETENCIAS**

- Aplica los conocimientos de computación y de matemáticas apropiadas para la disciplina
- Demuestra su capacidad de análisis identificando y definiendo los requerimientos computacionales apropiados para la solución de problemas reales
- Utiliza técnicas y herramientas actuales necesarias para el desarrollo de soluciones a problemas donde se requiera la aplicación de la computación gráfica

### **PROGRAMA ANALÍTICO**

1. **Introducción a los gráficos por ordenador - Sistemas Gráficos y Modelos**
  - a. Objetivos y definiciones básicas
  - b. Breve historia
  - c. Aplicaciones (CAD, Gráficas, esquemas y modelos, Arte y animación por ordenador, etc)
  - d. Procesamiento de Imágenes
  - e. El sistema visual humano. El modelo de cámara pin-hole
  - f. Interfaces gráficas
  - g. Arquitecturas gráficas
  - h. Introducción a OpenGL



2. **Transformaciones geométricas**
  - a. Objetivos y definiciones básicas
  - b. Sistemas de coordenadas
  - c. Transformaciones afines
  - d. Coordenadas Homogéneas
  - e. Concatenación de transformaciones
  - f. Implementación de transformaciones
3. **Transformaciones de visualización**
  - a. Proyecciones clásicas y visualización en el ordenador
  - b. Definición y posicionamiento de la cámara virtual
  - c. Proyecciones simples y en OpenGL
  - d. Matrices de proyección
4. **Técnicas de Realismo I**
  - a. Luz y Color: Propiedades de la Luz
  - b. Diagrama cromático CIE. Sistemas de color (RGB, CMY, YIQ, HSV, HLS)
  - c. Iluminación, Fundamentos y Modelo de Phong
  - d. Sombreado de polígonos
  - e. Funciones de OpenGL para Sombrear e Iluminación
  - f. Texturización
5. **Técnicas de Realismo II**
  - a. Conceptos Avanzados de Realismo
  - b. Modelos de Iluminación Global
  - c. Traza de Rayos
  - d. Modelo de Radiosidad
  - e. Fundamentos de Povray y RayShader
  - f. Shaders actuales. Lenguaje de programación de Shaders
6. **Modelamiento Geométrico**
  - a. Representación poligonal de objetos 3D
  - b. Curvas poligonales paramétricas y superficies
  - c. Representación de geometría sólida constructiva (CSG)
  - d. Representación implícita de curvas y superficies
  - e. Técnicas de subdivisión espacial
  - f. Modelos procedurales
  - g. Modelos deformables
  - h. Subdivisión de superficies
  - i. Modelamiento de multiresolución
  - j. Reconstrucción

## **BIBLIOGRAFÍA**

1. Hearn, Donald D.; Baker, M. Pauline. Computer Graphics with OpenGL. 3 edition. Pearson Education. 2003
2. Shreiner, Dave; Woo, Mason; Neider, Jackie; Davis, Tom. OpenGL Programming Guide: The Official Guide to Learning OpenGL. 6 edition. Addison-Wesley 2007
3. Libro Informática Gráfica. Materiales Didactics, UIB, Set 2012. F. J. Perales., J.. M. Buades, M. J. Abasolo. Interactive Computer Graphics. A Top-down approach with OpenGL. Ed. Angel. Addison Wesley 1997
4. OpenGL Reference Manual. Ed. Addison-Wesley. 1992

5. Programación en OpenGL. Ed. Anaya Multimedia. 1997
6. OpenGL en Fichas: Una introducción practica, J. Ribelles y J. Lluch. Treballs informatica i Tec. ,num14, UJI
7. Radiosity and realistic image synthesis / Michael F. Cohen, John , R. Wallace
8. OpenGL 4 Shading Language Cookbook, Second Edition. D. Wolff, Pact Open Spurge, 2011
9. Cohen, Michael F. Academic Press Professional, c1993
10. Ray Tracing II, Anaya Multimedia, 1994
11. OpenGL Programming Guide 8th Edition. Dave Shreiner et al. (Addison-Wesley, 2013)

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: PROGRAMACIÓN CONCURRENTES Y DISTRIBUIDA

CÓDIGO: CC741

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC642 - PROGRAMACIÓN PARALELA

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

El Curso de Programación Concurrentes y Distribuida está directamente ligada con la evaluación de los sistemas operativos y por ello subdivide en dos temáticas principales. Por un lado, la programación concurrentes está ligada a los conceptos de multiproceso y multiprocesador, por lo que se estudiará las técnicas básicas de gestión de la concurrencia, como por ejemplo los mecanismos de sincronización y comunicación entre procesos. Por otro lado, referente a la programación distribuida, está relacionada con el desarrollo de sistemas en los cuales la red es una infraestructura crítica. Por tanto, el alumno aprenderá los fundamentos básicos de programación de cliente/servidor bajo los protocolos UDP/TCP, hasta escalar al nivel más elevado automatizando las operaciones del sistema. En este contexto, se trabajarán los paradigmas RCP/RMI y la tecnología de componentes, una panorámica del tipo de aplicaciones y sistemas distribuidos modernos (Clustering, Grid, Cloud, P2P).

### **COMPETENCIAS**

- Programar en entornos de red con arquitectura cliente/servidor
- Aplicar las diferentes técnicas de comunicación entre procesos y grupos de procesos distribuidos
- Programar aplicaciones para entornos distribuidos
- Resolver problemas complejos y de respuesta en tiempo real utilizando conceptos y herramientas de programación concurrentes y tiempo real, planificando sus tareas, así como un uso eficiente de la memoria

### **PROGRAMA ANALÍTICO**

1. Conceptos fundamentales de la programación concurrentes y en tiempo real
2. **Sincronización con Espera Activa**
3. **Comunicación por Memoria Compartida**
  - a. Semáforos
  - b. Regiones Críticas Condicionales

- c. Monitores
- 4. **Paso de Mensajes**
- 5. **Tiempo Real**
- 6. **Introducción a la programación distribuida**
  - a. Nociones básicas
  - b. Arquitectura de sistemas distribuidos
  - c. Middleware
- 7. **Comunicación entre procesos distribuidos**
  - a. Primitivas básicas de comunicación
  - b. Comunicación por UDP y TCP en Java
  - c. Marshalling
  - d. Comunicación de grupos
- 8. **Objetos distribuidos e invocación remota**
  - a. Invocación remota
  - b. Comunicación entre objetos distribuidos
  - c. Llamada de procedimiento remoto
- 9. **Tecnologías Distribuidas Modernas**
  - a. Clustering
  - b. Tecnologías Grid y Cloud (virtualización)
  - c. Servicios Web
  - d. Computación móvil y ubicua
  - e. Redes P2P
- 10. **Sincronización en Sistemas Distribuidos**
  - a. Relojes físicos y lógicos
  - b. Estados globales
  - c. Exclusión mutua distribuida
  - d. Elecciones
- 11. **Transacciones y control de concurrencia**
  - a. Transacciones
  - b. Control de concurrencia en transacciones
  - c. Control optimista de concurrencia
  - d. Ordenación por marcas de tiempo
  - e. Recuperación de transacciones

## **BIBLIOGRAFÍA**

1. Andrew S. Tanenbaum. Sistemas Operativos Distribuidos. Publicado por Prentice-Hall, 1996
2. Andrew S. Tanenbaum. Redes de Ordenadores. Publicado por Prentice-Hall, 1991
3. Robert Orfali, Dan Harkey, Jeri Edwards. Client/Server Survival Guide. 3er ed. Publicado por John Wiley, 1999
4. George Coulouris, Jean Dollimore and Tim Kindberg. Distributed Systems: Concepts and Design. 4ta ed. Publicado por Addison-Wesley. 2005
5. Tom Sheldon . Lan Times - Guía de Interoperabilidad. Publicado por Osborne/McGraw Hill, 1995
6. Tom Sheldon. Lan Times - Enciclopedia de Redes – Networking. Publicado por McGraw Hill 1994
7. Reaz Hoque. Corba 3 developers guide - IDG Books, 1998

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: INTERACCIÓN HUMANO-COMPUTADORA

CÓDIGO : CC751

CRÉDITOS : 04 (CUATRO)

PRE-REQUISITOS:

- CC642 - DESARROLLO DE SOFTWARE

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

La interfaz de usuario es la parte visible de las aplicaciones informáticas. Por otro lado, la usabilidad de las aplicaciones implica la creación de interfaces de usuario para las aplicaciones fáciles de aprender, de usar y que satisfagan al usuario.

