

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



PLAN DE ESTUDIOS 2018

1. ANTECEDENTES DE LA FACULTAD DE CIENCIAS Y DE LA CARRERA DE CIENCIA DE LA COMPUTACIÓN

El 06 de agosto de 1960, promulgada la Ley Universitaria No. 13417, se creó la Facultad de Ciencias Físicas y Matemáticas en la UNI, a partir de la entonces Facultad de Ciencias Básicas y Nucleares. El año 1962 se crea el Instituto de Matemáticas Puras y Aplicadas (IMUNI).

2. FINES

El presente plan de estudios tiene la finalidad de reorganizar, actualizar e incorporar nuevos temas de enseñanza aprendizaje a la carrera de Ciencia de la Computación con el propósito de formar, en sus futuros egresados, las competencias necesarias para el logro de desempeños relevantes en el ejercicio de la profesión como licenciados en Ciencia de la Computación logrando un impacto positivo en el desarrollo del país.

La actualización de los planes de estudios de Ciencia de la Computación cumple con las directivas de adecuación de la nueva ley universitaria 30220 y el estatuto de la Universidad.

3. PERFIL DEL INGRESANTE

El ingresante a la Escuela profesional de Ciencia de la Computación debe contar con

- Habilidad para la programación científica
- Creatividad e innovación en la resolución de problemas
- Vocación para el trabajo científico
- Disposición para resolver problemas de la sociedad usando la Ciencia y la Tecnología
- Disposición, tenacidad y perseverancia en el estudio y en el trabajo computacional científico, teórico y experimental.
- Calidad y capacidad de autoaprendizaje.

4. PERFIL DEL EGRESADO

El alumno al egresar debe poseer las siguientes aptitudes y habilidades:

- Impulsar el desarrollo de nuevas técnicas computacionales.
- Generar puestos de empleo a través de la innovación permanente y de la formación de empresas de base tecnológica.
- Analizar un problema e identificar y definir los requerimientos computacionales apropiados para su solución.
- Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas.
- Aplicar la base matemática, principios de algoritmos y la teoría de la

Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales.

- Transformar sus conocimientos del área de Ciencia de la Computación en emprendimientos tecnológicos.
- Facilidad para participar en proyectos interdisciplinarios.
- Analizar, diseñar e implementar aplicaciones, en lo relativo a la eficiencia, correctitud y seguridad de las mismas, en entornos multiusuario y a través de red.
- Diseñar nuevas formas de utilizar computadoras, estudiando algoritmos eficientes que pueden involucrar robótica, bioinformática, redes sociales, entre otros.
- Desarrollar mecanismos efectivos para resolver problemas computacionales, como el almacenamiento masivo de información, el despliegue de imágenes complejas, la comunicación entre diversos sistemas, la aplicación de heurísticas a problemas de alta complejidad computacional.
- Impartir cátedra de temas relacionados con las ciencias de la computación.

5. CAMPO OCUPACIONAL

El campo ocupacional de un profesional de Ciencia de la Computación es amplio y está en continua expansión y cambio.

- Investigación y desarrollo de nuevas tecnologías computacionales: Procesamiento Paralelo y Sistemas Distribuidos, Computación Científica y Cómputo Visual.
- Formación de Empresas de Base Tecnológica
- Enseñanza de la Ciencia de la Computación a nivel universitario
- Soluciones e innovaciones computacionales para empresas
- Servicios informáticos "outsourcing" para empresas nacionales y/o extranjeras.

6. IMPORTANCIA DE LA CARRERA EN LA SOCIEDAD

Los actuales profesionales peruanos del área de computación se dedican a producir software o se integran a las empresas productoras de software. En el Perú, la entidad que agrupa a las empresas dedicadas a la producción de software es la Asociación Peruana de Productores de Software (APESOFT). Esta asociación ha tomado como política principal dedicarse a la producción de software para exportación. Siendo así no tiene sentido preparar a nuestros alumnos sólo para el mercado local o nacional. Nuestros egresados deben estar preparados para desenvolverse en el mundo globalizado que nos ha tocado vivir.

Nuestros futuros profesionales no deben estar orientados a competir en el mercado de trabajo con otros profesionales peruanos del área de computación y sistemas, ya que su orientación debe ser dirigida a crear nuevas empresas de base tecnológica que puedan incrementar las exportaciones de software peruano. Este nuevo perfil está orientado a generar industria innovadora. Si nosotros somos capaces de exportar software competitivo también estaremos en condiciones de atraer nuevas inversiones. Las nuevas inversiones generaría más puestos de empleo bien remunerados y con un costo bajo en relación a otros tipos de industria. Bajo esta perspectiva, podemos afirmar que esta carrera

será un motor que impulsará al desarrollo del país de forma decisiva con una inversión muy baja en relación a otros campos.

7. MISIÓN DE LA ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

- Generar recursos humanos orientados a la creación de nueva tecnología computacional.
- Ser el motor que complemente, impulse y consolide la industria de software peruana con base en la investigación científica y tecnológica en áreas innovadoras.
- Formar profesionales altamente capacitados, poseedores de un conjunto de habilidades y destrezas para la solución de problemas computacionales, que contribuyan al desarrollo de esta área estratégica, con un compromiso social.

8. VISIÓN DE LA ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

Ser una carrera profesional acreditada con estándares internacionales y que cuente con el reconocimiento en función de la calidad y competitividad de sus docentes y egresados.

Ser una carrera que trascienda por la relevancia y pertinencia de sus proyectos de investigación básica y aplicada.

Ser una carrera que comparta y difunda el conocimiento con todos los sectores de la población y contribuya a la solución de los problemas estratégicos de nuestra sociedad.

9. GRADOS Y TÍTULOS

Los estudios de esta especialidad tienen una duración de 5 años y llevan a la obtención del Grado Académico de Bachiller en Ciencias con mención en Ciencia de la Computación y Título Profesional de Licenciado en Ciencia de la Computación.

10. CAMPO OCUPACIONAL

El campo ocupacional de un profesional de Ciencia de la Computación es amplio y está en continua expansión y cambio.

- Investigación y desarrollo de nuevas tecnologías computacionales: Procesamiento Paralelo y Sistemas Distribuidos, Computación Científica y Cómputo Visual.
- Formación de Empresas de Base Tecnológica
- Enseñanza de la Ciencia de la Computación a nivel universitario
- Soluciones e innovaciones computacionales para empresas
- Servicios informáticos "outsourcing" para empresas nacionales y/o extranjeras

11. IMPORTANCIA DE LA CARRERA EN LA SOCIEDAD

Los actuales profesionales peruanos del área de computación se dedican a producir software o se integran a las empresas productoras de software. En el Perú, la entidad que agrupa a las empresas dedicadas a la producción de software es la Asociación Peruana de Productores de Software (APESOFT). Esta asociación ha tomado como política principal dedicarse a la producción de software para exportación. Siendo así no tiene sentido preparar a nuestros alumnos sólo para el mercado local o nacional. Nuestros egresados deben estar preparados para desenvolverse en el mundo globalizado que nos ha tocado vivir.

Nuestros futuros profesionales no deben estar orientados a competir en el mercado de trabajo con otros profesionales peruanos del área de computación y sistemas, ya que su orientación debe ser dirigida a crear nuevas empresas de base tecnológica que puedan incrementar las exportaciones de software peruano. Este nuevo perfil está orientado a generar industria innovadora. Si nosotros somos capaces de exportar software competitivo también estaremos en condiciones de atraer nuevas inversiones. Las nuevas inversiones generaría más puestos de empleo bien remunerados y con un costo bajo en relación a otros tipos de industria. Bajo esta perspectiva, podemos afirmar que esta carrera será un motor que impulsará al desarrollo del país de forma decisiva con una inversión muy baja en relación a otros campos.

B CIENCIA DE LA COMPUTACIÓN
CARLOS THAI LERATO

CHILLERATO

卷二

13. PLAN DE ESTUDIOS 2018

PLAN DE ESTUDIOS 2018 - CIENCIA DE LA COMPUTACIÓN												
PRIMER CICLO												
Código	Curso	T	P	L	S	E	T	H	C	T	C	Pre-requisito
BIC01	Introducción a la Computación	1	2		-	-	3	2	G	F	Ninguno	
BFI01	Física I	4	[2]	[2]	-	-	6	5	G	F	Ninguno	
BMA01	Cálculo Diferencial	4	2	-	-	-	6	5	G	G	Ninguno	
BMA03	Algebra Lineal	3	2	-	-	-	5	5	G	G	Ninguno	
BQU01	Química I	4	[2]	[2]	-	-	6	5	G	F	Ninguno	
		TOTAL:		26	22							
SEGUNDO CICLO												
Código	Curso	T	P	L	S	E	T	H	C	T	C	Pre-requisito
BMA02	Cálculo Integral	4	2	-	-	-	6	5	G	G	BMA01	
BRC01	Redacción y Comunicación	1	2	-	-	-	3	2	G	D	Ninguno	
BEI01	Idioma Extranjero o lengua nativa en el nivel intermedio	2						2	2	G	-	Ninguno
CM122	Algebra lineal I	3	2				5	5	G	G	BMA03	
CC212	Fundamentos de Programación	2		4			6	4	G	G	BIC01	
CM141	Cálculo de Probabilidades	2	2				4	3	G		Ninguno	
		TOTAL:		26	21							
TERCER CICLO												
Código	Curso	T	P	L	S	E	T	H	C	T	C	Pre-requisito
CM211	Cálculo Diferencial e Integral Avanzado	6	2	-	-	-	8	7	G		BMA02	
CM281	Matemática Discreta	3	2				5	4	G		BMA03	
CC311	Programación Orientada a Objetos	2		4			6	4	G	G	CC212	
CC321	Arquitectura de Computadores	2		4			6	4	G	G	CC212	
CM242	Estadística Inferencial	2	2				4	3	G		CM141	
		TOTAL:		29	22							
CUARTO CICLO												
Código	Curso	T	P	L	S	E	T	H	C	T	C	Pre-requisito
BEG01	Economía General	2	2	-	-	-	4	3	G	F	Ninguno	

CC412	Base de datos	2		4			6	4	G	G	CC311
CC422	Sistemas Operativos	2		4			6	4	G	G	CC321
CC432	Algoritmos y Estructuras de datos	2		4			6	4	E	G	CC311/CM281
CQ221	Introducción a la Electricidad y Magnetismo	3	2				5	5	G		BFI01/CM211
CL002	Ingles I	2					2	2	G		BEI01
							TOTAL:	29	22		

QUINTO CICLO												
Código	Curso	T	P	L	S	E	T	H	C	T	C	Pre-requisito
CM461	Análisis y Modelación Numérico I	4	2	2			6	6	E			CM211/CC432
CC511	Computación Centrada en Redes	2		4			6	4	E	G		CC422
CC521	Teoría de Autómatas, Lenguajes y Computación	2		4			6	4	E	G		CC432
CC531	Análisis y Diseño de Algoritmos	2		4			6	4	E	G		CC432
CC541	Ingeniería del Software	2		4			6	4	E	G		CC412
							TOTAL:	30	22			

SEXTO CICLO												
Código	Curso	T	P	L	S	E	T	H	C	T	C	Pre-requisito
BRN01	Realidad Nacional, Constitución y Derechos Humanos	2	2	-	-	-	4	3	G	D		Ninguno
BEF01	Etica y Filosofía Política	2					2	2	G	D		Ninguno
CC612	Administración de Redes	2		4			6	4	E	G		CC511
CC622	Matemática Computacional	2		4			6	4	E	G		CC531 / CM461
CC632	Programación Paralela	2		4			6	4	E	G		CC531
CC642	Desarrollo de Software	2		4			6	4	E	G		CC541
							TOTAL:	30	21			

SÉPTIMO CICLO												
Código	Curso	T	P	L	S	E	T	H	C			Pre-requisito
CC711	Seguridad en Sistemas Informáticos	2		4			6	4	E	G		CC612
CC721	Inteligencia Artificial	2		4			6	4	E	G		CC632
CC731	Computación Gráfica	2		4			6	4	E	G		CC622
CC741	Programación Concurrente y Distribuida	2		4			6	4	E	G		CC632
CC751	Interacción Humano-Computador	2		4			6	4	E	G		CC642
CC761	Metodología de la investigación	1	2				3	2	G	G		Ninguno

		TOTAL:	33	22				
--	--	---------------	-----------	-----------	--	--	--	--

OCATAVO CICLO

Código	Curso	T	P	L	S	E	T	H	C	Pre-requisito
CC812	Proyecto de Tesis I				8		8	4	E	CC761
CC822	Infraestructuras de Computación	2		4			6	4	E	G CC741
CC832	Robótica	2		4			6	4	E	G CC721 / CC731
CC842	Minería de datos	2		4			6	4	E	G CC721
					TOTAL:		26	16		

NOVENO CICLO

Código	Curso	T	P	L	S	E	T	H	C	Pre-requisito
CC911	Proyecto de Tesis II				8		8	4	E	CC812
CC921	Programación Evolutiva	2		4			6	4	E	G CC832
CC931	Análisis en Macrodatos	2		4			6	4	E	G CC842 / CC822
					TOTAL:		20	12		

DÉCIMO CICLO

Código	Curso	T	P	L	S	E	T	H	C	Pre-requisito
CC912	Proyecto de Tesis III				8		8	4	E	CC911
					TOTAL:		8	4		

14. CURSOS ELECTIVOS

El estudiante que haya aprobado los primeros seis ciclos o su equivalente en créditos puede llevar créditos de cursos electivos o complementarios. El estudiante debe completar como mínimo 22 créditos de cursos electivos. Con autorización del Consejo de Facultad podrá cursar asignaturas en otras facultades, otras universidades nacionales o extranjeras que tengan convenio recíproco (Art. 30 RR 1075), para tal caso debe solicitarlo por escrito al director de su correspondiente Escuela.

Cursos Electivos										
Código	Nombre de curso	T	P	L	S	E	T	H	C	Pre-requisito
CC001	Programación en tarjetas gráficas	2		4			6	4	E	G CC632
CC002	Base de datos Avanzadas	2		4			6	4	E	G CC412
CC003	Sistemas cognitivos	2		4			6	4	E	G CC721
CC004	Deep learning en visión artificial	2		4			6	4	E	G CC731/CC842

CC005	Programación de dispositivos móviles	2		4		6	4	E	G	CC531
CC006	Procesamiento del lenguaje natural	2		4		6	4	E	G	CC721
CC007	Programación declarativa	2		4		6	4	E	G	CC531
CC008	Compiladores	2		4		6	4	E	G	CC521
CC009	Multimedia	2		4		6	4	E	G	CC511
CC010	Diseño de sistemas basados en microordenador	2		4		6	4	E	G	CC511
CC011	Sistemas empotrados	2		4		6	4	E	G	CC612
CC012	Sistemas operativos avanzados	2		4		6	4	E	G	CC422
CC013	Simulación numérica	2		4		6	4	E	G	CC622
CC016	Calidad de Software	2		4		6	4	E	G	CC642
CC017	Auditoría de sistemas	2		4		6	4	E	G	CC711
CC018	Paradigmas de programación	2		4		6	4	E	G	CC531
CC019	Física Computacional	2		4		6	4	E	G	CC622
CC020	Algoritmos y Estructuras de Datos Avanzados	2		4		6	4	E	G	CC531
CC021	Biología Computacional	2		4		6	4	E	G	CC622
CC022	Sistemas Embebidos en Tiempo Real	2		4		6	4	E	G	CC741
CC023	Algoritmos bioinspirados	2		4		6	4	E	G	CC921
CC024	Programación de aplicaciones en redes	2		4		6	4	E	G	CC741
CC025	Lenguaje paralelo orientado a objetos	2		4		6	4	E	G	CC632
CC026	Tópicos de seguridad	2		4		6	4	E	G	CC711
CC027	Núcleo y Redes para Computación Paralela	2		4		6	4	E	G	CC632
CC028	Tópicos de Ciencia de la Computación I	2		4		6	4	E	G	Ninguno
CC029	Tópicos de Ciencia de la Computación II	2		4		6	4	E	G	Ninguno
CC030	Tópicos de Ciencia de la Computación III	2		4		6	4	E	G	Ninguno
CC031	Tópicos de Ciencia de la Computación IV	2		4		6	4	E	G	Ninguno
CC026	Tópicos de Computación Gráfica	2		4		6	4	E	G	CC731
CC027	Diseño Geométrico Asistido por Computador	2		4		6	4	E	G	CC731
CC028	Introducción a la Computación Cuántica	2		4		6	4	E		CQ221, CC622
CM561	Análisis y Modelamiento Numérico II	4	2	2		8	6	E		CM461
CM272	Introducción a la Ecuaciones Diferenciales Ordinarias	4	2			6	5	E		CM211
CM112	Lógica y Teoría de Conjunto	3	2			5	4	E		BMA01/BMA 03
CM352	Análisis Convexo	3	2			5	4	E		CM211
CM451	Programación Lineal y Flujos en Redes	4	2			6	5	E		CM352
CM454	Programación No Lineal	4	2		2	8	6	E		CM451

CM541	Introducción a los Procesos Estocásticos	3	2	2		5	4	E	CM242
CM096	Teoría de Grafos y Algoritmos	4	2			6	5	E	CC531
CM056	Modelos de Optimización	3	3			6	4	E	CM052
CM0E0	Programación Estocástica	4	[2]	[2]		6	5	E	CM454
CM221	Algebra Lineal II	4	2			6	5	E	CM122
CM442	Teoría de la Probabilidad	4	2	2		8	6	E	CM242
CM078	Programación Dinámica I	4	2			5	5	E	CM452
CM0D0	Programación Dinámica II	4	2			6	5	E	CM078
CM052	Optimización	4	2			6	5	E	CM454
CM0D9	Optimización Estocástica	4	2			6	5	E	CM454
CM0E0	Programación Estocástica	4	[2]	[2]		6	5	E	CM454
CM0E2	Programación Geométrica	4	[2]	[2]		6	5	E	CM454
CM0D4	Geoestadística	2		4		6	4	E	CM442
CM015	Series de Tiempo	4	2			6	5	E	CM242
CM019	Teoría de Colas	4	2			6	5	E	CM541
CM079	Inferencia Estadística No Paramétrica	3	3			6	4	E	CM242
CM080	Inferencia Estadística Paramétrica	4	2			6	5	E	CM242
CM085	Matemática Discreta I	4	2			6	5	E	CM281
CM017	Estadística Matemática	4	2			6	5	E	CM442
CM0A4	Criptografía y Seguridad	4	[2]	[2]		6	5	E	CC362/ CM221
CM031	Métodos Numérico para Ecuaciones Diferenciales Ordinarias	4	[2]	[2]		6	5	E	CM272/ CM461
CM035	Construcción Numérica - Diseño Geométrico	4	[2]	[2]		6	5	E	CC362
CM038	Métodos Numéricos del Álgebra	4	[2]	[2]		6	5	E	CM461/ CM221
CM039	Análisis Iterativo Matriciales	4	[2]	[2]		6	5	E	CM461/ CM221
CM048	Programación Científica	2	[2]	[2]		4	3	E	CC362
CM054	Métodos Numéricos de Optimización I	4	[2]	[2]		6	5	E	CM352/ CM461
CM055	Métodos Numéricos de Optimización II	4	[2]	[2]		6	5	E	CM054
CM033	Análisis Numérico III	3	[2]	[2]		5	4	E	CM561
CM0D5	Ánalisis de Datos	4		2		6	5	E	CC842
CM0D7	Teoría Algorítmica de Números	4	[2]	[2]		6	5	E	CC531
CM026	Sistemas Dinámicos	4	2			6	5	E	CM272
CL002	Inglés II	2				2	2	G	CL001
CC015	Inglés III	1		3		4	2	E	Inglés II
CF132	Física II	4	[3]	[3]		7	5	E	BFE01
CQ112	Química II	4	[4]	[4]		8	8	E	BQU01
CH061	Biología	2	[2]	[2]		4	3	E	Ninguno

CF382	Física III	4	[3]	[3]		7	5	E	CF132
CF382	Circuitos Electrónicos Analógicos	1		7		8	4	E	CC622
IF471	Sistemas Digitales	1		7		8	4	E	CF382

15. CRÉDITOS SOBRE ACTIVIDADES DIVERSAS

Créditos por actividades Diversas: Art. 27 RR 544, debe acumular 80 horas en 03 actividades distintas.

