

011

José C. Paz, 19 DIC 2018

VISTO:

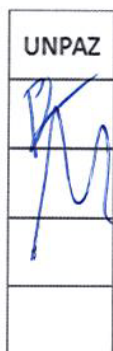
El Estatuto de la UNIVERSIDAD NACIONAL DE JOSÉ CLEMENTE PAZ aprobado por Resolución del entonces MINISTERIO DE EDUCACIÓN N° 584 del 17 de marzo de 2015, el REGLAMENTO DE FUNCIONAMIENTO DEL CONSEJO DEPARTAMENTAL DE ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA aprobado por Acta N° 2 del citado Consejo de fecha 26 de marzo de 2018, el Expediente Nro. 468/18, del Registro de esta UNIVERSIDAD NACIONAL DE JOSÉ CLEMENTE PAZ, y

CONSIDERANDO:

Que por el Expediente del VISTO tramitan las propuestas de programas de la carrera de Analista Programador Universitario, correspondientes a las siguientes asignaturas: Arquitectura de Computadoras I (602), Ingeniería de Software II (618) Sistemas Operativos II (622).

Que es competencia de este CONSEJO DEPARTAMENTAL aprobar y supervisar los programas curriculares de las carreras a su cargo, garantizando que aquellos se ajusten a los contenidos mínimos definidos en los correspondientes Planes de Estudios.

Que habiendo sido puestos a consideración del CONSEJO DEPARTAMENTAL en la Sesión N° 3, registrada en el Acta N° 3 del 30 de mayo de 2018, este Cuerpo Colegiado compartió los términos y contenidos de los referidos



instrumentos, por lo que resulta necesario aprobar los respectivos programas de las asignaturas detalladas.

Que la presente medida se adopta en ejercicio de las atribuciones conferidas por los artículos 77, inciso f), del Estatuto de la UNIVERSIDAD, y 1º, inciso d) y 7º, inciso c), del Reglamento de Funcionamiento de este Consejo Departamental.

Por ello,

**EL CONSEJO DEPARTAMENTAL
DE ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA
DE LA UNIVERSIDAD NACIONAL DE JOSÉ CLEMENTE PAZ**

DISPONE:

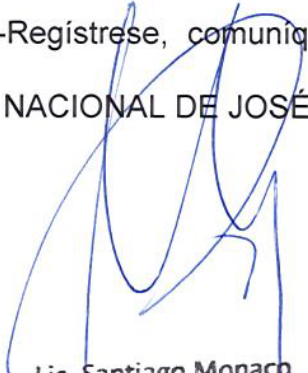
ARTÍCULO 1º.- Apruébanse los programas de la carrera de Analista Programador Universitario que se adjuntan como Anexo a la presente, correspondientes a las siguientes asignaturas: Arquitectura de Computadoras I (602), Ingeniería de Software II (618) Sistemas Operativos II (622).

ARTÍCULO 2º.-Establécese que los programas aprobados precedentemente, tendrán DOS (2) años de vigencia, contados a partir del semestre siguiente al de su aprobación.

ARTÍCULO 3º.-Regístrese, comuníquese, publíquese en el Boletín Oficial de la UNIVERSIDAD NACIONAL DE JOSÉ CLEMENTE PAZ y cumplido, archívese.

UNPAZ





Lic. Santiago Monaco
Director de Coordinación
de Órganos de Gobierno
Secretaría general UNPAZ


Mg. PABLO J. LÓPEZ
Director
Depto. de Economía, Prod. e Innov. Tec.
Universidad Nacional de José C. Paz

01



PROGRAMA UNIDAD CURRICULAR			
Departamento	ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA		
Carrera/s	ANALISTA PROGRAMADOR UNIVERSITARIO		
Nombre de la unidad curricular	ARQUITECTURA DE COMPUTADORAS I	Código	602
Docente responsable	JUAN FUNES - WALTER SALGUERO - FABIAN E. PALACIOS		
Año de presentación del programa	2018		
1. Carga horaria			
Horas de clase semanales	6		
Horas de clase totales	96	Horas totales clases teóricas	64
		Horas totales clases prácticas	32
		Otras horas totales (laboratorio, trabajo de campo, etc.)	0
2. Unidades correlativas precedentes en el Plan de Estudios			
Denominación		Código	
3. Contenidos mínimos según Plan de Estudios			
Computadoras digitales: Conceptos introductorios. Funcionamiento básico. Organización de un sistema de cómputo, modelo de von Neumann. Otros modelos de organización, clasificación de las computadoras de acuerdo al modelo. Ejemplos. Síntesis de la evolución histórica de las computadoras y la tecnología empleada en su fabricación. Costo y rendimiento. Análisis de la performance, métodos de medición, MIPS, MFLOPS, benchmarks. Concepto de niveles de abstracción. Aritmética de las computadoras. Sistemas de notación posicional. Sistema decimal, sistema binario, sistema octal, sistema hexadecimal. Operaciones fundamentales en todos los sistemas. Definición de bit, nibble, byte, palabra, palabra doble, relación con lenguajes de alto nivel. Representaciones numéricas: números enteros con y sin signo. Aritmética con enteros. Fundamentos de la representación en punto flotante, normalización, error de la representación. Representación normalizada IEEE 754. Representaciones alfanuméricas, ASCII, EBCDIC.			

[Handwritten signature]

Lógica Digital. Compuertas lógicas. Álgebra de Boole. Implementación de funciones booleanas, simplificación. Lógica combinatoria, codificadores, decodificadores, multiplexores. Lógica secuencial, registros, contadores. Concepto de memoria y lógica programable.

Unidad Central de Procesamiento (CPU). Organización. Descripción de los primeros microprocesadores. Concepto de instrucción. Conjunto de instrucciones del procesador. Ciclo de ejecución de microinstrucciones con y sin interrupción. Comunicación CPU – memoria, dato y dirección. Interconexión de subsistemas, buses, ejemplos. Organización de registros. Lenguaje de máquina y Assembly.

