



05

JOSÉ C. PAZ, 05 JUL 2021

**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 Resolución del citado CONSEJO N° 01 del 26 de junio de 2020, la Disposición N° 07/2019 del citado CONSEJO, el Expediente Nro. 782/2019 del Registro de esta UNIVERSIDAD NACIONAL DE JOSÉ CLEMENTE PAZ, y

**CONSIDERANDO:**

Que por Disposición del CONSEJO DEPARTAMENTAL DE ECONOMÍA, PRODUCCIÓN E INNOVACIÓN TECNOLÓGICA N° 07 del 29 de noviembre de 2019, se aprobó, entre otros, el programa de la asignatura *Paradigmas de Programación (cód.619)* perteneciente a la Carrera Analista Programador Universitario

Que por un error involuntario, se le asignó un código erróneo a la esta asignatura, ya que dónde debió decir *Paradigmas de Programación (cód.619)* se leyó *Paradigmas de Programación (cód.610)*.



Que es necesario rectificar el mencionado acto administrativo, en cuanto a esta asignatura, mediante una nueva disposición departamental para subsanar dicho error material.

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 puesta tal rectificación a consideración del Consejo DEPARTAMENTAL en la Sesión N° 24, de carácter ordinaria, registrada en el Acta N° 24 del 28 de mayo de 2021, este Cuerpo Colegiado compartió los términos y contenidos del referido instrumento, por lo que resulta necesario rectificar la Disposición N° 07 de fecha 29 de noviembre de 2019 en cuanto a este programa se refiere.

Que la presente medida se adopta en ejercicio de las atribuciones conferidas por los artículos 77, inciso f), del Estatuto de la UNIVERSIDAD, y 1°, inciso d) y 7°, 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 1°.- Rectifíquese la Disposición N° 07 de fecha 29 de noviembre de 2019**



mediante la presente Disposición, en cuanto lo establecido para el programa de la asignatura *Paradigmas de Programación (cód.619)*

ARTÍCULO 2º.- Apruébase el programa de la carrera de Analista Programador Universitario que se adjunta como Anexo a la presente, correspondiente a la siguiente asignatura ut supra mencionada.

ARTÍCULO 3º.- Establécese que los programas aprobados precedentemente, tendrán DOS (2) años de vigencia, contados a partir del semestre siguiente al de su aprobación.

ARTÍCULO 4º.- Regístrese, comuníquese, publíquese en el Boletín Oficial de la UNIVERSIDAD NACIONAL DE JOSÉ CLEMENTE PAZ y cumplido, archívese.



05

UNPAZ

M

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		
<b>1. Carga horaria</b>			
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.)	
<b>2. Unidades correlativas precedentes en el Plan de Estudios</b>			
Denominación			Código
ALGORITMOS Y PROGRAMACION III			610
TALLER DE INGLES II			
<b>3. Contenidos mínimos según Plan de Estudios</b>			
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 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 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.

**UNIDAD I: LENGUAJE DE PROGRAMACIÓN. CONCEPTO DE TIPO. REPRESENTACIÓN DE TIPOS**

Diferencia entre lenguaje y paradigma de programación. 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. Premio Nobel de Medicina Dr. César Milstein

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](http://www.object-arts.com)

Sitio oficial de Swi-Prolog:  
[www.swi-prolog.org](http://www.swi-prolog.org)

Sitio oficial de Haskell:  
[www.haskell.org](http://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.

#### 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; "2021 | AÑO DE HOMENAJE AL PREMIO NOBEL DE MEDICINA DR. CÉSAR MILSTEIN"
- 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 Haoin. 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 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	<p>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.</p> <p>Ejercitación sobre prolog. Ejercicios en grupo. Discusión sobre las distintas soluciones propuestas por los alumnos</p>
Semana 10	<p>Recuperatorio del 1er. Parcial</p> <p>Ejercitación sobre prolog. Ejercicios en grupo. Discusión sobre las distintas soluciones propuestas por los alumnos</p>
Semana 11	<p>Entrega Notas Recuperatorio. Unificación. Múltiples resultados. Negación. Listas. Pattern Matching. Predicados de orden superior. Functores. Polimorfismo.</p> <p>Ejercitación sobre prolog. Ejercicios en Grupo. Discusión sobre las distintas soluciones propuestas por los alumnos.</p>
Semana 12	<p>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.</p> <p>Presentación de Haskell. Ejercicios simples en grupo. Discusión sobre las distintas soluciones propuestas por los alumnos.</p>
Semana 13	<p>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.</p> <p>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</p>
Semana 14	<p>Parcial teórico y práctico sobre los temas vistos de los paradigmas lógico y Funcional. Unidades III, IV.</p> <p>Resolución práctica de 2do Parcial.</p>
Semana 15	<p>Recuperatorio del 2° parcial teórico y práctico sobre los temas vistos de paradigmas lógico y funcional.</p>
Semana 16	<p>Comprender la teoría de las Unidad III y IV de los contenidos de la materia.</p> <p>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</p>
<p><i>A partir de aquí a completar únicamente para unidades curriculares con régimen anual</i></p>	
Semana 17	
Semana 18	



Semana 19	"2021   AÑO DE HOMENAJE AL PREMIO NOBEL DE MEDICINA DR. CÉSAR MILSTEIN"
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	

<b>Firma docente responsable</b>	
<b>Firma Departamento Académico</b>	