Item	Crédito s	Especificación
01	01	<ul style="list-style-type: none"> ○ Actividades artísticas ○ Actividades física y deporte ○ Participación en cursos y conferencias ○ Actividades científicas, concursos ○ Proyección social: ferias y difusión vocacional en colegios ○ Cursos de desarrollo personal: Coaching, emprendedurismo, liderazgo

Total de Créditos Exigidos al alumno para egresar

Item	Crédito s	Especificación
01	184	<ul style="list-style-type: none"> ○ Créditos de cursos obligatorios
02	22	<ul style="list-style-type: none"> ○ Total de créditos de cursos electivos
03	1	<ul style="list-style-type: none"> ○ Créditos por actividades extracurriculares (BAE01)
04	2	<ul style="list-style-type: none"> ○ Créditos por prácticas pre-profesionales
05	1	<ul style="list-style-type: none"> ○ Ayudantía académica o de investigación
TOTAL	210	TOTAL PARA EGRESAR RESPECTO AL PLAN 2018

16. TABLA DE CONVALIDACIONES DEL PLAN DE ESTUDIOS 2018 CON EL PLAN 2011

Malla 2011	Malla 2018
Primer Ciclo	
Cálculo Diferencial	Cálculo Diferencial
Cálculo Vectorial I	Algebra Lineal
Química I	Química I
Física I	Física I
Introducción a la Ciencia de la Computación	Introducción a la Computación
Segundo Ciclo	
Cálculo Integral	Cálculo Integral

Cálculo Vectorial II	Álgebra Lineal I
Química II	Química II (e)
Física II	Física II (e)
Introducción a la Programación	Fundamentos de programación
Tercero Ciclo	
Física III	Introducción a la Electricidad y electromagnetismo
Introducción a la Matemática Discreta	Matemática discreta
Álgebra Lineal I	Algebra Lineal II (e)
Introducción a la Programación Orientada a Objetos	Programación Orientada a Objetos
Constitución y Derechos Humanos	Realidad Nacional, Constitución y Derechos Humanos
Inglés I	Inglés I
Cuarto Ciclo	
Calculo Diferencial e Integral Avanzado	Cálculo Diferencial e Integral Avanzado
Estructura de datos	Algoritmos y Estructuras de datos
Arquitectura de Computadoras	Arquitectura de Computadores
Algoritmos	Análisis y diseño de algoritmos
Inglés II	Ingles II
Quinto Ciclo	
Introducción a la Estadística y Probabilidades	Cálculo de probabilidades / Estadística inferencial
Algoritmos Paralelos	Programación paralela
Base de Datos	Base de datos
Sistemas Operativos	Sistemas Operativos
Biología	Biología (e)
Sexto Ciclo	
Análisis Numérico I	Análisis y Modelación Numérico I
Lenguaje Paralelo Distribuido Orientado a Objetos	Lenguaje Paralelo Distribuido Orientado a Objetos (e)
Computación Gráfica	Computación gráfica
Teoría de la Computación	Teoría de autómatas, Lenguajes y Computación
Computación Centrada en Redes	Computación centrada en redes
Séptimo Ciclo	
Programación de Aplicaciones en Redes	Programación de Aplicaciones en Redes (e)
Inteligencia Artificial	Inteligencia artificial

Compiladores	Compiladores
Biología Computacional	Biología Computacional (e)
Administración de Redes	Administración de Redes
Octavo Ciclo	
Sistemas Concurrentes y Distribuidos	Programación concurrente y distribuida
Ingeniería de Software I	Ingeniería del Sofware
Núcleo y Redes para Computación Paralela	Nucleo y Redes para Computación Paralela (e)
Ciencia, Tecnología y Sociedad	Ciencia, Tecnología y Sociedad
Noveno Ciclo	
Seminario de Tesis I	Proyecto de investigación I
Introducción a la Computación Grid y Cloud	Infraestructuras de computación
Sistemas Operativos Avanzados	Sistemas Operativos Avanzados
Décimo Ciclo	
Seminario de Tesis II	Proyecto de investigación II
Modelamiento y Simulación	Modelamiento y Simulación (e)

17. MÓDULOS DE CERTIFICACIÓN TÉCNICA

Modulo I: especialista en desarrollo de software

- Aprobar todos los cursos hasta el quinto ciclo
- Aprobar además:
 - CC642 Desarrollo de Software
 - CQ751 Interacción Humano-Computadora

Módulo II: Especialista en Administración y Seguridad en Sistemas Informáticos

- Aprobar todos los cursos hasta el quinto ciclo
- Aprobar además:
 - CC612 Administración de Redes
 - CC711 Seguridad en sistemas informáticos

Módulo III: Especialista en ciencia de datos

- Aprobar todos los cursos hasta el quinto ciclo
- Aprobar además:
 - CC632 Programación paralela
 - CC721 Inteligencia Artificial
 - CC842 Minería de datos

Modulo IV: Especialista en visión gráfica y robótica

- Aprobar todos los cursos hasta el quinto ciclo
- Aprobar además:
 - CC622 Matemática computación
 - CC721 Inteligencia Artificial

- CC731 Computación gráfica
- CC832 Robótica

18. PROYECTO PARA ALCANZAR LA CERTIFICACIÓN

Recibida la solicitud del alumno interesado, el Director de la Escuela profesional asignará un docente de la Escuela quien le asignará al solicitante un proyecto que realizará a lo largo del semestre y presentará el correspondiente informe a la Escuela profesional con la respectiva nota del docente asignado, validado por el director de Escuela.

19. RECONOCIMIENTOS

- Comisión Central para la Elaboración del plan de Estudios 2018
- Comisión para establecer el perfil de ingreso y egreso de la escuela profesional de Ciencia de la Computación
- Comisión para elaboración de la malla curricular:
- Profesores que participaron elaborando los sílabos de cursos:

20. SÍLABOS DE CURSOS OBLIGATORIOS

21. SÍLABOS DE CURSOS ELECTIVOS

RECONOCIMIENTOS:

Comisión Central para la Elaboración del plan de Estudios 2018

Comisión para establecer el perfil de ingreso y egreso

Comisión para elaboración de la malla curricular:

Profesores que participaron elaborando los sílabos de cursos:

-



**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-REQUISITO	:	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 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

I: lenguajes naturales y artificiales. II: Elementos y evolución. III: Diseño e implementación. IV: Taxonomía, paradigmas y popularidad.

2. Representación de datos simple en memoria

I: Definición de programa y sus partes. II: Conceptos básicos de la programación: variables, tipos de datos, constantes, operadores y sentencias. III: Almacenando información e identificadores.

3. Entrada y salida de datos

I: Escribiendo y leyendo información: funciones de entrada y salida de datos.

4. Sentencias de control

I: Introducción a la programación estructurada. II: Estructuras selectivas. III: Estructuras iterativas.

5. Funciones y Recursividad

I: Empleo de funciones: paso por valores. II: Alcance de variables: variables locales y globales. III: Empleo de la técnica de recursividad.

6. Estructuras de datos fundamentales 1

I: Arreglos numéricos: vectores y matrices. II: Registros.

7. Punteros

I: Declaraciones. II: Operadores. III: Paso por referencia. IV: Memoria dinámica.

8. Archivos

I: Métodos de acceso. II: Tipos de archivos: secuenciales, binarios. III: Directorios. IV: Binarios y ASCII.

9. Estructuras de datos fundamentales 2

I: Estructuras y uniones. II: Listas enlazadas. III: 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 (6th 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-REQUISITO : CC212 – FUNDAMENTOS DE PROGRAMACIÓN

CONDICIÓN : OBLIGATORIO

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

SISTEMA DE EVALUACIÓN : G

SUMILLA

Para el estudiante de Ciencias 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 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:** Introducción a la programación orientada a objetos. Introducción a la programación estructurada y orientado a objetos. Clases, objetos diagramas de clases. Constructores y destructores. Objetos y mensajes. Implementa clase valorando el paradigma anterior. Resuelve e implementa programas utilizando una herramienta orientado a objetos.
- 2. Relaciones entre clases y objetos:** Modela programas utilizando las relaciones entre objetos: de uso, asociación, agregación y composición. Relaciones entre clases, valora la reutilización. Modelamiento utilizando relaciones entre clases. Implementa estructuras de datos estática y dinámica. Implementación soluciones con relaciones entre clases y objetos utilizando una herramienta orientado a objetos.
- 3. Interfaces, polimorfismo y clases genéricas:** Interfaces y Clases Abstractas. Clases polimórficas utilizando la herencia de Clases, Interfaces y Clases abstractas. Clases y métodos genéricos. Colecciones genéricas paramétricas. Métodos de ordenamiento y de búsqueda genéricos. Archivos directos, Archivo de índices: Implementa soluciones utilizando herramientas orientadas a objetos.
- 4. Excepciones y persistencia:** Excepciones, Excepciones Java. Excepciones predefinidas y Herencia de excepciones creadas por el usuario. Archivos en java. Persistencia serializarán y des socialización de objetos. Acceso a base de datos desde Java. Implementación de programas valorando el manejo de excepciones. Gestión de la información en memoria principal y secundaria.
- 5. Hilos y GUIs:** Programación concurrente, programación de hilos en Java. Arquitectura de las tres capas: datos, lógica e interfaz. Interfaces graficas de Usuario (GUI) de java. Ventanas, componentes y administradores de distribución de componentes. Implementación de programas utilizando lenguaje de programación orientado a objetos
- 6. Estructuras dinámicas de datos:** I: Introducción. II: Asignación dinámica de memoria. III: Listas enlazadas simples.
- 7. Programación orientada a eventos**

LABORATORIOS Y EXPERIENCIAS PRÁCTICAS

Laboratorio: Todas y cada una de las unidades de aprendizaje tienen sus prácticas y/o laboratorio alusivo al tema.

METODOLOGÍA

El curso se desarrolla en sesiones de teoría y práctica. En las sesiones de teoría, el docente presenta los conceptos, fundamentos y aplicaciones, mientras que en las sesiones prácticas se resuelven casos hipotéticos o reales cuya solución es analizada en detalle.

El trabajo en equipo, el desarrollo de la comunicación y la expresión es promovido y evaluado mediante trabajos grupales que serán expuestos en clase, para lo cual se conformarán equipos de no más de 4 integrantes.

En todas las sesiones se promueve la participación activa del alumno.

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 básico del repertorio de instrucciones.

Garantizar el buen desempeño, la eficiencia de la futura codificación y conocer los fundamentos de la programación de bajo nivel será 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.

PROGRAMA ANALÍTICO

1. Computadores de 8 bits

I:Presentación del Curso. II:Historia (breve) y Tecnología de los computadores. III: Bajo los programas y bajo la cubierta. IV:El Microcontrolador 8051. V:Arquitectura del 8051. VI:Conjunto de Instrucciones y Lenguaje ensamblador. VII:El TMC51. VIII:Simulador EDSIM51 y programación del 8051. IX: Timers/Contadores. X:Puerto Serie. XI:Interrupciones. XII:Display LCD.

Computadores de 16 y 32 bits

I:Introducción al MSP430 de Texas Instruments.
II:Introducción a MIPS. III:Instrucciones del computador. IV:El procesador MIPS. V:Programación de MIPS

2. Organización de la Memoria y Arquitectura

I: Codificación, compresión de datos e integridad de datos. II: Jerarquía de memoria. III: Memorias caché, (mapeo de direcciones, tamaño de bloques, políticas de reemplazo y almacenamiento). IV: Memoria Virtual (tablas de paginación, TLB).

3. Comunicación e Interfase

I: Fundamentos de entrada y salida: buffering, handshaking, entradas y salidas, entradas y salidas manejadas por interrupciones. II: Manejo de interrupción. III: Almacenamiento externo, organización física y drivers. IV: Buses: protocolos de buses, arbitraje, acceso directo a memoria (DMA). V: Arquitecturas RAID.

4. Multinúcleos, multiprocesadores y clústeres.

I: Introducción a SIMD, MIMD, VLIW, EPIC. II: Arquitectura sistólica. . III: Sistemas de memoria compartida. IV:GPUs para gráficos y cálculos.

5. Herramientas de Diagnóstico y seguridad

I: Evaluación de los sistemas de cómputo. II: Diagnóstico de los sistemas de procesamiento. III: 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-REQUISITO	:	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

I: Qué es una base de datos. II: Qué es un Sistema de gestión de base de datos. III: Independencia de los datos. IV: Arquitectura de un sistema de base de datos.

2. Modelo de Entidad relación (E/R)

I: Introducción al modelo de datos ER. II: Conjuntos de entidades y de relaciones. III: Dominios. IV: Representaciones equivalentes de una relación. V: Cuestiones de diseño. VI: Especialización. VII: Agregación. VIII: Reducción a tablas.

3. El modelo relacional. Conversión de E/R

I: Introducción. II: Esquemas, Tuplas, Tablas, Dominios. III: Conversión a tablas desde un modelo con relaciones. IV: Conversión a tablas desde un modelo con generalización. V: Descubrimiento de claves en las relaciones.

4. Dependencias Funcionales

I: Definición. II: Axiomas de Armstrong. III: Reglas Adicionales. IV: Clave de un conjunto de atributos. V: No redundancia. VI: Determinación de las claves de un esquema y cálculo de las claves de un esquema Relacional..

5. Normalización. Dependencias de múltiple valores.

I: Redundancia, Anomalías de actualización y Eliminación. II: Primera y Segunda forma Normal. III: Descomposición sin Pérdida. IV: Preservación de dependencias. V: Forma Normal de Boyce-Codd (BCNF). VI: Tercera forma Normal, BCNF vs 3NF. VII: Cuarta y Quinta forma Normal.

6. Introducción a SQL

I: Aplicaciones de las bases de datos. II: Sistemas de bases de datos frente a sistemas de archivos. III: Visión de los datos. IV: Modelos de bases de datos. V: Lenguajes de bases de datos. VI: Gestión de transacciones. VII: Estructura de un sistema de bases de datos. VIII: Usuarios de bases de datos.

7. Agregaciones, modificaciones, y operaciones avanzadas

I: Agregaciones y diseño de agregaciones. II: Modificación de datos. III: Clasificación, Insertar tuplas en una relación, Eliminar tuplas de una relación y Actualizar el valor de algunos componentes de tuplas existentes.

8. Vistas y definición de datos

I: Concepto de vista. II: Aplicaciones de vistas. III: Vistas en SQL. IV: Sintaxis. V: Motivos por que una lista no es actualizable. VI: Vista sobre una tabla Básica. VII: Vista sobre una concatenación de relaciones. VIII: Definir tablas, filas y columnas. IX: Insertar claves de índice. X: Crear relaciones entre tablas. XI: Asignar tipos de datos.

9. Restricciones y disparadores (triggers)

I: Restricciones en procedimientos almacenados y disparadores. II: Restricciones en subconsultas. III: Restricciones en vistas. IV: Definición, uso y sintaxis del trigger. V: Componentes y nombres de disparadores. VI: Tipos de disparadores. VII: Orden de activación de disparadores.

10. Módulos de almacenamiento persistentes

I: Almacenamiento y estructura de archivos. II: Indexación, asociación, hashing. III: Índices. IV: Procesamiento de consultas. V: Optimización de consultas.

11. Autorización

I: Violaciones de la seguridad. II: Control de acceso a la base de datos. III: Tipos de autorización. IV: Autorizaciones y vistas. V: Concesión de privilegios. VI: Eliminación de privilegios. VII: El concepto de rol o papel. VIII: Limitaciones de la autorización SQL.

12. Transacciones

I: Introducción. II: Sentencias para una transacción. III: Transacciones anidadas. IV: 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-REQUISITO	:	CC321 – ARQUITECTURA DE COMPUTADORES
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

OBJETIVO

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 Sistemas Operativos:** Funciones del Sistema Operativo. Componentes del Sistema Operativo. Diagrama de casos de uso. Funciones y componentes como casos de uso.
2. **Descripción de procesos:** Ejecución de un proceso. Ejemplo de ejecución Estados de un proceso. Como se crean los procesos. Como terminan los procesos. El estado bloqueado. El estado suspendido. Tablas de control del sistema operativo. Tabla de procesos. Bloque de control de procesos BCP. Identificación del proceso. Identificación del contexto del CPU. Información de control del proceso. El proceso en el kernel. Modos de ejecución. Creación de procesos. Intercambio de procesos. Cambio en el estado del proceso.
3. **Planificación de procesos:** Cola de procesos. Reloj de tiempo real. Tipos de planificación. Planificación a largo plazo. Planificación a mediano plazo. Planificación a corto plazo. Tipos de procesos. Procesos en tiempo real. Procesos Normales. Procesos no expropiativos. Procesos expropiativos. Políticas de planificación. SCHED_FIFO. SCHED_RR. SCHED_OTHER. Algoritmo de planificación en Linux.
4. **Concurrencia de procesos:** Principios Generales. Exclusión mutua. Sección crítica. Algoritmo de Peterson. Exclusión mutua por Hardware. Por inhabilitación de interrupciones. Por instrucciones en lenguaje de máquina. Semáforos. El estado Espera. Algoritmo para un semáforo binario. Exclusión mutua con semáforos. Ejemplo de semáforo binario. Mensajes. Direccionamiento. Sincronización. Partes de un mensaje. Exclusión Mutua.
5. Interbloqueo de procesos: Principios Generales. Procesos con interbloqueo, Procesos sin interbloqueo. Recursos reutilizables. Recursos consumibles. Condiciones para el interbloqueo. Prevención del interbloqueo. Exclusión mutua. Retener y Esperar. Expropiación. Círculo de espera. Predicción del interbloqueo. Estado seguro. Estado inseguro. Detección del interbloqueo. Recuperación luego del interbloqueo. Estrategias integradas para evitar el interbloqueo.
6. **Gestión de Memoria:** Requisitos de la gestión de memoria. Reubicación. Protección. Compartimiento. Organización Lógica. Organización física. Partición de la memoria. Particiones estáticas de igual tamaño. Particiones estáticas de diferente tamaño. Particiones Dinámicas. Algoritmo de Ubicación. Sistema de Bloques 2n (Buddy System). Reubicación de procesos en la memoria. Tipos de Direcciones. Registros usados durante la ejecución. Paginación. Asignamiento de marcos libres a páginas de un proceso. Cálculo de la dirección física de memoria. Segmentación. Cálculo de la dirección física de memoria. Direccionamiento lógico.
7. **Gestión de Memoria Virtual:** Estructuras de hardware y control. Que ocurre al ejecutar un programa? Ventajas de la fragmentación de procesos. Tipos de Memoria. Hiperpaginación. Principio de Cercanía. Soporte para el uso de memoria virtual. Paginación. La tabla de páginas. Buffer de traducción adelantada. El problema del tamaño de página. Segmentación. Tablas de segmentos. Política de Carga. Política de Reemplazo. Algoritmos de reemplazo. Asignación de páginas para un proceso. Asignación variable. Política de limpieza. Control de carga. Suspensión de procesos.
8. **Gestión de E/S:** Dispositivos de Entrada/Salida. Tipos de dispositivos E/S. Diferencias entre dispositivos E/S. Organización de las funciones E/S. Evolución de las funciones E/S. Técnicas para ejecutar la E/S. Memoria de Acceso Directo (DMA). Aspectos del diseño del Sistema Operativo. Características para diseñar un sistema operativo.
9. **Gestión de Almacenamiento:** Almacenamiento intermedio de la E/S. Buffering de E/S. Buffer simple. Buffer Doble. Buffer Circular. Planificación de Discos. Perfomance del Disco Duro. Raid. Raid 0. Raid 1. Raid 2. Raid 3. Raid 4. Raid 5. Raid 6. Cache de disco. Usado menos recientemente. Usado menos frecuentemente.
10. **Gestión de Archivos:** Conceptos iniciales. Términos más usados. Operaciones típicas con archivos. Objetivos de un sistema de manejo de archivos. Conjunto de requerimientos

- mínimos. Drivers de dispositivos (Disco y Tape). Sistema Básico de Archivos. Supervisor Básico de E/S. E/S Lógico. Métodos de acceso. Funciones de la gestión de archivos. Organización y acceso de archivos. Criterios para la Organización de archivos.
11. **Gestión de Directorios:** Organización de directorios. Directorio de archivos. Estructura simple para un directorio. Estructura de dos niveles de archivos. Estructura jerárquica de archivos. Compartiendo archivos. Privilegios de acceso a archivos. Acceso simultáneo. Agrupación de registros. Bloques y Registros. Bloques fijos. Bloques variables. Gestión del almacenamiento secundario. Preasignación. Métodos de ubicación de archivos. Sistema de archivos Linux y Windows. Ext2. NTFS.

