

- 07 -
José C. Paz, 29 NOV 2019

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. 782/2019, 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: Álgebra y Análisis I (cod. 601), Algoritmos y Programación I (cod. 603), Paradigmas de Programación (cod. 610), Matemática Discreta (cod. 612).

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° 15, de carácter ordinaria, registrada en el Acta N° 15 del 6 de noviembre de 2019, este Cuerpo Colegiado compartió los términos y

UNPAZ
M
R

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**

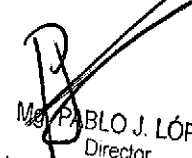
DISPONE:

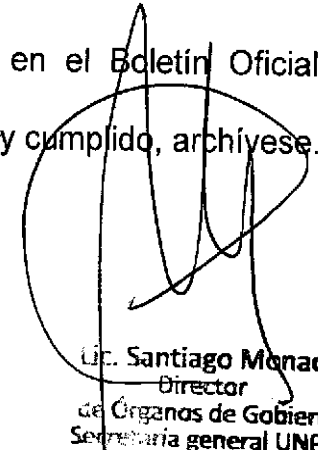
ARTÍCULO 1º.- Apruébanse los programas de la carrera de Analista Programador Universitario que se adjuntan como Anexo a la presente, correspondientes a las siguientes asignaturas: Álgebra y Análisis I (cod. 601), Algoritmos y Programación I (cod. 603), Paradigmas de Programación (cod. 610), Matemática Discreta (cod. 612).

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
M
P


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


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



PROGRAMA UNIDAD CURRICULAR			
Departamento	ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA		
Carrera/s	ANALISTA PROGRAMADOR UNIVERSITARIO		
Nombre de la unidad curricular	ÁLGEBRA Y ANÁLISIS I	Código	01
Docente responsable	GISELE AZUL OLMOS		
Año de presentación del programa	2019		
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	
3. Contenidos mínimos según Plan de Estudios			
Álgebra. Números naturales. Inducción. Combinatoria. Polinomios. Números complejos. Vectores en el plano y el espacio. Ángulo entre vectores, distancia, norma, desigualdad de Cauchy-Schwarz. Producto escalar, vectorial y mixto. Ecuación de la recta y del plano. Cónicas y cuádricas. Sistemas de ecuaciones lineales. Matrices. Determinantes. Regla de Cramer. Métodos de escalonamiento (pivote, triangulación, Gauss, factorización LU). Análisis Matemático. Funciones en R. Sucesiones. Límite, continuidad y derivada en R. Aproximaciones de números reales. Teoremas del Valor Medio. Desarrollo en serie de potencias (Taylor). Convergencia. Máximos y mínimos. Integral definida e indefinida. Técnicas de integración. Aplicaciones geométricas y físicas (áreas, volúmenes, longitud de curvas).			

27



4. Fundamentación

La asignatura Álgebra y Análisis I corresponde al primer cuatrimestre de la carrera de pregrado Analista Programador Universitario y aborda algunos fundamentos matemáticos necesarios para que el estudiante pueda trabajar con los contenidos, tanto de las materias que tienen a la presente como correlativa, como así también para su futura labor profesional.

Los fundamentos y prácticas de matemática son esenciales en la formación y trabajo cotidiano de un programador, no obstante es conocido el rechazo que en general existe hacia la matemática en la sociedad producto, en parte, a una percepción de que la comprensión de la misma le corresponde a algunos pocos "iluminados". Por este motivo, esta asignatura tendrá como uno de sus principales objetivos acercar conceptos matemáticos al quehacer cotidiano y a la intuición.

Los contenidos son seleccionados y organizados de manera tal que sigan una secuencia que permita a los estudiantes vincular los contenidos matemáticos con conceptos y prácticas vinculadas con el eje central de su carrera. Por otro lado, se tiene en consideración, que ésta es la primera asignatura de matemática con la que los estudiantes se enfrentan en la Universidad, y por lo tanto se privilegian abordajes de contenidos "ceranos" a los trabajados en la escuela media, principalmente al comienzo del cronograma. En este sentido, siempre que fue posible, se ubicaron al inicio los contenidos cuyo tratamiento requiera menor nivel de abstracción, en todos los casos preservando una coherencia lógica entre los mismos.

5. Objetivos

Que los estudiantes logren:

- Desarrollar la capacidad de interpretar definiciones y resultados matemáticos desconocidos.
- Realizar justificaciones matemáticas sencillas: métodos directos e indirectos.
- Adquirir la capacidad de utilizar algunas heurísticas para resolver problemas.
- Aprender a argumentar con razonamientos matemáticamente válidos.
- Vincularse con la matemática naturalmente y entenderla como parte cotidiana del trabajo de un analista programador universitario.
- Aplicar conocimientos matemáticos al desarrollo de softwares

6. Contenidos (organizados por unidades)

Unidad 1 Números y vectores: Números naturales. Inducción. Combinatoria. Números enteros, racionales, reales y complejos. Operaciones básicas. Vectores. Álgebra vectorial en R^2 y R^3 . Ángulo entre vectores, distancia y norma. Producto escalar, vectorial y mixto. Ecuación vectorial de la recta y el plano. Cónicas y cuádricas.



Unidad 2 Sistemas de ecuaciones lineales y matrices: Polinomios. Sistemas de ecuaciones lineales. Matrices. Determinantes. Regla de Cramer. Métodos de escalonamiento (pivote, triangulación, Gauss, factorización LU).

Unidad 3 Estudio de funciones reales: Análisis Matemático. Funciones en $\mathbb{R}$. Sucesiones. Límite, continuidad y derivada en $\mathbb{R}$. Aproximaciones de números reales. Teoremas del Valor Medio. Desarrollo en serie de potencias (Taylor). Convergencia. Máximos y mínimos.

Unidad 4 Integrales: Integral definida e indefinida. Técnicas de integración. Aplicaciones geométricas y físicas (áreas, volúmenes, longitud de curvas).

7. Bibliografía obligatoria y complementaria (organizada por unidades)

Obligatoria:

Unidades 1 y 2

- Lang, Serge (1990). Introducción al álgebra lineal. Addison-Wesley iberoamericana, Argentina.

Unidades 3 y 4

- Bianco, M., García, R. y Zorzoli, G. (2000). Análisis Matemático I, Notas de teoría y práctica. Buenos Aires: Eudeba.

Complementaria:

- Noriega, Ricardo J. (1991). Cálculo Diferencial e Integral. Edit. Docencia. S.A. Buenos Aires.
- Stewart James. (2008). CÁLCULO de una variable, México, Editorial Progresso.
- Grossman Stanley I. (2004), Álgebra lineal, México, Ed. Mc Graw Hill.
- Lheman, Charles H. (1980). Geometría Analítica. Ed. Limusa.
- Novelli, Alfredo; Elli, Eduardo. (2003). Cálculo. Ed. UNLU.

8. Metodología de trabajo

Los estudiantes dispondrán de una serie de guías de actividades que abarcan la totalidad de los temas tratados en la asignatura.

Los docentes iniciarán los temas con exposiciones dialogadas de carácter, esencialmente teóricos aunque eventualmente se resolverán ejercicios y problemas a modo de ejemplo. Luego, los docentes alternarán su rol, entre asistentes en la resolución de actividades que les propongan a los estudiantes y explicaciones en el pizarrón sobre cómo resolver actividades.