Memoria: tipos y clasificación. Parámetros característicos, tamaño, tiempo de acceso, costo, otros. Direccionamientos a Memoria principal. Memoria secundaria, organización y formato de datos. Otros dispositivos de almacenamiento externo, discos SSD, cinta, disco óptico, otros. Periféricos. Dispositivos de interacción típicos: terminales, pantallas, teclado, mouse, impresora, scanner, audio. Características de interconexión a cada uno de ellos, errores de comunicación.

4. Fundamentación

El presente espacio curricular está ubicado en el Primer cuatrimestre del primer año de la carrera de pregrado Analista Programador Universitario.

Arquitectura de Computadoras es una de las materias del área de Computación que provee los conocimientos básicos para la formación científica, tecnológica y complementaria, sobre la organización y funcionamiento de las computadoras y sus periféricos.

La arquitectura de las computadoras se interesa por la estructura y desempeño de los diferentes módulos funcionales de las computadoras y como interactúa para atender las necesidades de los procesamiento del usuario.

La comprensión que los analistas programadores universitarios tengan de las técnicas básicas de hardware (procesador, memoria, dispositivos de entrada y salida) como así también del lenguaje de máquina, serán fundamentales para la optimización del rendimiento de los futuros sistemas de software que deban desarrollar.

5. Objetivos

Comprender la organización, la estructura y los servicios proporcionados por las Arquitecturas de las computadoras.

Comprender temas vinculados con códigos numéricos binarios, su representación interna, y temas relacionados con la codificación y decodificación de información en sistemas de computadoras.

Comprender los conceptos básicos sobre álgebra booleana, sistemas combinacionales y secuenciales.

Adquirir el vocabulario técnico, propio de los sistemas informáticos, incorporando definiciones y expresiones que caracterizan un sistema informático.

2



Adquirir un panorama general sobre las estructuras de hardware, a partir de un análisis completo de una configuración clásica tipo Von Neumann, y su relación con el software que le da funcionalidad. Comprender los conceptos sobre lenguaje ensamblador, su entorno de desarrollo y utilidad.
6. Contenidos (organizados por unidades)
UNIDAD 1: DIGITALES Señales analógicas y digitales . Sistemas electrónicos digitales y su conversión analógico-digital. Numeración decimal, binario, octal y hexadecimal. CÓDIGOS BINARIOS - Código Gray - Conversión entre binario y Gray, y viceversa - Suma módulo 2 - Codificación de los caracteres alfanuméricos - El código ASCII - Los códigos ASCII extendidos de 8 bits - Unicode y código UCS - Otros códigos alfanuméricos - Codificación de los números - Operaciones aritméticas entre magnitudes binarias - Códigos para números binarios enteros - Codificación en Signo y magnitud - Codificación en Complemento a 1 - Codificación en Complemento a 2 - Codificación en Binario desplazado - Códigos binarios para numeros decimales - BCD Natural - BCD Aiken - BCD Exceso 3 o XS3 -. Código 2 de 5 - Código 7 segmentos - Representación de coma flotante, Norma IEEE-754. ÁLGEBRA DE CONMUTACIÓN - Álgebra de Boole - Operaciones básicas entre proposiciones lógicas - La conjunción - La disyunción - La negación. - Aplicación a conjuntos - El Álgebra de conmutación aplicada a contactos - Contactos en serie: producto lógico - Propiedades del producto lógico - Contactos en paralelo: suma lógica, operación OR - Propiedades de la suma lógica - La inversión - Propiedades de la inversión - Precedencia de los operadores - Dualidad y el Principio de dualidad - Teoremas del álgebra de conmutación - Tablas de verdad de funciones lógicas - Dualidad y ley de Shannon - Expresiones canónicas - Productos canónicos o minitérminos - Teorema general los minitérminos - Sumas canónicas o maxitérminos - Teorema general de los maxitérminos - Conjunto completo de operaciones. Diagrama de Venn, tabla de verdad y minitérminos - Introducción a la simplificación de funciones lógicas - Fundamentos para la simplificación de funciones - Tabla de verdad y la representacion en el mapa de Karnaugh - Simplificación de funciones lógicas como Sumas de Productos - Simplificación como Producto de Sumas. UNIDAD 2: CIRCUITOS COMBINACIONALES Y SECUENCIALES CIRCUITOS COMBINACIONALES CON COMPUERTAS - Las compuertas - Las compuertas fundamentales - La compuerta inversora - Las compuertas derivadas - La compuerta buffer - Introducción al control de flujo de señales - La compuerta de transmisión - Simbología alternativa para las compuertas - Simbología IEEE-91 - Análisis de circuitos lógicos combinacionales y graficación - Síntesis de circuitos

lógicos combinacionales - Análisis del funcionamiento y obtención de las expresiones algebraicas - Implementación de las expresiones algebraicas con compuertas - Implementación de expresiones tipo Suma de Productos - Implementación de expresiones tipo Producto de Sumas - Comentarlos finales sobre la implementación de expresiones - Formas degeneradas.

COMPONENTES DIGITALES - Niveles de integración - Multiplexores - Demultiplexores - Decodificadores - Codificadores de prioridad - Matrices lógicas programables - Lógica secuencial - El circuito biestable (flip flop) S R - El flip flop S R síncronico - El flip flop D y la configuración maestro esclavo - Flip flops J K y T - Registros - Contadores.

UNIDAD 3: COMPUTADORAS DIGITALES - ORGANIZACIÓN INTERNA DEL COMPUTADOR:

Organización Y Arquitectura - Estructura y Funcionamiento - Estructura-Organización Interna Del Computador - Funcionamiento Del Computador en general y del microprocesador, en particular - Estructuras De Interconexión - Interconexión Con Buses - Bus PCI.