LABORATORIOS Y EXPERIENCIAS PRÁCTICAS

Laboratorio: Uso de herramientas y desarrollo de programas que aplican los conceptos teóricos estudiados.

METODOLOGÍA

El curso se desarrolla en sesiones de teoría y práctica. En las sesiones de teoría, el docente presenta los conceptos, fundamentos y aplicaciones, mientras que en las sesiones prácticas se resuelven casos hipotéticos o reales cuya solución es analizada en detalle.

El trabajo en equipo, el desarrollo de la comunicación y la expresión es promovido y evaluado mediante trabajos grupales que serán expuestos en clase, para lo cual se conformarán equipos de no más de 4 integrantes.

En todas las sesiones se promueve la participación activa del alumno.

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.
9. <http://svnbook.red-bean.com/nightly/en/svn-book.html>
10. http://sources.redhat.com/autobook/autobook/autobook_toc.html
11. <http://tldp.org/LDP/lame/LAME/linux-admin-made-easy/>



**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-REQUISITO	:	CC311 – PROGRAMACIÓN ORIENTADA A OBJETOS XXX - 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 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 del Algorítmica y ejecuta los principios de esta ciencia de la computación.
- Describe los procesos de construcción Algoritmos identificando las propiedades asociadas que se generan de ellos apreciando su influencia en la ciencia e ingeniería.

PROGRAMA ANALÍTICO

1. **CAPÍTULO 1:** Introducción. Definición de estructura de datos. Clasificación de estructura de datos. Operaciones sobre Estructura de Datos, Estructuras de datos y eficiencia, Algoritmos para encontrar máximos y mínimos, Algoritmos para realizar búsquedas secuenciales y binarias, Algoritmos de ordenamiento, de peor caso cuadrático (selección, inserción), Algoritmos de ordenamiento caso promedio (Quicksort, Heapsort, Mergesort)
2. **CAPÍTULO 2:** Descripción formal de TAD: Conjuntos, Implementación de un TAD, Listas enlazadas simples. Noción de lista enlazada. Definición formal de lista enlazada. Características. Representación de listas enlazadas mediante arreglos, mediante cursores y mediante apuntadores. Listas doblemente enlazadas y listas circulares. Definición formal de lista doblemente enlazada y listas circulares. Características. Representación de listas enlazadas. Pilas y colas. Noción de pila. Definición formal de pila. Características. Acciones primitivas. Representación de pilas. Operaciones básicas. Aplicación y ejemplos. Noción de cola. Definición formal de cola. Características. Acciones Primitivas. Representación de colas. Tipos de colas.
3. **CAPÍTULO 3:** Árboles. Noción de árbol. Definición de árbol. Características. Acciones primitivas. Representación de Árboles mediante arreglos y mediante apuntadores. Tipos de árboles. Árboles binarios. Definición formal de árbol binario características. Acciones primitivas. Representación de árboles binarios. Ordenamiento, Búsqueda, Inserción y Eliminación.
4. **CAPÍTULO 4:** Grafos. Noción de grafo. Tipos: Grafos no dirigidos, Grafos dirigidos. Definición Formal de grafos no dirigidos. Representación. Operaciones básicas. Aplicaciones y ejemplos. Grafos. Recorrido de grafos. Matriz de adyacencia, Matriz de caminos. Caminos mínimos. Definición Formal de Grafos dirigidos. Búsqueda en Profundidad. Árboles de recubrimiento. Camino Hamiltoniano. Paseo Euleriano. Algoritmo de Dijkstra. Algoritmo de Warshall. Grafos acíclicos. 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. (En línea.)
4. Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J.A.: Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos. 2^a 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, 3^a edición, Ed. Edicions UPC, 2001.
8. Mehta, D.P. y Sahni, S.: Handbook of Data Structures and Applications, Chapman & Hall/CRC, 2005.



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	COMPUTACIÓN CENTRADA EN REDES
CÓDIGO	:	CC511
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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

Tema 1.: Introducción. Uso de las redes de computadoras. Hardware de Red (PAN, LAN, MAN, WAN). Software de Red. Modelos de Referencia (OSI, TCP/IP). Redes de Ejemplo (Redes LAN inalámbricas, Redes RFID y de sensores). Estandarización de Redes. Introducción. Una breve historia de las redes. Calidad de servicio. Redes TCP/IP y Redes ATM.

Tema 2.: Capa Física: Medios de transmisión guiados (Medios magnéticos, par trenzado, cable coaxial, líneas eléctricas, fibra óptica). Transmisión inalámbrica (microondas, infrarroja, ondas de luz). Satélites de comunicación. Modulación digital y multiplexión (transmisión en banda base). La Red Telefónica Pública Conmutada (ADSL y Fibra óptica). Sistema de telefonía móvil (1G, 2G, 3G y 4G). Protocolos y la arquitectura TCP/IP. El modelo OSI. Interconexión de redes.

Tema 3.: Capa de Enlace de Datos: Diseño. Detección y Corrección de Errores. Protocolos elementales de Enlace de Datos. Protocolos de Ventanas Deslizantes. TCP e IP. El protocolo de control de transmisión (TCP). Ipv4 e Ipv6.

Tema 4.: Subcapa de Control de Acceso al Medio: Protocolo de Acceso Múltiple. Ethernet (Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet, Ethernet Conmutada). Redes LAN Inalámbricas (802.11 y la pila de protocolos). Banda Ancha Inalámbrica (Capa física estándar 802.16) . Bluetooth (Arquitectura). RFID (Formatos de los mensajes de identificación de etiquetas). Conmutación de la Capa de Enlace de Datos (Usos de los puentes, repetidores, hubs, enrutadores, y puertas de enlaces o gateways, Redes LAN virtuales) . Redes y Arquitectura de Frame Relay.

Tema 5.: La Capa de Red: Aspectos del Diseño de la Capa de Red. Algoritmos de Enrutamiento (Enrutamiento por vector distancia, enrutamiento jerárquico, enrutamiento multidifusión, enrutamiento por la ruta más corta, anycast). Algoritmos de Control de Congestión (Control de admisión). Calidad de Servicio. Interconexión de Redes. La Capa de Internet (Protocolo Ip, Ip versión 4, Ip versión 6, Protocolos de control en Internet, multidifusión en Internet, Ip móvil). Modo de transferencia asíncrono (ATM). Conexiones lógicas, celdas ATM.

Tema 6.: La Capa de Transporte: El servicio del transporte. Elementos del los Protocolos de Transporte (Direccionamiento, establecimiento de una conexión, multiplexación, control de errores). Control de Congestión. Los Protocolos de Transporte en Internet (Introducción a UDP, llamada de procedimiento remoto, protocolos en tiempo real). 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). Aspectos del Desempeño (Diseño de los hosts para redes rápidas). Redes tolerantes al retardo. TCP, FTP y Telnet.

Tema 7.: La Capa de Aplicación: DNS, el sistema de nombres de dominio (el espacio de los nombres del DNS, registros de los recursos de dominio, servidores de nombres). Correo electrónico. WWW (web estáticas y dinámicas, HTTP, web móvil). Audio y Video de Flujo Continuo. Entrega de Contenido (Contenido y tráfico en Internet, redes de igual a igual). Gestión de la congestión y del tráfico. Control y gestión del tráfico. Control de la congestión en Frame Relay.

Tema 8.: Introducción a la Seguridad de Redes: Criptografía. Algoritmo de Clave Simétrica. Algoritmos de Clave Pública (RSA y otros algoritmos de clave pública). Firmas Digitales. Administración de Claves Públicas (Certificados). Seguridad en la Comunicación (Ipsec, Firewalls, Redes privadas virtuales, seguridades inalámbricas). Protocolos en la Autenticación . Seguridad de Correo Electrónico. Seguridad en Web.

BIBLIOGRAFÍA

1. ARIGANELLO, ERNESTO - “Redes Cisco”. Ra-Ma
2. TANEMBAUM/WETHERALL - “Redes de Computadoras” . Pearson Quinta edición, 2012. *Obligatorio
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-REQUISITO	:	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. Cual 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

- 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.
- 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

- **U1.- Introducción a la TC:** Conjuntos en TC. Relaciones y Funciones en TC. Teoría de Grafos. Semigrupos y Grupos.
- **U2. Cadenas y Lenguajes:** Símbolo. Alfabeto. Cadena. Operaciones con cadenas. Lenguajes. Operaciones con Lenguajes. Técnicas Básicas de Demostración.

- **U3. Lenguajes Regulares:** Definición recursiva de LR. Ejemplos. Expresiones Regulares(ER). Equivalencia de ER. Propiedades de ER. Derivadas. Sistemas de Ecuaciones.
- **U4.- Máquinas de Estado Finito:** Definición MEF. Representación Gráfica. Sumador Binario. Clasificación de MEF: Moore y Mealy
- **U5.- Autómatas de Estados Finitos:** Autómatas Finitos Determinísticos(AFD). Minimización de AFD. AFD Equivalentes. Autómatas Finitos no Determinísticos(AFND). Representación de AFND. Equivalencia entre AFD y AFND. Autómatas con transiciones épsilon: representación, función de transición extendida.
- **U6.- Gramáticas:** Regla. Derivaciones. Definición de Gramática Formal.Gramáticas Regulares. Conversión de GR a AFD. Conversión de GR a AFND.
- **U7.- Lenguajes libres de contexto:** Gramáticas Libres de Contexto. Árboles de Derivación. Simplificación de Gramáticas: Eliminación de Símbolos Inútiles. Eliminación de Producciones épsilon. Eliminación de Producciones Unitarias.Forma Normal de Chomsky. Eliminación de Factores Comunes izquierdos. Eliminación Recursividad Izquierda. Eliminación de ambigüedad. Forma Normal de Greibach. Autómata de Pila Determinístico. Autómata de Pila No Determinístico
- **U8.- Máquina de Turing:** Definición formal de máquina de Turing. Restricciones a la máquina de Turing. Construcción Modular máquina de Turing.
- **U9.- Decibilidad:** Lenguajes Decibles. Los problemas de Halting. Teorías Lógicas de decibilidad
- **U10.- Reducibilidad:** Problemas Insolubles en la teoría de lenguajes. Un problema simple que es insoluble. Funciones Computables. 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-REQUISITO	:	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

Tema 1. Análisis de la eficiencia de los algoritmos: Complejidad de los algoritmos. Notaciones asintóticas. Análisis de Tiempos de Ejecución para cada caso. Ecuaciones de Recurrencia. Medir eficiencia a través de metaheurísticas.

Tema 2. Introducción a la optimización simple: Análisis de algoritmos iterativos y recursivos. Análisis de algoritmos de ordenación y búsqueda. Dividir-y-Conquistar. Análisis Probabilístico y Algoritmos aleatorios

Tema 3. Clasificación y estadísticas de orden: Heapsort. Quicksort. Ordenamiento en tiempo lineal. Medianas y estadísticas de orden.

Tema 4. Diseño Avanzado y Técnicas de Análisis: Programación dinámica. Algoritmos golosos. Análisis amortizado.

Tema 5. Algoritmos de grafos: Algoritmos de grafo elementales. Árboles de expansión de mínimos. Caminos más cortos de un solo proveedor. Todos los pares de rutas más cortas. 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 FÍSICA

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	INGENIERÍA DE SOFTWARE
CÓDIGO	:	CC541
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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

- **Tema 1.: Introducción a la Ingeniería del Software:** 1.1. Origen y definiciones básicas. 1.2. Conceptos fundamentales
- **Tema 2: Ingeniería de Requisitos Software:** 2.1. Introducción, 2.2. Definición de Requisito, Tipos y Propiedades, 2.3. Modelos de Proceso de IR. Propuesta metodológica,

2.4. Etapa de Adquisición de Requisitos, 2.5. Etapa de Análisis de Requisitos, 2.6. Etapa de Validación y Verificación de Requisitos

- **Tema 3. Modelado de Requisitos con UML 2.0. - Diagramas de Casos de Uso:** 3.1. Introducción, 3.2. Definición y especificación de Casos de Uso, 3.3. Notación gráfica, 3.4. Tipos de Relaciones en un Diagrama de Casos de Uso, 3.5. Ejemplos y casos prácticos.
- **Tema 4. Análisis Orientado a Objetos con UML 2.0:** 4.1. Introducción al modelado de SW con UML 2.0, 4.2. Diagramas de Clases de Dominio, 4.3. Diagramas de Clases de Análisis, 4.4. Diagramas de Actividad
- **Tema 5. Diseño Orientado a Objetos con UML 2.0:** 5.1. Diagramas de Estados, 5.2. Diagramas de Secuencia, 5.3. Diagramas de Comunicación, 5.4. Diagramas de Tiempos, 5.5. Diagramas de Componentes, 5.6. Diagramas de Despliegue
- **Tema 6. Desarrollo de Sistemas Orientado a Objetos. Proceso Unificado de Desarrollo (RUP):** 6.1. Introducción, 6.2. Principios básicos de RUP, 6.3. Fases y Etapas de RUP.
- **Tema 7. Gestión y Pruebas de software:** 7.1. Gestión de Configuración Software, 7.2. Pruebas de Software: Introducción y Principios, 7.3. Pruebas de Software: Estrategias y Procesos
- **Tema 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-REQUISITO	:	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.

Todo 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 como 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 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. **Tema 1.** Administración básica a nivel de sistema operativo: 1.1. Comando básicos e instalación de S.O., 1.2. Gestión de aplicaciones, 1.3. Gestión de usuarios, 1.4. Administración de Ficheros.
2. **Tema 2.** Protocolos de la capa de red: 2.1. Funcionalidad y datagrama, 2.2. Encaminamiento IP, 3.3. Protocolos de encaminamiento dinámico, 3.4. ICMP
3. **Tema 3.** Protocolos de la capa de transporte: 4.1. UDP, 4.2. TCP
4. **Tema 4.** Administración avanzadas a nivel de sistema operativo: 4.1. Discos redundantes, 4.2. Monitorización del sistema, 4.3. Administración de red y encaminamiento.
5. Tema 5. Administración básica de servidores: 5.1. CIDR, VLSM y SSH, 5.2. HTTP, 5.3. DHCP, 5.4. NAT
6. **Tema 6.** Administración avanzada de servidores: 6.1. DNS, 6.2. DDNS, 6.3. Servidor de correo.
7. **Tema 7.** Control de acceso a servicios: 7.1. Proxy, 7.2. Firewall.

BIBLIOGRAFÍA

1. "UNIX. Interconexión de redes". U.O. Pabral. 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. [Disponible en línea: <http://es.tldp.org/Manuales-LuCAS/SEGUNIX/unixsec-2.1.pdf>]
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. <http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/GARL2/garl2/index.html>
15. L. Parziale. "TCP/IP Tutorial and Technical Overview". 8th edition. IBM RedBooks. 2006.
16. Q. Li. "IPv6 Core Protocols Implementation". 1st edition. Morgan Kaufmann Publishers. 2005.
17. F. Halsall. "Redes de Computadores e Internet". 5^a edición. Addison-Wesley. 2006.
18. B. Sosinsky. "Networking Bible". 1st edition. Wiley Publishing. 2009.
19. E. Cole. "Network Security Bible". 2nd edition. John Wiley & Sons. 2009.



**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION**

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	MATEMÁTICA COMPUTACIONAL
CÓDIGO	:	CC622
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC531 - ANÁLISIS Y DISEÑO DE ALGORITMOS CCXXX - ANÁLISIS NUMÉRICO
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 resuelven 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 .
- Estabilidad. Un algoritmo estable no es sensible a pequeños cambios en la 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. Tema 1. Matemáticas para la Computación Gráfica: 1.1. Transformaciones geométricas, 1.2. Interpolación, 1.3. Curvas y superficies, 1.4. Geometría analítica, 1.5. Álgebra geométrica.
2. Tema 2. Optimización Computacional: 2.1. Preliminares: ejemplos de problemas de optimización concretos, 2.2. Programación lineal: formato de los problemas, soluciones admisibles y optimalidad, Método Simplex y Método del Punto-Interior, 2.3. Programación no lineal: tipos de problemas, condiciones de optimalidad, direcciones de búsqueda y búsqueda lineal, 2.4. Algoritmos de programación diferenciable y no diferenciable.
3. Tema 3. Probabilidad Computacional: 3.1. Preliminares, 3.2. Algoritmos para variables aleatorias continuas, 3.3. Algoritmos para variables aleatorias discretas, 3.4. Simulación estocástica.
4. Tema 4. Métodos Numéricos: 4.1. Análisis lineal, 4.2. El Métodos de Galerkin: formulación y solución de PDEs, 4.3. 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 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 all. Cambridge University Press. Cambridge Texts in Applied Mathematics. 2014.



**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION**

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	PROGRAMACIÓN PARALELA
CÓDIGO	:	CC632
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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 basado en memoria o procesamiento distribuido.

PROGRAMA ANALÍTICO

1. **Tema 1.** Introducción a la programación de altas prestaciones
2. **Tema 2.** Programación orientada a bloques
3. **Tema 3.** Introducción a las arquitecturas paralelas
4. **Tema 4.** Paradigmas de computación paralela
5. **Tema 5.** Diseño de programas en arquitecturas paralelas
6. **Tema 6.** Software de programación de arquitecturas de altas prestaciones
7. **Tema 7. Tipo de datos abstractos:** 1 Stacks; 2. Colas; 3. Colas con prioridad; 4. Conjuntos; 5. Tablas; 6. Árboles; 7. Búsqueda en árboles binarios, Heaps y árboles AVL.
8. **Tema 9. Algoritmos de grafos:** 1. Búsqueda siguiendo depth-first and breath-first; 2. Mínimo spanning tree; 3. Búsqueda en árboles.
9. **Tema 10. 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 FÍSICA

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	DESARROLLO DE SOFTWARE
CÓDIGO	:	CC642
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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 apropiados.
- Conocer cuáles son los estándares para contenidos digitales.
- Gestionar la seguridad en sistemas informáticos.

PROGRAMA ANALÍTICO

1. **Tema 1.** Introducción y conceptos básicos: I. Introducción al currículo de estándares web/contenidos, La historia de Internet y la web y la evolución de los estándares web; II. Arquitectura OSI; III. Protocolos de internet

2. **Tema 2. Texto etiquetado y hojas de estilo:** 1. Conceptos básicos de HTML; 2. El elemento del HTML; 3. Elegir el doctype correcto para los documentos HTML; 4. Etiquetar contenido textual en HTML; 5. CSS y su importancia en un buen diseño; 6. Herencia y cascada; 7. Creación de estilos de texto con CSS; 8. Construcción del esqueleto de una página web.
3. **Tema 3. Lenguajes de programación interpretado y su recursos en desarrollo web.**
4. **Tema 4. Lenguajes para entornos de programación multiplataforma.**
5. **Tema 5: Programación del lado servidor:** 1. Arquitectura y funcionamiento de un desarrollo basado en cliente-servidor; 2. Sintaxis; 3. Variables y operadores 4. Uso de para aplicaciones web; 5. Programación Orientada a Objetos; 6. Programación de aplicaciones y proyectos.
6. **Tema 6. Acceso a base de datos:** 1. Controladores y direcciones; 2. Sintaxis; 3. Acceso básico a los datos; 4. Transacciones; 5. Metadatos; 6. Integración de aplicaciones con PHP.
7. **Tema 7. Servicios Web:** 1. Introducción a los servicios Web; 2. XML-RPC; 3. SOAP; 4. WSDL y UDDI.
8. **Tema 8. Seguridad en internet:** 1. Cifrado; 2: Firma digital, 3. Vulnerabilidad y ataques.

