Ministerio de Cultura y Educación **(Programa del año 2022)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION III	ING. INFORM.	026/12-08/15	2022	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
NECCO, CLAUDIA MONICA	Prof. Responsable	P.Asoc Exc	40 Hs
SANCHEZ, ALEJANDRO	Prof. Colaborador	P.Adj Exc	40 Hs
ESCUDERO, LEONARDO RUBEN	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
2 Hs	Hs	2 Hs	2 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
08/08/2022	18/11/2022	15	90

IV - Fundamentación

Si algo caracteriza al panorama actual del desarrollo de sistemas es su complejidad y heterogeneidad. No existe una sola forma de pensar y encarar las soluciones, no son uniformes los conceptos que fundamentan los lenguajes, no es única la manera de programar. Los modelos de datos, estructuras de control, mecanismos de evaluación, sentencias, enlaces, expresiones, declaraciones y tantos otros elementos que conforman los lenguajes de programación actuales son muy diferentes entre unos y otros, hasta opuestos, pero es posible detectar cuáles son los conceptos que marcan diferencias mayores o menores, muestran puntos de contacto o de inflexión, establecen criterios de clasificación, y finalmente, aportan elementos teóricos y prácticos para optimizar el desarrollo de sistemas concretos.
Desde la ingeniería en Sistemas de Información, los lenguajes pueden ser considerados como herramientas para formular, representar y resolver problemas. Desde esta perspectiva el paradigma que sustenta un lenguaje y la programación en el mismo, estaría dictando una forma de modelar o ver un problema, por lo tanto el desarrollo de código en diferentes paradigmas, además de revelar las características de los mismos, permite analizar distintos enfoques con que se puede representar un problema en términos de programación. Esta actividad de desarrollo y análisis debería conducir a la elección del paradigma mas adecuado para resolver cada problema o por lo menos, si no existe la posibilidad de elección, a una mejor utilización de los recursos del paradigma que se debe utilizar.
Otra aplicación del conocimiento de los recursos que ofrece cada paradigma, es el desarrollo de sistemas utilizando los

denominados lenguajes multiparadigma, que integran y/o soportan características de más de un modelo de desarrollo. Finalmente, otro componente importante a la hora de programar es el entorno de programación. Nadie discute la relevancia de internet en el mundo actual. En este contexto, un entorno de programacion web hace referencia a un ambiente de desarrollo y/o ejecución programas o servicios en el marco de la web en general. Existen herramientas, programas, lenguajes de programación y desarrollo que son específicos para el diseño de aplicaciones dentro de un entorno web que deberían ser sujetos de estudio del ingeniero en Informatica.

V - Objetivos / Resultados de Aprendizaje

Utilizar correctamente las singularidades del Paradigma de Programación Lógica y desarrollar aplicaciones de hasta mediana complejidad en dicho enfoque. Aprovechar razonablemente las posibilidades del Paradigma de Programación Funcional y construir programas de mediano nivel de dificultades en este contexto. Reconocer las oportunidades de utilización de la Programación Multiparadigma utilizando correctamente los recursos asociados a cada paradigma.
Desarrollar aplicaciones en entornos Web, utilizando herramientas y lenguajes de programación y desarrollo que específicos para para dicho entorno o adaptando soluciones provistas por diferentes modelos de desarrollo.

VI - Contenidos

Unidad I: Paradigma de Programación Funcional 1
Paradigma de Programación Funcional. Descripción del modelo de programación funcional. Características principales de los lenguajes funcionales: transparencia referencial, funciones de orden superior y currificación, sistemas de tipos. Reducción y evaluación en programas funcionales. Ordenes de reducción: reducción aplicativa y reducción normal.

Unidad II: Paradigma de Programación Funcional 2
Introducción a Haskell. Expresiones. Declaraciones. Estructuras de datos. Tipos simples y tipos numéricos. Tipos funcionales. Definición de funciones. Implementación de soluciones de problemas en lenguaje Haskell en el laboratorio. Listas por comprensión. Definición y ejemplos. Listas como tipo inductivo. Funciones básicas sobre listas. Funciones de orden superior sobre listas: map, filter, reduce, traversals, etc. Patrones de recursión: foldr y foldl. Polimorfismo. Tipos de Polimorfismo. Definición de Tipos de datos abstractos. Mecanismos de definición de tipos y funciones sobre ellos. Tipos algebraicos polimorficos.

Unidad III: Paradigma de Programación Lógica 1
Introducción. Variables, constantes y estructuras: Desarrollo de ejemplos. Átomos, literales y hechos: Ejemplos y ejercicios en clase. Reglas. Cláusulas y predicados: Ejemplos y ejercicios en clase.
Programas, Preguntas y Ejecución: Desarrollo de ejemplos. Semántica operacional y árboles lógicos: Ejemplos y ejercicios en clase. Unificación: Concepto, ejemplos y ejercicios en clase.