En este contexto, en este curso se pretende profundizar en el desarrollo de interfaces de usuario que cumplan con unos altos estándares de calidad, siendo esta una extensión fundamental de la Ingeniería del Software y, por tanto, fundamental para cualquier profesional que se quiera dedicar al desarrollo web.

Con el estudio y aplicación de esta materia, el alumno desarrollará aplicaciones teniendo en cuenta consideraciones fundamentales, básicas y previas tales como la propia persona, los mecanismos de interacción o algunas reglas de diseño.

### **COMPETENCIAS**

- Conocer y aplicar el concepto de la interacción persona ordenador en un entorno de desarrollo de software
- Comprender los elementos principales de la IPO: las personas, la tecnología y el diseño
- Conocer el diseño centrado en el usuario
- Analizar, diseñar y evaluar productos interactivos centrados en el usuario
- Identificar los aspectos principales y la relación entre tecnología, diversidad y accesibilidad, y saber evaluar la accesibilidad de lugares web

### **PROGRAMA ANALÍTICO**

1. **Introducción a la interacción persona-ordenador**
  - a. Definición del concepto
  - b. Multidisciplinariedad
  - c. Historia

- d. Conceptos fundamentales en IHC
  - e. Diseño centrado en el usuario
- 2. **Elementos de la IPO: diseño, personas y tecnología**
  - a. Tecnología de la interacción
  - b. El factor humano
  - c. El diseño
- 3. **Diseño centrado en el usuario**
  - a. Diseño centrado en el usuario
  - b. Principios clave para el diseño con usabilidad
  - c. Ingeniería de la Usabilidad
  - d. El ciclo de vida de la Ingeniería de la Usabilidad
  - e. Diseño centrado en el usuario
  - f. Propuesta de Greenberg
  - g. Propuesta de diseño de la UPA (Asociación de profesionales de la usabilidad)
  - h. Diseño centrado en el uso
  - i. OVID: Object, View and Interaction Design
  - j. Tendencia: El desarrollo de interfaces de usuario basado en modelos (Mb-UIDE)
- 4. **Tecnología, diversidad y accesibilidad**
  - a. El reto de la diversidad
  - b. Estrategias para afrontar la diversidad
  - c. Accesibilidad web
  - d. Evaluación de la accesibilidad
- 5. **HCI en el Proceso de la Ingeniería del Software**
- 6. **Usabilidad y modelos de calidad centrados en la usabilidad**
  - a. Definiciones previas y estándares internacionales
  - b. Principios de diseño
  - c. La usabilidad como proceso y como producto
  - d. La calidad en uso y su relación con la usabilidad
  - e. Factores, atributos y métricas
  - f. Usabilidad y diseño
- 7. **Conoce al usuario y sus tareas**
  - a. Análisis de tareas
  - b. Personas
- 8. **Especificación de la calidad**
  - a. La experiencia disponible: las guías de estilo y los patrones de interacción/usabilidad
  - b. Tendencia: El desarrollo de interfaces de usuario basado en modelos (Mb-UIDE)
  - c. Especificación de la interfaz de usuario: un ejemplo el lenguaje usiXML
- 9. **Evaluación de la usabilidad: métodos y técnicas**
  - a. Métodos de evaluación de la usabilidad: clasificación
  - b. Otros métodos de evaluación: en búsqueda de la evaluación automática
- 10. **Accesibilidad: criterios y herramientas de evaluación**
  - a. Accesibilidad: criterios y herramientas de evaluación
  - b. Criterios WAI
  - c. Section 508

- d. Herramientas
- e. Recomendaciones
- f. Disposiciones legales

## **BIBLIOGRAFÍA**

1. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale. Human-Computer Interaction. PrenticeHall. 2004
2. Krug, Steve. No me hagas pensar : una aproximación a la usabilidad en la. Pearson Prentice Hall. 978-84-8322-286-7. 2006
3. CONSTANTINE, L., LOCKWOOD, L. Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. ACM Press/Addison-Wesley Publishing Co. 1999
4. GALITZ, W. O. The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Technique. Wiley. 2007
5. GHAOUI, C. Encyclopedia Of Human Computer Interaction. Idea group Publishing. 2005
6. Guías de estilo de Gnome. <http://library.gnome.org/devel/hig-book/stable>
7. GOULD, J. D. and LEWIS, C. Designing for usability: key principles and what designers think. ACM. 1985
8. ISO/IEC Estándar ISO/IEC 9126-4
9. ISO/IEC. Norma ISO/IEC 9126-1
10. JARRETT, C. and GAFFNEY, G. Forms that Work: Designing Web Forms for Usability Morgan Kaufmann Publishers Inc. 2008
11. LÓPEZ JAQUERO, V., MONTERO, F., MOLINA, J.P., VANDERDONCKT, J. Computer-Aided Design of User Interfaces VI. Springer. 2009

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: METODOLOGÍA DE LA INVESTIGACIÓN

CÓDIGO: CC761

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS: NINGUNO

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 03 (TEORÍA: 01, PRÁCTICA: 02)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

La asignatura Metodología de la Investigación tiene como finalidad el desarrollar un enfoque conceptual y metodológico de la actividad investigadora en los campos propios de la Ciencia de la Computación en la formación en investigación y realización de actividades doctorales en las Tecnologías Informáticas.

Ha sido ampliamente estudiada la investigación como actividad profesional, así como los investigadores como individuos integrados en entidades académicas o productivas. Por otro lado, un desarrollo adecuado de la actividad de investigación se caracteriza por la necesidad de comunicar los resultados mediante artefactos escritos (artículos, informes, tesis) o presentaciones orales (en congresos, reuniones o talleres).

### **COMPETENCIAS**

- Capacidad para aplicar técnicas y metodologías que permitan abordar desde nuevas perspectivas los problemas de interés
- Buscar bibliografía científica, herramientas software y recursos para validación de resultados
- Plantear un tema de investigación de interés
- Estructurar un trabajo de investigación en los diferentes campos científicos
- Abordar y planificar un trabajo de investigación
- Redactar un trabajo científico

### **PROGRAMA ANALÍTICO**

1. Investigación científica y tecnológica
2. Búsqueda y difusión de la investigación
3. Diseño de proyectos de investigación
4. Desarrollo de un tema de investigación
5. Redacción y presentación de trabajos científicos

### **BIBLIOGRAFÍA**



1. Sabino, Carlos, El proceso de investigación
2. Briones G. “Métodos y Técnicas de Investigación”. Trillas 1995
3. Cea d’Ancona Ángeles, Métodos y Técnicas de Investigación cuantitativa”, Editorial Síntesis Madrid 1997
4. Festinger y Katz. “Los Métodos de Investigación en Ciencias Sociales”. Piados 1992
5. Flórez Ochoa Rafael y Alonso Tobón Restrepo. Investigación Educativa y Pedagógica. Bogotá: McGraw Hill. 2001
6. Grawitz M. “Métodos y Técnicas de las Ciencias Sociales I-II.” Editorial Mexicana 1984, México
7. Hernández, Fernández Baptista. “Metodología de la Investigación”. McGraw Hill 1994. Colombia
8. Padua J. “Técnicas de Investigación” FCE-Colegio de México 1982, México
9. Sabino, Carlos A. El Proceso de Investigación. Buenos Aires: Edit. Lumen.199
10. Salkind, Neil J. Métodos de Investigación. México: Prentice Hall. 1999

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: INFRAESTRUCTURAS DE COMPUTACIÓN

CÓDIGO : CC822

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC741 - PROGRAMACIÓN CONCURRENTES Y DISTRIBUIDA

CONDICIÓN : OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, PRÁCTICA: 02, LABORATORIO: 02)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Los sistemas Grid son entornos muy variables, compuestos por una serie de organizaciones independientes que comparten sus recursos. Durante la década pasada, la Computación Grid ha surgido como una tecnología que ha hecho posible muchas aplicaciones intensivas en datos y/o en cálculo. A través del empleo de tecnologías Grid, es posible agregar recursos dispersos y heterogéneos para resolver varios tipos de aplicaciones paralelas en ciencia, ingeniería y comercio. Los usuarios de Grid deberán percibir alguna calidad de servicio (QoS) en las prestaciones de sus aplicaciones, en términos de tiempo de respuesta o de número de trabajos finalizados por unidad de tiempo. Sin embargo, esto es muy difícil de conseguir debido a la gran cantidad de redes interconectadas que forman parte de un Grid.