Los docentes indicarán, a finalizar cada clase, actividades de las prácticas que los estudiantes están en condiciones de resolver para retomar la clase siguiente.

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



La asignatura será regularizada cuando el/la estudiante haya cumplido con un mínimo del 75% (setenta y cinco por ciento) de la asistencia y haya obtenido en las instancias evaluatorias parciales (o sus recuperatorios) una calificación de 4 (cuatro) puntos o superior.

Se tomarán dos EXÁMENES PARCIALES y de dos EXÁMENES RECUPERATORIOS. Que serán tomados durante las semanas de clases, en el horario usual de cursada. Las notas de los recuperatorios reemplazan la de sus respectivos parciales

Los/as estudiantes que mantengan la condición regular, podrán aprobar mediante:

i. Promoción directa. Estarán aprobados mediante promoción directa, aquellos/as estudiantes que: hayan mantenido su condición de regularidad al final del curso y, hayan obtenido una calificación de 7 (siete) o más puntos como promedio de todas las instancias evaluativas, sean éstas parciales o sus recuperatorios, debiendo obtener una nota igual o mayor a 6 (seis) puntos en cada una de éstas.

ii. Evaluación Integradora. Quedarán habilitados automáticamente para rendir la evaluación integradora aquellos/as estudiantes que: hayan mantenido su condición de regularidad al final del curso y, hayan obtenido una calificación entre 4 (cuatro) y 6 (seis) puntos en promedio de las instancias parciales y como mínimo un 4 (cuatro) en cada instancia o en sus respectivos recuperatorios.

Estarán aprobados mediante evaluación integradora, aquellos/as estudiantes que obtengan en dicha instancia una calificación de 4 (cuatro) o más puntos.

iii. Examen Final. Podrán aprobar mediante examen final los/as estudiantes que: hayan mantenido su condición de regularidad al final del curso, hayan obtenido una calificación entre 4 (cuatro) y 6 (seis) en los respectivos exámenes parciales y/o sus recuperatorios, pero no hubieren aprobado o asistido a la instancia del examen integrador. Para rendir examen final los estudiantes deberán inscribirse en las fechas previstas en el Calendario Académico. La nota de aprobación del examen será de 4 (cuatro) puntos o más.

Régimen de aprobación en exámenes libres:

La modalidad del examen libre será escrita y oral, siendo la primera instancia de carácter previa y eliminatoria. Se evaluarán todos los contenidos establecidos en el programa correspondiente a la fecha del examen. La calificación mínima establecida para la aprobación de la asignatura en examen libre es de 4 (cuatro) puntos.



11. Cronograma de actividades teóricas y prácticas	
Semana 1	Desarrollo de la Unidad 1
Semana 2	Desarrollo de la Unidad 1
Semana 3	Desarrollo de la Unidad 1
Semana 4	Desarrollo de la Unidad 2
Semana 5	Desarrollo de la Unidad 2
Semana 6	Desarrollo de la Unidad 2
Semana 7	Repaso y Primer Parcial
Semana 8	Desarrollo de la Unidad 3
Semana 9	Desarrollo de la Unidad 3
Semana 10	Desarrollo de la Unidad 3
Semana 11	Desarrollo de la Unidad 3
Semana 12	Desarrollo de la Unidad 4
Semana 13	Desarrollo de la Unidad 4
Semana 14	Repaso y Segundo Parcial Entrega de notas y consultas
Semana 15	Recuperatorio del Segundo Parcial Entrega de notas y consultas
Semana 16	Recuperatorio del Primer Parcial Entrega de notas y cierre de actas
<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	

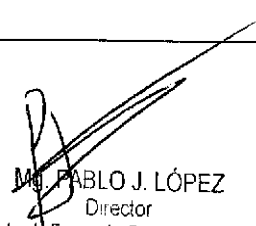
[Handwritten signature]



UNPAZ
Universidad Nacional de José C. Paz

3.7

Semana 26	
Semana 27	
Semana 28	
Semana 29	
Semana 30	
Semana 31	
Semana 32	

Firma docente responsable	
Firma Departamento Académico	 MR. PABLO J. LÓPEZ Director Depto. de Economía, Prod. e Innov. Tec. Universidad Nacional de José C. Paz



07

PROGRAMA UNIDAD CURRICULAR			
Departamento	ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA		
Carrera/s	ANALISTA PROGRAMADOR UNIVERSITARIO		
Nombre de la unidad curricular	MATEMÁTICA DISCRETA	Código	612
Docente responsable	EDGARDO SAMPAOLES		
Año de presentación del programa	2019		
1. Carga horaria			
Horas de clase semanales	6		
Horas de clase totales	96	Horas totales clases teóricas	3
		Horas totales clases prácticas	3
		Otras horas totales (laboratorio, trabajo de campo, etc.)	
2. Unidades correlativas precedentes en el Plan de Estudios			
Denominación		Código	
ALGORITMOS Y PROGRAMACIÓN I		03	
ÁLGEBRA Y ANÁLISIS II		05	
3. Contenidos mínimos según Plan de Estudios			
Elementos de la Lógica Formal. Noción y objeto de la Lógica Formal. Reseña Histórica de la Lógica Formal. Principios Fundamentales de la Lógica. Estructuras de la Lógica: términos, proposiciones, razonamientos. Clasificación: Términos, proposiciones y razonamientos. Lógica Proposicional. Presentación y especificación del sistema formal proposicional. Evaluación semántica de formulas. Equivalencias lógicas y leyes lógicas. Teoría de la demostración: Deducción natural o método derivativo. Reducción al absurdo. Fórmulas bien formadas, definición recursiva de fórmula. Prueba de Invalidez. Propiedades de los sistemas axiomáticos: consistencia, independencia y completitud. Lógica de Predicados. Objeto de la lógica de predicado. Lenguaje formal. Tipo de proposiciones predicativas. Cuadro de la oposición de la lógica de predicados. Definición semántica de cuantificadores y evaluación semántica de formulas. Reglas. Métodos de Deducción. Teoría de Grafos. Grafos y multigrafos. Caminos, trayectorias y ciclos. Conexión. Propiedades. Grafos especiales. Grafos y			

B



matrices. Matrices de arcos, de adyacencia, de incidencia y de conexión. Multigrafos recorribles. Trayectorias y circuitos: de Euler y Hamilton. Propiedades. Existencia de trayectos y circuitos. Algoritmo de Fleury. Grafos dirigidos. Grafos etiquetados. Grafos isomorfos. Árboles: elementos y definición. Reticulados. Definición de orden. Propiedades. Conjuntos parcialmente ordenados. Diagramas de Hasse. Reticulados. Definición. Reticulados distribuidos. Reticulados acotados. Complementos. Reticulados complementados. Álgebras de Boole. Propiedades.

4. Fundamentación

Las matemáticas se basan en la teoría de conjuntos y su conocimiento básico para la informática es fundamental. La lógica, para un informático, debe ser un lenguaje formal más, para el que existe una interpretación (una semántica) muchas veces estándar. En la sintaxis del lenguaje se incluyen símbolos para variables numéricas y elementos de estructuras de datos informáticos. Hay que saber qué la lógica que usan los informáticos tiene esta interpretación conjuntista, lo que la hace fácil de entender y de usar cuando se adquieren esos conocimientos.