UNIDAD 4: LENGUAJE ENSAMBLADOR DEL MICROPROCESADOR 8086/88

Fundamentos - Lenguaje de Bajo Nivel - Ventajas de los lenguajes ensambladores - Unidad central de proceso - Repertorio de Instrucciones - Registros internos del procesador, Funcionalidad. - Unidad de memoria principal - Modos de Direccionamiento - Unidad de Entradas y Salidas - Interconexion de las unidades funcionales - Registros de segmento - Registros de propósito general - Segmentos - Límites de los segmentos.-Desplazamiento - Concepto de interrupcion - Tabla de servicio de interrupcion - Elementos funcionales básicos: - La pila - Flujo de programa - Transferencia de datos. - Definicion de procedimientos. - Definición de funciones - Definicion de una macro - Instalación y uso del emulador Emu8086 - Reconocimiento del entorno funcional - Carga y ejecución de programas ejemplo - Nuestro propio programa en Assembly - Interrupciones, entrada por teclado, salida por pantalla, formatos.

7. Bibliografía obligatoria y complementaria (organizada por unidades)

- Miles J Murdocca, Principios de Arquitectura de Computadoras, Ed. Pearson, Prentice Hall
- William Stalling, Organización y Arquitectura de Computadoras, Ed. Prentice Hall
- Jorge Sinderman, TÉCNICAS DIGITALES, DISPOSITIVOS, CIRCUITOS, DISEÑO Y APLICACIONES, 2da. Edición, Nueva Librería, Marzo del 2007, ISBN 978 -987-1104-51-2.

De consulta:

- M. Morris Mano, Arquitectura de Computadoras, Ed. Prentice Hall.
- Thomas Floyd, Fundamento de los sistemas digitales, Ed. Prentice Hall
- Enrique Mandado, Sistemas Electrónicos Digitales, Ed. Marcombo
- Roger Tckheim, Principios Digitales, Ed. Shaums - Mc Graw Hill
- AC1 2018 - APUNTE de la cátedra.

RV

- AC1 2018 - Instalando EMU8086 - Guía del docente para instalar aplicativo.
- AC1 2018 - Introducción a Assembly - Apunte de la cátedra.

8. Metodología de trabajo

A partir de la fundamentación y de las expectativas de logro de la asignatura, daremos un enfoque metodológico teniendo en cuenta los conocimientos previos que fue adquiriendo el alumno durante su trayecto en la educación media y como usuario de computadoras. De esta manera a partir de las técnicas-instrumentales que cuenta el alumnado para la lecto/comprensión de textos, el vocabulario técnico/académico será lo suficientemente flexible como para desarrollar en las clases, tanto los aspectos teóricos como los aspectos técnico-instrumentales de la práctica. En este sentido se propone, en primer lugar, el estudio y la reflexión sobre las teorías y, posteriormente, su aplicación.

Las clases teóricas desarrollarán los contenidos de cada unidad con ejemplos diversos. Luego de cada tema teórico y ejemplos prácticos en pizarra, los alumnos resolverán ejercicios propuestos en varios trabajos domiciliarios. La cantidad de problemas a resolver y la variedad de los mismos será suficiente para considerar un adecuado aprendizaje significativo.

Se fomentará el trabajo en grupo para resolver los problemas prácticos, discutiendo estrategias, formulando conjeturas, reflexionando sobre procedimientos y resultados.

Los recursos metodológicos que se utilizarán en cada Unidad Didáctica son los siguientes:

- Planteamiento de la necesidad del estudio del tema a partir de problemas basados en situaciones reales.
- Explicación del tema por parte del profesor, con la intervención y participación de los alumnos/as y la realización de algunas actividades que sirvan para profundizar determinados aspectos del tema.
- Se proponen cuestionarios guía para abordar el material de consulta.
- Realización de tareas de investigación en equipo que posteriormente, expondrá en clase y se compartirá el material obtenido entre los mismos.

9. Evaluación (Requisitos de aprobación y criterios de evaluación)

Para alumnos regulares:

Cumplir con el cincuenta por ciento (50%) de asistencia a las actividades fijadas como obligatorias en el programa.

La nota de aprobación será de 4 (cuatro) para los dos exámenes parciales y el final, considerando que cumple con un 60% de conocimiento de los temas evaluados.

Aprobar los trabajos prácticos propuestos.

PV

Sólo existirá un recuperatorio para cada examen parcial. A esta instancia podrán acceder todos aquellos estudiantes que hubieran obtenido en el examen parcial una calificación inferior a siete (7) puntos o bien no hubieran asistido al mismo. En este caso la nota que los estudiantes obtengan en el recuperatorio, será la nota definitiva a los efectos de la evaluación.

Para aprobar la materia en forma Promocional se debe:

Cumplir con el ochenta por ciento (80 %) de asistencia a las actividades fijadas como obligatorias en el programa.

Aprobar los trabajos prácticos propuestos.

Aprobar cada parcial o su recuperatorio, con una nota no inferior a siete (7) puntos.

Criterios de Aprobación del Examen Final

Final de alumnos regulares:

- Para acceder al final en condición de alumno regular se deberá tener aprobada la cursada.

Caso contrario rendirá en condición de libre.

- Con examen escrito, teórico y práctico.
- Aprobación con nota igual o mayor a 4 (cuatro).

10. Instancias de práctica (opcional)

La práctica se abordará tanto en clase como con trabajos domiciliarios.

Los trabajos prácticos domiciliarios son de carácter obligatorio y deberán ser entregados en tiempo pautado, para poder corregidos y detectar errores comunes, que nos permitirán volver a abordar en detalle esos conceptos o resoluciones prácticas, antes de ser evaluados.