Cloud Computing proviene de la idea de un nuevo paradigma informático donde se incorpora tanto la computación distribuida, diversos centros de datos, aplicaciones, un consumo computacional y sobretodo el término virtualización como también plataformas de sistemas operativos y proveedores de servicio de acuerdo a una demanda extensa de los clientes, a través del Internet.

Por último, el curso verá también el concepto novedoso que surge de los dos anteriores para el internet of things, el llamado Fog Computing. Fog Computing es una extensión del Cloud Computing que pretende descentralizar y balancear los recursos a través de la red de sensores o dispositivos de almacenamiento.

### **COMPETENCIAS**

- Adquirir los conocimientos y desarrollar las aptitudes necesarias en relación al funcionamiento y programación en tecnologías de altas prestaciones en Grid como en Cloud Computing; como los mecanismos presentes en su funcionamiento y mejoras en el rendimiento
- Diferenciar la tecnología de cliente de la tecnología de servidor; sus respectivos costes, implicaciones técnicas y funcionales; y cómo la Web 2.0 ha supuesto, en

algunos ámbitos, una migración progresiva de la primera hacia la segunda

- Explicar los conceptos básicos del cloud computing: software as a service (SaaS), platform as a service (PaaS) e infrastructure as a service (IaaS)
- Incorporar en la caja de herramientas del profesional las herramientas básicas para almacenar información en la Red, compartir archivos y trabajar en red
- Analizar el impacto en la productividad, los costes y la organización del trabajo del uso de soluciones de cloud computing en el día a día de la empresa

## **PROGRAMA ANALÍTICO**

### **1. Introducción a un Grid**

- a. Complejidad de los sistemas distribuidos
- b. Definir la computación en la Nube. Examinar las características de la computación en la Nube
- c. Evaluar la proposición de valor

### **2. Entender la arquitectura de la Nube**

- a. Infraestructura, plataformas, appliance virtuales, protocolos de comunicación, aplicaciones
- b. Conectarse a la Nube
- c. Computación bajo demanda (On-demand Computing)

### **3. Servicios y aplicaciones por tipo**

- a. Infraestructura como servicio (IaaS)
- b. Plataforma como Servicio (PaaS)
- c. Software como Servicio (SaaS)
- d. Autenticación de usuario Grid
- e. Conceptos, arquitecturas y servicios

### **4. Entender la abstracción y la virtualización**

- a. Utilizar tecnologías de virtualización
- b. Equilibrio de carga
- c. La nube de Google
- d. Hipervisores
- e. Trasladar aplicaciones
- f. Grid Middleware

### **5. Planificación de capacidad**

- a. Definir una línea de base y una métrica
- b. Tipos de instancias y servidores
- c. Capacidad de red
- d. Globus

### **6. Administrar la Nube**

- a. Administración de ciclos de vida
- b. Productos de administración de Nube
- c. Estándares de administración de nubes emergentes
- d. Recursos de administración de un Grid

### **7. Entender la seguridad en la Nube**

- a. Asegurar la Nube
- b. Asegurar los datos, cifrado
- c. Establecer identidad y presencia
- d. Web Services Resource Framework (WSRF)

8. Servicios y aplicaciones
  - a. Entender la arquitectura orientada a servicios
  - b. SOA Problemas de rendimiento en un Grid
9. **Trasladar aplicaciones a la Nube**
  - a. Creación de un mapa de funcionalidad
  - b. Atributos de aplicaciones
  - c. Economía en un Grid
10. **Trabajar con el almacenamiento basado en la Nube**
  - a. Aprovisionar el almacenamiento en la Nube
  - b. Explorar las soluciones de copia de seguridad en la Nube
  - c. Interoperabilidad del almacenamiento en la Nube
  - d. Trabajar con software de productividad
  - e. Utilizar aplicaciones de productividad
  - f. Cloud público, privado e híbrido
11. **Fog Computing**
  - a. Edge level
  - b. Core level
  - c. Optimizar recursos y almacenamiento
  - d. Protocolos de comunicación
  - e. Procesamiento de eventos complejos
  - f. Despliegue de tecnologías en Fog Computing
  - g. Internet of things - Fog Computing

## **BIBLIOGRAFÍA**

1. ¿Qué es la Nube? El futuro de los sistemas de información – Barrie Sosinsky – Anaya -Wiley
2. Computación en la nube. Estrategias de Cloud Computing en las Empresas. Luis Joyanes Aguilar (2012)
3. Cloud Computing: Principles and Paradigms. Rajkumar Buyya
4. Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS) by Michael J. Kavis (2014)
5. Cloud Computing for Programmers: Software Development in the Age of Cloud by D. Casal (2014)
6. Cloud Computing: Concepts, Technology & Architecture (The Prentice Hall Service Technology Series from Thomas... by Thomas Erl, Ricardo Puttini and Zaigham Mahmood (2013)
7. OpenStack Cloud Computing Cookbook - Second Edition by Kevin Jackson and Cody Bunch (2013)
8. Advances in Cloud Computing Research by Muthu Ramachandran (2014)

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ROBÓTICA

CÓDIGO: CC432

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC721 - INTELIGENCIA ARTIFICIAL
- CC731 - COMPUTACIÓN GRÁFICA

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Este curso presentará a los estudiantes las limitaciones, tecnologías y algoritmos fundamentales de la robótica autónoma. El enfoque se centrará en los aspectos computacionales de los robots móviles con ruedas autónomos. Los temas más importantes serán movilidad, percepción y navegación.

### **COMPETENCIAS**

- Conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas
- Aplicar los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes
- Desarrollar técnicas de aprendizaje computacional y diseñar e implementar aplicaciones y sistemas dedicados a extracción automática de información y conocimiento a partir de grandes volúmenes de datos

### **PROGRAMA ANALÍTICO**

1. Introducción a la robótica autónoma
  - a. Tipos de robots
  - b. Partes del robot
  - c. Sensores
  - d. Potencia
  - e. Control
  - f. Inteligencia
  - g. Comunicación
2. **Movilidad**

- a. Métodos de locomoción
  - b. Conceptos de locomoción
  - c. Cinemática de robots móviles
  - d. Sistemas de control simple
3. **Detección y percepción**
- a. Sensor y tecnologías de medición
  - b. Robot Vision
  - c. Incertidumbre de modelado
4. **Localización y Planificación de trayectorias**
- a. Detección de la posición de un robot
  - b. Representación de creencias y modelo de error de odometría
  - c. Localización basada en mapas probabilísticos y localización de Markov
  - d. Métodos de localización de Montecarlo
  - e. Métodos de localización del filtro de Kalman
5. **Robots autónomos y navegación**
- a. Evitación de obstáculos
  - b. Planificación de ruta
  - c. Localización y mapeo simultáneos (SLAM)