Las relaciones entre conjuntos son más generales que las funciones. Es bueno tener claro que una relación es un subconjunto de un producto cartesiano y que hay propiedades de las relaciones que las organizan y conectan con otros conceptos matemáticos; de tal forma que se puede lograr una estructura del pensamiento lógico que será óptimo para trabajar en informática.

Las congruencias aritméticas permiten resolver problemas que, de otra forma, tendrían soluciones complicadas de explicar y entender. La aritmética de congruencias se puede mirar como una aplicación de la teoría de grupos cíclicos. De hecho, conocer álgebra abstracta de estructuras finitas es una ventaja para entender varios conceptos fundamentales en informática.

Otorgarle al alumno los elementos necesarios para la confección de modelos lógicos matemáticos en las materias aplicativas, en especial en lo referente a problemas de estructuras y algoritmos.

El enfoque del dictado se orienta a proveer al alumno la capacidad de plantear y resolver problemas expresando ideas con claridad y concisión, los fundamentos matemáticos para tratar con medios informáticos y los conceptos básicos para adentrarse en el conocimiento del cálculo.

Dentro del plan de estudio, la materia está ubicada en el primer cuatrimestre del segundo año de la carrera, previamente debió superar dos correlativas como son Álgebra y Análisis II y Algoritmo y Programación II, a su vez le aportará las herramientas necesarias para sortear algunas dificultades de las dos últimas materias y la Práctica Integradora. Matemática Discreta es materia correlativa de las tres últimas materias del plan de estudio.

5. Objetivos

Favorecer la inserción de contenidos no abordados en la formación básica homogénea; tales como



Lógica, Teoría de Conjuntos, estructuras algebraicas, atendiendo siempre a la resolución de problemas.

Potenciar prácticas investigativas como un eje curricular transversal en donde relacionar e integrar conocimientos provenientes tanto de otras asignaturas, como del contexto natural y social, para aprender de este modo a tratar la complejidad de las situaciones.

Reconocer la aplicación de métodos inductivos y deductivos en la resolución de situaciones problemáticas y las demostraciones matemáticas.

Presentar un análisis adecuado de los temas para el estudiante de informática, que le servirán de base para realizar cursos más avanzados en áreas específicas de su carrera.

En general, el objetivo es que el alumno logre comprender y aplicar los conceptos y propiedades de las estructuras matemáticas, para la representación y estudio de fenómenos discretos.

6. Contenidos (organizados por unidades)

Unidad I: Teoría de conjunto: principios y definiciones. Operaciones con conjuntos. Principio de inclusión-exclusión. Conjuntos borrosos. Partición de un conjunto. Conteo.

Unidad II: Relaciones: pares y producto cartesiano. Relaciones binarias. Representación gráfica de las relaciones. Propiedad de las relaciones. Operaciones con relaciones. Relaciones de equivalencia

Unidad III: ¿Qué es la Lógica? Premisas y conclusiones. El reconocimiento de razonamientos. Deducción e inducción. Verdad y validez. Los usos del lenguaje. Falacias. Elementos de Lógica Formal. Lógica elemental de proposiciones. Proposición. Operación lógica binaria. Tabla de verdad. Negación. Disyunción. Conjunción. Implicaciones y equivalencias. Implicación. Equivalencia. Tautología. Teorema. Demostración por reducción al absurdo. El principio de inducción.

Unidad IV: Teoría de Grafos: Introducción. Definiciones básicas. Caminos y conceptos relacionados. Grafo conexo y subgrafo. Grafos y multigrafos. Caminos, trayectorias y ciclos. Conexión. Propiedades. Grafos especiales. Grafos y matrices. Matrices de arcos, de adyacencia, de incidencia y de conexión. Multigrafos recorribles. Trayectorias y circuitos: de Euler y Hamilton.

Unidad V: Árboles: elementos y definición. Reticulados. Definición de orden. Propiedades. Conjuntos parcialmente ordenados. Diagramas de Hasse. Reticulados. Definición. Reticulados distribuidos. Reticulados acotados. Complementos. Reticulados complementados. Búsqueda de árboles maximales. Árbol generador minimal. Árbol generador máximo. Árboles con raíz. Recorrido de árboles. Formas polacas. Álgebras de Boole. Propiedades.

P



7. Bibliografía obligatoria y complementaria (organizada por unidades)

Obligatoria:

- Matemática Discreta – Felix Garcia Merayo – Paraninfo – 3ª edición 2015.
Unidad I: cap. 3; Unidad II: cap. 4; Unidad III: cap. 10; Unidad IV: cap. 7; Unidad V: cap. 8.
- Matemáticas Discretas – Richard Johnsonbaugh – Grupo Editorial Iberoamérica – Edición 1988.
Unidad I: cap. 2; Unidad II: cap. 2; Unidad III: cap. 1; Unidad IV: cap. 4; Unidad V: cap. 5.
- Matemáticas Discreta y Combinatoria – Ralph P. Grimaldi – Edit. Addison Wesley Iberoamericana. – Edición 1997.
Unidad I: cap. 3 (P1); Unidad II: cap. 5 (P1); Unidad III: cap. 2 (P1); Unidad IV: cap. 11(P3); Unidad V: cap. 12 (P3).

Complementaria:

- Matemática Discreta – Susana Granado Peralta – Edit. CEIT - 4ª edición 2009.
Unidad I: Cap. 2; Unidad II: cap. 7; Unidad III: cap. 1; Unidad IV: cap. 17; Unidad V: cap. 17.
- Estructuras de Matemáticas Discretas para la Computación – Bernard Kolman – Robert C. Busby – Edit. Prentice Hall – Edición 1984
Unidad I: cap. 1; Unidad II: s/d; Unidad III: Apéndice; Unidad IV: cap. 2; Unidad V: cap. 5.

8. Metodología de trabajo

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

De acuerdo con el Régimen General de Estudios, los/as estudiantes que mantengan la condición regular, conforme a lo previsto en el artículo 21 del RGE, podrán aprobarla:

- (i) mediante promoción directa;
- (ii) mediante aprobación de examen integrador;
- (iii) mediante examen final

En cualquiera de los casos se requiere un mínimo del 75% de asistencia a clase o tener aprobado un Régimen Especial de Cursada.

Para aprobar la asignatura por promoción directa se requiere que:

- (i) hayan mantenido su condición de regularidad al final del curso, conforme a lo previsto en el artículo 21 del RGE y,
- (ii) hayan obtenido una calificación de 7 (siete) o más puntos como promedio de las dos instancias evaluativas parciales presenciales o sus recuperatorios, debiendo obtener una nota igual o mayor a 6 (seis) puntos en cada una de éstas.

Para aprobar a través de examen integrador se requiere:

[Handwritten signature]



(i) hayan mantenido su condición de regularidad al final del curso, conforme a lo previsto en el artículo 21 del RGE y,

(ii) hayan obtenido una calificación entre 4 (cuatro) y 6 (seis) puntos en promedio de las dos instancias evaluativas parciales presenciales y como mínimo un 4 (cuatro) en cada instancia o sus respectivos recuperatorios.