11. Cronograma de actividades teóricas y prácticas

Semana 1	El rol del analista - U1: Señales analógicas y Digitales. Conversión analógico-digital, ventajas y desventajas. - Sistemas de numeración: Decimal - Binario - Octal - Decimal - Hexadecimal - Cambios de base.
Semana 2	Códigos binarios y de cambio único. Código GRAY, conversión a binario. Codificación de alfanuméricos ASCII, ASCII extendido, Unicode y UCS. Operaciones aritméticas en binario entero Magnitudes: Suma, Resta, Producto y Cociente. - Codificación Signo y Magnitud - CA1 y CA2. Suma y resta de binarios codificados. Codificación de binario desplazado. Códigos binarios para decimales -

Handwritten signature



UNPAZ
Universidad Nacional de José C. Paz

= 01 =

	BCD Natural - BCD AIKEN - Exceso 3 - 2 de 5 - 7 Segmentos.
Semana 3	Representación de binarios en coma flotante normalizados en IEEE-754, 32bits. Lógica proposicional - Álgebra de BOOLE - Ley de De Morgan - Dualidad y Principio de dualidad - Teoremas del Álgebra de Conmutación - Tablas de verdad de funciones lógicas - Dualidad y Ley de SHANNON.
Semana 4	Expresiones canónicas de funciones - Minitérminos y Maxitérminos - Simplificación analítica y gráfica por KARNAUGH.
Semana 5	U2: Circuitos combinacionales con compuertas: FUNDAMENTALES, DERIVADAS Y EQUIVALENTES. - Control de flujo de señales - Representación y análisis de expresiones algebraicas con compuertas. Aplicación práctica aplicando de Morgan: Analítica y con Gráficamente.
Semana 6	Componentes digitales: Multiplexores, Demultiplexores, Decodificadores, Codificadores. Circuitos secuenciales: FLIP-FLOP Básico y sus mejoras - Aplicación en Circuitos Semisumadores y Sumadores.
Semana 7	REPASO - PARCIAL 1
Semana 8	U3: Computadoras digitales, Arquitectura y Organización. Estructura y funcionamiento de los componentes básicos del MICRO 8086 - Ciclo de Captación y ejecución, con y sin Interrupciones. Repertorio de instrucciones.
Semana 9	Interrupciones al Microprocesador. Módulos de E/s Entradas y Salidas a periféricos. Estructuras de Interconexión: BUSES - ARBITRAJE - El Bus PCI.
Semana 10	U4: Lenguajes de Alto y Bajo Nivel, ventajas y desventajas - ASSEMBLY - El microprocesador 8086 desde el emulador "Emu-8086" EMU8086: Instalación y entorno de funcionamiento - Estructura de un programa.
Semana 11	Registros de multipropósito y reservados - Modos de direccionamiento a M.P.- Instrucciones básicas aritméticas y lógicas - Flujo de programa con etiquetas y referencias a las banderas Registro de FLAGS.
Semana 12	Interrupciones: Salida a pantalla y captura por teclado - Formatos de salida.
Semana 13	Trabajo práctico final en EMU8086
Semana 14	Repaso y Parcial 2
Semana 15	Recuperatorio
Semana 16	Cierre de la materia.
<i>A partir de aquí a completar únicamente para unidades curriculares con régimen anual</i>	
Semana 17	

Semana 18	
Semana 19	
Semana 20	
Semana 21	
Semana 22	
Semana 23	
Semana 24	
Semana 25	
Semana 26	
Semana 27	
Semana 28	
Semana 29	
Semana 30	
Semana 31	
Semana 32	

Firma docente responsable	
Firma Departamento Académico	

[Handwritten signature]



PROGRAMA UNIDAD CURRICULAR			
Departamento	ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA		
Carrera/s	ANALISTA PROGRAMADOR UNIVERSITARIO		
Nombre de la unidad curricular	INGENIERÍA DE SOFTWARE II	Código	618
Docente responsable	VICTOR HUGO CONTRERAS		
Año de presentación del programa	2018		
1. Carga horaria			
Horas de clase semanales	6		
Horas de clase totales	96	Horas totales clases teóricas	48
		Horas totales clases prácticas	48
		Otras horas totales (laboratorio, trabajo de campo, etc.)	
2. Unidades correlativas precedentes en el Plan de Estudios			
Denominación		Código	
TALLER DE INGLES II		611	
INGENIERÍA DE SOFTWARE I		614	
3. Contenidos mínimos según Plan de Estudios			
Diseño Arquitectónico. Organización de la validación del diseño. Implementación. Estándares de programación y procedimientos. Verificación y Técnicas de Prueba. Pruebas de caja blanca. Pruebas de caja negra. Estrategias de prueba. Defectos y fallas. Validación. Sistema. Descomposición modular. Control. Diseño de interfaces de usuario. Diseño a nivel de componentes. Técnicas para la mejora del diseño. Evaluación y Planificación. Diseño de casos de prueba. Resultados. Documentación de las pruebas. La depuración: proceso, estrategia, corrección del error. Mantenimiento. Tipos de mantenimiento: correctivo, adaptativo, perfectivo, preventivo. Técnicas y herramientas para el mantenimiento. Rejuvenecimiento del software. Ingeniería Inversa. Reingeniería. Gestión de Proyectos. Diseño e Implementación. Métricas y Estimaciones. Planificación Temporal. Planificación Organizativa. Gestión del Riesgo. Gestión de la configuración del software: Línea base, gestión del cambio, control de versiones. Auditoría y Peritaje. Conceptos Objetivos.			