BIBLIOGRAFÍA

1. Consorcio gestor de la web. <http://www.w3.org/>
2. Servidor web de la asignatura. <http://tutatis.albacete.org/info.html>
3. Una colección de buenos tutoriales. <http://www.w3schools.com/>



**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-REQUISITO	:	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 riesgo que coloque 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 autentificación más importantes, identificando sus características.

PROGRAMA ANALÍTICO

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

LABORATORIOS Y EXPERIENCIAS PRÁCTICAS

Laboratorio: Uso de herramientas y desarrollo de programas que aplican los conceptos teóricos estudiados.

METODOLOGÍA

El curso se desarrolla en sesiones de teoría y práctica. En las sesiones de teoría, el docente presenta los conceptos, fundamentos y aplicaciones, mientras que en las sesiones prácticas se resuelven casos hipotéticos o reales cuya solución es analizada en detalle.

El trabajo en equipo, el desarrollo de la comunicación y la expresión es promovido y evaluado mediante trabajos grupales que serán expuestos en clase, para lo cual se conformarán equipos de no más de 4 integrantes.

En todas las sesiones se promueve la participación activa del alumno.

BIBLIOGRAFÍA

Plan de Estudios 2018

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. Oppiger, 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-REQUISITO	:	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 de 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:** I: Historia de la inteligencia artificial II: Cuestiones filosóficas. III: La prueba de Turing. IV: Experimento de pensamiento del “Cuarto Chino” de Searle. V: Temas típicos en IA. VI: Definiciones fundamentales. VII: Comportamiento óptimo vs. comportamiento actuando como humano. VIII: Razonamiento óptimo vs. razonamiento actuando como humano. IX: Preguntas filosóficas. X: Modelando el mundo. XI: El rol de la heurística.

2. **Búsqueda y Satisfacción de la Restricción:** I: Problemas de espacio. II: Búsqueda de fuerza bruta (respiro primero, profundidad primero, profundidad primero con profundización iterativa). III: Búsqueda del mejor primero (mejor primero genérico, algoritmo de Dijkstra, A*, admisibilidad de A*). IV: Juegos de dos jugadores (búsqueda mínima , poda alfa-beta). V: Satisfacción de la restricción (backtracking o métodos de búsqueda local y seguimiento).
3. **Representación del Conocimiento y Razonamiento:** I: Inferencia no monotécnica. II: Razonamiento probabilístico. III: Teorema de Bayes
4. **Búsqueda Avanzada:** I: Algoritmos genéticos II: Simulated annealing. III: Búsqueda local.
5. **Representación Avanzada del Conocimiento y Razonamiento:** I: Incertezas. a) Razonamiento probabilístico. b) Redes Bayesianas. c) Conjuntos difusos y teoría de la posibilidad. d) Teoría de la decisión.
6. **Agentes:** I: Definición de agentes. II: Arquitectura de agentes. a) Agentes reactivos simples. b) Planeadores reactivos. c) Arquitecturas de capas. d) Ejemplos de arquitecturas y aplicaciones. III: Teoría de agentes. a) Acuerdos. b) Intenciones. c) Agentes de decisión teórica. d) Procesos de decisión de Markov (PDM). IV: Agentes que aprenden. V: Sistemas multiagente. a) Sistemas multiagente inspirados económico. b) Agentes colaborativos. c) Equipos de agentes. d) Modelando agentes. e) Aprendizaje multiagente
7. **Procesamiento de Lenguaje Natural:** I: Gramáticas determinísticas y estocásticas. II: Algoritmos de parsing. III: Métodos basados en corpus. IV: Recuperación de información. V: Traslación de lenguaje. VI: Reconocimiento del habla.
8. **Aprendizaje de Máquina y Redes Neuronales :** I: Definición y ejemplos de aprendizaje de máquina. II: Aprendizaje supervisado. III: Árboles de aprendizaje por decisión. IV: Redes neuronales de aprendizaje. V: Redes de aprendizaje por creencia. VI: Algoritmo del vecino mas cercano. VII: Teoría de aprendizaje. VIII: El problema del sobreajuste. IX: Aprendizaje no supervisado. X: 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 y 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

I. INFORMACIÓN GENERAL

CURSO	:	COMPUTACIÓN GRÁFICA
CÓDIGO	:	CC731
PRE-REQUISITO	:	CC622 - MATEMÁTICA COMPUTACIONAL
DPTO ACADÉMICO	:	
CONDICIÓN	:	OBLIGATORIO
CICLO ACADÉMICO	:	2018-1
CRÉDITOS	:	04 (CUATRO)
HORAS TEÓRICAS	:	02 HORAS SEMANALES
HORAS PRÁCTICAS	:	04 HORAS SEMANALES
SISTEMA DE EVALUACIÓN	:	G
PROFESOR DEL CURSO	:	

II. SUMILLA

El presente curso está concebido para estudiantes del séptimo semestre. Esta asignatura ofrece una Introducción general de los principios, técnicas y herramientas necesarios 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.

III. COMPETENCIAS

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

IV UNIDADES DE APRENDIZAJE

1. **Unidad 1. Introducción a los gráficos por ordenador - Sistemas Gráficos y Modelos:** 1. Objetivos y definiciones básicas; 2. Breve historia 3. Aplicaciones (CAD, Gráficas, esquemas y modelos, Arte y animación por ordenador, etc.); 4. Procesamiento de Imágenes; 5. El sistema visual humano. El modelo de cámara pin-hole. 6. Interfaces gráficas. 7. Arquitecturas gráficas. 8. Introducción a OpenGL.
2. **Unidad 2. Transformaciones geométricas:** 1. Objetivos y definiciones básicas; 2. Sistemas de coordenadas; 3. Transformaciones afines; 4. Coordenadas Homogéneas; 5. Concatenación de transformaciones; 6. Implementación de transformaciones; 7. Posibilidades de OpenGL.
3. **Unidad 3. Transformaciones de visualización:** 1. Proyecciones clásicas y visualización en el ordenador; 2. Definición y posicionamiento de la cámara virtual; 3. Proyecciones simples y en OpenGL; 4. Matrices de proyección; 5. Posibilidades de OpenGL.
4. **Unidad 4. Técnicas de Realismo I:** 1. Introducción. 2. Luz y Color. Propiedades de la Luz. Diagrama cromático CIE. Sistemas de color (RGB, CMY, YIQ, HSV, HLS); 3. Iluminación. Fundamentos. Modelo de Phong. 4. Sombreado de polígonos; 5. Funciones a OpenGL para Sombreamiento e Iluminación; 6. Texturización; 7. Posibilidades de OpenGL.
5. **Unidad 5. Técnicas de Realismo II:** 1. Conceptos Avanzados de Realismo; 2. Modelos de Iluminación Global 3. Traza de Rayos. 4. Modelo de Radiosidad; 5. Fundamentos de Povray y RayShader; 6. Shaders actuales. Lenguaje de programación de Shaders.
6. **Unidad 6. Modelamiento Geométrico.** 1. Representación poligonal de objetos 3D; 2. Curvas poligonales paramétricas y superficies; 3. Representación de geometría sólida constructiva (CSG); 4. Representación implícita de curvas y superficies; 5. Técnicas de subdivisión espacial; 7. Modelos procedurales; 8. Modelos deformables; 9. Subdivisión de superficies. 10. Modelamiento de multiresolución; 11. Reconstrucción.

V METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, y es complementada con casos y soluciones reales relacionados a la especialidad. Incluye tutoría académica permanente en forma semanal según horarios fuera de clase.

VI SISTEMA DE EVALUACIÓN

Sistema de Evaluación G:

Cantidad de Prácticas o Trabajos Calificados: (06)

El Promedio Final (PF) se calcula tal como se muestra a continuación:

EP: Examen Parcial. (Peso 1)

EF: Examen Final. (Peso 1)

PP: Promedio de Prácticas (Peso 1). Se obtiene del promedio aritmético de las cinco (05) mejores notas de las prácticas o trabajos calificados.

$$PF = (EP + EF + PP) / 3$$

VIII 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 Informatica Grafica. 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 Languaje Cookbook, Second Edition. D. Wolff, Pact Open Spurce, 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 CONCURRENTE Y DISTRIBUIDA
CÓDIGO	:	CC741
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC642 - PROGRAMACIÓN PARALELA
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

El Curso de Programación Concurrente y Distribuida está directamente ligada con la evaluación de los sistemas operativos y por ello subdivida en dos temáticas principales.

Por un lado, la programación concurrente 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 concurrente y tiempo real, planificando sus tareas, así como un uso eficiente de la memoria.

PROGRAMA ANALÍTICO

1. **Tema 1. Introducción:** Conceptos fundamentales de la programación concurrente y en tiempo real.
2. **Tema 2. Sincronización con Espera Activa.**
3. **Tema 3. Comunicación por Memoria Compartida:** 1. Semáforos, 2. Regiones Críticas Condicionales, 3. Monitores
4. **Tema 4. Paso de Mensajes.**
5. **Tema 5. Tiempo Real.**
6. **Tema 6. Introducción a la programación distribuida:** 1. Nociones básicas,.2. Arquitectura de sistemas distribuidos, 3. Middleware,
7. **Tema 7. Comunicación entre procesos distribuidos:** 1. Primitivas básicas de comunicación, 2. Comunicación por UDP y TCP en Java, 3. Marshalling, 4. Comunicación de grupos
8. **Tema 8. Objetos distribuidos e invocación remota:** 1. Invocación remota, 2. Comunicación entre objetos distribuidos, 3. Llamada de procedimiento remoto.
9. **Tema 9. Tecnologías Distribuidas Modernas:** 1. Clustering, 2. Tecnologías Grid y Cloud (virtualización), 3. Servicios Web, 4. Computación móvil y ubícuo.,5. Redes P2P
10. **Tema 10. Sincronización en Sistemas Distribuidos:** 1. Relojos físicos y lógicos, 2. Estados globales, 3. Exclusión mutua distribuida, 4. Elecciones
11. **Tema 11. Transacciones y control de concurrencia:** 1. Transacciones, 2. Control de concurrencia en transacciones, 3. Control optimista de concurrencia, 4. Ordenación por marcas de tiempo, 5. 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-REQUISITO	:	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. Tema 1. Introducción a la interacción persona-ordenador:** 1. Definición del concepto; 2. Multidisciplinariedad; 3. Historia; 4. Conceptos fundamentales en IHC; 5. Diseño centrado en el usuario
- 2. Tema 2. Elementos de la IPO: diseño, personas y tecnología:** 1. Tecnología de la interacción; 2. El factor humano; 3. El diseño.
- 3. Tema 3. Diseño centrado en el usuario:** 1. Diseño centrado en el usuario; 2. Principios clave para el diseño con usabilidad; 3. Ingeniería de la Usabilidad; 4. El ciclo de vida de la Ingeniería de la Usabilidad; 5. Diseño centrado en el usuario; 6. Propuesta de Greenberg; 7. Propuesta de diseño de la UPA (Asociación de profesionales de la usabilidad); 8. Diseño centrado en el uso; 9. OVID: Object, View and Interaction Design; 10. Tendencia: El desarrollo de interfaces de usuario basado en modelos (Mb-UIDE).
- 4. Tema 4. Tecnología, diversidad y accesibilidad:** 1. El reto de la diversidad; 2. Estrategias para afrontar la diversidad; 3. Accesibilidad web; 4. Evaluación de la accesibilidad.
- 5. Tema 5. HCI en el Proceso de la Ingeniería del Software.**
- 6. Tema 6. Usabilidad y modelos de calidad centrados en la usabilidad:** 1. Definiciones previas y estándares internacionales; 2. Principios de diseño; 3. La usabilidad como proceso y como producto; 4. La calidad en uso y su relación con la usabilidad; 5. Factores, atributos y métricas; 6. Usabilidad y diseño.
- 7. Tema 7. Conoce al usuario y sus tareas:** 1. Análisis de tareas; 2. Personas.
- 8. Tema 8. Especificación de la calidad:** 1. La experiencia disponible: las guías de estilo y los patrones de interacción/usabilidad; 2. Tendencia: El desarrollo de interfaces de usuario basado en modelos (Mb-UIDE); 3. Especificación de la interfaz de usuario: un ejemplo el lenguaje usiXML.
- 9. Tema 8. Evaluación de la usabilidad: métodos y técnicas:** 1. Métodos de evaluación de la usabilidad: clasificación; 2. Otros métodos de evaluación: en búsqueda de la evaluación automática
- 10. Tema 9. Accesibilidad: criterios y herramientas de evaluación:** 1. Accesibilidad: criterios y herramientas de evaluación; 2. Criterios WAI; 3. Section 508; 4. Herramientas; 5. Recomendaciones; 6. 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. GnomeGuías 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-REQUISITO	:	NINGUNA
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (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. Tema 1. Investigación científica y tecnológica.

Plan de Estudios 2018

- 2. Tema 2.** Búsqueda y difusión de la investigación.
- 3. Tema 3.** Diseño de proyectos de investigación.
- 4. Tema 4.** Desarrollo de un tema de investigación.
- 5. Tema 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.1996
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-REQUISITO	:	CC741 - PROGRAMACIÓN CONCURRENTE 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 término 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. **Tema 1. Introducción a un Grid:** 1. Complejidad de los sistemas distribuidos; 2. Definir la computación en la Nube. Examinar las características de la computación en la Nube; 3. Evaluar la proposición de valor
2. **Tema 2. Entender la arquitectura de la Nube:** 1. Infraestructura, plataformas, appliance virtuales, protocolos de comunicación, aplicaciones; 2. Conectarse a la Nube; 3. Computación bajo demanda (On-demand Computing)
3. **Tema 3. Servicios y aplicaciones por tipo.** Infraestructura como servicio (IaaS): 1. Plataforma como Servicio (PaaS); 4. Software como Servicio (SaaS); 5. Autentificación de usuario Grid; 6. Conceptos, arquitecturas y servicios
4. **Tema 4. Entender la abstracción y la virtualización:** 1. Utilizar tecnologías de virtualización; 2. Equilibrio de carga; 3. La nube de Google; 4. Hipervisores; 5. Trasladar aplicaciones; 6. Grid Middleware
5. **Tema 5. Planificación de capacidad:** 1. Definir una línea de base y una métrica; 2. Tipos de instancias y servidores; 3. Capacidad de red; 4. Globus
6. **Tema 6. Administrar la Nube:** 1. Administración de ciclos de vida; 2. Productos de administración de Nube; 3. Estándares de administración de nubes emergentes; 4. Recursos de administración de un Grid
7. **Tema 7. Entender la seguridad en la Nube.** Asegurar la Nube. Asegurar los datos, cifrado. Establecer identidad y presencia. Web Services Resource Framework (WSRF)
8. Tema 8. Entender servicios y aplicaciones: Entender la arquitectura orientada a servicios. SOA Problemas de rendimiento en un Grid
9. **Tema 9. Trasladar aplicaciones a la Nube:** Creación de un mapa de funcionalidad. Atributos de aplicaciones. Economía en un Grid
10. **Tema 10. Trabajar con el almacenamiento basado en la Nube.** Aprovisionar el almacenamiento en la Nube. Explorar las soluciones de copia de seguridad en la Nube. Interoperabilidad del almacenamiento en la Nube. Trabajar con software de productividad. Utilizar aplicaciones de productividad. Cloud público, privado e híbrido.
11. **Tema 11. Fog Computing:** Edge level. Core level. Optimizar recursos y almacenamiento. Protocolos de comunicación. Procesamiento de Eventos complejos. Despliegue de tecnologías en Fog Computing. 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	:	CC8323
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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. **Tema 1. Introducción:** 1. Introducción a la robótica autónoma, tipos de robots, partes del robot, sensores, potencia, control, inteligencia, comunicación.
2. **Tema 2. Movilidad:** Conocer los distintos medios de locomoción aplicables a la robótica, sus particularidades dinámicas y campos de aplicación más adecuados. Métodos

de locomoción; Conceptos de locomoción; Cinemática de robots móviles; Sistemas de control simple.

3. **Tema 3. Detección y percepción:** Ser capaz de aplicar métodos de reconocimiento de patrones y de aprendizaje computacional en el análisis de datos sensoriales y para la toma de decisiones en sistemas robóticos. Sensor y tecnologías de medición; Robot Vision; Incertidumbre de modelado
4. **Tema 4. Localización y Planificación de trayectorias:** La detección de la posición de un robot con respecto a su medio ambiente, utilizando la información sobre el entorno recogida por el robot. Representación de creencias y modelo de error de odometría; Localización basada en mapas probabilísticos y localización de Markov; Métodos de localización de Montecarlo; Métodos de localización del filtro de Kalman.
5. **Tema 5. Robots autónomos y navegación:** Un agente (inteligente) percibe el medio ambiente a través de sensores y actúa racionalmente sobre el entorno con sus efectores. Evitación de obstáculos, Planificación de ruta, 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. 2^a 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-REQUISITO	:	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. **CAPÍTULO 1:** Introducción a la minería de datos, Tecnologías relacionadas: Aprendizaje automático, DBMS, OLAP, Estadísticas, Objetivos de minería de datos, Etapas del proceso de minería de datos Técnicas de minería de datos, Métodos de representación del conocimiento.
2. **CAPÍTULO 2:** Almacén de datos y OLAP, Data Warehouse y DBMS, Modelo de datos multidimensionales, Preprocesamiento de datos, Limpieza de datos, Transformación de datos, Reducción de datos Discretización y generación de jerarquías conceptuales.
3. **CAPÍTULO 3:** Algoritmos de minerías de datos, Reglas de asociación, clasificación Predicción, Regresión Lineal, Clasificación estadística(bayesiana), redes bayesianas, modelos lineales, modelos no lineales, métodos nearest neighbor.
4. **CAPÍTULO 4:** Redes neuronales, Máquinas de vectores soporte , Descubrimiento de asociaciones y detección de anomalías, Cluster.
5. **CAPÍTULO 5:** Minería de datos en textos: Naive Bayes multinomial. 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. 978-3-319-10246-72015.
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.
11. Jiawey Han y Micheline Kamber Data Mining: Concepts and techniques. Morgan Kauffmann. 2006.
12. M. Berry, G. Linoff. Data mining techniques for marketing, sales and customer support. Whiley. 1997.
13. 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-REQUISITO	:	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. Tema 1. Introducción a la programación evolutiva.

Plan de Estudios 2018

2. **Tema 2. Conceptualización dentro de la programación evolutiva.**
3. **Tema 3. Técnicas de selección.**
4. **Tema 4. Técnicas de cruza.**
5. **Tema 5. Mutación y ajuste de parámetros.**
6. **Tema 6. Manejo de restricciones y operadores avanzados.**
7. **Tema 7. Algoritmos basados en enjambres.**
8. **Tema 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/~ccarlo/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



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	ANÁLISIS EN MACRODATOS
CÓDIGO	:	CC931
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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.

PROGRAMA ANALÍTICO

1. **Tema 1. Introducción al Big Data:** 1. Introducción. Gestión y análisis y visualización del Big Data; 2. Bases de datos NoSQL; 3. Escalabilidad, eficiencia y efectividad; 4. Arquitectura de BBDD e independencia de datos; 5. Large small data.
2. **Tema 2. Aprendizaje automático en grandes volúmenes de datos:** 1. Función de pérdida (Loss function), estrategias de validación (leaving-one-out, cross-validation); 2. Clustering paramétrico y no paramétrico; 3. Deep Learning: RBM, Auto-encoders, DBN/DNN; Learning from Data Streams.