Los estudiantes que no aprueben por promoción o por examen integrador tendrán la posibilidad de aprobar la asignatura mediante examen final, siempre que:

(i) hayan mantenido su condición de regularidad al final del curso, conforme lo previsto en el artículo 21 del RGE y,

(ii) hayan obtenido una calificación entre 4 (cuatro) y 6 (seis) puntos en los respectivos exámenes parciales presenciales y/o sus respectivos recuperatorios, pero no hubiesen aprobado o asistido a la instancia del examen integrador.

10. Instancias de práctica (opcional)

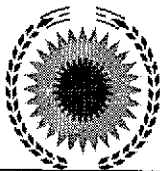
11. Cronograma de actividades teóricas y prácticas

Semana 1	Teórico: Conjuntos. Igualdad de conjuntos. Conjunto Vacio. Conjunto Universal. Operaciones con conjuntos. Práctico: Realización de Ejercicios de aplicación.
Semana 2	Teórico: Representación de las relaciones. Propiedad de las Relaciones en un conjunto. Reflexiva, Simétrica Transitiva. Relación de Equivalencia. Relación de orden amplio. Relación de orden extricto. Conteo. Práctico: Realización de Ejercicios de aplicación.
Semana 3	Teórico: Elementos de la Lógica Formal. Noción y objeto de la Lógica Formal. Principios Fundamentales de la Lógica. Estructuras de la Lógica: términos, proposiciones, razonamientos. Clasificación: Términos, proposiciones y razonamientos. Lógica Proposicional. Presentación y especificación del sistema formal proposicional. Equivalencias lógicas y leyes lógicas. Práctico: Ejercicios de aplicación.
Semana 4	Teórico: Lógica de Predicados. Objeto de la lógica de predicado. Lenguaje formal. Tipo de proposiciones predicativas. Cuadro de la oposición de la lógica de predicados.

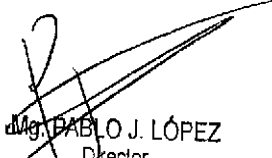


	Definición semántica de cuantificadores y evaluación semántica de fórmulas. Reglas. Métodos de Deducción. Práctico: Ejercicios de aplicación.
Semana 5	Teórico: Teoría de grafos: Introducción. Definiciones básicas. Caminos y conceptos relacionados. Grafo conexo y subgrafo. Grafos y multigrafos. Caminos, trayectorias y ciclos. Conexión. Práctico: Ejercicios de aplicación.
Semana 6	Consultas y repaso para el parcial. Primera evaluación parcial
Semana 7	Revisión de los temas incluidos en la primera evaluación parcial. Recuperatorio de la primera evaluación parcial.
Semana 8	Teórico: Teoría de grafos. Grafos especiales. Grafos y matrices. Matrices de arcos, de adyacencia, de incidencia y de conexión. Multigrafos recorribles. Trayectorias y circuitos: de Euler y Hamilton. Práctico: Realización de Ejercicios de aplicación.
Semana 9	Teórico: Teoría de grafos. Propiedades. Existencia de trayectos y circuitos. Algoritmo de Fleury. Grafos dirigidos. Grafos etiquetados. Grado de un vértice. Sucesión de grados. Grafos isomorfos. Práctico: Realización de Ejercicios de aplicación.
Semana 10	Teórico: Complementos de un grafo. Tipos de distancias entre vértices. Articulaciones y puentes. Representación matricial de grafos. Grafos eulerianos. Grafos hamiltonianos. Emparejamientos. Práctico: Realización de Ejercicios de aplicación.
Semana 11	Teórico: Árboles. Elementos y definición. Reticulados. Definición de orden. Propiedades. Conjuntos parcialmente ordenados. Diagramas de Hasse. Reticulados. Definición. Reticulados distribuidos. Práctico: Realización de Ejercicios de aplicación.
Semana 12	Teórico: Árboles. Reticulados acotados. Complementos. Reticulados complementados. Búsqueda de árboles maximales. Árbol generador minimal. Árbol generador máximo. Árboles con raíz. Recorrido de árboles. Formas polacas. Práctico: Realización de Ejercicios de aplicación.
Semana 13	Consultas y repaso para el parcial. Práctica: Realización de Ejercicios de aplicación..

2



Semana 14	Segunda evaluación parcial.
Semana 15	Revisión de los temas incluidos en la segunda evaluación parcial. Recuperatorio segunda evaluación parcial.
Semana 16	Clase de consulta para examen Integrador y/o Finales.
<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	 Mag. PABLO J. LÓPEZ Director Depto. de Economía, Prod. e Innov. Tec. Universidad Nacional de José C. Paz



PROGRAMA UNIDAD CURRICULAR			
Departamento	ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA		
Carrera/s	ANALISTA PROGRAMADOR UNIVERSITARIO		
Nombre de la unidad curricular	ALGORITMOS Y PROGRAMACION I	Código	603
Docente responsable	GONZALEZ TULIAN GERARDO		
Año de presentación del programa	2019		
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	
3. Contenidos mínimos según Plan de Estudios			
Introducción. Sistemas de información. Esquema funcional de una computadora. Conceptos básicos. Algoritmos. Acciones elementales. Estructuras de control. Modelo de máquina abstracta. Estructura esquemática de un programa para una máquina abstracta. Corrección de algoritmos. Importancia de la verificación. Eficiencia de un algoritmo. Importancia de la documentación de un algoritmo. Relación de los conceptos anteriores con el modelo de máquina abstracta. Ejemplos. Datos y Tipos de datos. Constantes y variables. Tipos de datos simples y compuestos. Tipos de datos primitivos. Tipos de datos definidos por el usuario. Funciones predefinidas. Tipos ordinales. Modularización. Procedimientos y funciones. Parámetros. Descomposición de problemas. Utilidad e importancia. La noción de reusabilidad. Subprogramas o módulos. Procedimientos. Funciones. Conceptos de argumentos y parámetros. Conceptos de variables locales y variables globales. Procedimientos y funciones con parámetros. Manejo de memoria en ejecución. Estructuras de datos. Introducción y clasificación de las			

2



estructuras de datos. Registros. Arreglos. Operaciones con arreglos de una dimensión. Matrices. Tratamiento de información estructurada en vectores y matrices. Algoritmos de búsqueda. Algoritmos de ordenación. Ordenación por índice.

4. Fundamentación

La asignatura Algoritmos y Programación I se ubica en el primer cuatrimestre del primer año según el plan de estudios vigente y se enmarca en el área de tecnologías básicas. Los contenidos son la base para las asignaturas correlativas: Algoritmos y Programación II, Bases de Datos I y Matemática Discreta.

La asignatura reviste vital importancia para el perfil de egresado, ya que a partir de su cursada el estudiante adquiere los fundamentos del paradigma de programación estructurada y la forma de implementar en un lenguaje de programación problemas de pequeña y mediana complejidad.

Los contenidos trabajados aportan un valor esencial en el pensamiento sistémico y algorítmico que es fundamental en la etapa inicial de la formación de los estudiantes.