## **BIBLIOGRAFÍA**

1. Robótica, Control, Detección, Visión e Inteligencia; Fu, K.S., González, R.C. y Lee, C.S.G. Mc Graw-Hill, 1988
2. Ollero, A. Robótica, Manipuladores y Robots Móviles. Marcombo, 2002
3. Sensors for mobile robots. Theory and application. H.R. Everett. A.K. Peters. Wellesley, 1995
4. Introduction to Robotics. P.J. McKerrow. Addison-Wesley, 1991
5. Introducción a la robótica. Principios teóricos, construcción y programación de un robot educativo. J.M. Angulo Usategui, S. Romero, I. A. Martínez. Ed. Thomson, 2005
6. Fundamentos de Robótica A. Barrientos, L.F. Peñin, C. Balaguer, R. Aracil. Mc. Graw Hill, 1997
7. Robots y Sistemas sensoriales. Fernando Torres, Jorge Pomares y otros. Prentice Hall, 2002
8. Robot motion planning. J.C. Latombe. Kluwer Academic Publishers, 1991
9. Introductory Computer Vision and Image Processing. A. Low. Mc. Graw-Hill, 1991
10. Visión por computador: imágenes digitales y aplicaciones. 2ª edición. G. Pajares y J. M. de la Cruz. RA-MA, 2008
11. Ejercicios resueltos de visión por computador. G. Pajares y J. M. de la Cruz. RA-MA, 2007

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: MINERÍA DE DATOS

CÓDIGO: CC842

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC721 - INTELIGENCIA ARTIFICIAL

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

La asignatura se ubica en la intensificación de Computación, en el ámbito de las asignaturas de la Inteligencia Artificial. La Minería de datos, y el aprendizaje automático en general, están ligados al campo de la estadística y de la algorítmica, y abordan las técnicas para la extracción de conocimiento implícito en conjuntos de datos.

La minería de datos es una herramienta poderosa para el proceso de descubrir nuevas correlaciones, patrones y tendencias mediante el análisis de grandes cantidades de datos almacenados en repositorios. En el curso, los estudiantes aprenden a aplicar los principios de data mining para manejar y analizar conjuntos grandes y complejos de datos, incluyendo aquellos que se encuentran en la web.

### **COMPETENCIAS**

- Detección, interpretación y predicción de patrones cuantitativos y cualitativos en los datos
- Proceso de extraer información o patrones interesantes (no triviales, implícitos, previamente desconocidos y potencialmente útiles) desde grandes repositorios
- Comprender los conocimientos base de datos y los principios de la ciencia de la computación
- Decidir ante un problema práctico concreto qué tarea de minería de datos conviene utilizar, qué modelo se quiere obtener, qué técnica resultaría más adecuada de utilizar y cómo evaluar los resultados obtenidos

### **PROGRAMA ANALÍTICO**

1. Introducción a la minería de datos
2. Aprendizaje automático
3. DBMS
4. OLAP
5. Estadísticas

6. Objetivos de minería de datos
7. Etapas del proceso de minería de datos
8. Técnicas de minería de datos
9. Métodos de representación del conocimiento
10. Almacén de datos y OLAP
11. Data Warehouse y DBMS
12. Modelo de datos multidimensionales
13. Preprocesamiento de datos
14. Limpieza de datos
15. Transformación de datos
16. Reducción de datos Discretización y generación de jerarquías conceptuales
17. Algoritmos de minería de datos
18. Reglas de asociación, clasificación y predicción
19. Regresión Lineal
20. Clasificación estadística (bayesiana)
21. Redes bayesianas
22. Modelos lineales
23. Modelos no lineales
24. Métodos nearest neighbor
25. Redes neuronales
26. Máquinas de vectores soporte
27. Descubrimiento de asociaciones
28. Detección de anomalías
29. Cluster
30. Minería de datos en textos: Naive Bayes multinomial
31. Minería de datos en grafos y web: Page rank

## **BIBLIOGRAFÍA**

1. Basilio Sierra. Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software weka. Prentice-Hall. 9788483223185. 2006
2. García, Salvador, Luengo, Julián, Herrera, Francisco. Data Preprocessing in Data Mining. Springer. 978-3-319-10246-7 2015
3. José Hernández Orallo, M.José Ramírez Quintana, Cèsar Ferri Ramírez. INTRODUCCIÓN A LA MINERÍA DE DATOS. Pearson. 84 205 4091 9. 2004
4. Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to Data Mining. Addison Wesley. Longman Publishing Co. 0321321367. 2005
5. Witten, Frank & Hall. Data Mining: Practical Machine Learning Tools and Techniques. Morgan & Kauffmann. 978-0-12-374856-0. 2011
6. Xindong Wu, Vipin Kumar The Top Ten Algorithms in Data Mining Chapman and Hall/CRC. 9781420089646. 2009
7. D. Hand, P. Smyth, H. Mannila. Principles of DataMining. MIT press. 2001
8. D. Pyle. Data preparation for data mining. Morgan Kauffmann. 1999
9. Ian H. Witten y Eibe Frank Data Mining: Practical machine tools and techniques. Elsevier. 2005. 10. J. Hernández Orallo, M.J. Ramírez Quintana, C. Ferri Ramírez. Introducción a la minería de datos. Pearson. 2003
10. Jiawei Han y Micheline Kamber Data Mining: Concepts and techniques. Morgan Kauffmann. 2006
11. M. Berry, G. Linoff. Data mining techniques for marketing, sales and customer support. Wiley. 1997



12. Varios autores. Data Mining: Know it all. Morgan Kauffmann. 2009

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: PROGRAMACIÓN EVOLUTIVA

CÓDIGO: CC921

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC832 - ROBÓTICA

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

La computación evolutiva es una de las ramas de la Inteligencia Artificial que se aplica para la resolución de problemas de optimización combinatorial. Estas técnicas están inspiradas en los mecanismos de evolución biológica propuestos por Darwin, Medel y Lamark. Sin entrar mucho en detalle sobre los estudios que hicieron estos científicos, solo vamos a mencionar brevemente lo que propusieron. Darwin propuso la “Selección natural de los más adaptados”, Medel propuso la “Teoría corpuscular de la herencia” y Lamark propuso la “Herencia de caracteres adquiridos”.

En este curso se aprenderá la importancia de estas técnicas, entre las que se encuentran, algoritmos genéticos, programación evolutiva, algoritmos bioinspirados o programación de enjambres, entre otras.

### **COMPETENCIAS**

- Aprender los fundamentos de la programación evolutiva y su complejidad basada en metaheurísticas
- Aplicar las técnicas de computación evolutiva a problemas reales
- Comprender la potencialidad de la técnica y sus limitaciones más importantes
- Comparar a los algoritmos genéticos con otras técnicas de optimización y búsqueda de soluciones evaluando la calidad y performance

### **PROGRAMA ANALÍTICO**

1. **Introducción a la programación evolutiva**
2. **Conceptualización dentro de la programación evolutiva**
3. **Técnicas de selección**
4. **Técnicas de cruce**
5. **Mutación y ajuste de parámetros**
6. **Manejo de restricciones y operadores avanzados**
7. **Algoritmos basados en enjambres**
8. **Programación de expresión del gen (GEP)**

## **BIBLIOGRAFÍA**

1. A.E. Eiben & J.E.Smith. Introduction to Evolutionary Computing Springer, 2nd printing, 2007. ISBN 3540401849
2. Carlos Artemio Coello Coello. Introducción a la Computación Evolutiva. CINVESTAV IPN, 2014. <http://delta.cs.cinvestav.mx/~ccoello/compevol/apuntes.pdf>
3. Melanie Mitchell. An Introduction to Genetic Algorithms MIT Press, 1996. ISBN 0262133164
4. David E. Goldberg. Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley, 1989. ISBN 0201157675

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ANÁLISIS EN MACRODATOS

CÓDIGO: CC931

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC424 - INFRAESTRUCTURAS DE COMPUTACIÓN
- CC421 - MINERÍA DE DATOS

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

En la actualidad se ha encontrado el problema principal de cómo gestionar grandes volúmenes de datos de manera eficiente. Para ellos se han desarrollado nuevas tecnologías hardware y software para poder gestionar grandes volúmenes de datos y poder analizarlos con el fin de obtener la información relevante que nos permita interpretar mejor la realidad y tomar decisiones es lo que conocemos como Big Data.

A tenor del fenómeno Big Data han surgido otras áreas de conocimiento especializadas como Social Business Intelligence, Text Mining, o Cloud Computing, y otras existentes han experimentado un importante desarrollo como Data Mining, Knowledge Discovery for Big Data, Business Intelligence o Data Science.