3. **Tema 3. Aplicaciones con datos intensivos.** 1. Modelo de programación en Hadoop Map-Reduce; 2. Aplicaciones en el paradigma Map-Reduce; 3. Algoritmos en grandes volúmenes.
4. **Tema 4. Visualización de datos:** 1. Visualización de datos crudos; 2. Representaciones gráficas de datos procesados; 3. Tipos de visualización de datos según las necesidades del análisis
5. **Tema 5. Herramientas de última generación:** 1. Pig Latin y Funciones definidas por el usuario; 2. Scala; 3. Apache Spark; 4. GraphX, Bagel, Spark SQL, MLib
6. **Tema 6. Big Data y Tiempo Real:** 1. Arquitecturas de propósito general: combinar procesamiento en lotes y stream analytics; 2. Arquitecturas Lambda; 3. 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 FÍSICA

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	INGENIERÍA DE SOFTWARE
CÓDIGO	:	CC541
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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

- **Tema 1.: Introducción a la Ingeniería del Software:** 1.1. Origen y definiciones básicas. 1.2. Conceptos fundamentales
- **Tema 2: Ingeniería de Requisitos Software:** 2.1. Introducción, 2.2. Definición de Requisito, Tipos y Propiedades, 2.3. Modelos de Proceso de IR. Propuesta metodológica,

2.4. Etapa de Adquisición de Requisitos, 2.5. Etapa de Análisis de Requisitos, 2.6. Etapa de Validación y Verificación de Requisitos

- **Tema 3. Modelado de Requisitos con UML 2.0. - Diagramas de Casos de Uso:** 3.1. Introducción, 3.2. Definición y especificación de Casos de Uso, 3.3. Notación gráfica, 3.4. Tipos de Relaciones en un Diagrama de Casos de Uso, 3.5. Ejemplos y casos prácticos.
- **Tema 4. Análisis Orientado a Objetos con UML 2.0:** 4.1. Introducción al modelado de SW con UML 2.0, 4.2. Diagramas de Clases de Dominio, 4.3. Diagramas de Clases de Análisis, 4.4. Diagramas de Actividad
- **Tema 5. Diseño Orientado a Objetos con UML 2.0:** 5.1. Diagramas de Estados, 5.2. Diagramas de Secuencia, 5.3. Diagramas de Comunicación, 5.4. Diagramas de Tiempos, 5.5. Diagramas de Componentes, 5.6. Diagramas de Despliegue
- **Tema 6. Desarrollo de Sistemas Orientado a Objetos. Proceso Unificado de Desarrollo (RUP):** 6.1. Introducción, 6.2. Principios básicos de RUP, 6.3. Fases y Etapas de RUP.
- **Tema 7. Gestión y Pruebas de software:** 7.1. Gestión de Configuración Software, 7.2. Pruebas de Software: Introducción y Principios, 7.3. Pruebas de Software: Estrategias y Procesos
- **Tema 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	:	ALGORITMOS BIOINSPIRADOS
CÓDIGO	:	CC
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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 Bio - Inspiración 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 cibernetica.

PROGRAMA ANALÍTICO

1. Computación Basada en Modelos Naturales:

Los algoritmos bioinspirados simulan el comportamiento de sistemas naturales para el diseño de métodos heurísticos no determinísticos de búsqueda, aprendizaje, comportamiento.

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 ESTRUCTURA DE DATOS AVANZADO
CÓDIGO	:	CC020
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC531 – ANALISIS Y DISEÑO DE ALGORITMOS
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

Este curso es una curso donde se exponen temas avanzados de algoritmos y estructura de datos que sirve la ____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 del Algorítmica y ejecuta los principios de esta ciencia de la computación.
- Describe los procesos de construcción Algoritmos identificando las propiedades asociadas que se generan de ellos apreciando su influencia en la ciencia e ingeniería.

PROGRAMA ANALÍTICO

1. **CAPÍTULO 1:** Introducción. Definición de problemas básicos de estructura de datos avanzados: fibonacci Heaps, Van Emde Boas Priority Queues, Estructura de Datos Dinámicos.
2. **CAPÍTULO 2:** Descripción de Algoritmos para Cadenas con el tema de el algoritmo Rabin-Karp Fingerprinting.
3. **CAPÍTULO 3:** Flujo Máximo con los temas de Flujo de costo mínimo y Bipartie Matching
4. **CAPÍTULO 4:** Programación Lineal, formulación de problemas de programación lineal, Dualidad, Simplex, Punto Interior, el algoritmo Ellipsoid.
5. **CAPÍTULO 5:** Algoritmos en Línea, Ski Rental,Problemas de River Search, el problema de los k-server.
6. **CAPÍTULO 6:** Algoritmos de Aproximación, NP-Hardness, Algoritmos de aproximaciòn Greedy, Programaciòn Dinámica.
7. **CAPÍTULO 7:** Geometria Computacional, Convex Hull, Line- Intersection, Sweep Lines

BIBLIOGRAFÍA

1. Shaffer, Clifford A.: Data Structures and Algorithm Analysis in C++, Third Edition, Dover Publications, 2013. (En línea.)
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.
- 5.



UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ciencias

SÍLABO

I. INFORMACIÓN GENERAL

Curso	: Auditoría de Sistemas
Código	:
Pre-requisito	: Calidad de software
Dpto. Académico	:
Condición	: Electivo
Ciclo Académico	: 2018-1
Créditos	: 4
Horas teóricas	: 2 horas semanales
Horas prácticas	: 4 horas semanales
Sistema de Evaluación	: G
Profesor del curso	: ...

II. 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.

En el curso se tratarán los siguientes contenidos:

- Conceptos Generales
- Sistema de Control de Tecnologías de Información
- Examen de Gestión TI
- Examen de Seguridad Informática
- Examen de Sistemas Aplicativos
- Examen de Proyectos Informáticos
- MODELO COBIT
- Examen de Aseguramiento de Calidad

III. COMPETENCIAS

- 1) Desarrollar en los estudiantes su capacidad crítica y evaluadora.
- 2) Dar a conocer los sistemas de control que existen en las empresas que hacen uso intensivo de las tecnologías de información.
- 3) 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.
- 4) Conocer el software de auditoria existente en el mercado, sus aplicaciones y formas de utilización.
- 5) Elaborar los dictámenes de auditoria para que pueda realizar funciones de auditora de sistemas.

IV. UNIDADES DE APRENDIZAJE

CAPÍTULO 1: CONCEPTOS GENERALES / 4 HORAS

Introducción a la Auditoria, Similitudes y diferencias de los conceptos: evaluación, control y supervisión, Aplicaciones prácticas del uso de estos conceptos. Tipos y Formas.

Auditoría de Sistemas: conceptos, objetivos y ámbito de trabajo. Clases de Auditoría de Sistemas

Revisión de los conceptos de sistemas, teoría de sistemas y el enfoque integral de la auditoria, desarrollo de proyectos de innovación tecnológica, Organización, administración y funcionamiento de una empresa computarizada. Sistemas de control: conceptos y tipos. Discusión caso práctico

CAPÍTULO 2: SISTEMA DE CONTROL DE TECNOLOGÍAS DE INFORMACIÓN / 2 HORAS

Etapas y procesos de la auditoria. Metodología a desarrollar en el curso. Estructura del Sistema de Control de Tecnologías de Información. Presentación de las Normas a utilizar en el desarrollo del curso. Presentación y exploración de las Normas a utilizar en Internet. Ejercicios prácticos. Herramientas de software en cada proceso de auditoria

CAPÍTULO 3: EXAMEN DE GESTIÓN TI / 4 HORAS

Primer Examen de Auditoría: Gestión de la Gerencia de Sistemas y de los servicios informáticos existentes. Revisión del sistema de control:

- Controles organizacionales y Controles administrativos
- Controles operativos al uso de los recursos de cómputo.
- Controles operativos al uso de la información.
- Controles operativos a la confiabilidad de la información. Revisión de las Normas aplicables a este tipo de examen.

Desarrollo de un caso práctico. Programa de Trabajo. Determinación de Criterios a aplicar Identificación de Observaciones Conclusiones y Recomendaciones

CAPÍTULO 4: EXAMEN DE SEGURIDAD INFORMÁTICA / 4 HORAS

Segundo Examen: Evaluación del sistema de seguridad informático. Revisión de los conceptos de Seguridad informática y de las normas existentes. Ejercicios de aplicación de Norma NTP INDECOPI/ISO17799

Desarrollo de un caso práctico. Técnicas de Auditoría de Sistemas: Cuestionario general, Listas de chequeo, Muestreo, Transacciones erróneas, Violación de seguridad.

CAPÍTULO 5: EXAMEN DE SISTEMAS APLICATIVOS / 6 HORAS

Auditoría de sistemas aplicativos. Revisión de:

- Controles de entrada y salida.
- Controles de proceso.
- Controles de librería.
- Controles de acceso: lógico y físico.

Uso de herramientas de Software. Revisión de las Normas aplicables a este tipo de examen.

Revisión de sistemas informáticos tipo: Análisis del sistema y de los procedimientos operativos. Puntos fuertes y débiles de los controles internos y del sistema de seguridad. Determinación de Criterios a aplicar. Recomendaciones y análisis comparativo entre grupos.

Desarrollo de un caso práctico: Programa de trabajo. Determinación de Criterios a aplicar Identificación de Observaciones Conclusiones y Recomendaciones

CAPÍTULO 6: EXAMEN DE PROYECTOS INFORMÁTICOS / 2 HORAS

Auditoría de Proyectos informáticos: Ciclo de vida de un proyecto. Controles de la fase de planeamiento. Controles de la fase de ejecución. Controles de la fase de implantación o lanzamiento. Controles administrativos y de seguimiento. Determinación de Criterios a aplicar

CAPÍTULO 7: MODELO COBIT / 4 HORAS

COBIT: Normas de Gestión TI, Dominios y Puntos de Control. Diferencia de aplicación con la metodología seguida.

Desarrollo de Caso práctico: Análisis del sistema y de los procedimientos operativos. Puntos fuertes y débiles de los controles internos. Recomendaciones y análisis comparativo.

Elaboración del Dictamen de Auditoría: Formato de presentación. Metodología para la elaboración, seguimiento y control de un Dictamen de Auditoría. Seguimiento de Medidas correctivas.

CAPÍTULO 8: EXAMEN DE ASEGURAMIENTO DE CALIDAD / 2 HORAS

Nuevo enfoque de auditoria preventiva que permite el Aseguramiento de la calidad de los sistemas de información. Aplicación de Norma ISO.

V. METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor expone los conceptos, buenas prácticas, herramientas y técnicas, complementa con:

- Uso de casos reales sobre calidad de software y de información.
- Ejercicios grupales en aula para el desarrollo de habilidades mediante casos prácticos.
- Investigación en temas de interés.
- Tutoría académica permanente en forma semanal según horarios fuera de clase.

BIBLIOGRAFÍA

Soy Aumatell, Cristina; “Auditoría de la Información”, Editorial UOC, Barcelona, 2003.

Tecnología de la información. Código de buenas prácticas para la gestión de la seguridad de la información NTP - ISO/IEC 17799:2004.

Comisión de Reglamentos Técnico y Comerciales –CRT de **INDECOP**I. Presidencia del Consejo de Ministros – Oficina Nacional de Gobierno Electrónico del Perú – ONGEI. 2004.

Muñoz Razo, Carlos; “Auditoría en Sistemas Computacionales”, Universidad del Valle de México, Pearson Educación, 2002.

COBIT Management Guidelines.

COBIT Steering Committee and the IT Governance InstituteTM. 1 T GOVERNANCE INSTITUTE. 2002.

Echenique, José Antonio; “Auditoría en Informática”, Universidad Nacional Autónoma de México. Mc Graw Hill, 2001

Metodología para la auditoría de sistemas automatizados **Weights**. 2001. Deuts M. 2000

Normas de la **ONGEI** sobre sistemas, seguridad y sistemas de información Oficina Nacional de Gobierno Electrónico e Informática. 2003.

Normas de Control Interno de la **Contraloría General de la República**. Contraloría General de la República. 1998.

Normas del **INEI** sobre seguridad y sistemas de información Instituto Nacional de Estadística e Informática. 1995.

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

I. INFORMACIÓN GENERAL

CURSO	:	BIOLOGÍA COMPUTACIONAL
CÓDIGO	:	CC021
PRE-REQUISITO	:	CH061 - BIOLOGÍA CC622 - MATEMÁTICA COMPUTACIONAL
DPTO ACADÉMICO	:	
CONDICIÓN	:	ELECTIVO
CICLO ACADÉMICO	:	2018-1
CRÉDITOS	:	04 (CUATRO)
HORAS TEÓRICAS	:	02 HORAS SEMANALES
HORAS PRÁCTICAS	:	04 HORAS SEMANALES (4 LABORATORIO)
SISTEMA DE EVALUACIÓN	:	G
PROFESOR DEL CURSO	:	

II. SUMILLA Y DESCRIPCIÓN DEL CURSO

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 incluyen 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.

III. COMPETENCIAS

1. Aplica los conocimientos de biología, computación y matemáticas apropiadas para el análisis y generación de información utilizadas en el campo de la biología computacional.
2. Demuestra su capacidad de análisis identificando y definiendo los requerimientos computacionales apropiados para la solución de problemas reales.

3. Utiliza técnicas y herramientas adecuadas para el análisis y generación de información bioinformática.

IV UNIDADES DE APRENDIZAJE

1. **Unidad 1. Introducción a la Biología Computacional**
1. Introducción; 2. Conceptos básicos de Biología Molecular; 3. Problemas clásicos de Bioinformática; 4. Herramientas de recolección y almacenamiento de secuencias de laboratorio. 5. Recursos de Software, introducción a BLAST, CLUSTAL Omega; 6. Cadenas, Grafos y algoritmos
2. **Unidad 2. Alineamiento de Secuencias:** 11. Introducción al alineamiento de secuencias; 2 comparación de pares de secuencias; 3. Alineamiento de secuencias Global; 4. Alineamiento de secuencias múltiples; 5. Cadenas Ocultas de Markov; 6. Métodos exactos, aproximados y heurísticos del alineamiento de secuencias; 7. Problemas derivados del alineamiento de secuencias.
3. **Unidad 3. Clustering:** 1. El Problema del Clustering; 2. Clustering Jerárquico. 3. Algoritmo Neighbour Joining; Algoritmo Average Linkage. 4. Clustering no Jerárquico o K- Means
4. **Unidad 4. Árboles Filogenéticos:** 1. Introducción a la Filogenia. 2. Algoritmos Comunes 3. Aplicaciones Biológicas 4. Algoritmos Exactos; 5. Algoritmos Probabilísticos.
5. **Unidad 5. Mapeo de Secuencias:** 1. El Problema de Double Digest y Partial Digest, 2; Técnicas utilizadas en el mapeo de secuencias; 3. Mapeo con non – Unique Probes; 4. Mapeo con Unique Probes; 5. Grafos de Intervalos; 6. Mapeo con señales de frecuencias de restricción.
6. **Unidad 6. Introducción a la estructura de proteínas.** 1. Fundamentos biológicos de las proteínas; 2. Motivación para la predicción de estructuras de las proteínas; 3. Alineamiento rígido de proteínas; 4. Técnica de alineamiento por Hashing geométrico

V METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, y es complementada con casos y soluciones reales relacionados a la especialidad. Incluye tutoría académica permanente en forma semanal según horarios fuera de clase.

VI SISTEMA DE EVALUACIÓN

Sistema de Evaluación G:

Cantidad de Prácticas o Trabajos Calificados: (06)

El Promedio Final (PF) se calcula tal como se muestra a continuación:

EP: Examen Parcial. (Peso 1)

EF: Examen Final. (Peso 1)

PP: Promedio de Prácticas (Peso 1). Se obtiene del promedio aritmético de las (05) mejores notas de las prácticas o trabajos calificados.

$$PF = (EP + EF + PP)/3$$

VIII 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 INGENIERIA

Facultad de Ciencias

SÍLABO

I. INFORMACIÓN GENERAL

Curso	: Calidad de software
Código	:
Pre-requisito	: Desarrollo de Software
Dpto. Académico	:
Condición	:
Ciclo Académico	: 2018-1
Créditos	: 4
Horas teóricas	: 2 horas semanales
Horas prácticas	: 2 horas semanales
Sistema de Evaluación	: G
Profesor del curso	: ...

II. 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

III. COMPETENCIAS

- 1) Lograr una visión detallada de los conceptos, herramientas y métodos de calidad en el ciclo de vida del desarrollo de software.
- 2) Desarrollar habilidades para realizar el aseguramiento de la calidad de software en el ciclo de vida del desarrollo de software.
- 3) 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.
- 4) Evaluar la calidad de los resultados del proyecto de software y diseñar el proceso de sostenibilidad del proyecto.

IV. UNIDADES DE APRENDIZAJE

CAPÍTULO 1: FUNDAMENTOS DE LA CALIDAD DE SOFTWARE / 2 HORAS

Introducción a la calidad de software: Introducción a la calidad. La calidad del software. Ingeniería de software y calidad. Estándares y modelos de evaluación y mejora de los

procesos software. Introducción a la calidad total. Costos de errores de SW. Entorno de la Calidad de SW.

CAPÍTULO 2: MODELOS DE PROCESOS DE SOFTWARE / 12 HORAS

Gestión de la calidad: Definiciones. Introducción en la gestión de la calidad: planificar, organizar, gestionar el personal, liderar, y controlar. Gestión de riesgos: definición, identificación, análisis, planificación, y monitoreo. Gestión de proyectos de SW: seguimiento y control, mediciones, estimación, análisis post-mortem, verificación y validación, gestión de configuración, métricas, análisis y diseño. Plan de desarrollo de SW: introducción, modelización de procesos, estimación de esfuerzo, planificación de proyectos, seguimiento de procesos, demostración de SMP, y conclusión.

Procesos y Métodos de calidad de software: Factores que Determinan la Calidad de Software (Mc Call). Situación Actual de la Industria del Software. Estándares y Modelos de Evaluación y Mejora de los Procesos de Software. ISO 9000 (Objetivos, Principios). ISO 9126:2001 Modelo de Calidad de Software (Externa e Interna). ISO/IEC 15504 (SPICE): Componentes, Dimensiones, Evaluación de Atributos y Certificación.

Métricas de calidad de software: Introducción. Teoría y Ontología de la medición. Estándares y metodologías. Método de definición. Métricas de software. PSP (Personal Software Process). Planeación en PSP. Métricas del PSP.

Configuración de Software: Línea de producto. Áreas de maduración técnica: ingeniería de SW, gestión técnica, y gestión de la empresa. Gestión de la configuración de software (SCM). Planeamiento de la SCM.

CAPÍTULO 3: ASEGURAMIENTO DE CVALIDAD DEL SOFTWARE / 8 HORAS

El SQA (Aseguramiento de Calidad de SW): Introducción. Organización. Actividades. Plan SQA. Implementación SQA.

El CMMI: Modelo CMMI: Foco, Qué es y Qué no es CMMI? Cómo utilizar CMMI. Conceptos claves: Capacidad, Desempeño, Madurez e Institucionalización. Análisis de costos y beneficios de la implementación del modelo CMMI. Estructura y representación del modelo CMMI. Áreas de procesos, Metas y Prácticas Específicas de CMMI. Verificación y Calidad.

El TMMI: El proceso de prueba. EL TPI. TMMI versus TPI. Niveles del TMMI. Modelos de Mejora de Procesos. Prioridades en TI. ¿Qué es y Qué no es ITIL?. ITIL: Características, Objetivos, Fundamentos, Mejora Progresiva. Áreas Claves. Relación con Otras Metodologías. Aspectos Importantes. Organización y Componentes de una Organización. Perspectivas. Implementación del ITIL.

CAPÍTULO 4: CALIDAD DE INFORMACIÓN / 2 HORAS

Calidad de información: Introducción. Aproximación a la calidad. Medición de los modelos. Calidad de datos. Aspectos de gestión. Estudio de casos prácticos

CAPÍTULO 5: ESTÁNDARES Y NORMAS DE SOFTWARE / 4 HORAS

Modelo de calidad NTP- ISO/IEC 12207 – Procesos del ciclo de vida del software. Procesos principales, procesos de apoyo, procesos organizativos. Estrategias de Implementación. Casos prácticos.