4. Fundamentación
Una regla de medida significativa que permite evaluar el grado de comprensión de los conocimientos es la elaboración de un sistema de información. Contribuyendo con un criterio analítico y unificando los conocimientos y conduciendo el saber, la práctica guiada permite robustecer el perfil profesional del futuro egresado. La construcción de sistemas de información adaptados a un contexto determinado, es promovida por la asignatura a través del uso de las técnicas modernas de modelización, estimulando las destrezas personales que faciliten las técnicas necesarias para aprovechar las diferentes métricas de calidad, con la finalidad de construir componentes de software y evaluar su rendimiento.
5. Objetivos
Sobre técnicas de desarrollo de software, el egresado debe afianzar los conocimientos con la finalidad de generar sistemas de información para las organizaciones e integrar los saberes sobre relevamiento, análisis y diseño de software. Recurrir a metodologías avanzadas para gestión de proyectos y percibir los compendios que se utilizan para lograr una óptima gestión del riesgo para construir aplicaciones que posean sólidos mecanismos de control y gestión de calidad. Implantar las técnicas que permitan incorporar los conceptos de ingeniería del software orientado a objetos, como también lograr conocimientos alusivos al ciclo de vida del producto (software) así como el análisis estratégico sobre la eficacia tiempo y forma del mantenimiento del mismo. Desarrollar las destrezas que se necesitan para proporcionar una planificación y seguimiento del proyecto conforme a las necesidades del mercado. Vincular los conocimientos de producción de software a las métricas concernientes, a fin de alcanzar indicadores que mejoren la rentabilidad en la producción.
6. Contenidos (organizados por unidades)
Unidad 1: Procesos para la creación de software. El Software. Del Producto. Definición y alcance de un sistema. Selección de herramientas. Análisis estratégico sobre la vida del producto (software). Pautas de Mantenimiento. Reconvertir Vs. Mantener. Del Proceso. Métodos y herramientas. Etapas en la construcción de un software. Modelos de procesos. Modelo lineal secuencial. Modelo de prototipos. El modelo DRA. Modelos evolutivos: modelo



incremental, modelo espiral, modelo de desarrollo concurrente. ensamblaje de componentes.

Unidad 2: Métricas

Diseño: del mapa de navegación, de las interfaces de usuario, de la gestión de errores, de la seguridad, del inicio y cierre de sesión, de los perfiles de usuarios, del sistema de Backup, del servicio de múltiples idiomas y de las bitácoras.

Métricas de proyectos. Medida, métrica e indicadores. Mediciones del software. Integración de las métricas al proceso de ingeniería del software. Adaptación de las métricas a pequeñas y medianas empresas. Métricas técnicas del software. Métricas para la calidad. Métricas del modelo de análisis. Métricas del modelo de diseño. Métricas del código fuente. Métricas para la prueba. Métricas de Mantenimiento.

Unidad 3: Gestión del riesgo y garantía de calidad

Riesgo del software. Identificación del riesgo. Proyección del riesgo. Tablas para la evaluación del riesgo. Evaluación del impacto. Refinamiento del riesgo. Reducción y gestión del riesgo. Riesgos y peligros para la seguridad. Plan de reducción, supervisión y gestión del riesgo RSGR.

Concepto de calidad. Garantía de calidad del software. Revisiones del software. Concepto de error y defecto. Amplificación y eliminación de errores. Revisiones técnicas formales. Fiabilidad del software. Fiabilidad y disponibilidad. Estándares de Calidad. ISO-9000. ISO-12207. CMM (Modelo de madurez de capacidades). CMMI (Integración del modelo de madurez de capacidades).

Unidad 4: Introducción a los patrones de diseño.

Concepto de arquitectura. Concepto de patrón de diseño. Consideraciones a tener en cuenta cuando se trabaja con patrones de diseño. Relación entre los patrones de diseño y el desarrollo de software. Análisis y detección de los problemas que pueden ser resueltos por patrones. Aspectos a considerar para seleccionar un patrón de diseño. Formas adecuadas y buenas prácticas para la utilización de un patrón de diseño.

Unidad 5: Prueba del Software

Técnicas de prueba. Fundamento de las pruebas del software. Diseño de casos de prueba. Pruebas de caja blanca. Pruebas del camino básico. Complejidad ciclomática. Prueba de la estructura de control. Pruebas de caja negra. Prueba de entornos especializados, arquitecturas y aplicaciones. Estrategias de prueba. Enfoque estratégico para la prueba de software. Aspectos estratégicos. Prueba de unidad.



Pruebas de integración. Integración ascendente. Integración descendente. Prueba de regresión. Prueba de humo. Pruebas de validación. Pruebas alfa. Pruebas beta. Prueba del sistema. Prueba de recuperación. Prueba de seguridad. Prueba de resistencia. Prueba de rendimiento. Depuración. Enfoques de la depuración.

7. Bibliografía obligatoria y complementaria (organizada por unidades)

- Pressman, Roger S. Ingeniería del software: un enfoque práctico. 7ta.ed.-- México,DF : McGraw-Hill, c2005. xxxiv, 958 páginas.
- Erich Gamma y otros. Patrones de Diseño. Elementos de software orientado a objetos reutilizable. Ed:Addison wesley, 2002 384 páginas.
- Sommerville, Ian; Alfonso Galipienso, María Isabel (Traductor); Botía Martínez, Antonio (Traductor) y otros. Ingeniería del software. 7a. ed.-- Madrid: Pearson Educación, 2005. xviii, 687 páginas
- Booch, Grady. Análisis y diseño orientado a objetos con aplicaciones. 2a. ed.-- México, DF: Addison Wesley, 1996. xix, 638 páginas
- Braude, Eric J. Ingeniería de software: una perspectiva orientada a objetos.-- México, DF : Alfaomega, c2003. xxiii, 539 páginas
- Pfleeger, Shari Lawrence; Ruíz de Mendarozqueta, Alvaro; Quiroga, Elvira(Traductor). Ingeniería de software : teoría y práctica.-- Buenos Aires : Pearson Education, c2002. xxv, 760 páginas
- Piattini, Mario; Villalba, José; Ruiz, Francisco; Bastanchury, Teresa; Polo, Macario; Martínez, Miguel Ángel ; Nistal, César. Mantenimiento del software : modelos, técnicas y métodos para la gestión del cambio. Alfaomega, 2001.

8. Metodología de trabajo

Las clases teóricas desarrollaran los contenidos de cada unidad con ejemplos diversos de aplicación en la vida cotidiana. Concluido el sustento teórico los alumnos resolverán un trabajo práctico, conteniendo problemas con todas las variantes posibles de los temas tratados. La cantidad de problemas a resolver y la variedad de los mismos será suficiente para considerar un adecuado aprendizaje significativo.

Se fomentará el trabajo en grupo para resolver los problemas prácticos, discutiendo estrategias, formulando conjeturas, reflexionando sobre procedimientos y resultados.