### **COMPETENCIAS**

#### **PROGRAMA ANALÍTICO**

##### **1. Introducción al Big Data**

- a. Gestión y análisis y visualización del Big Data
- b. Bases de datos NoSQL
- c. Escalabilidad, eficiencia y efectividad
- d. Arquitectura de BBDD e independencia de datos
- e. Large small data

##### **2. Aprendizaje automático en grandes volúmenes de datos**

- a. Función de pérdida (Loss function)
- b. Estrategias de validación (leaving-one-out, cross-validation)
- c. Clustering paramétrico y no paramétrico
- d. Deep Learning: RBM, Auto-encoders, DBN/DNN
- e. Learning from Data Streams

##### **3. Aplicaciones con datos intensivos**

- a. Modelo de programación en Hadoop Mapreduce
  - b. Aplicaciones en el paradigma Mapreduce
  - c. Algoritmos en grandes volúmenes
4. **Visualización de datos**
- a. Visualización de datos crudos
  - b. Representaciones gráficas de datos procesados
  - c. Tipos de visualización de datos según las necesidades del análisis
5. **Herramientas de última generación**
- a. Pig Latin y Funciones definidas por el usuario
  - b. Scala
  - c. Apache Spark
  - d. GraphX
  - e. Bagel
  - f. Spark SQL
  - g. MLlib
6. **Big Data y Tiempo Real**
- a. Arquitecturas de propósito general: combinar procesamiento en lotes y stream analytics
  - b. Arquitecturas Lambda
  - c. Lambdoop: middleware y framework de desarrollo sobre una arquitectura Lambda

## **BIBLIOGRAFÍA**

1. Data-intensive supercomputing: The case for disc. Technical report, Carnegie Mellon University, School of Computer Science
2. Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. Commun. ACM , 51(1):107
3. Hey, T., Tansley, S., and Tolle, K., editors (2009). The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, Redmond, Washington

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ALGORITMOS BIOINSPIRADOS

CÓDIGO: CC023

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC632 - PROGRAMACIÓN PARALELA

CONDICIÓN: OBLIGATORIO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Este curso trata de algoritmos que se inspiran en fenómenos naturales y los aplica a problemas de optimización, diseño y aprendizaje. La atención se centra en el proceso de abstracción de algoritmos del fenómeno observado, su análisis de resultados y comparación. Este enfoque se hará principalmente a través de la computación evolutiva, inteligencia de enjambre (colonia de hormigas y métodos basados en partículas) y redes neuronales.

### **COMPETENCIAS**

- La computación BioInspirada es un campo dedicado a abordar problemas complejos utilizando métodos computacionales basados en los principios de diseño encontrados en la naturaleza. Este curso está fuertemente basado en los fundamentos de los sistemas complejos y la biología teórica
- El objetivo es una comprensión profunda de las arquitecturas distribuidas de los sistemas complejos naturales, y cómo se pueden usar para producir herramientas informáticas con mayor robustez, escalabilidad y flexibilidad, y que puedan interactuar de manera más efectiva con los humanos
- Es un campo multidisciplinario fuertemente basado en la biología, la complejidad, la informática, la informática, la ciencia cognitiva, la robótica y la cibernética

### **PROGRAMA ANALÍTICO**

#### **1. Computación Basada en Modelos Naturales**

- 2. Algoritmos evolutivos**
- 3. Redes neuronales**
- 4. Algoritmos inmunológicos**
- 5. Algoritmos basados en enjambres (swarm intelligence)**
- 6. Algoritmos basados en colonias de hormigas**
- 7. Algoritmos Clustering de hormigas**
- 8. Optimización Basada en Nubes de Partículas (Particle Swarm Optimization)**
- 9. Mapas de autoorganización**
- 10. Aprendizaje profundo**

## **BIBLIOGRAFÍA**

1. Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, L. N. de Castro (2006), CRC Press
2. Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies, D. Floreano and C. Mattiussi (2008), MIT Press
3. Evolutionary Optimization Algorithms, D. Simon (2013), Wiley

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: ALGORITMOS Y ESTRUCTURAS DE DATOS AVANZADAS

CÓDIGO: CC020

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC531 - ANÁLISIS Y DISEÑO DE ALGORITMOS

CONDICIÓN: ELECTIVO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

En este curso se exponen temas avanzados de algoritmos y estructura de datos que sirven como continuación del curso de programación orientada a objetos pero centrado en la programación de estructuras de datos simples y complejas. Por tanto, los aspectos que se estudiarán serán principalmente cómo se definen y cómo se implementan las estructuras de datos e incorporarlos a cualquier entorno de programación.

Así mismo, como puede observarse, este curso continúa con la formación de un programador en cualquier ámbito o especialidad, continuando posteriormente en el curso de “Análisis y diseño de algoritmos”, donde se verá algoritmia y optimización en estructuras de datos complejas.

### **COMPETENCIAS**

- Evalúa las propiedades de las estructuras de datos estableciendo su utilidad en las aplicaciones en ciencia e ingeniería comprometiéndose en el uso adecuado que conlleve a resolver problemas de la vida real
- Demuestra su capacidad de análisis diseñando, implementando estructuras que permitan dar solución, trabajando en equipo en los diversos problemas en la ciencia e ingeniería
- Comprende los conocimientos básicos de Algorítmica y ejecuta los principios de esta ciencia de la computación
- Describe los procesos de construcción de Algoritmos identificando las propiedades asociadas que se generan de ellos apreciando su influencia en la ciencia e ingeniería

### **PROGRAMA ANALÍTICO**



1. Definición de problemas básicos de estructura de datos avanzados: fibonacci Heaps, Van Emde Boas Priority Queues
2. Estructuras de Datos Dinámicos
3. Descripción de Algoritmos para Cadenas con el tema del algoritmo Rabin Karp Fingerprinting
4. Flujo Máximo con los temas de Flujo de costo mínimo y Bipartite Matching
5. Programación Lineal
6. Formulación de problemas de programación lineal
7. Dualidad
8. Simplex
9. Punto Interior
10. Algoritmo Ellipsoid
11. Algoritmos en Línea
12. Ski Rental
13. Problemas de River Search
14. El problema de los k-server
15. Algoritmos de Aproximación
16. NP-Hardness
17. Algoritmos de aproximación Greedy
18. Programación Dinámica
19. Geometría Computacional
20. Convex Hull
21. Line-Intersection
22. Sweep Lines

## **BIBLIOGRAFÍA**

1. Shaffer, Clifford A.: Data Structures and Algorithm Analysis in C++, Third Edition, Dover Publications, 2013
2. Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J.A.: Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos. 2ª Edición, Ed. Garceta, 2013
3. Joyanes, L., Zahonero, I., Fernández, M. y Sánchez, L.: Estructura de datos. Libro de problemas, McGraw Hill, 1999
4. Campos Laclaustra, J.: Estructuras de Datos y Algoritmos, Prensas Universitarias de Zaragoza, Colección Textos Docentes, 1995

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: AUDITORÍA DE SISTEMAS

CÓDIGO: CC017

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC016 - CALIDAD DE SOFTWARE

CONDICIÓN: ELECTIVO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

La asignatura Auditoría de Sistemas es de naturaleza es de tipo teórico-práctico en un proceso de auditoría de sistemas.

En esta asignatura se estimula el estudio, trabajo e investigación individual y en equipo, así como la rotación de roles de los estudiantes para completar su formación profesional dentro del contexto científico y tecnológico actual.