Planes de la calidad de software. Modelo de calidad NTP- ISO/IEC 15504 – Evaluación y Mejora Continua del software. Casos prácticos.

V. METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor expone los conceptos, buenas prácticas, herramientas y técnicas, complementa con:

- Uso de casos reales sobre calidad de software y de información.
- Ejercicios grupales en aula para el desarrollo de habilidades mediante casos prácticos.
- Investigación en temas de interés.
- Tutoría académica permanente en forma semanal según horarios fuera de clase.

VI. SISTEMA DE EVALUACIÓN

Sistema de evaluación G:

1) Las prácticas (P1, P2, P3, P4) y los exámenes parcial (EP), final (EF) y sustitutorios (ES) se calificarán sobre 20 puntos.

2) El promedio de prácticas (PP) resulta de la siguiente fórmula:

$$PP = [(P_1 + P_2 + P_3 + P_4 - \min(P_1, P_2, P_3, P_4))/3].$$

3) El examen sustitutorio (ES) es opcional. El promedio del curso (PC) resulta de la siguiente fórmula:

PC = [(PP + EP + EF)/3].	Si no se da ES
PC = [(PP + max(EP, EF) + ES)/3]	Si se da ES.

1. BIBLIOGRAFÍA

SOMMERVILLE, Ian; “Ingeniería de Software”, 9na. Edición, Addison Wesley, México, 2011.

PIATTINI, Miguel & al; “Calidad de Sistemas Informáticos”; 2da. Edición, RA-MA Editorial, Madrid, 2011.

PRESSMAN, Roger; “Ingeniería de Software: Un Enfoque Práctico”; 7ma. Edición, McGraw-Hill, Madrid, 2010.

Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual (Indecopi)

NTP-ISO/IEC 12207:2004 Tecnología de la Información. Procesos del ciclo de vida del software, 2004.

Guía Técnica sobre evaluación de software para la Administración Pública, 2004.

Software Engineering Institute - Carnegie Mellon University - (CMU SEI).

Capability Maturity Model Integration (CMMI) version 1.2, 2007.

TMMi Foundation

Test Maturity Model Integration (TMMi) Version 1.0 (dd. February, 17th 2008)

Páginas Web:

- www.cmu-sei.edu (SEI = Software Engineering Institute de la CMU)
- www.asqc.org (American Society for Quality Control de la NASA)
- www.indecopi.gob.pe (Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual)
- www.sqi.gu.edu.au/spice/ (SPICE = Software Process Improvements and Capability dEtermination)
- www.ieee.org (Institute of Electrical and Electronic Engineers)
- <http://www.rspa.com/> (R.S. Pressman & Associates, Inc.)
- <http://www.comp.lancs.ac.uk/computing/resources/IanS/> (Ian Sommerville)
- <http://www.calidadadelsoftware.com/> (Calidad de Software)
- <http://www.utilidades-utiles.com/>



**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-REQUISITO	:	TEORÍA DE AUTÓMATAS, LENGUAJES Y COMPUTACIÓN.
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

OBJETIVO

Que el alumno conozca y comprenda los conceptos y principios fundamentales de la teoría de compilación para realizar la construcción de un compilador. 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

I: Historia de los lenguajes de programación. II: Breve revisión de los paradigmas de programación. III: Lenguajes procedurales. IV: Lenguajes orientados a objetos. V: Lenguajes funcionales. VI: Lenguajes declarativos y no algorítmicos. VII: Lenguajes de *scripts*. VIII: Los efectos de la escalabilidad en las metodologías de programación.

2. Introducción a la Traducción de Lenguajes

I: Comparación entre intérpretes y compiladores. II: Fases de traducción del lenguaje (análisis léxico, análisis sintáctico, generación de código, optimización). III: Aspectos de traducción dependientes e independientes de la máquina.

3. Sistemas de Traducción del Lenguaje

I: Aplicación de las expresiones regulares en analizadores léxicos. II: Análisis sintáctico (sintaxis concreta y abstracta, árboles de sintaxis abstracta). III: Aplicación de las gramáticas libres de contexto en un parseo dirigido por tablas o recursivo descendente. IV: Administración de tablas de símbolos. V: Generación de código por seguimiento de un árbol. VI: Operaciones específicas de la arquitectura: selección de instrucciones y alocación de registros. VII: Técnicas de optimización. VIII: El uso de herramientas como soporte en el proceso de traducción y las ventajas de éste. IX: Librerías de programas y compilación separada. X: Construcción de herramientas dirigidas por la sintaxis.

BIBLIOGRAFÍA

1. Aho, A. Compiladores Principios, tecnicas y herramientas. Addison Wesley. 1990.
2. Aho, A., Lam, M., Sethi, R., and Ullman, J. D. Compiladores. Principios, tecnicas y herramientas. Addison Wesley, 2nd edition. ISBN:10-970-26-1133-4. 2008.
3. A.Lemone, K. Fundamentos de Compiladores. CECSA-Mexico. 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 Programación Diseño e Implementación. 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-REQUISITO	:	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. Tema 1. Introducción al deep learning.
2. Tema 2. Recordando el aprendizaje supervisado. Métodos lineales de clasificación, métodos no lineales y métodos ensamblados.
3. Tema 3. Representaciones de alto nivel, características en imágenes.
4. Tema 4. Formulación del aprendizaje como un problema de optimización. Algoritmo de "backpropagation".
5. Tema 5. Redes neuronales conectadas. Multilayer Perceptron (MLP). Fase de entrenamiento y validación en imágenes. Hiperparámetros. Evaluación del rendimiento de los modelos deep learning. Entender el comportamiento del modelo durante la fase de entrenamiento. Reducción de overfitting.
6. Tema 6. Redes neuronales de convolución, arquitectura, capas y operadores.
7. Tema 7. Análisis de redes neuronales, visualización de representaciones. Propiedades: invarianza, covarianza, redundancia e invertibilidad. Mejorar el rendimiento del modelo con "Image augmentation".
8. Tema 8. Redes neuronales recurrentes (R-NN). Predicción de series de tiempo con MLP.
9. Tema 9. Redes de corta-larga memoria (LSTM). Predicción de series de tiempo con R-NN - LSTM. 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	:	PROGRAMACIÓN EN DISPOSITIVOS MÓVILES
CÓDIGO	:	CC010
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	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 medionivel (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.

En este curso se tratarán los siguientes temas:

- Sistemas digitales basados en microprocesador.
- Modelo de programación del 80x86 de Intel.
- Interfaz del ensamblador con el lenguaje C.
- Recursos de programación.
- Entrada / Salida.
- Programación de los recursos hardware básicos del PC.

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

- **CAPÍTULO 1: SISTEMAS DIGITALES BASADOS EN MICROPROCESADOR. MODELO DE PROGRAMACIÓN DEL 80X86 DE INTEL.** Arquitectura básica de un sistema digital basado en microprocesador. Arquitectura básica de un microprocesador. Funcionamiento de un sistema basado en microprocesador. Familia 80x86 como caso particular. Registros internos y arquitectura del 80x86. Acceso y organización de la memoria. Modos de direccionamiento. Directivas y operadores del ensamblador del 80x86. Estructura de un programa en ensamblador. Instrucciones del ensamblador: transferencia de datos, aritméticas y lógicas, de control, de interrupción, etc. Mapa de Memoria del sistema PC. Interrupciones: mecanismo y vectores de interrupción.
- **CAPÍTULO 2: INTERFAZ DEL ENSAMBLADOR CON EL LENGUAJE C. RECURSOS DE PROGRAMACIÓN.** Características generales. El ejemplo del lenguaje C. Los distintos modelos del lenguaje C. Convenios de nomenclatura, paso de parámetros, devolución de resultados. Interrupciones BIOS. Interrupciones DOS. Ejecución de programas desde el DOS. PSP (Prefijo de Segmento de Programa). Tipos de programas: EXE, COM, y residentes (TSR).
- **CAPÍTULO 3: ENTRADA/SALIDA.** Técnicas de programación de entradas y salidas (E/S). Sondeo. Interrupción. DMA. Gestión y programación de las interrupciones en el 80x86: el controlador programable de interrupciones 8259A.
- **CAPÍTULO 4: PROGRAMACIÓN DE LOS RECURSOS HARDWARE BÁSICOS DEL PC.** Teclado. Timer. Reloj de Tiempo Real (RTC). Controladora de Vídeo y Pantalla. Puerto Paralelo. Impresora.

METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, complementada con aplicaciones preferentemente relacionadas a la especialidad respectiva. Tutoría académica permanente en forma semanal según horarios fuera de clase.

SISTEMA DE EVALUACIÓN

Sistema de evaluación G:

Cantidad de prácticas o trabajos calificados: cuatro (04)

El promedio final (PF) se calcula tal como se muestra a continuación:

$$PF = (EP + EF + PP)/3$$

EP: Examen Parcial (Peso 1)

EF: Examen Final (Peso 1)

PP: Promedio de Practicas (Peso 1)

PP: Se obtiene del promedio aritmético de las tres (03) mejores notas de las prácticas o trabajos calificados.

BIBLIOGRAFÍA

- Los microprocesadores Intel. Barry B. Brey, Ed. Prentice-Hall
- IBM PC & XT, Assembly Language. Leo. J. Scallop, Ed. Brady

- Arquitectura, programación y diseño de sistemas basados en microprocesadores (80x86/80186/80286). Yu-Cheng Liu y Glenn A. Gibson, Ed. Anaya



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

SÍLABO

I. INFORMACIÓN GENERAL

CURSO	:	DISEÑO GEOMÉTRICO ASISTIDO POR COMPUTADOR
CÓDIGO	:	CC027
PRE-REQUISITO	:	COMPUTACIÓN GRÁFICA
DPTO ACADÉMICO	:	
CONDICIÓN	:	ELECTIVO
CICLO ACADÉMICO	:	2018-1
CRÉDITOS	:	04 (CUATRO)
HORAS TEÓRICAS	:	02 HORAS SEMANALES
HORAS PRÁCTICAS	:	04 HORAS SEMANALES
SISTEMA DE EVALUACIÓN	:	G
PROFESOR DEL CURSO	:	

II. SUMILLA

Esta asignatura se proyecta hacia la solución de problemas planteados por el ser humano en su adaptación al medio que lo rodea, para la satisfacción de sus necesidades, para lo cual utiliza recursos como la tecnología CAD/CAE/CAM.

III. COMPETENCIAS

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

IV UNIDADES DE APRENDIZAJE

- Aplicaciones del CAGD. Sistemas de representación.
- Entidades de forma libre. Modelos polinomiales. Curvas de Bézier y Bspline. Superficies de Bézier y Bspline.
- Modelos racionales. Curvas y superficies NURBS. Curvas sobre superficies. Software de modelado
- Elementos Finitos

V METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, y es complementada con casos y soluciones reales relacionados a la especialidad. Incluye tutoría académica permanente en forma semanal según horarios fuera de clase.

VI SISTEMA DE EVALUACIÓN

Sistema de Evaluación G:

Cantidad de Prácticas o Trabajos Calificados: (06)

El Promedio Final (PF) se calcula tal como se muestra a continuación:

EP: Examen Parcial. (Peso 1)

EF: Examen Final. (Peso 1)

PP: Promedio de Prácticas (Peso 1). Se obtiene del promedio aritmético de las cinco (05) mejores notas de las prácticas o trabajos calificados.

$$PF = (EP + EF + PP) / 3$$

VIII 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 Didácticos, 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 práctica, J. Ribelles y J. Lluch. Treballs informàtica 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, Pack Open Source, 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)
12. Geometric Modeling M.E. Mortenson John Wiley & Sons. 1997. > CAD/CAM B. Hawkes Paraninfo S.A. 1989. > An Introduction to Computer-Aided Design and Manufacture M.J. Haigh Blackwell Scientific Publications, Computer Science Texts. 1985.



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	FÍSICA COMPUTACIONAL
CÓDIGO	:	CC019
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC622 - MATEMÁTICA COMPUTACIONAL
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	06 (TEORÍA: 02; LABORATORIO 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

El propósito del curso es enseñar al estudiante las ideas de computabilidad usadas en distintas áreas de la física para resolver un conjunto de problemas modelo. A partir de planteamientos analíticos se pretende obtener resultados numéricos reproducibles consistentes, y que predigan situaciones físicas asociadas al problema bajo estudio. El alumno debe asimilar las ideas básicas del análisis numérico, como son las de estabilidad en el cálculo y la sensibilidad de las respuestas a las perturbaciones en la estructura del problema.

En el curso se tratarán los siguientes contenidos:

- ESCALAS, CONDICIÓN Y ESTABILIDAD.
- OPERACIONES MATEMÁTICAS BÁSICAS.
- ECUACIONES DIFERENCIALES ORDINARIAS.
- ANÁLISIS NUMÉRICO DE PROBLEMAS MATRICIALES.
- PROBLEMAS CLÁSICOS Y CUÁNTICOS DE VALORES PROPIOS.
- SIMULACIÓN COMPUTACIONAL.
- ECUACIONES DE EVOLUCIÓN.

COMPETENCIAS

- Resuelve sistema de ecuaciones, ecuaciones diferenciales de una y varias variables que representan fenómenos físicos a través de la aplicación de los métodos numéricos.
- Calcula errores y convergencia de los algoritmos.
- Escribe programas para resolver sistema de ecuaciones, ecuaciones diferenciales de una y varias variables.
- Aplica los métodos numéricos para resolver problemas de la Física.

PROGRAMA ANALÍTICO

- **CAPÍTULO 1: OPERACIONES MATEMÁTICAS BÁSICAS.** Introducción. Sistemas numéricos de punto flotante y lenguajes. Dimensiones y escalas. Errores numéricos y su amplificación. Condición de un problema y estabilidad de un método. Interpolación y extrapolación. Diferenciación numérica. Integración numérica. Evaluación numérica de soluciones.
- **CAPÍTULO 2: ECUACIONES DIFERENCIALES ORDINARIAS. ANÁLISIS NUMÉRICO DE PROBLEMAS MATRICIALES.** Métodos simples. Métodos implícitos y de multipasos. Métodos de RungeKutta. Estabilidad de las soluciones. Orden y caos en el movimiento de dos dimensiones. Inversión de matrices y número de condición. Valores propios de matrices tridiagonales. Discretización de la ecuación de Laplace y métodos iterativos de solución. Solución numérica de ecuaciones diferenciales elípticas en una y dos dimensiones.
- **CAPÍTULO 3: PROBLEMAS CLÁSICOS Y CUÁNTICOS DE VALORES PROPIOS. SIMULACIÓN COMPUTACIONAL.** Algoritmo de Numerov. Integración de problemas con valores en la frontera. Formulación matricial para problemas de valores propios. Formulaciones variacionales. Método de Monte Carlo. Dinámica molecular. Otros algoritmos de simulación. Aplicación a problemas de física de interés actual.
- **CAPÍTULO 4: ECUACIONES DE EVOLUCIÓN.** La ecuación de ondas y su discretización en diferencias finitas. Criterio de Courant. La ecuación de Fourier para el calor y su discretización en diferencias finitas. Estabilidad del esquema.

METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, complementada con aplicaciones preferentemente relacionadas a la especialidad respectiva. Tutoría académica permanente en forma semanal según horarios fuera de clase.

SISTEMA DE EVALUACIÓN

Sistema de evaluación G:

Cantidad de prácticas o trabajos calificados: cuatro (04)

El promedio final (PF) se calcula tal como se muestra a continuación:

$$PF = (EP + EF + PP)/3$$

EP: Examen Parcial (Peso 1)

EF: Examen Final (Peso 1)

PP: Promedio de Practicas (Peso 1)

PP: Se obtiene del promedio aritmético de las tres (03) mejores notas de las prácticas o trabajos calificados.

BIBLIOGRAFÍA

- Kahaner, D., Moler, C., Nash, S., 1989, Numerical methods and software, PrenticeHall, USA.
- Koonin, S. E., Meredith, D. C., 1990, Computational physics (Fortran version), Addison Wesley Publishing Company, USA.
- Gould, H., Tobochnik, J., 1988, An introduction to computer simulation methods: Applications to physical systems, Addison Wesley Publishing Company, USA
- Vesely, F., 1994, Computational physics: An introduction, Plenum Press, USA.



SÍLABO

I. INFORMACIÓN GENERAL

Curso	: Introducción a la Computación Cuántica
Código	: CC028
Pre-requisito	: CQ221, CC622
Dpto. Académico	:
Condición	:
Ciclo Académico	: 2018-1
Créditos	: 4
Horas teóricas	: 4 horas semanales
Horas prácticas	: 2 horas semanales
Sistema de Evaluación	: G
Profesor del curso	:

II. SUMILLA

La computación cuántica es un campo de investigación emergente e interdisciplinario que abarca las matemáticas, la física y la informática, y que explota la naturaleza cuántica de la materia de existir simultáneamente en múltiples estados.

El curso consta de cuatro partes:

1. Fundamentos cuánticos.
2. Algoritmos cuánticos.
3. Algoritmos de factorización y búsqueda cuántica.
4. Criptografía cuántica.

III. COMPETENCIAS

1. Interpreta los principios fundamentales de la mecánica cuántica aplicándolo a la computación cuántica.
2. Comprende la lógica los bits cuánticos para diseñar circuitos cuánticos.
3. Entiende los algoritmos cuánticos y programa simulaciones.
4. Comprende la criptografía cuántica aplicándola a casos reales.

IV. UNIDADES DE APRENDIZAJE

CAPÍTULO 1: FUNDAMENTOS CUÁNTICOS.

Sistemas cuánticos de un solo qubit. Sistemas cuánticos de múltiples qubit. Medición de estados de múltiples qubit. Transformaciones de estados cuánticos. Versiones cuánticas de cálculos clásicos.

CAPÍTULO 2: ALGORITMOS CUÁNTICOS.

Calculando con superposiciones. Noción de complejidad. Algoritmo de Deutsh. Subrutinas cuánticas. Algoritmo de Deutsh-Jozsa. Algoritmo de Bernstein-Vazirani. Algoritmo de Simon. Algoritmo de la computación distribuida. Transformada de Fourier cuántica.

CAPÍTULO 3: ALGORITMOS DE FACTORIZACIÓN Y BÚSQUEDA CUÁNTICA.

Algoritmo de Shor. Algoritmo de Grover.

CAPÍTULO 4: CRIPTOGRAFÍA CUÁNTICA.

Criptografía de clave privada. Amplificación de privacidad y reconciliación de información. Distribución de clave cuántica: Protocolos BB84, B92, E91 y BBM92.

V. METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, complementada con aplicaciones preferentemente relacionadas a la especialidad respectiva. Tutoría académica permanente en forma semanal según horarios fuera de clase.

VI. SISTEMA DE EVALUACIÓN

Sistema de evaluación G:

Cantidad de prácticas o trabajos calificados: cuatro (04)

El promedio final (**PF**) se calcula tal como se muestra a continuación:

$$\text{PF} = (\text{EP} + \text{EF} + \text{PP}) / 3$$

EP: Examen Parcial (Peso 1)

EF: Examen Final (Peso 1)

PP: Promedio de Prácticas (Peso 1)

PP: Se obtiene del promedio aritmético de las tres (03) mejores notas de las prácticas o trabajos calificados.

VII. BIBLIOGRAFÍA

- Quantum Computation and Quantum Information. 10th Anniversary Edition. Michael A. Nielsen & Isaac L. Chuang. 2010
- Quantum Computing. A gentle introduction. Eleanor Rieffel and Wolfgang Polak. 2011
- Quantum Computer Science. An Introduction. N. David Mermin. 2007
- Problems and solutions in quantum computing and quantum information. Willi-Hans Steeb and Yorick Hardy. 2004



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	MULTIMEDIA
CÓDIGO	:	CC009
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC511 - COMPUTACIÓN CENTRADA EN REDES
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	06 (TEORÍA: 02; LABORATORIO 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

Multimedia es la disciplina relacionada con el diseño, implementación y evaluación de sistemas informáticos interactivos para uso de seres humanos buscando optimizar el desempeño de los mismos por medio de los componentes que la conforman (Imagen, Sonido, Texto, Animación y Video), por lo que la asignatura aporta conocimientos que serán aplicados en cualquier otra asignatura relacionada con el desarrollo de aplicaciones que utilicen interfaz gráfica de usuario (GUI).