Los recursos metodológicos que se utilizan en cada Unidad Didáctica son los siguientes:

- Planteamiento de la necesidad del estudio del tema a partir de problemas basados en situaciones reales.
- Explicación del tema por parte del profesor/a con la intervención y participación de los alumnos/as y la realización de algunas actividades que sirvan para desarrollar determinados aspectos del tema.
- Realización de actividades de consolidación del tema.
- Resolución de problemas y actividades de refuerzo o ampliación según sea el caso.

Realización de tareas de investigación en equipo. Posteriormente, los resultados de cada grupo en el trabajo de investigación serán expuestos en clase, debatidos los resultados diferentes entre los grupos,





etc.	
9. Evaluación (Requisitos de aprobación y criterios de evaluación)	
Para aprobar la materia como alumno regular, se calificará: • Nota numérica (1 a 10) • La nota de aprobación será de 4 (cuatro), considerando que cumple con un 60% de conocimiento de los temas evaluados Para aprobación de Final de alumnos regulares: • Examen escrito. • Aprobación con nota igual o mayor a 4 (cuatro) Para aprobar la materia en forma Promocional se debe: • Aprobar las evaluaciones parciales teóricas, o su respectivo recuperatorio, con una nota igual o superior a 7 (siete) y se deberán tener aprobadas las correlativas • Aprobar los trabajos prácticos con una nota igual o superior a la nota 7 (siete) Para rendir el examen parcial se debe: • Asistir por lo menos al 80% de las clases teóricas – prácticas.	
10. Instancias de práctica (opcional)	
11. Cronograma de actividades teóricas y prácticas	
Semana 1	Diagnóstico - Presentación de la materia - Entrega de Guías teóricas
Semana 2	UNIDAD Nº 1 Procesos para la creación de software
Semana 3	UNIDAD Nº 1 Procesos para la creación de software
Semana 4	UNIDAD Nº 2 Métricas - Corrección Guías teóricas -Entrega practica taller
Semana 5	UNIDAD Nº 3 Gestión del riesgo y garantía de calidad
Semana 6	UNIDAD Nº 3 Gestión del riesgo y garantía de calidad
Semana 7	Primer examen parcial
Semana 8	Devolución de notas y corrección de parcial - Corrección Guías teóricas

[Handwritten signature]



UNPAZ
Universidad Nacional de José C. Paz

101

	UNIDAD Nº 4. Introducción a los patrones de diseño
Semana 9	Examen recuperatorio 1 Parcial
Semana 10	UNIDAD Nº 4-5. Introducción a los patrones de diseño. - Intro. Prueba del Software
Semana 11	UNIDAD Nº 5. Prueba del Software
Semana 12	UNIDAD Nº 5. Prueba del Software
Semana 13	Segundo examen parcial
Semana 14	Devolución de notas y corrección de parcial - Corrección de Trabajo de taller
Semana 15	Examen recuperatorio del Segundo parcial
Semana 16	Cierre de materia y Guías teóricas
A partir de aquí a completar únicamente para unidades curriculares con régimen anual	
Semana 17	
Semana 18	
Semana 19	
Semana 20	
Semana 21	
Semana 22	
Semana 23	
Semana 24	
Semana 25	
Semana 26	
Semana 27	
Semana 28	
Semana 29	
Semana 30	
Semana 31	
Semana 32	

Firma docente responsable	
---------------------------	--

[Handwritten signature]



UNPAZ
Universidad Nacional de José C. Paz

= 01

Firma Departamento Académico	

PV



PROGRAMA UNIDAD CURRICULAR			
Departamento	ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA		
Carrera/s	ANALISTA PROGRMADOR UNIVERSITARIO		
Nombre de la unidad curricular	Sistemas Operativos II	Código	622
Docente responsable	FERNANDO J. MARTINI		
Año de presentación del programa	2018		
1. Carga horaria			
Horas de clase semanales	5		
Horas de clase totales	80	Horas totales clases teóricas	55
		Horas totales clases prácticas	25
		Otras horas totales (laboratorio, trabajo de campo, etc.)	
2. Unidades correlativas precedentes en el Plan de Estudios			
Denominación		Código	
MATEMÁTICA DISCRETA		612	
SISTEMAS OPERATIVOS I		616	
3. Contenidos mínimos según Plan de Estudios			
Repaso de conceptos vistos en Sistemas operativos I. Bloqueo Mutuo. Uso de recursos. Colas de procesos esperando por un recurso. Exclusión mutua. Grafo de asignación de recursos. Gestión de interbloqueos. Prevención. Estado seguro. Recuperación. Interbloqueo en ambientes distribuidos. Concurrencia entre procesos. Concepto de Thread (hilo de ejecución). Estructura del thread. Arquitecturas Microkernel. Comunicación entre procesos. Operaciones. Comunicación directa simétrica. Comunicación asimétrica. Comunicación indirecta. Casillas de Correo. Tamaño de la casilla. Errores en la comunicación. Conceptos avanzados sobre procesos. Sincronización entre procesos. Protección. Dominios. Matriz de acceso. Seguridad. Requisitos de seguridad. Amenazas. Conceptos asociados: identidad y autenticación. Medidas de seguridad. Programas malignos. Nociones de arquitecturas multiprocesador. Convivencia de diferentes sistemas operativos. Emuladores. Sistemas Operativos Distribuidos. Diferencia entre Sistemas Operativos de Red y Distribuidos. Características de			

[Handwritten signature]



un sistema distribuido. Servicios remotos. Middleware. Concepto de plataforma. Portabilidad e interoperabilidad. Llamada a Procedimiento Remoto (RPC). Transacciones Distribuidas. Seguridad en sistemas Distribuidos. Virtualización. Computación en la nube.

4. Fundamentación

El hardware de una computadora, su circuitería y el software de aplicación, es decir los programas que el usuario emplea, necesitan de una interface. Esta interface es el Sistema Operativo.

El estudiante debe conocer adecuadamente este entorno sobre el cual desarrollará sus aplicaciones. Esta asignatura brinda los saberes correspondientes a las distintas arquitecturas existentes.