### **COMPETENCIAS**

- Desarrollar en los estudiantes su capacidad crítica y evaluadora
- Dar a conocer los sistemas de control que existen en las empresas que hacen uso intensivo de las tecnologías de información
- Evaluar la operatividad de los sistemas computarizados, el uso de recursos de confiabilidad de la información y la seguridad que debe existir en la instalación
- Conocer el software de auditoría existente en el mercado, sus aplicaciones y formas de utilización
- Elaborar los dictámenes de auditoría para que pueda realizar funciones de auditoría de sistemas

### **PROGRAMA ANALÍTICO**

1. Introducción a la Auditoría
  - a. Similitudes y diferencias de los conceptos: evaluación, control y supervisión, aplicaciones prácticas del uso de estos conceptos
  - b. Tipos y Formas de auditorías
  - c. Auditoría de Sistemas: conceptos, objetivos y ámbito de trabajo
  - d. Clases de Auditoría de Sistemas

- e. Revisión de los conceptos de sistemas, teoría de sistemas y el enfoque integral de la auditoría
  - f. Desarrollo de proyectos de innovación tecnológica
  - g. Organización, administración y funcionamiento de una empresa computarizada
  - h. Sistemas de control: conceptos y tipos
- 2. Sistema de Control de Tecnologías de Información
  - a. Etapas y procesos de la auditoría
  - b. Estructura del Sistema de Control de Tecnologías de Información
  - c. Presentación de las Normas a utilizar en el desarrollo del curso
  - d. Presentación y exploración de las Normas a utilizar en Internet
  - e. Herramientas de software en cada proceso de auditoría
- 3. Examen de gestión de TI: Gerencia de Sistemas y de los servicios informáticos
  - a. Controles organizacionales y Controles administrativos
  - b. Controles operativos al uso de los recursos de cómputo
  - c. Controles operativos al uso de la información
  - d. Controles operativos a la confiabilidad de la información
- 4. Examen de Seguridad Informática
  - a. Revisión de los conceptos de Seguridad informática y de las normas existentes
  - b. Aplicación de Norma NTP INDECOPI/ISO17799
- 5. Examen de Sistemas Aplicativos
  - a. Controles de entrada y salida
  - b. Controles de proceso
  - c. Controles de librería
  - d. Controles de acceso: lógico y físico
  - e. Análisis del sistema y de los procedimientos operativos
  - f. Puntos fuertes y débiles de los controles internos y del sistema de seguridad
  - g. Determinación de Criterios a aplicar
  - h. Recomendaciones y análisis comparativo entre grupos
- 6. Examen de Proyectos Informáticos
  - a. Ciclo de vida de un proyecto
  - b. Controles de la fase de planeamiento
  - c. Controles de la fase de ejecución
  - d. Controles de la fase de implantación o lanzamiento
  - e. Controles administrativos y de seguimiento
  - f. Determinación de Criterios a aplicar
- 7. Modelo Cobit
  - a. Normas de Gestión TI
  - b. Dominios y Puntos de Control
  - c. Diferencia de aplicación con la metodología seguida
- 8. Examen de Aseguramiento de Calidad
  - a. Nuevo enfoque de auditoría preventiva que permite el Aseguramiento de la calidad de los sistemas de información
  - b. Aplicación de Norma ISO

## BIBLIOGRAFÍA

1. **Soy Aumatell, Cristina**; “Auditoría de la Información”, Editorial UOC, Barcelona, 2003
2. Tecnología de la información. Código de buenas prácticas para la gestión de la seguridad de la información NTP - ISO/IEC 17799:2004
3. Comisión de Reglamentos Técnico y Comerciales –CRT de **INDECOPI**. Presidencia del Consejo de Ministros – Oficina Nacional de Gobierno Electrónico del Perú – ONGEI. 2004
4. **Muñoz Razo, Carlos**; “Auditoría en Sistemas Computacionales”, Universidad del Valle de México, Pearson Educación, 2002

5. COBIT Management Guidelines
6. COBIT Steering Committee and the IT Governance InstituteTM. IT GOVERNANCE INSTITUTE. 2002
7. **Echenique, José Antonio**; “Auditoría en Informática”, Universidad Nacional Autónoma de México. Mc Graw Hill, 2001
8. Metodología para la auditoría de sistemas automatizados **Weights**. 2001. Deuts M. 2000
9. Normas de la **ONGEI** sobre sistemas, seguridad y sistemas de información Oficina Nacional de Gobierno Electrónico e Informática. 2003
10. Normas de Control Interno de la **Contraloría General de la República**. Contraloría General de la República. 1998
11. Normas del **INEI** sobre seguridad y sistemas de información Instituto Nacional de Estadística e Informática. 1995
12. **Piattini, Mario - Del Peso, Emilio**; “Auditoría Informática. Un enfoque práctico”, Computec rama Madrid España 1998

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: BIOLOGÍA COMPUTACIONAL

CÓDIGO : CC021

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CH061 - BIOLOGÍA
- CC622 - MATEMÁTICA COMPUTACIONAL

CONDICIÓN: ELECTIVO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

El presente curso está concebido para estudiantes del noveno semestre. Esta asignatura ofrece una introducción general de los principios, técnicas y herramientas de la biología computacional. Los tópicos que toca el curso incluye principios y métodos necesarios para el estudio y análisis del alineamiento de secuencias de ADN y de aminoácidos, la aplicación de métodos estadísticos y probabilísticos en el análisis de información biológica, el análisis y modelamiento de árboles filogenéticos, el mapeo de secuencias, y una introducción al estudio de la estructura de las proteínas.

### **COMPETENCIAS**

- Aplica los conocimientos de biología, computación y matemáticas apropiadas para el análisis y generación de información utilizadas en el campo de la biología computacional
- Demuestra su capacidad de análisis identificando y definiendo los requerimientos computacionales apropiados para la solución de problemas reales
- Utiliza técnicas y herramientas adecuadas para el análisis y generación de información bioinformática

### **PROGRAMA ANALÍTICO**

#### **1. Introducción a la Biología Computacional**

- a. Conceptos básicos de Biología Molecular
- b. Problemas clásicos de Bioinformática
- c. Herramientas de recolección y almacenamiento de secuencias de laboratorio
- d. Recursos de Software, introducción a BLAST, 43

- e. Omega
- f. Cadenas, Grafos y algoritmos
- 2. Alineamiento de Secuencias**
  - a. Introducción al alineamiento de secuencias
  - b. comparación de pares de secuencias
  - c. Alineamiento de secuencias Global
  - d. Alineamiento de secuencias múltiples
  - e. Cadenas Ocultas de Markov
  - f. Métodos exactos, aproximados y heurísticos del alineamiento de secuencias
  - g. Problemas derivados del alineamiento de secuencias
- 3. Clustering**
  - a. El Problema del Clustering
  - b. Clustering Jerárquico
  - c. Algoritmo Neighbor Joining
  - d. Algoritmo Average Linkage
  - e. Clustering no Jerárquico o K- Means
- 4. Árboles Filogenéticos**
  - a. Introducción a la Filogenia
  - b. Algoritmos Comunes
  - c. Aplicaciones Biológicas
  - d. Algoritmos Exactos
  - e. Algoritmos Probabilísticos
- 5. Mapeo de Secuencias**
  - a. El Problema de Double Digest y Partial Digest
  - b. Técnicas utilizadas en el mapeo de secuencias
  - c. Mapeo con non – Unique Probes
  - d. Mapeo con Unique Probes
  - e. Grafos de Intervalos
  - f. Mapeo con señales de frecuencias de restricción
- 6. Introducción a la estructura de proteínas**
  - a. Fundamentos biológicos de las proteínas
  - b. Motivación para la predicción de estructuras de las proteínas
  - c. Alineamiento rígido de proteínas
  - d. Técnica de alineamiento por Hashing geométrico

## **BIBLIOGRAFÍA**

1. Clote, P. and Backofen, R. Computational Molecular Biology, An Introduction. Wiley, 2000
2. Jones, N. A. and Pevzner, P. A. An Introduction to Bioinformatics Algorithms. The MIT Press. 2004
3. Mount, D. W. Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, 2001

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: CALIDAD DE SOFTWARE

CÓDIGO : CC016

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC642 - DESARROLLO DE SOFTWARE

CONDICIÓN: ELECTIVO

HORAS POR SEMANA: 04 (TEORÍA: 02, LABORATORIO: 02)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

La asignatura Calidad de Software es de tipo teórico-taller en un proceso de desarrollo de software. En esta asignatura se estimula el estudio, trabajo e investigación individual y en equipo, así como la rotación de roles de los estudiantes para completar su formación profesional dentro del contexto científico y tecnológico actual. En el curso se tratarán los siguientes contenidos: Fundamentos de la Calidad de Software, Modelos de Procesos de Software, Aseguramiento de Calidad del Software, Calidad de información, Estándares y Normas de Software.