5. Objetivos

- Formar el razonamiento para la resolución de distintos problemas que nos presenta la programación, aplicando en cada uno de ellos la solución más eficaz.
- Adquirir habilidad en la resolución de los problemas informáticos.
- Aplicar conceptos en Programación estructurada.
- Resolver problemas lógicos de baja y media complejidad.
- Conocer las herramientas de la programación
- Incentivar el espíritu de investigación e incursión en diversos lenguajes de programación.

6. Contenidos (organizados por unidades)

UNIDAD I: Sistemas de Información y Algoritmos

Introducción a los sistemas de Información. Breve revisión de los antecedentes históricos. Hardware y software. Arquitectura del hardware. Diagrama de Von Neumann. Unidad central de proceso. Memoria principal. Memorias auxiliares. Periféricos. Esquema funcional de una computadora. Modelo de máquina abstracta. Estructura esquemática de un programa para una máquina abstracta. Conceptos de Algoritmos. Distintas técnicas de representación de algoritmos. Diagramas Estructurados. Representación en pseudocódigo de los algoritmos. Etapas en la construcción y prueba



de algoritmos Acciones elementales. Estructuras de control. Corrección de algoritmos. Importancia de la verificación. Eficiencia de un algoritmo. Importancia de la documentación de un algoritmo. Relación de los conceptos anteriores con el modelo de máquina abstracta. Ejercitación práctica de los temas de la unidad.

UNIDAD II: Estructuras de Control, Datos y Tipos de Datos

Estructura de un programa. Variables y Constantes. Declaraciones e identificadores de variables, alcance de las variables, globales y locales. Declaración e identificación de las constantes. Operadores y expresiones.

Algoritmos lineales y no lineales. Ruptura de la secuencialidad. Estructuras de decisión. Decisión simple. Decisiones anidadas. Decisión múltiple. Estructuras iterativas. Estructura "repetir-mientras". Estructura "mientras-hacer". Estructura "para-hacer". Iteraciones anidadas. Condiciones que deben cumplirse. Tipos de datos Simples y Compuestos. Tipos de datos primitivos. Tipos de datos definidos por el usuario. Clasificación de las estructuras de datos. Registros. Funciones predefinidas en las bibliotecas. Tipos ordinales. Ejercitación práctica de los temas de la unidad.

UNIDAD III: Arreglos y Matrices

Arreglos unidimensionales o vectores. Definición y generación de arreglos. Operaciones con arreglos de una dimensión. Arreglos de tipos de datos definidos por el usuario y registros. Algoritmos de búsqueda sobre un arreglo. Algoritmo de Ordenamiento sobre un arreglo. Ordenación por índice.

Matrices o Arreglos bidimensionales Definición y generación de matrices. Operaciones con matrices, recorrido por filas y columnas, propiedades de las matrices de $N \times N$, diagonal principal y contra diagonal. Matrices de tipos de datos definidos por el usuario y registros.

Tratamiento de información estructurada en vectores y matrices. Asignación y manejo de memoria en ejecución en arreglos y para matrices.

UNIDAD IV: Modularización, Procedimientos y funciones

Descomposición de problemas en problemas menores. Utilidad e importancia de la modularización y noción de reusabilidad. Subprogramas o módulos. Conceptos de argumentos y parámetros. Procedimientos y funciones con parámetros. Prototipo, implementación y llamadas. Parámetros

[Handwritten signature]



formales por valor y parámetros por referencia. Desarrollo de algoritmos y programas en forma modular. Utilización de las funciones predefinidas o estándar de los lenguajes. Funciones escalares, funciones que no retornan un valor llamadas procedimientos. Anidamiento de funciones y/o procedimientos. Principio de integridad referencial. Funciones o procedimientos reutilizables implementadas en bibliotecas implementadas por el usuario. Ejercitación práctica de los temas de la unidad.

7. Bibliografía obligatoria y complementaria (organizada por unidades)

Kernighan, B. W. & Ritchie, D. M. (1991). El lenguaje de programación C. (2da Edición). Mexico: Person Educación.

Aguilar, L. J. & Zahonero Martinez, I. (2004). Algoritmos y estructuras de datos: una perspectiva en C. Madrid: McGrawHil Interamericana de España.

De Giusti, Madoz, Bertone, Naiouf, Lanzarini, Gorga, Russo –2001- "Algoritmos, Datos y Programas. Conceptos Básicos" - Prentice-Hall.

Joyanes Aguilar, L. – 1996 - 2003 (3ra. Edición) - "Fundamentos de Programación. Algoritmos y Estructuras de Datos" - McGraw-Hill/Interamericana de España

Aguilar, L. J. (1993) Programación en turbo Pascal Versiones 5.5, 6.0 y 7.0 (2da. Edición). Madrid: McGraw-Hill/Interamericana de España S.A.

Deitel, H. M. & Deitel, P. J. (2012). Como Programar en C (9na Edición). Madrid: Pearson Educación.

Wirth. –1987- "Algoritmos y Estructuras de Datos". Prentice-Hall.

8. Metodología de trabajo

La asignatura está dividida en 4 unidades, las cuales se dictarán durante un cuatrimestre en clases teóricas y prácticas, con resolución de problemas a cargo de los alumnos.

Las guías de trabajos prácticos propuestos por el docente, se resolverán en forma individual. Se utilizarán las herramientas informáticas adecuadas para la resolución de los problemas y la justificación de las respuestas obtenidas.

Se realizarán prácticas en laboratorio de computadoras relacionadas a la unidad temática de la asignatura.



Guía de ejercicios: En sincronía con los temas que se vayan dando en cada unidad, existe la guía de ejercicios que ayuda a consolidar los conocimientos vistos en la clase teórica. La intención es que el alumno desarrolle ejercicios cortos y enfocados en cada tema específico, según dicte la unidad que se esté viendo.

Trabajos Prácticos

Las prácticas de la materia se abordará a través de la resolución de ejercicios obligatorios de la guía de Ejercicios

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

El régimen de aprobación podrá ser por:

- promoción directa;
- evaluación integradora;
- examen final.

En el caso de la promoción directa se requiere aprobar dos instancias parciales o sus respectivos recuperatorios con un promedio de 7 (siete) puntos, siendo 6 (seis) la nota mínima.

Para el examen integrador se requiere aprobar dos instancias parciales o sus respectivos recuperatorios con una nota no inferior a 4 (cuatro) puntos. Es una instancia que se desarrolla luego de finalizada la cursada, no requiere inscripción previa y es llevada adelante por el o la profesor/a de la comisión, quien indica a cada estudiante los contenidos a evaluar y su modalidad (escrito, oral, trabajo práctico, etc.). El examen integrador se aprueba con 4 (cuatro) puntos.

Las/os estudiantes que no aprueben por promoción o por examen integrador tendrán la posibilidad del examen final.

Para acceder a la instancia de examen final, se requiere obtener no menos de 4 (cuatro) puntos en cada instancia parcial o sus respectivos recuperatorios. Luego, realizar previamente la inscripción a rendir en las fechas estipuladas en el calendario académico.

Se sugiere leer para más información los artículos 31 a 39 del Régimen General de Estudios.