En el curso se tratarán los siguientes contenidos:

- Multimedia. Conceptos.
- La imagen.
- El sonido.
- La animación.
- El video.

COMPETENCIAS

- Reconoce los elementos de una interacción humano – máquina.
- Crea videos, sonidos, imágenes y animaciones.
- Reconoce los componentes de una interfaz gráfica de usuario
- Diseña el mapa de navegación de una herramienta multimedia.
- Aplica estándares de diseño de interfaces gráficas de usuario.
- Usa herramientas y metodologías para medir la usabilidad y accesibilidad.

PROGRAMA ANALÍTICO

- **CAPÍTULO 1: MULTIMEDIA. CONCEPTOS. IMAGEN.** Definición de Multimedia. Etapas de un proyecto multimedia. Hardware y Software que se utiliza en multimedia. Definición de imagen. Formas de almacenar y crear imágenes. Tipos de imagen. Formatos de imagen. Practica con Gimp.
- **CAPÍTULO 2: SONIDO.** Definición de sonido. Tipos de sonido. Formatos de sonido. Practica con la grabadora de sonido y audacity.
- **CAPÍTULO 3: ANIMACIÓN.** Definición de animación. Tipos de animación. Formatos de animación. Practica con Adobe Director.
- **CAPÍTULO 4: VIDEO.** Definición de video. Formatos de video. Práctica Adobe Premiere y Windows Movie Maker.

METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, complementada con aplicaciones preferentemente relacionadas a la especialidad respectiva. Tutoría académica permanente en forma semanal según horarios fuera de clase.

SISTEMA DE EVALUACIÓN

Sistema de evaluación G:

Cantidad de prácticas o trabajos calificados: cuatro (04)

El promedio final (PF) se calcula tal como se muestra a continuación:

$$PF = (EP + EF + PP) / 3$$

EP: Examen Parcial (Peso 1)

EF: Examen Final (Peso 1)

PP: Promedio de Practicas (Peso 1)

PP: Se obtiene del promedio aritmético de las tres (03) mejores notas de las prácticas o trabajos calificados.

BIBLIOGRAFÍA

- Burgos, Daniel. Director practico. Mc Graw Hill. 2002.
- Peña, Jaime. Flash practico. Mc Graw Hill. 2002.
- Vaughan. Tay. Multimedia. Mc Graw Hill. 2002.
- Winn L. Rosch. Todo sobre multimedia. Ed.Prentice Hall.
- Haskin,David. Multimedia fácil. Ed.Prentice Hall.
- Wodaski, Ron. Multimedia para todos. Ed.Prentice Hall.
- Martín, Nacho. Multimedia. Ed.Anaya.



UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE FÍSICA

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	NUCLEO Y REDES PARA COMPUTACION PARALELA
CÓDIGO	:	CC027
CRÉDITOS	:	02 (DOS)
PRE-REQUISITO	:	CC461 - COMPILADORES CC481 – ADMINISTRACION DE REDES
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	04 (TEORÍA: 01, LABORATORIO: 03)
SISTEMA DE EVALUACIÓN	:	G

OBJETIVO

Construir un sistema operativo y las redes de comunicaciones que permitan la utilización de múltiples núcleos.

PROGRAMA ANALÍTICO

- 1. Arquitectura, diseño e implementación del ExoKernel.**
- 2. Arquitectura, diseño e implementación de librerías que pueden formar parte del sistema operativo (libases).**
- 3. Definición del Control y la Seguridad en las librerías de sistema operativo (libOS)**
- 4. Multiplexación de redes**
- 5. Protección de discos**
- 6. Protocolos: Arquitectura, diseño, e implementación**
- 7. Diseño e implementación del protocolo HTTP)**
- 8. Diseño e implementación de la arquitectura del Web Server**
- 9. Sistema de archivos de Redes**
- 10. Arquitectura, diseño, e implementación del Router**

11. Arquitectura, diseño, e implementación del Switch**12. Sistema de archivos con cambios garantizados (Journal file system)****13. Sistema Global de archivos (Global File system)****14. Motor de búsqueda****15. Base de Datos****BIBLIOGRAFÍA**

1. <http://www.globus.org><http://pdos.csail.mit.edu/exo.html>
2. <http://en.wikipedia.org/wiki/Exokernel>
3. <http://pdos.csail.mit.edu/exo/distrib.html>
4. <http://www.cs.berkeley.edu/~brewer/cs262b-2004/Lec-Exokernel.pdf>
5. <http://www.cs.utexas.edu/users/dahlin/Classes/UGOS/reading/engler95exokernel.pdf>
6. <http://pdos.csail.mit.edu/PDOS-papers.html>
7. P. J. Hatcher y Michael J. Quinn. Data-Parallel Programming on MIMD Computers. Publicado por MIT Press, 1991
8. C. Xavier y S.S. Ivengar. Introducción to parallel algoritmos. Publicado por Wiley-Interscience, 1998
9. J. Reinders. Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism. Publicado por O'Reilly, 2007.
10. Shameem Akhter y Jason Roberts. Multi-Core Programming Increasing Performance through. Publicado por Intel Corporation; 1ST edition 2006.
11. Andrew S. Tanenbaum y Maarten van Oteen. Distributed Systems: Principles and paradigms, 2da. ed. Publicado por Prentice Hall, 2006.
12. John L. Hennessy y David A. Patterson. Computer Architecture: A Quantitative Approach. 4ta. ed. Publicado por Morgan Kaufmann, 2006.
13. Maurice J. Bach. Design of the UNIX Operating System. Publicado por Prentice Hall PTR, 1986.
14. Kaare Christian y Susan Richter, The UNIX Operating System. Publicado por Wiley Professional Computing, 1993.
15. Stephen W. Keckler, Kunle Olukotun y H. Peter Hofstee. Multicore Processors and Systems (Integrated Circuits and Systems). Publicado por Springer .2009



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	PARADIGMAS DE PROGRAMACIÓN
CÓDIGO	:	CC018
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC531-Análisis y Diseño de Algoritmos
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

Estudiar diferentes tipos de lenguajes de programación. Si bien la mayoría de lenguajes de programación están usando la programación orientada a objetos o procedimental, existen otros tipos de lenguajes de programación con ciertas fortalezas y debilidades, según se tenga su uso. El conocimiento de estos Lenguajes de Programación es útil según la eficiencia de ciertos tipos de aplicaciones en el mercado del software. El estudio de estos implica diferentes tipos de corrección de errores, como el uso de lenguajes compilados e interpretados.

COMPETENCIAS

PROGRAMA ANALÍTICO

- 1. Tema 1. Introducción:** Evolución de los lenguajes de Programación
- 2. Tema 2. Lambda Calculus.**
- 3. Tema 3. Sintaxis como forma de lenguaje**
- 4. Tema 4. Programación Imperativa.**
- 5. Tema 5. Programación Orientado a Objetos.**
- 6. Tema 6. Programación Logica.**
- 7. Tema 7. Programación Funcional.**

8. Tema 8. Otros tipos de lenguaje de programación.

BIBLIOGRAFÍA

1. R. Sebesta, Concepts of Programming Languages, Addison Wesley, 2005.
2. P. Van Roy, S. Haridi, Concepts, Techniques, and Models of Computer Programming, MIT Press, 2004.
3. K. Arnold, J. Gosling, The Java Programming Language Addison Wesley, 2005.
4. R. Bird, Introduction to Functional Programming using Haskell, Prentice Hall, 1988.
5. M. Abadi, L. Cardelli, A Theory of Objects, Springer, 1996.
6. J. Reynolds, Theories of Programming Languages, Cambridge University Press, 1998
7. U. Nilsson, J. Małuszyński, Logic, Programming and Prolog, John Wiley & Sons, 1995.
8. Apple, Inc., [The Swift Programming Language](#).
9. Axel Rauschmayer, [Exploring ES6](#).
10. James Gosling, Bill Joy, Guy L. Steele, Jr., Gilad Bracha, and Alex Buckley, [The Java Language Specification, Java SE 8 Edition](#).
11. [Dave Thomas, Andy Hunt, and Chad Fowler, Programming Ruby 1.9 & 2.0](#), 4th edition, O'Reilly, 2013.
12. Bjarne Stroustrup, [The C++ Programming Language](#), 4th Edition, Addison-Wesley Professional, 2013.
13. Brian W. Kernighan and Dennis M. Ritchie, [The C Programming Language](#), 2nd edition. Prentice Hall, 1988.
14. [The Rust Programming Language](#) (online book).
15. Martin Odersky, Les Spoon, and Bill Venners, [Programming in Scala: Updated for Scala 2.12](#), Artima, 2016.
16. Jeffrey Ullman, [Elements of ML Programming, ML97 Edition](#), Prentice Hall, 1998.

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad: Ciencias.

Curso: Procesamiento de Lenguaje Natural (NLP)

Horas semanales: 8 horas

Trata del procesamiento de información codificada en lenguaje natural (sea texto o sea voz), para toda tecnología que requiera el tratamiento del lenguaje humano (Español, Inglés, etc.). Este curso se ocupa fundamentalmente de los enfoques estadísticos para el procesamiento del lenguaje natural, pero también abordaremos la necesidad de modelar el conocimiento lingüístico, el conocimiento del dominio y la necesidad de realizar un procesamiento eficiente. Las técnicas y herramientas utilizadas para el procesamiento del lenguaje natural son fundamentales para aplicaciones como minería de texto, extracción de información, procesamiento automático de documentación, traducción automática o asistida, generación de texto, redacción asistida o correctores automáticos. El objetivo de esta materia es comprender cuáles son los principales problemas que presenta el procesamiento del lenguaje natural y cuáles las principales herramientas y técnicas utilizadas. Al finalizar el alumno debe ser capaz de entender y utilizar los métodos de procesamiento de lenguaje natural para extraer significado y valor de textos.

Temario

1. Introducción al PLN. Enfoques racionalista y empirista del lenguaje. Breve Reseña histórica. Conceptos lingüísticos básicos. El lenguaje como fenómeno probabilístico. Algunas aplicaciones básicas de PLN.
2. Principales conceptos de lingüística. Distintos niveles de análisis; Léxico, categorías gramaticales y morfología. Sintaxis. Semántica y Pragmática. Los corpora. Construcción y anotación de un corpus. Tokenization (segmentación, morfología y oraciones). Marcación de documentos y etiquetado.
3. El nivel de la Palabra. Expresiones Regulares y Autómatas de estados finitos. Análisis morfológico y Traductores.
4. N-gramas. Modelos Markovianos visibles y ocultos. Estimación de parámetros.
5. Etiquetado (tagging) de categorías sintácticas. Principales técnicas de etiquetamiento. Aplicaciones del etiquetamiento. Evaluación: Precision, Recall y medida-F.
6. El nivel de las frases. Chunking: parsing con automatas finitos, o parsing parcial.
7. Gramáticas libres de Contexto, Compiladores, Gramáticas probabilísticas, Gramáticas Lexicalizadas.
8. Nivel Semántico. Representación del significado. Desambiguación léxica.
9. Clases de ambigüedad léxica. Adquisición léxica. Papel y aplicaciones de la adquisición léxica.
10. Adquisición de distintas propiedades de las palabras: subcategorización, preferencias de selección, rasgos semánticos y otros.
11. Extracción de significado y valor a tipos de datos utilizando el lenguaje R y sus librerías de NLP.

Bibliografia

Speech and Language Processing (SLP). Jurafsky, Martin, 2da edición, 2009. Prentice Hall.
Foundations of Statistical Natural Language Processing (FSNLP). Manning, Schütze. MIT Press, 1999

Finite State Morphology (FSM). Beesley, Karttunen
Introduction to Automata Theory, Languages, and Computation(ATLC). Hopcroft, Ullman. Addison Wesley.

Daniel Jurafsky and James H. Martin. "Speech and language processing an introduction to natural language processing, computational linguistics, and speech." (2000)

Handbook of Natural Language Processing, Second Edition (Chapman & Hall/CRC: Machine Learning & Pattern Recognition) 2nd Edition. Fred J. Damerau.

Programación Neurolingüística (PNL). Editorial AMAT, 11 feb. 2011 - 278 páginas.
Steve Bavister, Amanda Vickers

Foundations of Statistical Natural Language Processing 1st Edition.
Hinrich Schütze



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	PROGRAMACIÓN EN DISPOSITIVOS MÓVILES
CÓDIGO	:	CC005
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC531 - ANÁLISIS Y DISEÑO DE ALGORITMOS
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

Los dispositivos móviles son en la actualidad los sistemas de comunicación e información líderes gracias al rápido crecimiento en el número de dispositivos vendidos.

El objetivo principal del curso es conducir al alumno en la programación de aplicaciones para dispositivos móviles conociendo las herramientas que ponen en marcha una aplicación avanzada, adquiriendo los conocimientos avanzados para la programación de grandes aplicaciones y su publicación y venta.

Podemos resumir comentando que la asignatura se centra en los objetivos principales de aprender a programar para los sistemas operativos predominantes en los dispositivos móviles. Necesitaremos trabajar desde los aspectos de diseño que tiene una aplicación hasta técnicas algorítmicas avanzadas para la puesta en marcha de una aplicación novedosa.

COMPETENCIAS

- Conocimiento, administración y mantenimiento 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.
- Identificar una panorámica general de la tecnología de programación en smartphones.
- Implementar aplicaciones y servicios generales para los dispositivos móviles comerciales.

- Desarrollar aplicaciones en plataformas de programación, lenguajes de ejecución y de desarrollo más usados en la actualidad..
- Esquematizar las herramientas que ofrece Android para el desarrollo de proyectos en su plataforma.
- Generar aplicaciones avanzadas para Smartphones.

PROGRAMA ANALÍTICO

1. Módulo 1: introducción y puesta a punto del entorno de desarrollo: Android Studio.
2. Módulo 2: Componentes para el desarrollo de aplicaciones móviles. Fundamentos de aplicaciones.
3. Módulo 3: Diseño de interfaz. Layouts. Formularios, Eventos, Colecciones de datos. Menús. Estilos.
4. Módulo 4. Diseño avanzado de interfaces de usuario.
5. Módulo 5. Almacenamiento de datos.
6. Módulo 6: Servicios web. Arquitectura cliente-servidor. Almacenamiento externo. Extracción y modificación en BBDD externa.
7. Módulo 7: Sensores en un dispositivo móvil. Programación de periféricos en dispositivos móviles.
8. Módulo 8: API's a disposición de un desarrollador y su programación. Ubicaciones y posicionamiento en mapas.
9. Módulo 9: Multimedia y Widgets.

BIBLIOGRAFÍA

1. Android. Guía para desarrolladores (Segunda Edición); Ed: Anaya.
2. Curso de Android (Spanish Edition); Ed. Maestros de la web
3. Curso de Android de la Universidad Politécnica de Valencia. www.androidcurso.com.
4. Android: guía de desarrollo de aplicaciones para Smartphones y Tabletas (2a edición); Sylvain HÉBUTERNE - Sébastien PÉROCHON.
5. Desarrollo de juegos para Android (Títulos Especiales); Ed: Anaya.
6. El gran libro de programación avanzada con Android; Ed: Marcombo.
7. Mobile Design and Development; Brian Fling; O'REILLY; Agosto 2009.
8. Android Essentials; Chris Haseman; Apress; Julio 2008.
9. Android Programming Tutorials; Mark Lawrence Murphy; CommonsWare; Marzo 2010.
10. Unlocking Android: A developer's Guide; Frank Ableson, Robi Sen; Manning; 2010 (2^a edición).
11. Android: a programmer's guide; Jerome DiMarzio; McGraw-Hill; 2008; <http://bit.ly/bBeI18>.
12. The Busy Coders Guide to Android Development; Mark Lawrence Murphy; CommonsWare; Septiembre 2008; <http://bit.ly/9gOdMy>.
13. Beginning Android Games; Richard Taylor; Apress; 2010.
14. Hello, Android: Introducing Google's Mobile Development Platform; Ed Burnette; Pragmatic Bookshelf; 2010.
15. Professional Android Application Development; Reto Meier; Wrox; 2008.
16. The Busy Coders Guide to Advanced Android Development; Mark Lawrence Murphy; CommonsWare; 2009.

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad: Ciencias.

Curso: Programación declarativa

Horas semanales: 8 horas

La Programación Declarativa (p.d.) se caracteriza porque los programas describen relaciones entre objetos/valores; pueden ser relaciones funcionales o relaciones más generales, y dejan que el ejecutor del lenguaje (intérprete o compilador) trate de satisfacerlas aplicando un algoritmo fijo de cómputo.

Estudiaremos los estilos Programación Funcional (en Haskell), Programación Lógica (en Prolog), Programación Algebraica, Lógico-Funcional y Visual, para disponer de un lenguaje que sea suficientemente expresivo, de una semántica operacional (un mecanismo de cómputo que permite ejecutar los programas) y de una semántica declarativa que permita dar un significado a los programas de forma independiente a su posible ejecución. La clasificación de todos estos lenguajes que tratan la programación de computadoras puede hacerse desde varios puntos de vistas: bajo y alto nivel, de propósito general o especialistas, interpretados o compilados, etc. El alumno al terminar el curso debe tener la capacidad de indicar qué debe computarse, que resultados queremos obtener, pero no necesariamente se le indica al computador cómo hacerlo y es prerrogativa del computador elegir el orden de las sentencias.

Temario

1. Lógica y programación lógica. Conceptos de Programación Declarativa.
2. Estilos de programación declarativa, Ecuacional Funcional, Clausal Relacional y Heterogenea Tipos.
3. Estilos de programación declarativa con Géneros ordenados Herencia, Modal S.B.C.y temporal concurrencia.
4. Programación Lógica. Lógica de cláusulas de Horn.
5. Predicados extralógicos. Programación lógica con estructura.
6. Lenguajes algebraicos con Maude.
7. Lenguajes algebraicos con SQL.
8. El lenguaje de programación funcional Prolog y la programación eficiente en Prolog.
9. Programación Funcional con LISP. Lógica ecuacional y sistemas de reescritura.
10. Programación Funcional con el entorno Hugs para Haskell.
11. Programación Funcional. Programación eficiente en Hugs.
12. Programación Lógico-Funcional con Curry.
13. Programación Visual. El lenguaje Visual Prolog.
14. Programación Lógica Difusa con F-Prolog.

Bibliografia

Rafael del Vado Vírseda (Departamento de Sistemas Informáticos y Computación – U.C.Madrid)

The Art of Prolog (2ed) Leon Sterling y Ehud Shapiro. MIT Press, 1994

Prolog Programming for Artificial Intelligence (3ed). Ivan Bratko Addison-Wesley, 2000

Clause and Effect. William F. Clocksin. Springer-Verlag, 1997.

Programación en Prolog (2ed). William F. Clocksin y Chris S. Mellish. Gustavo Gili, 1993

Carlos Alberto Romero Díaz (Departamento de Sistemas Informáticos y Computación – U.C.Madrid)

Richard Bird. Introducción a la Programación Funcional con Haskell. Prentice-Hall, 2000.

Seif Haridi and Peter van Roy. Concepts, Techniques, and Models of Computer Programming. The MIT Press, 2004.

Paul Hudak. The Haskell School of Expression. Learning functional programming through multimedia. Cambridge University Press, 2000.

Ulf Nilsson and Jan Maluszynski Logic, Programming and Prolog (2ed) John Wiley & Sons Ltd, 2000.

Rinus Plasmeijer and Marko van Eekelen. Functional Programming and Parallel Graph Rewriting. Addison-Wesley, 1993.

Ricardo Peña Marí (Sistemas Informáticos y Computación-U.C.Madrid)

Razonando con Haskell. Un curso sobre programación funcional”
B.C. Ruiz, F. Gutiérrez, P. Guerrero y J.E. Gallardo
Ed. Thomson , 2004.

Introducción a la programación funcional con Haskell (2ed) R. Bird Ed. Prentice-Hall, 1999.