### **COMPETENCIAS**

- Lograr una visión detallada de los conceptos, herramientas y métodos de calidad en el ciclo de vida del desarrollo de software
- Desarrollar habilidades para realizar el aseguramiento de la calidad de software en el ciclo de vida del desarrollo de software
- Trabajar en equipo: Participar, analizar y dirigir una etapa de un proyecto de software y conocer los roles y procedimientos de acuerdo a las mejores prácticas nacionales e internacionales en materia de calidad de software
- Evaluar la calidad de los resultados del proyecto de software y diseñar el proceso de sostenibilidad del proyecto

### **PROGRAMA ANALÍTICO**

1. **Fundamentos de la Calidad de Software**
  - a. La calidad del software
  - b. Ingeniería de software y calidad
  - c. Estándares y modelos de evaluación y mejora de los procesos software
  - d. Introducción a la calidad total
  - e. Costos de errores de SW

- f. Entorno de la Calidad de SW
- 2. **Modelos de procesos de software - Gestión de la calidad**
  - a. Introducción en la gestión de la calidad: planificar, organizar, gestionar el personal, liderar, y controlar
  - b. Gestión de riesgos: definición, identificación, análisis, planificación, y monitoreo
  - c. Gestión de proyectos de SW
  - d. Plan de desarrollo de SW
- 3. **Procesos y Métodos de calidad de software**
  - a. Factores que Determinan la Calidad de Software (Mc Call)
  - b. Situación Actual de la Industria del Software
  - c. Estándares y Modelos de Evaluación y Mejora de los Procesos de Software
  - d. ISO 9000 (Objetivos, Principios)
  - e. ISO 9126:2001 Modelo de Calidad de Software (Externa e Interna)
  - f. ISO/IEC 15504 (SPICE): Componentes, Dimensiones, Evaluación de Atributos y Certificación
- 4. **Métricas de calidad de software**
  - a. Teoría y Ontología de la medición
  - b. Estándares y metodologías
  - c. Método de definición
  - d. Métricas de software
  - e. PSP (Personal Software Process)
  - f. Planeación en PSP. Métricas del PSP
- 5. **Configuración de Software**
  - a. Línea de producto
  - b. Áreas de maduración técnica: ingeniería de SW, gestión técnica, y gestión de la empresa
  - c. Gestión de la configuración de software (SCM)
  - d. Planeamiento de la SCM
- 6. **Aseguramiento de Calidad de Software**
  - a. Organización
  - b. Actividades
  - c. Plan SQA
  - d. Implementación SQA
- 7. **El CMMI**
  - a. Modelo CMMI: Foco, ¿Qué es y qué no es CMMI?
  - b. Cómo utilizar CMMI
  - c. Conceptos claves: Capacidad, Desempeño, Madurez e Institucionalización
  - d. Análisis de costos y beneficios de la implementación del modelo CMMI
  - e. Estructura y representación del modelo CMMI
  - f. Áreas de procesos, Metas y Prácticas Específicas de CMMI
  - g. Verificación y Calidad
- 8. **El TMMi**
  - a. El proceso de prueba
  - b. EL TPI. TMMi versus TPI
  - c. Niveles del TMMi
  - d. Modelos de Mejora de Procesos
  - e. Prioridades en TI
  - f. ¿Qué es y qué no es ITIL?
  - g. ITIL: Características, Objetivos, Fundamentos, Mejora Progresiva, Áreas Claves
  - h. Relación con Otras Metodologías
  - i. Aspectos Importantes
  - j. Organización y Componentes de una Organización



- k. Perspectivas
- l. Implementación de ITIL
- 9. **Calidad de Información**
  - a. Aproximación a la calidad
  - b. Medición de los modelos
  - c. Calidad de datos
  - d. Aspectos de gestión
  - e. Estudio de casos prácticos
- 10. Estándares y normas de Software
  - a. Modelo de calidad NTP- ISO/IEC 12207 – Procesos del ciclo de vida del software
  - b. Procesos principales, procesos de apoyo, procesos organizativos
  - c. Estrategias de Implementación
  - d. Planes de calidad de software
  - e. Modelo de calidad NTP- ISO/IEC 15504 – Evaluación y Mejora Continua del software

## **BIBLIOGRAFÍA**

1. **SOMMERVILLE, Ian**; “Ingeniería de Software”, 9na. Edición, Addison Wesley, México, 2011
2. **PIATTINI, Miguel & al**; “Calidad de Sistemas Informáticos”; 2da. Edición, RA-MA Editorial, Madrid, 2011
3. **PRESSMAN, Roger**; “Ingeniería de Software: Un Enfoque Práctico”; 7ma. Edición, McGraw-Hill, Madrid, 2010
4. NTP-ISO/IEC 12207:2004 Tecnología de la Información. Procesos del ciclo de vida del software, 2004
5. Guía Técnica sobre evaluación de software para la Administración Pública, 2004
6. **Software Engineering Institute - Carnegie Mellon University - (CMU SEI)**. Capability Maturity Model Integration (CMMI) version 1.2, 2007
7. **TMMi Foundation** Test Maturity Model Integration (TMMi) Version 1.0 (dd. February, 17th 2008)

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: COMPILADORES

CÓDIGO: CC008

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS:

- CC521 - TEORIA DE AUTÓMATAS, LENGUAJES Y COMPUTACIÓN

CONDICIÓN: ELECTIVO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO: 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

Conocer las técnicas básicas empleadas durante el proceso de generación intermedio, optimización y generación de código. Aprender a implementar pequeños compiladores.

### **COMPETENCIAS**

- Comprende los conceptos y principios fundamentales de los compiladores
- Reconoce las técnicas básicas para construir un compilador
- Diseña compiladores
- Implementa pequeños compiladores

### **PROGRAMA ANALÍTICO**

#### **1. Visión General de los Lenguajes de Programación**

- a. Historia de los lenguajes de programación
- b. Breve revisión de los paradigmas de programación
- c. Lenguajes procedurales
- d. Lenguajes orientados a objetos
- e. Lenguajes funcionales
- f. Lenguajes declarativos y no algorítmicos
- g. Lenguajes de *scripts*
- h. Los efectos de la escalabilidad en las metodologías de programación

## **2. Introducción a la Traducción de Lenguajes**

- a. Comparación entre intérpretes y compiladores
- b. Fases de traducción del lenguaje (análisis léxico, análisis sintáctico, generación de código, optimización)
- c. Aspectos de traducción dependientes e independientes de la máquina

## **3. Sistemas de Traducción del Lenguaje**

- a. Aplicación de las expresiones regulares en analizadores léxicos
- b. Análisis sintáctico (sintaxis concreta y abstracta, árboles de sintaxis abstracta)
- c. Aplicación de las gramáticas libres de contexto en un parseo dirigido por tablas o recursivo descendente
- d. Administración de tablas de símbolos
- e. Generación de código por seguimiento de un árbol
- f. Operaciones específicas de la arquitectura: selección de instrucciones y asignación de registros
- g. Técnicas de optimización
- h. El uso de herramientas como soporte en el proceso de traducción y las ventajas de éste
- i. Librerías de programas y compilación separada
- j. Construcción de herramientas dirigidas por la sintaxis

## **BIBLIOGRAFÍA**

1. Aho, A. Compiladores Principios, técnicas y herramientas. Addison Wesley. 1990
2. Aho, A., Lam, M., Sethi, R., and Ullman, J. D. Compiladores. Principios, técnicas y herramientas. Addison Wesley, 2nd edition. ISBN:10-970-26-1133-4. 2008
3. A.Lemone, K. Fundamentos de Compiladores. CECSA-México. 1996
4. Appel, A. W. Modern compiler implementation in Java. Cambridge University Press, 2.a edición. 2002
5. Louden, K. C. Construcción de Compiladores Principios y Práctica. Publicado por Thomson. 2004
6. Louden, K. C. Lenguajes de Programación. Publicado por Thomson. 2004
7. Pratt, T. W. and V.Zelkowitz, M. Lenguajes de Programacion Diseno e Implementacion. Prentice-Hall Hispanoamericana S.A. 1998
8. Teufel, B. and Schmidt, S. Fundamentos de Compiladores. Addison Wesley Iberoamericana. 1998