10. Instancias de práctica (opcional)



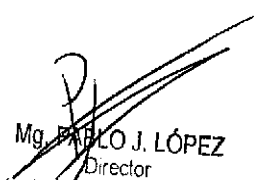
11. Cronograma de actividades teóricas y prácticas	
Semana 1	Hardware y software. Arquitectura del hardware. Diagrama de Von Neumann. Unidad central de proceso. Memoria principal. Memorias auxiliares. Periféricos. Concepto de algoritmo.
Semana 2	Distintas técnicas de representación de algoritmos. Diagramas Estructurados. Pseudocódigo. Etapas en la construcción y prueba de algoritmos. Resolución de problemas simples en pseudocódigo. Propiedades de los Algoritmos. Representación de los algoritmos
Semana 3	Características esenciales. Estructura de un programa. Funciones. Concepto de Programa. Función main. Formatos. Operadores, Palabras Claves. Compilador, Preprocesador
Semana 4	Datos. Variables. Alcance de las variables, globales y locales. Constantes. Identificadores de variables. Datos escalares o simples. Tipos de Datos. Declaraciones.
Semana 5	Algoritmos lineales. Ruptura de la secuencialidad. Enunciación de las estructuras de control. Estructuras de decisión. Decisión simple. Decisiones anidadas. Decisión múltiple.
Semana 6	Estructuras iterativas. Estructura "repetir-mientras". Estructura "mientras-hacer". Estructura "para-hacer". Iteraciones anidadas. Condiciones que deben cumplirse
Semana 7	Resolución práctica y generalizada de los ejercicios obligatorios de la guía de trabajos Prácticos. Práctica para el primer parcial de la materia. Repaso de teoría .
Semana 8	Primer Parcial - Resolución del primer Parcial. Práctica de ejercicios complementarios
Semana 9	Arreglos unidimensionales o vectores. Definición por tipo, Inicialización, Carga de valores por posición. Recuperación de Valores por posición. Recorrido y Longitud.
Semana 10	Recuperatorio del Primer Parcial. Rezagados del primer parcial. Repaso de Arrays unidimensionales.
Semana 11	Arrays de Caracteres, String, Funciones para String. Resolución de ejercicios prácticos de la guía. Propiedad de primer-ultimo elemento, segundo y ante-ultimo elemento, Ejercicio obligatorio de palíndromos.
Semana 12	Recorrido Secuencial en búsqueda de Máximos, Mínimos, Promedio de valores de un Array de Enteros. Operaciones matemáticas con Arrays. Introducción a Arrays bidimensionales o Matrices.
Semana 13	Módulos. Ventajas de la programación modular. Desarrollo de programas de forma modular. Procedimientos. Funciones definidas por el usuario. Diferencias y semejanzas entre procedimientos y funciones. Variables globales y locales.



UNPAZ
Universidad Nacional de José C. Paz

07

	Parámetros formales
Semana 14	Resolución práctica y generalizada de los ejercicios obligatorios de la guía de trabajos Prácticos segunda parte. Práctica para el segundo parcial de la materia.
Semana 15	Segund Parcial - Resolución del Segundo Parcial.
Semana 16	Recuperatorio del Segundo Parcial. Cierre de notas y Firma de Libretas a los alumnos.
<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	 Mg. PABLO J. LÓPEZ Director Depto. de Economía, Prod. e Innov. Tec. Universidad Nacional de José C. Paz



- 107 -

PROGRAMA UNIDAD CURRICULAR			
Departamento	ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA		
Carrera/s	ANALISTA PROGRAMADOR UNIVERSITARIO		
Nombre de la unidad curricular	PARADIGMAS DE PROGRAMACION	Código	619
Docente responsable	GONZALEZ TULIAN GERARDO		
Año de presentación del programa	2019		
1. Carga horaria			
Horas de clase semanales	5		
Horas de clase totales	80	Horas totales clases teóricas	40
		Horas totales clases prácticas	40
		Otras horas totales (laboratorio, trabajo de campo, etc.)	
2. Unidades correlativas precedentes en el Plan de Estudios			
Denominación		Código	
ALGORITMOS Y PROGRAMACION III		610	
TALLER DE INGLES II		611	
3. Contenidos mínimos según Plan de Estudios			
Lenguajes de Programación como herramientas para producir software. Evaluación de los lenguajes de programación a través de las características del software que producen. Principios de diseño de los lenguajes. Evolución histórica de los lenguajes de programación. Lenguajes funcionales. Lenguajes Orientados a objetos. Estructura de un lenguaje: sintaxis y semántica. Sintaxis: Características de las sintaxis. Elementos de las sintaxis. Estructuras sintácticas. Reglas léxicas y sintácticas. Tipos de sintaxis. Tipos de semánticas. Formas de definir la semántica de un lenguaje de programación. Semántica operacional. Entidades y ligaduras. Parámetros. Datos y Subprogramas. Pasaje de Rutinas como parámetros. Sistema de tipos. Tipos predefinidos, tipos definidos por el usuario, tipos estructurados, tipos abstractos. Equivalencias de tipos. Procesamiento de un lenguaje. Interpretación y traducción. Tipos de traductores. Comparación entre Traductor e Intérprete. Compiladores. Etapas de Análisis y Síntesis. Optimización. Unidades. Atributos. Representación en ejecución. Elementos. Unidades			

8



recursivas. Unidades genéricas. Alias y sobrecarga. Procesamiento de un lenguaje. Clasificación. Lenguaje estático. Entidades locales. Rutinas internas. Compilación separada. Lenguajes basados en pila. Unidades recursivas, implementación. Estructura de bloque. Datos semidinámicos y dinámicos. Lenguajes dinámicos. Paradigma imperativo. Paradigma funcional. Características. Comparación lenguaje imperativo con lenguaje funcional. Paradigma Orientado a Objetos. Características. Elementos básicos de la programación orientada a objetos. Paradigma lógico. Características. Elementos de la programación lógica.

4. Fundamentación

La asignatura paradigmas de programación se ubica en el primer cuatrimestre del tercer año según el plan vigente, además tiene como correlativa a la asignatura Programación Concurrente y Paralela.

La asignatura reviste vital importancia para el perfil de egresado, ya que a en los entornos dinámicos y cambiante de las organizaciones los problemas requieren soluciones de sistemas de información que pueden ser de muy variado tipo y con características muy distintas unos de otros. Seleccionar y conocer el estilo de programación adecuado es el primer paso en el camino de la solución del problema. Es indispensable que el alumno comprenda y conozca los distintos enfoques y estilos de programación que se pueden aplicar para la solución de estos problemas. Los paradigmas de programación son justamente estos enfoques y estilos de programación que el alumno necesita conocer y comprender.

5. Objetivos

- Comprender la base teórica y los fundamentos de los paradigmas que son utilizados en los lenguajes de programación.
- Aplicar los diferentes paradigmas en la resolución de problemas.
- Conocer los modelos de cada paradigma como así también la sintaxis y semántica los lenguajes de programación que realizan la implementación de cada paradigma en forma correcta.
- Utilizar herramientas de programación que implementan cada uno de los paradigmas.
- Formar el razonamiento para la resolución de distintos problemas que nos presenta la programación, aplicando en cada uno de ellos la solución más eficaz.
- Resolver problemas lógicos de mediana y alta complejidad.
- Incentivar el espíritu de investigación e incursión en diversos lenguajes de programación.