The Craft of Functional Programming. S. Thompson. Ed. Addison-Wesley, 1999.

Blas Carlos Ruiz Jiménez (Departamento de Lenguajes y Ciencias de la Computación – U.Malaga)

Federico Barber (Departamento de Sistemas Informáticos y Computación – U.P.Valencia)

Miguel A. Salido (Departamento de Ciencias de la Computación e Inteligencia Artificial– U.Alicante)



SÍLABO

I. INFORMACIÓN GENERAL

Curso	: Introducción a la Computación Cuántica
Código	: CC028
Pre-requisito	: CQ221, CC622
Dpto. Académico	:
Condición	:
Ciclo Académico	: 2018-1
Créditos	: 4
Horas teóricas	: 4 horas semanales
Horas prácticas	: 2 horas semanales
Sistema de Evaluación	: G
Profesor del curso	:

II. SUMILLA

La computación cuántica es un campo de investigación emergente e interdisciplinario que abarca las matemáticas, la física y la informática, y que explota la naturaleza cuántica de la materia de existir simultáneamente en múltiples estados.

El curso consta de cuatro partes:

1. Fundamentos cuánticos.
2. Algoritmos cuánticos.
3. Algoritmos de factorización y búsqueda cuántica.
4. Criptografía cuántica.

III. COMPETENCIAS

1. Interpreta los principios fundamentales de la mecánica cuántica aplicándolo a la computación cuántica.
2. Comprende la lógica los bits cuánticos para diseñar circuitos cuánticos.
3. Entiende los algoritmos cuánticos y programa simulaciones.
4. Comprende la criptografía cuántica aplicándola a casos reales.

IV. UNIDADES DE APRENDIZAJE

CAPÍTULO 1: FUNDAMENTOS CUÁNTICOS.

Sistemas cuánticos de un solo qubit. Sistemas cuánticos de múltiples qubit. Medición de estados de múltiples qubit. Transformaciones de estados cuánticos. Versiones cuánticas de cálculos clásicos.

CAPÍTULO 2: ALGORITMOS CUÁNTICOS.

Calculando con superposiciones. Noción de complejidad. Algoritmo de Deutsh. Subrutinas cuánticas. Algoritmo de Deutsh-Jozsa. Algoritmo de Bernstein-Vazirani. Algoritmo de Simon. Algoritmo de la computación distribuida. Transformada de Fourier cuántica.

CAPÍTULO 3: ALGORITMOS DE FACTORIZACIÓN Y BÚSQUEDA CUÁNTICA.

Algoritmo de Shor. Algoritmo de Grover.

CAPÍTULO 4: CRIPTOGRAFÍA CUÁNTICA.

Criptografía de clave privada. Amplificación de privacidad y reconciliación de información. Distribución de clave cuántica: Protocolos BB84, B92, E91 y BBM92.

V. METODOLOGÍA

Método presencial de aprendizaje, en el cual el profesor deduce e induce las bases teóricas, complementada con aplicaciones preferentemente relacionadas a la especialidad respectiva. Tutoría académica permanente en forma semanal según horarios fuera de clase.

VI. SISTEMA DE EVALUACIÓN

Sistema de evaluación G:

Cantidad de prácticas o trabajos calificados: cuatro (04)

El promedio final (**PF**) se calcula tal como se muestra a continuación:

$$\text{PF} = (\text{EP} + \text{EF} + \text{PP}) / 3$$

EP: Examen Parcial (Peso 1)

EF: Examen Final (Peso 1)

PP: Promedio de Prácticas (Peso 1)

PP: Se obtiene del promedio aritmético de las tres (03) mejores notas de las prácticas o trabajos calificados.

VII. BIBLIOGRAFÍA

- Quantum Computation and Quantum Information. 10th Anniversary Edition. Michael A. Nielsen & Isaac L. Chuang. 2010
- Quantum Computing. A gentle introduction. Eleanor Rieffel and Wolfgang Polak. 2011
- Quantum Computer Science. An Introduction. N. David Mermin. 2007
- Problems and solutions in quantum computing and quantum information. Willi-Hans Steeb and Yorick Hardy. 2004



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	SIMULACIÓN NUMÉRICA
CÓDIGO	:	CC013
CRÉDITOS	:	4
PRE-REQUISITO	:	CC 622
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

El objetivo general del presente curso es motivar la programación de computadoras como una herramienta muy poderosa para hacer matemáticas. Sin embargo, para resolver problemas matemáticos a través de la programación de computadoras, se necesita de métodos numéricos que se explicaran de una manera básica, así algunos modelos matemáticos como las ecuaciones diferenciales que constituyen el tipo más frecuente de modelos. En consecuencia, el enfoque numérico se orienta hacia las ecuaciones diferenciales. Se inicia con integración numérica, para luego tratar temas de hallazgo de raíces que es importante para la solución numérica en ecuaciones diferenciales no lineales. Es decir, el contenido se inicia con un breve introducción sobre métodos numéricos dado que el objetivo es implementar algoritmos numéricos, pero para desarrollar programas de trabajo confiables, el programador debe tener confianza en las ideas básicas de las aproximaciones numéricas involucradas.

PROGRAMA ANALÍTICO

- Tema 1. Cálculo de Integrales:** Ideas Básica de Integración Numérica; Método Trapezoide Compuesto; Método Punto Medio Compuesto; Testing; Vectorización; Midiendo la Velocidad Computacional; Integrales Dobles y Triples
- Tema 2. Resolviendo Ecuaciones Diferenciales Ordinarias:** Crecimiento Poblacional; Dispersión de Enfermedades; Sistemas de Oscilamiento Unidimensionales.

3. **Tema 3. Resolviendo Ecuaciones Diferenciales Parciales.** Métodos de Diferencia Finita.
4. **Tema 4. Resolviendo Ecuaciones Algebraicas No Lineales:** Método de la Fuerza Bruta; Método de Newton; Método de la Secante; Método de la Bisección; Tasa de Convergencia; Resolviendo Ecuaciones Algebraicas No Lineales Múltiples.

BIBLIOGRAFÍA

1. Linge, S. and Langtangen, H. (2016) “Programming for Computations – A Gentle Introduction to Numerical Simulations with MATLAB/Octave”. SpringerOpen

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad: Ciencias.

Curso: SISTEMAS COGNITIVOS

Horas semanales: 8 horas

El cerebro humano y las capacidades cognitivas que genera configuran el dispositivo de procesamiento de información más complejo y sofisticado que se conoce. Para abordar dicha complejidad sólo es posible a partir del uso combinado de conocimiento experimental que viene de ámbitos como la neurobiología, la neurociencia o la psicología, con la fuerza teórica de los modelos y desarrollos de las ciencias de la computación y de planteamientos integradores como los que pretende alcanzar la neurociencia computacional o la ciencia cognitiva.

Temario

1. Comportamiento inteligente

Introducción a la noción de inteligencia. Marcos históricos de investigación: Cartesianismo, Teoría de la evolución, Cibernética, Teoría dinámica de sistemas, GOFAI, conexionismo, inteligencia basada en el comportamiento, vida Artificial, etc. Propiedades de los sistemas inteligentes: relaciones entre fisiología y función, relaciones entre biología e ingeniería.

2. Ciencias de la mente

Información y sistemas de símbolos: modelos de representación. Contenido no-conceptual de las representaciones. Estudio de los sistemas intencionales: enfoque representacional vs. no-representacional. ¿Representaciones sin símbolos?: Cerebro vs. computador. Ideas sobre el lenguaje. Modelos sobre otras mentes.

3. Inteligencias simples y cognición mínima

Inteligencia artificial a partir del enfoque Brooksiano: sistemas situados, corporizados y embebidos, inteligencia emergente. Teoría de la cognición mínima. Estrategias de navegación y orientación en insectos (fonotáxis, quimiotáxis, infotáxis, etc.). Soluciones evolutivas. Diseño de animats.

4. Cerebro y evolución

Relación entre redes neuronales, y anatomía cerebral. Evolución y desarrollo del cerebro. Comparación entre el cerebro del hombre y otros primates. Crecimiento alométrico y coste metabólico de los cerebros complejos. Especialización cortical. Sistemas complejos, propiedades emergentes y aplicaciones a las neurociencias. Fenómenos caóticos y críticos de la dinámica neuronal. Bases neurológicas de la conciencia. Modelos de conciencia artificial.
Lecturas recomendadas:

5. Vida Artificial

Propiedades de lo vivo y definiciones de vida artificial. Auto-replicación: Búcles auto-replicantes de Langton. Evolución interactiva y Coevolución. Morfogénesis: Redes de Kauffman, Simulación del desarrollo, Sistemas de Lindenmayer.

6. Sistemas sociales inteligentes

Sistemas colectivos: organizaciones de insectos sociales y organizaciones humanas. Sistemas de control distribuido, comportamiento global vs. conocimiento local, planes. Interacción. Cooperación. Acciones y representaciones. Cooperación en el tiempo.

7. Sistemas autónomos y adaptativos

Desarrollo de modelos de adaptación dinámica, mecanismos de regulación sensomotriz, controladores adaptativos de estabilidad. Relaciones de autopoiesis de los organismos establecen con sus entornos para garantizar condiciones de supervivencia. Espacio de soluciones evolutivas posibles, adaptación homeostática, señales propioceptivas.

Lecturas recomendadas:

8. Emociones y sistemas artificiales

Teorías de emociones cognitivas, anti-cognitivas, e interactivas; Arquitecturas de agentes emocionales, Dualismo vs. Funcionalismo, Origen y evolución de las emociones, mecanismos de exaptación, emociones como mecanismos de apoyo a la planificación.

9. Simulación del comportamiento inteligente

Ciencias de lo artificial, modelos a nivel de información vs modelos de conocimiento, sistemas naturales vs sistemas funcionales, principio de simulación, modelos de adaptación en sistemas naturales y artificiales, adaptación vs. Auto-adaptación, simulación de modelos de cognición mínima, modelos de vida artificial.

Bibliografia

Brooks, R.A. (1995). "Intelligence Without Reason". In Steels, L. and Brooks, R. (eds.) *The Artificial Life Route to Artificial Intelligence: Building embodied, situated agents*, 25-81, Lawrence Erlbaum Associates.

Arkin R.C. (1998) *Behavior-based robotics*. MIT Press

Ashby, W.R. (1960). "Design for a brain: the origin of adaptive behaviour", Chapman. Gibson, J.J. (1979). "The ecological approach to visual perception", Houghton Mifflin

Johnson-Laird. (1993). *El ordenador y la mente*, Barcelona, Paidós.

Dennett D. (1992) *Darwin's Dangerous Idea: Evolution and the Meanings of Life*, Editorial Simon and Schuster, Editorial Simon and Schuster New York, 1995

David Casacuberta, *La mente humana. Cinco enigmas y cien preguntas* Ángel Rivière, Objetos con mente. Alianza

Webb, B. (1996). "A Cricket Robot". *Scientific American*, December 1996, 62-67.

Franceschini, N., Pichon, J.M., and Blanes, C. (1997). "Bionics of Visuo motor Control". In: *Evolutionary Robotics: From intelligent robots to artificial life (ER'97)*, Gomi, T. (ed.), 49-67, AAI Books.

Dave Cliff, Philip Husbands, Jean-Arcady Meyer and Stewart W. Wilson (1995). From Animals to Animats. Proceedings of the Third International Conference on Simulation of Adaptive Behavior. ISBN-10:0-262-53122-4

Holland, O. (2003) Machine Consciousness, Special Issue of the Journal of Consciousness Studies ,Exeter: Imprint Academic, 2003

Dunbar, R. (2003). The social brain: mind, language and society in evolutionary perspective. Annual Review of Anthropology 32: 163-181

Dennett, D. (1992) Consciousness Explained. Back Bay Books. (1992). ISBN-10: 0316180661

Beer, R. (1995). A dynamical systems perspective on agent-environment interaction. Artificial Intelligence, pages 173-215.

Clark, A. (1997). Being there. Putting brain, body and world together again. MIT Press.

Keijzer, F. (2002). Representation in dynamical and embodied cognition. Cognitive Systems Research, 3:275-288

Koza, John R . Genetic programming: on the programming of computers by means of natural selection. Editorial MIT Press ISBN: 0262111705

Langton, C.G., Artificial Life, SFI Studies in the Sciences of Complexity, Addison-Wesley, 1989

Adami, C., Introduction to Artificial Life, Telos-Springer Verlag, 1998Emmeche, C, La Ciencia Naciente de la Vida Artificial, Gedisa Ed., 1998

Dorigo, M., Di Caro,G (1995). ,Ant Algorithms for Discrete Optimization, Artificial Life, 1999, Vol. 5,

Michelsen, A., et al. (1992) How honeybees perceive communication dances, studied by means of a mechanical model. Behav. Ecol. Sociobiol. 130:143-150.

Wooldridge, M. (2002) An Introduction to MultiAgent Systems. John Wiley and Sons, Chichester, England.

Brooks. Artificial life and real robots. In F. J. Varela and P. Bourgine, editors, Towards a practice of autonomous systems: Proc. 1st Eur. Conf. on Artificial Life (ECAL91), pages 3-10. MIT Press, 1992.

J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. Rethinking Innateness: A connectionist perspective on development. MIT Press: Bradford Books, 1996.

N. Jakobi. Evolutionary robotics and the radical envelope of noise hypothesis. Adaptive Behaviour, 6(2):326-368, 1998.

Damasio, A. (1996). Descartes' error. Drakontos, Crítica.

De Sousa, R. (1987). The Rationality of Emotion. Cambridge, Mass., MIT Press.

Frijda, N. (1986). The emotions. Sudies in Emotion and Social Interactions. Cambridge University Press, Cambridge, UK.

LeDoux, J. (2000). The Emotional Brain, Nueva York, 2000

Petta, P. (2002). The role of emotions in tractable architectures for situated cognizers. In Trappi, R., Petta, P. and Payr, S. (Eds.). Emotions in Humans and Artifacts. Cambridge, M.A., MIT Press.



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	SISTEMAS EMBEBIDOS EN TIEMPO REAL
CÓDIGO	:	CC
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC632 – PROGRAMACIÓN PARALELA
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (TEORÍA: 02, LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

SUMILLA

El curso presenta al estudiante la problemática particular de los sistemas de tiempo real, y las características que los diferencian de otros sistemas informáticos. Se aprenderán los métodos más importantes que se utilizan para desarrollar sistemas de tiempo real con un elevado grado de fiabilidad, especialmente los que se refieren a la medida del tiempo, la planificación del uso de los recursos, la prevención y la tolerancia a fallos, y la organización del software y su aplicación. Se contemplarán de forma especial las aplicaciones de control de procesos. Se plantearán varias técnicas para desarrollar estas aplicaciones y se establecerán los mecanismos necesarios para evaluar sus prestaciones.

COMPETENCIAS

- Dominar los conceptos específicos de los sistemas embebidos de tiempo real, a fin de identificar los problemas de este tipo dentro de sistemas más complejos y desarrollar las soluciones adecuadas.
- Manejar recursos informáticos como herramientas auxiliares para la resolución de problemas específicos.
- Aplicar la creatividad, la iniciativa personal y la innovación en el área tecnológica

PROGRAMA ANALÍTICO

1. Introducción a los sistemas informáticos de tiempo real. I Definición de sistema de tiempo real. II Ejemplos de sistemas de tiempo real. III Características de los sistemas de tiempo real. IV Tipo de

sistemas de tiempo real. V Programación de los sistemas de tiempo real: sistemas operativos y lenguajes de programación.

2. Sistemas operativos multiprogramados I Introducción. II RTOS. Idea de deadline. III Memoria. IV Entrada/Salida. V Procesos, concurrencia y comunicación. VI Lenguajes y sistemas operativos de tiempo real.

3. Sistemas cíclicos. I Conceptos y metodología II Planificación cíclica III Segmentación de tareas IV Construcción del plan de ejecución cíclico V Programación del ejecutivo cíclico

4. Gestión del tiempo I Sistemas de referencia de tiempo II Relojes, retardos y límites temporales III Requerimientos temporales IV Tolerancia a fallos

5. Planificación de tareas I Conceptos y metodología II Planificación con prioridades fijas III Interacción entre tareas IV Planificación con prioridades dinámicas. V Modelo de tareas generalizado VI. Multiprogramación VI Gestión de procesos y memoria VI Sincronización y comunicación entre procesos

6. Ingeniería del Software para un Sistemas de Tiempo Real.

7. Aplicacion Sistemas Embebidos. (Embedded Systems).

BIBLIOGRAFÍA

1. Mohammad Ayoub Khan, Saqib Saeed Embedded and Real Time System Development: A Software Engineering Perspective Springer, 2013
2. Berger, Arnold. Embedded Systems Design: an Introduction to Processes, Tools, and Techniques, CMP Books ; Berkeley, CA: Distributed in the U.S. and Canada by Publishers Group W. ISBN: 1578200733.
3. Daniel W. Lewis "Fundamentals of Embedded Software: Where C and Assembly Meet", Prentice Hall, ISBN: 0130615897, 2001.
4. Alan Burns - Andy Wellings "Sistemas de Tiempo Real y Lenguajes de Programacion", Pearson Educacion, ISBN: 8478290583, 2003.



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	SISTEMAS EMPOTRADOS
CÓDIGO	:	CC019
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC612 - ADMINISTRACIÓN DE REDES
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	06 (TEORÍA: 02; LABORATORIO 04)
SISTEMA DE EVALUACIÓN	:	G

OBJETIVO

Este curso tiene por objetivo permitir al alumno aprender adecuadamente las tecnologías que se usan para el desarrollo de una aplicación usando sistemas empotrados.

PROGRAMA ANALÍTICO

- 1. Programación en lenguaje C del microcontrolador 8051**
I: Tarjeta TMC51 . II: Características del 8051 de ATMEL III: El IDE de KEIL. IV: Lenguaje C. V: Manejo de leds, motores, etc.
- 2. Programación en lenguaje C del MSP430**
I: Descripción de la tarjeta. II: Lenguaje C. III: Manejo de sensores, display LCD..
- 3. Programación en lenguaje Python del Raspberry Pi**
I:Placa Raspberry Pi. II: Instalación del SO. III: El GPIO IV: Proyectos con Raspberry Pi.
- 4. Programación de Arduino Uno**
I: Placa Arduino Uno y otros modelos de placas. II. Forma de uso del lenguaje en la placa arduino. III. Proyectos con Arduino.

BIBLIOGRAFÍA

1. Q. Li & C. Yao. Real-Time Concepts for Embedded Systems. CMP Books, 2003
2. M.J. Pont, Embedded C.Addison-Wesley, 2002



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

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	SISTEMAS OPERATIVOS AVANZADOS
CÓDIGO	:	CC012
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITOS	:	CC612 - ADMINISTRACIÓN DE REDES
CONDICIÓN	:	ELECTIVO
HORAS POR SEMANA	:	04 (TEORÍA: 02; LABORATORIO: 04)
SISTEMA DE EVALUACIÓN	:	G

OBJETIVO

Este curso tiene por objetivo permitir al alumno aprender a adecuadamente las tecnologías que se usan para la mejora de un sistema operativo.

PROGRAMA ANALÍTICO

1. **Introducción a Sistemas Operativos**
I: Instalación de un sistema operativo en Raspberry . II:Sistema de Archivos. .
2. **Organización Funcional de un Sistema Operativo**
I: Principio de abstracción. II: Recursos de Hardware. III: Funcionalidad de un Sistema Operativo.
3. **Estructuras de un Sistema Operativo**
I: Abstracciones de un SO. II: Estructuras del SO.
4. **Enfoques hacia la extensibilidad**
I: Spin del SO. II: Exokernel. III. Microkernel L3.
5. **Virtualización**
I: Tipos de Virtualización. II.Importancia de la virtualización.

BIBLIOGRAFÍA

1. Silberschatz and Galvin. Operating System Concepts.
2. Maekawa, Oldehoeft. OS: Advanced Concepts. Addison Wesley.
3. Sape Mullender. Distributed Systems. Addison Wesley.
4. Andrew Tanenbaum. Distributed Operating Systems. Prentice Hall.