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: DEEP LEARNING EN VISIÓN ARTIFICIAL

CÓDIGO : CC004

CRÉDITOS : 04 (CUATRO)

PRE-REQUISITOS:

- CC421 - MINERÍA DE DATOS
- CC422 - COMPUTACIÓN GRÁFICA

CONDICIÓN: ELECTIVO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

En los últimos años, el aprendizaje profundo (deep learning) se ha convertido en una herramienta fundamental en el aprendizaje de máquinas para una amplia variedad de dominios y aplicaciones. En sí se puede decir, que es una técnica que emplea redes neuronales para aprender representaciones a partir de una serie de datos observados, que puedan ser de utilidad para resolver problemas de alto nivel como ser predicción, restauración o clasificación de señales. El aprendizaje profundo pretende descubrir las propiedades intrínsecas de grandes volúmenes de datos construyendo representaciones distribuidas, tanto en contextos supervisados como no supervisados.

### **COMPETENCIAS**

- Ilustrar a los estudiantes los principales aspectos de modelado, algorítmicos y de optimización de forma de que ellos mismos puedan implementar (diseñar, entrenar y validar) sus propios modelos
- Analizar modelos simples basados en redes convolucionales que permiten explicar matemáticamente algunas de las propiedades claves para su éxito
- Ilustrar aplicaciones específicas en los campos de visión artificial y procesamiento de imágenes

### **PROGRAMA ANALÍTICO**

1. Introducción al deep learning
2. Aprendizaje supervisado
  - a. Métodos lineales de clasificación
  - b. Métodos no lineales
  - c. Métodos ensamblados

3. Representaciones de alto nivel, características en imágenes
4. Formulación del aprendizaje como un problema de optimización
  - a. Algoritmo de backpropagation
5. Redes neuronales conectadas
  - a. Multilayer Perceptron (MLP)
  - b. Fase de entrenamiento y validación en imágenes
  - c. Hiperparámetros
  - d. Evaluación del rendimiento de los modelos deep learning
  - e. Entender el comportamiento del modelo durante la fase de entrenamiento
  - f. Reducción de overffiting
6. Redes neuronales de convolución
  - a. Arquitectura
  - b. Capas
  - c. Operadores
7. Análisis de redes neuronales
  - a. Visualización de representaciones
  - b. Propiedades: invarianza, covarianza, redundancia e invertibilidad
  - c. Image augmentation
8. Redes neuronales recurrentes (R-NN)
  - a. Predicción de series de tiempo com MLP
9. Redes de corta-larga memoria (LSTM)
  - a. Predicción de series de tiempo con R NN - LSTM
  - b. Comprensión de R-NN LSTM con estado
10. Redes adversarias generativas (GAN)

## BIBLIOGRAFÍA

1. Goodfellow, I., Bengio, Y., Courville, A. “Deep learning”. MIT Press, 2016
2. Hastie, T, Tibshirani, R. Friedman, J. “The Elements of Statistical Learning”. NY Springer, 2001
3. LeCun, Y., Bengio, Y., Hinton, G. “Deep learning”. Nature 521.7553 (2015): 436-444
4. Mallat, S., “Understanding deep convolutional networks” Phil. Trans. R. Soc. A 374.2065 (2016): 20150203
5. Krizhevsky, A., Sutskever, I., Hinton, G. “Imagenet classification with deep convolutional neural networks”. NIPS, 2012
6. Zeiler, M. D., and Fergus, R. “Visualizing and understanding convolutional networks”. ECCV, 2014
7. Li, Fei-Fei, L., Karpathy, A., Johnson, J. “CS231n: Convolutional neural networks for visual recognition”. Stanford University, notas de curso, 2015
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. “Generative adversarial nets”. NIPS, 2014
9. Anh, N., Yosinski, J., Clune, J. “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.” CVPR, 2015
10. Johnson, J., Alahi, A., Fei-Fei, L. “Perceptual losses for real-time style transfer and super-resolution”. ECCV. 2016

UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## **SÍLABO**

### **INFORMACIÓN GENERAL**

ASIGNATURA: DISEÑO DE SISTEMAS BASADOS EN MICROPROCESADOR

CÓDIGO: CC010

CRÉDITOS: 04 (CUATRO)

PRE-REQUISITOS: CC511 - COMPUTACIÓN CENTRADA EN REDES

CONDICIÓN: ELECTIVO

HORAS POR SEMANA: 06 (TEORÍA: 02, LABORATORIO 04)

SISTEMA DE EVALUACIÓN: G

### **SUMILLA**

En este curso se aprende el modelo de programación de bajo nivel de los sistemas digitales basados en microprocesador. En concreto, se estudian y analizan los conceptos básicos relacionados con los recursos de programación de bajo nivel, así como la integración de programas desarrollados como mezcla de lenguajes de medio nivel (lenguaje C) y bajo nivel (ensamblador). Se analizan las distintas estrategias de programación de los recursos del subsistema de entrada y salida, haciendo énfasis en la programación de los controladores de dispositivo más importantes. Todo ello se complementa con el estudio y análisis de los buses e interfaces de entrada y salida.

### **COMPETENCIAS**

- Diseña y escribe programas en lenguaje ensamblador del 80x86
- Diseña y escribe programas utilizando las interrupciones del 80x86
- Diseña y escribe programas combinando lenguaje ensamblador y lenguaje C (medio nivel)
- Utilizar recursos software proporcionados por la BIOS y el Sistema Operativo
- Diseña y escribe programas residentes en memoria (drivers)
- Programa los recursos hardware básicos de E/S del PC
- Utiliza un entorno de desarrollo y depuración de bajo nivel

### **PROGRAMA ANALÍTICO**

1. **Sistemas digitales basados en microprocesador**
  - a. **Modelo De Programación Del 80x86 De Intel**
  - b. Arquitectura básica de un sistema digital basado en microprocesador
  - c. Arquitectura básica de un microprocesador
  - d. Funcionamiento de un sistema basado en microprocesador
  - e. Familia 80x86 como caso particular
  - f. Registros internos y arquitectura del 80x86

- g. Acceso y organización de la memoria
  - h. Modos de direccionamiento
  - i. Directivas y operadores del ensamblador del 80x86
  - j. Estructura de un programa en ensamblador
  - k. Instrucciones del ensamblador: transferencia de datos, aritméticas y lógicas, de control, de interrupción, etc
  - l. Mapa de Memoria del sistema PC
  - m. Interrupciones: mecanismo y vectores de interrupción
2. **Interfaz del ensamblador con el lenguaje C**
- a. Distintos modelos del lenguaje C
  - b. Convenios de nomenclatura, paso de parámetros, devolución de resultados
  - c. Interrupciones BIOS
  - d. Interrupciones DOS
  - e. Ejecución de programas desde el DOS
  - f. PSP (Prefijo de Segmento de Programa)
  - g. Tipos de programas: EXE, COM, y residentes (TSR)
3. **Entrada/Salida**
- a. Técnicas de programación de entradas y salidas (E/S)
  - b. Sondeo
  - c. Interrupción
  - d. DMA
  - e. Gestión y programación de las interrupciones en el 80x86: el controlador programable de interrupciones 8259A
4. **PROGRAMACIÓN DE LOS RECURSOS HARDWARE BÁSICOS DEL PC**
- a. Teclado
  - b. Timer
  - c. Reloj de Tiempo Real (RTC)
  - d. Controladora de Vídeo y Pantalla
  - e. Puerto Paralelo
  - f. Impresora

## **BIBLIOGRAFÍA**

- 1. Los microprocesadores Intel. Barry B. Brey, Ed. Prentice-Hall
- 2. IBM PC & XT, Assembly Language. Leo. J. Scalon, Ed. Brad