6. Contenidos (organizados por unidades)

UNIDAD I: Lenguajes y Paradigmas de Programación en General

Lenguajes de Programación como herramientas para producir software. Evaluación de los lenguajes de programación a través de las características del software que producen. Principios de diseño de los

[Handwritten signature]



lenguajes. Evolución histórica de los lenguajes de programación.

Concepto de paradigma de programación. Necesidad de la existencia de diferentes paradigmas de programación. Etapas de Análisis y Síntesis. Optimización y selección del paradigma de programación. Diferencia entre lenguaje y paradigma de programación. Concepto de tipo: representación de los tipos en los diferentes lenguajes de programación. Importancia del concepto de tipo en relación a la implementación de sistemas complejos y cambiantes. Comparación de los diferentes esquemas de chequeo de tipos. Ubicación de los mecanismos de control de flujo en un programa.

Estructura de un lenguaje: sintaxis y semántica. Sintaxis: Características de las sintaxis. Elementos de las sintaxis. Estructuras sintácticas. Reglas léxicas y sintácticas. Tipos de sintaxis. Tipos de semánticas. Formas de definir la semántica de un lenguaje de programación. Semántica operacional. Entidades y ligaduras. Parámetros. Datos y Subprogramas. Pasaje de Rutinas como parámetros. Sistema de tipos. Tipos predefinidos, tipos definidos por el usuario, tipos estructurados, tipos abstractos. Equivalencias de tipos. Procesamiento de un lenguaje. Interpretación y traducción. Tipos de traductores. Comparación entre Traductor e Intérprete. Compiladores

UNIDAD II: Lenguaje y paradigmas orientado a objetos

Características y elementos básicos de la programación orientada a objetos. Concepto de Objeto. Unidades. Atributos. Representación en ejecución. Estructura de bloque. Datos semidinámicos y dinámicos. Concepto de mensaje, estado y comportamiento. Unidades genéricas. Alias y sobrecarga. Elementos Encapsulamiento. Visión de programa entendido como un conjunto de objetos que envían mensajes. Ambientes de objetos: diferencia con la programación tradicional. Los métodos como mecanismo de resolución de mensajes. Concepto de polimorfismo. Concepto de Clase como modelo/molde de objetos. Delegación y responsabilidad. Concepto de referencia. Interfaz e implementación: encapsulamiento del estado interno, ocultamiento de datos. Tipos de mensaje. Herencia. Variables y métodos de clase. Igualdad e identidad. Relaciones entre clases: asociación, composición; relación con delegación. Aplicación del concepto de tipo en el paradigma de objetos. Efecto de lado y declaratividad en el paradigma de objetos. Concepto de orden superior en la programación orientada a objetos. Introducción al manejo de errores.

Lenguaje asociado: Smalltalk. Imagen, ambiente de objetos, definición y uso de clases y objetos. Herramientas de navegación (object browser, class browser, otros). Uso de workspaces. Estudio de algunas clases propias de Smalltalk: String, Integer, Date, otras. Estudio del protocolo de Colecciones. Bloques. Garbage collection.



UNIDAD III: Paradigma Lógico

Características y elementos de la programación lógica. Fundamentación lógica. Predicados. Razonamientos y silogismos. Relaciones, hechos y reglas. Consultas. Tipos de consultas. Definición de programa en Paradigma Lógico. Motor de inferencia, ubicación del control en un programa lógico.

Diferencia entre una función y una relación. Concepto de variable o incógnita.

Unificación. Múltiples resultados. Inversibilidad. Aritmética, evaluación de expresiones aritméticas.

Negación. Listas. Pattern Matching. Predicados de orden superior. Functores. Polimorfismo. Unidades recursivas.

Lenguaje asociado: Prolog. Entorno de trabajo, manejo de archivos. Realización de Consultas. Ayuda. Trace y debug. Limitaciones de inversibilidad: generación de valores

UNIDAD IV: Lenguaje y paradigma funcional

Concepto de función. La función como bloque de construcción de programas. Concepto de programa en el paradigma funcional. Efecto de lado. Concepto de variable. Definición de tipo y valor. Definición de funciones. Funciones definidas por ramas. Pattern matching. Inferencia de tipos. Implementación de Unidades recursivas. Prueba por inducción. Entidades locales. Rutinas internas. Compilación separada y diferida. Manejo de listas. Listas por comprensión. Funciones de orden superior. Aplicación parcial de funciones. Evaluación diferida y listas infinitas. Composición de funciones. Sistemas de tipos. Polimorfismo y tipos genéricos. Tuplas. Comparación lenguaje imperativo con lenguaje funcional

Lenguaje asociado: Haskell. Entorno de trabajo, definición de programas, uso del intérprete. Notación bidimensional. Módulos.

7. Bibliografía obligatoria y complementaria (organizada por unidades)

Alonso Amo, F. & Segovia Perez, F. (1995). Entornos y Metodologías de Programación. Madrid: Paraninfo.

Watt, D. (1990). Programming Languages Concepts and Paradigms. New York, NY: Prentice Hall.

Programming Languages Concepts and Paradigms, David Watt, Prentice Hall. 1990

Inside Smalltalk. Vol. I, II: Lalonde y J. Pugh; Prentice Hall International; 1991

Carlos Varela. "Programming Languages". Rennselaer Polytechnic Institute. USA. 2011



Richard Bird. "Introduction to Functional Programming using Haskell". Prentice Hall International, 2nd Ed. New York, 1997

Smalltalk, Objects and Design, Chamond Liu., Prentice Hall., 2000

Prolog, Giannesini, Kanoui, Pasero y Van Caneghem, Addison, Wesley Iberoamericana. 1989

Kent, B. (1995). Smalltalk Best Practice Patterns. Prentice Hall.

Chamond, L. (1996) Smalltalk, Objects and Design. New York, NY: To Excel

Giannesini, F.; Kanouri, R.H.; Pasarero, R. & Van Caneghem, M. (1989). Prolog (Versión traducida). Addison-Wesley Iberoamericana

Labra G. J (1998). Introducción al lenguaje Haskell. Universidad de Oviedo

Sitio oficial de Dolphin Smalltalk:
www.object-arts.com

Sitio oficial de Swi-Prolog:
www.swi-prolog.org

Sitio oficial de Haskell:
www.haskell.org

8. Metodología de trabajo

La asignatura está constituida por 4 Unidades, las cuales se dictarán durante un cuatrimestre en clases teóricas y prácticas, con resolución de problemas a cargo de los alumnos.

Las guías de trabajos prácticos propuestos por el docente, se resolverán en forma individual. Se utilizarán las herramientas informáticas adecuadas para la resolución de los problemas y la justificación de las respuestas obtenidas.

Se realizarán prácticas en laboratorio de computadoras relacionadas a la unidad temática de la asignatura.

Guía de ejercicios: En sincronía con los temas que se vayan dando en cada unidad, existe la guía de ejercicios que ayuda a consolidar los conocimientos vistos en la clase teórica. La intención es que el alumno desarrolle ejercicios cortos y enfocados en cada tema específico, según dicte la unidad que se esté viendo.