5. Objetivos

Que el alumno conozca detalladamente las funciones y características generales de un Sistema Operativo sea concentrado o distribuido.

Objetivos Específicos.

- Identificar las particularidades, características y algoritmos de los Sistemas Operativos.
- Identificar ventajas y desventajas de los diversos algoritmos utilizados en la administración de los -- distintos recursos de un sistema informático.
- Evaluar y comparar distintos Sistemas Operativos que brinda el mercado actual.
- Comprender el funcionamiento del software de base.
- Evaluar las posibilidades de los distintos sistemas operativos.
- Aplicar su operatoria en el diseño e implementación de aplicaciones.

6. Contenidos (organizados por unidades)

UNIDAD Nº 1. INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS.

Sistema Operativo:

- Definición
- Conceptos Básicos: Procesos, archivos, Llamadas al Sistema.
- Funciones y Objetivos.

Requisitos para el diseño de Sistemas Operativos.

Estructuras de los Sistemas Operativos:

- Sistemas Monolíticos.
- Sistemas con capas.
- Máquinas Virtuales
- Modelo Cliente-Servidor

Tipos de Sistemas Operativos

B

- Sistemas de Multiprocesamiento.
- Sistemas de Tiempo Compartido.
- Sistemas Distribuidos.
- Sistemas de Tiempo Real.

UNIDAD Nº 2. ADMINISTRACIÓN DE PROCESOS.

Procesos.

Definición:

- Características del modelo de procesos.
- Jerarquía de Procesos

Estados de un Proceso. Transiciones de estados.

Bloque de Control de Procesos. Bloque de Control de Sistemas.

Operaciones sobre Procesos.

Procesamiento de interrupciones

Tipos de Procesos.

Planificación de Procesos

Objetivos de la Planificación

Criterios de Planificación

Planificación Apropiativa Versus No Apropiativa

Algoritmos de Planificación. Evaluación de Algoritmos.

Tipos de planificación:

- Planificación del Primero en Entrar Primero en Salir (FIFO)
- Planificación de Asignación en Rueda (RR: Round Robin)
- Planificación del Trabajo Más Corto Primero (SJF)
- Planificación del Tiempo Restante Más Corto (SRT)
- Planificación el Siguiendo con Relación de Respuesta Máxima (HRN o HPRN)
- Planificación por Prioridad
- Colas con y sin Retroalimentación de Niveles Múltiples
- Política vs. Mecanismo de Planificación. Planificación de dos Niveles

Planificación en Tiempo Real

UNIDAD Nº 3. CONCURRENCIA. COORDINACIÓN DE PROCESOS. INTERBLOQUEO

Principios Generales de la Concurrencia.

8

- Coordinación entre Procesos.

- Condición de Carrera.

- Sección Crítica.

- Exclusión Mutua.

Exclusión Mutua. Soluciones por software:

- Algoritmos de Espera Activa. Decker, Peterson, Lamport

- Algoritmos de Espera no Activa. Semáforos. Monitores. Pasaje de Mensajes.

Exclusión Mutua. Soluciones por Hardware:

- Inhabilitación de Interrupciones

- Instrucciones Especiales de la Máquina

Problemas de Comunicación entre Procesos.

Definición y Ejemplos de Bloqueo (o Interbloqueo)

Conceptos de Recursos

Bloqueos y Condiciones Necesarias Para el Bloqueo

Modelación de Bloqueos

El Algoritmo del Avestruz o de Ostrich

Detección de Bloqueos

- Gráficas de Asignación de Recursos

- Reducción de Gráficas de Asignación de Recursos

- Detección de Bloqueos de Forma "Un Recurso de Cada Tipo"

- Detección de Bloqueos de Forma "Varios Recursos de Cada Tipo"

Recuperación de Bloqueos

- Recuperación Mediante la Apropiación

- Recuperación Mediante la Eliminación de Procesos

Evasión de Bloqueos

- Trayectorias de Recursos

- Estados Seguros e Inseguros

- El Algoritmo del Banquero (de Dijkstra) Para Solo Un Recurso

- El Algoritmo del Banquero (de Dijkstra) Para Varios Recursos

- Asignación de Recursos por el Algoritmo del Banquero

Debilidades del Algoritmo del Banquero. Prevención de Bloqueos

UNIDAD Nº 4. SEGURIDAD Y PROTECCIÓN.

PV



Introducción a la Seguridad de los Sistemas Operativos.

Principios de Diseño de Seguridad

Supervisión de amenazas

- Verificación de Amenazas

- Amplificación

Vigilancia

Penetración al Sistema Operativo

Modalidad de penetración

Asincronismo

Rastreo y Código clandestino

UNIDAD Nº 5. SISTEMAS OPERATIVOS DISTRIBUIDOS.

Introducción.

- Diferencia entre Sistemas Operativos de Red y Distribuidos. Características de un sistema distribuido.

- Servicios remotos. Middleware.

Concepto de plataforma. Portabilidad e interoperabilidad. Llamada a Procedimiento Remoto (RPC).

Transacciones Distribuidas. Seguridad en sistemas Distribuidos. Virtualización.

Concepto de middleware. Clúster.

- Concepto. Ejemplos.

- Concepto de Cloud Computing.

- Aplicaciones Grids Computacionales. Aplicaciones.

Sistemas operativos Móviles. Teléfonos y Tablets. Kernel. Ejemplos.

7. Bibliografía obligatoria y complementaria (organizada por unidades)

Obligatoria: Para todas las unidades

- Tanenbaum, Andrew S.; Escalona García, Roberto (Traductor); Levine Gutiérrez, Guillermo. Sistemas operativos modernos. 2a.ed.-- México, DF: Pearson Educación de México, 2003. xxiv, 951 páginas

- Stallings, William; Peña Sánchez, José María (Traductor); Pérez Costoya, Fernando (Traductor) y otros. Sistemas operativos: aspectos internos y principios de diseño. 5a.ed.-- Madrid: Pearson Educación, 2005, 848 páginas

Complementaria:

Para todas las unidades:

- Fundamento de los Sistemas Operativos (A.M.Lister - G.GILI 1986).