7



Trabajos Prácticos

Las prácticas de la materia se abordará a través de la resolución de ejercicios de la guía de Ejercicios

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

El régimen de aprobación podrá ser por:

- promoción directa;
- evaluación integradora;
- examen final.

En el caso de la promoción directa se requiere aprobar dos instancias parciales o sus respectivos recuperatorios con un promedio de 7 (siete) puntos, siendo 6 (seis) la nota mínima.

Para el examen integrador se requiere aprobar dos instancias parciales o sus respectivos recuperatorios con una nota no inferior a 4 (cuatro) puntos. Es una instancia que se desarrolla luego de finalizada la cursada, no requiere inscripción previa y es llevada adelante por el o la profesor/a de la comisión, quien indica a cada estudiante los contenidos a evaluar y su modalidad (escrito, oral, trabajo práctico, etc.). El examen integrador se aprueba con 4 (cuatro) puntos.

Las/os estudiantes que no aprueben por promoción o por examen integrador tendrán la posibilidad del examen final.

Para acceder a la instancia de examen final, se requiere obtener no menos de 4 (cuatro) puntos en cada instancia parcial o sus respectivos recuperatorios. Luego, realizar previamente la inscripción a rendir en las fechas estipuladas en el calendario académico.

Se sugiere leer para más información los artículos 31 a 39 del Régimen General de Estudios.

10. Instancias de práctica (opcional)

11. Cronograma de actividades teóricas y prácticas

Semana 1	Definición de paradigma, de paradigmas de programación. Diferencia entre lenguaje de programación y paradigma de programación. Introducción a los distintos tipos de paradigmas de programación y estilos de programación por cada paradigma. Introducción al paradigma Imperativo o por procedimientos. Conceptos básicos del
----------	---



	paradigma concurrente. Tipos de paradigmas de programación. Imperativo o por procedimientos (continuación clase 1). Declarativo (programación lógica y funcional). Paradigma orientado a objetos.
Semana 2	Presentación de las herramientas a utilizar en cada paradigma. Smalltalk, Haskell, Prolog. Comparación entre los diferentes paradigmas. Características básicas y particulares de cada uno. Introducción al paradigma de objetos. Definición de objetos. Clasificación y clases de objetos. Modelo y representación de los objetos. Comportamiento de los objetos. Mensaje y método. Estado interno.
Semana 3	Apliación de conceptos sobre paradigmas. Recursividad de grado 1 y grado dos. Ejercicio Factorial y torres de Hanoi. Ejercitación variada sobre recursividad. Introducción al manejo del Software prolog, escribir consultas sencillas y armar una base de conocimiento con Hechos y Reglas.
Semana 4	Repaso temas de la clase Nro 4. Clase abstracta y clase concreta. Encapsulamiento. Mensajes a si mismo (self). Relaciones entre clases: asociación, composición; relación con delegación. Polimorfismo. Delegación y responsabilidad. Introducción al manejo de errores. Ambiente y software. Práctica. Instalación de la herramienta del Dolphin/Smalltalk 6x. Introducción al ambiente Dolphin Smalltalk X6. Sintaxis de smalltalk (Un sub conjunto de Smalltalk). Creación de un ejemplo del paradigma de objetos con la herramienta.
Semana 5	Repaso temas teóricos de las clases Nro. 4, 7 y 8.. Imagen, ambiente de objetos, definición y uso de clases y objetos. Herramientas de navegación (object browser, class browser, otros). Uso de workspaces. Estudio de algunas clases propias de Smalltalk: String, Integer, Date, otras. Estudio del protocolo de Colecciones. Bloques. Garbage collection. Práctica en grupo sobre un ejercicio en la herramienta Dolphin Smalltalk 6X. Discusión sobre las distintas soluciones propuestas por los alumnos.
Semana 6	Trabajo con Colecciones de Smalltalk. Revision de métodos size; isEmpty; notEmpty; at, add, addAll; addFirst; addLast; remove; removeFirst; removeLast; removeAll; first; last; asSortedCollection; select, do, reject, detect; anySatisfy; allSatisfy; includes. Clase de ejercicios prácticos. Ejercicio práctico de Taller Mecánico y de Productora de Cine sobre Smalltalk. Ayuda a los alumnos en la codificación de los métodos

12



	directamente sobre las PC.
Semana 7	Repaso general de teoría de paradigmas de objetos. Respuesta a consulta y dudas sobre temas para el parcial. Practica en grupo con ejercicios en la herramienta Dolphin Smalltalk 6X. Discusión sobre las distintas soluciones propuestas por los alumnos. Practica para el primer parcial.
Semana 8	1° Parcial teórico y práctico sobre los temas vistos de paradigmas de objetos. Comprende las Unidad I y II de los contenidos de la materia. Resolución práctica del 1er. Parcial
Semana 9	Principales características del paradigma lógico. Tipos de consultas. Definición de programa en paradigma lógico. Motor de inferencia, ubicación del control en un programa lógico. Entrega de notas 1° parcial. Ejercitación sobre prolog. Ejercicios en grupo. Discusión sobre las distintas soluciones propuestas por los alumnos
Semana 10	Recuperatorio del 1er. Parcial Ejercitación sobre prolog. Ejercicios en grupo. Discusión sobre las distintas soluciones propuestas por los alumnos
Semana 11	Entrega Notas Recuperatorio. Unificación. Múltiples resultados. Negación. Listas. Pattern Matching. Predicados de orden superior. Functores. Polimorfismo. Ejercitación sobre prolog. Ejercicios en Grupo. Discusión sobre las distintas soluciones propuestas por los alumnos.
Semana 12	Concepto de función. Efecto de lado. Concepto de variable. Definición de tipo y valor. Funciones definidas por ramas. Pattern Matching. Inferencia de tipos. Funciones recursiva. Diferencia entre una función y una relación. Concepto de variable o incógnita. Resolución de distintos tipos de ejercicios sobre la herramienta Prolog. Discusión general con los alumnos sobre los temas tratados. Presentación de Haskell. Ejercicios simples en grupo. Discusión sobre las distintas soluciones propuestas por los alumnos.
Semana 13	Tupas, Evaluación diferida y listas infinitas. Composición de funciones. Sistemas de tipos. Prueba por inducción. Manejo de listas. Resolución de ejercicios varios en Haskell. Trabajo en grupo en la resolución de problemas. Repaso general de los paradigmas lógico y funcional de las unidades III y IV respectivamente. Trabajo en grupo con ejercicios de práctica para el parcial. Respuesta a dudas y consultas
Semana 14	Parcial teórico y práctico sobre los temas vistos de los paradigmas lógico y Funcional.

2



	Unidades III, IV. Resolución práctica de 2do Parcial.
Semana 15	Recuperatorio del 2° parcial teórico y práctico sobre los temas vistos de paradigmas lógico y funcional.
Semana 16	Comprender la teoría de las Unidad III y IV de los contenidos de la materia. Entrega de notas del segundo recuperatorio de las unidades III y IV. Firma de libretas faltantes. Consulta para el final de la materia. 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	



UNPAZ
Universidad Nacional de José C. Paz

-07

Firma Departamento
Académico

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