- Operating Systems (Madnik Donovan - McGraw Hill 1974).

- Sistemas Operativos (Francisco Rueda - McGraw Hill 1989).

- Sistemas Operativos - Diseño e Implementación (Andrew Tanenbaum - Prentice Hall 1998).

- The Design of the Unix Operating System (Bach - Prentice Hall 1986).

- Construcción de Sistemas Operativos (Jorge Boria - Kapeluz (IV EBAI 1989)).

PV



- Sistemas Operativos - Concepto y Diseño (Milan Millenovic-McGraw Hill 1988).

Unidad 1:

- Sistemas de Tiempo Compartido (Watson - El Ateneo 1977).

- Sistemas de Tiempo Compartido (Watson - El Ateneo 1977).

8. Metodología de trabajo

Las clases teóricas desarrollaran los contenidos de cada unidad con ejemplos diversos de aplicación en la vida cotidiana. Concluidos el sustento teórico los alumnos resolverán un trabajo práctico, conteniendo problemas con todas las variantes posibles de los temas tratados. La cantidad de problemas a resolver y la variedad de los mismos será suficiente para considerar un adecuado aprendizaje significativo.

Se fomentará el trabajo en grupo para resolver los problemas prácticos, discutiendo estrategias, formulando conjeturas, reflexionando sobre procedimientos y resultados.

Los recursos metodológicos que se utilizan en cada Unidad Didáctica son los siguientes:

- Planteamiento de la necesidad del estudio del tema a partir de problemas basados en situaciones reales.
- Explicación del tema por parte del profesor/a con la intervención y participación de los alumnos/as y la realización de algunas actividades que sirvan para desarrollar determinados aspectos del tema.
- Resolución de problemas y actividades de refuerzo o ampliación según el caso.

La actividad curricular se organiza en: clases teóricas, clases prácticas donde los alumnos trabajan sobre los ejercicios propuestos en la guía de trabajos prácticos, explicaciones de práctica y clases de consulta (de teoría y práctica).

Periódicamente se publican actividades (teórico-prácticas) que los alumnos pueden resolver.

El reglamento y cronograma tentativo son conocidos por los alumnos desde el inicio de la actividad curricular.

Se utiliza un entorno virtual de información-aprendizaje (Google Site), donde están disponibles clases, guías de TP, avisos, etc.

Para las clases teóricas y las explicaciones de práctica se utilizan PC, proyector y pizarrón.

Los alumnos pueden realizar prácticas en PC usando distintos lenguajes/bibliotecas que soportan concurrencia.

(Handwritten signature)

9. Evaluación (Requisitos de aprobación y criterios de evaluación)	
Para aprobar la materia en forma Promocional se debe: - Cumplir con el ochenta por ciento (80 %) de asistencia a las actividades fijadas como obligatorias en el programa. - Aprobar el taller correspondiente al estudio del Sistema Operativo GNU/LINUX, el cual tendrá la siguiente temática: . Sistema Operativo GNU/Linux. Características Generales. . Aspectos importantes para su implementación. Llamadas de Sistema. Ejemplos de desarrollo. . Administración de recursos. Este taller se cursará paralelamente al desarrollo de la asignatura. - Aprobar cada parcial o su recuperatorio con una nota no inferior a siete (7) puntos. Para aprobar la materia en condición de regular: • La nota de aprobación de las instancias parciales será de 4 (cuatro), considerando que cumple con un 60% de conocimiento de los temas evaluados Para aprobación de Final de alumnos regulares: - Se debe rendir un examen escrito (teórico y práctico). - Aprobación con nota igual o mayor a 4 (cuatro).	
10. Instancias de práctica (opcional)	
Aprobar el taller correspondiente al estudio del Sistema Operativo GNU/LINUX, el cual tendrá la siguiente temática: • Sistema Operativo GNU/Linux. Características Generales. • Aspectos importantes para su implementación. Llamadas de Sistema. Ejemplos de desarrollo. . Administración de recursos. Este taller se cursará paralelamente al desarrollo de la asignatura.	
11. Cronograma de actividades teóricas y prácticas	
Semana 1	INTRODUCCIÓN A LOS SISTEMAS OPERATIVO - Taller linux
Semana 2	ADMINISTRACIÓN DE PROCESOS - Taller linux
Semana 3	ADMINISTRACIÓN DE PROCESOS- Taller linux
Semana 4	CONCURRENCIA. COORDINACIÓN DE PROCESOS. INTERBLOQUEO Taller linux
Semana 5	CONCURRENCIA. COORDINACIÓN DE PROCESOS. INTERBLOQUEO - Taller linux

[Handwritten signature]



Semana 6	Primer examen parcial
Semana 7	Devolución de notas y corrección de parcial - SEGURIDAD Y PROTECCIÓN. Taller linux
Semana 8	Examen recuperatorio
Semana 9	SISTEMAS OPERATIVOS DISTRIBUIDOS Taller linux
Semana 10	SISTEMAS OPERATIVOS DISTRIBUIDOS - Taller linux
Semana 11	SEGURIDAD Taller linux
Semana 12	SEGURIDAD - Taller linux
Semana 13	Segundo examen parcial
Semana 14	Devolución de notas del segundo parcial
Semana 15	Examen recuperatorio.
Semana 16	Devolucion de nota de recuperatorio
<i>A partir de aquí a completar únicamente para unidades curriculares con régimen anual</i>	
Semana 17	
Semana 18	
Semana 19	
Semana 20	
Semana 21	
Semana 22	
Semana 23	
Semana 24	
Semana 25	
Semana 26	
Semana 27	
Semana 28	
Semana 29	
Semana 30	
Semana 31	
Semana 32	

--	--

[Handwritten signature]



01

Firma docente responsable	
Firma Departamento Académico	

PV