Unidad IV: Paradigma de Programación Lógica 2
Ejemplos de notación estándar en Prolog: Ejercicios en clase. Uso del interfaz de Prolog: Desarrollo mediante ejemplos en laboratorio. Aritmética: Ejemplos y ejercicios en clase. Factorial. Estructuras internas: functor y arg. Ejemplos de su utilización. Listas explotando Prolog: Desarrollo de ejercicios en laboratorio. Uso del 'cut'. Metapredicados: Ejemplos y ejercicios.

Unidad V: Programación Multiparadigma 1
Concepto de Programación Multiparadigma: Ejemplos. Características y propiedades. Biblioteca estándar de lenguajes multiparadigma. Tipos de datos, Cadenas de texto. Tuplas y listas. Conjuntos. Controles de flujo. Funciones. Clases. Módulos. Archivos.

Unidad VI: Programación Multiparadigma 2
Análisis y comparación de lenguajes multiparadigma actuales (Python, Swift, Scheme, Ruby, Javascript, etc.)

Unidad VII: Programación en entorno Web 1
Programación en entorno Web: Programación en la Web con lenguajes del tipo PHP. Uso de lenguajes del tipo PHP y Programación Orientada a Objetos. AJAX: Asynchronous JavaScript and XML. Lenguajes del tipo PHP y Web Services.

Unidad VIII: Programación en entorno Web 2

Introducción a la programación en Internet con lenguajes multiparadigmas. Network Scripting en Python. Socket. Manejo de multiples clientes. Script del lado del cliente. Protocolos. Transferencia de Archivos. Distribución de información. Envío de mensajes básicos. Procesamiento de correo electrónico. Acceso a grupos de noticias. Acceso a sitios Web. Scripts del lado del Servidor. CGI Script. Ejecución de scripts del lado del servidor. Utilidades.

VII - Plan de Trabajos Prácticos

Práctico 1: Programación funcional en lenguaje Haskell.
Ejercicios prácticos. Laboratorio a ser presentado.

Práctico 2: Programación Lógica en lenguaje Prolog.
Ejercicios prácticos. Laboratorio a ser presentado.

Práctico 3: Programación en la Web.
Ejercicios prácticos. Laboratorio a ser presentado.

Práctico 3: Programación multiparadigma lenguaje a elección (Python, Swift, Scheme, Ruby, Javascript, etc.).
Elaboración de informe escrito y video sobre un lenguaje a elección.

VIII - Regimen de Aprobación

CONDICIONES PARA DICTADO REGULAR

- Para regularizar la asignatura:

El alumno debe aprobar tres parciales o sus recuperaciones, y presentar en tiempo y forma los laboratorios, el informe y el video solicitados por la cátedra. Se tomarán dos recuperaciones por cada evaluación tal como menciona la OCS 32/14.

El alumno debe presentar en tiempo y forma las evaluaciones On-Line solicitadas por la cátedra en el aula Virtual de la materia.

El alumno debe participar en tiempo y forma en los foros de discusión abiertos por la cátedra en el aula Virtual de la materia.

El alumno debe asistir al 80% de las clases correspondientes al crédito horario.

- Para promocionar la asignatura:

El alumno debe cumplir con las condiciones de regularización con un nivel superior o igual al 70% del total y aprobar la sección especial de integración de cada parcial o su recuperación.
Se tomarán dos recuperaciones por cada evaluación tal como menciona la OCS 32/14.

EXAMEN REGULAR

Se tomará un exámen escrito u oral en las mesas habilitadas del calendario académico

EXAMEN LIBRE:

Según lo dispuesto por Art. 27 de Ord. 13/03 CS.

IX - Bibliografía Básica

[1] Sterling, L., Shapiro, E., “The Art of Prolog", MIT Press, 1994 (2a edición).
[2] Apt, K, ”From Logic Programming to Prolog", Prentice--Hall, 1997.
[3] Bratko, I., ”Prolog Programming for Artificial Intelligence", Addison-Wesley, 1990.
[4] Escrig, T. 2015. El Lenguaje de Programación PROLOG.S. (En línea).Formato PDF. Disponible en: http://mural.uv.es/mijuanlo/PracticasPROLOG.pdf
[5] Guía de Estudio del Lenguaje Haskell. apunte proporcionado por la Cátedra. Disponible en Aula Virtual de la materia.
[6] Thompson, S., “Haskell-The Craft of Functional Programming, 3/E”, ISBN-10: 0201882957 • ISBN-13: 9780201882957. ©2012 • Addison-Wesley
[7] Learn You a Haskell for Great Good!A Beginner's Guide by Miran Lipovača. April 2011, 400 pp. ISBN-13: 978-1-59327-283-8.Versión On Line: http://learnyouahaskell.com/

[8] Programación Funcional. Jeroen Fokker.Universidad de Utrecht, 90 páginas. 1996. Versión On-Line: https://openlibra.com/es/book/programacion-funcional-ed-fokker
[9] PHP Reference Book Blog http://www.phpreferencebook.com/
[10] Larry E. Ullman, PHP for the World Wide Web, Third Edition, Peachpit Press; 3 edition, 2008

X - Bibliografia Complementaria

[1] Nixon, Robin, “Learning PHP, MySQL, and JavaScript: A Step-By Step Guide to Creating Dynamic Websites (Animal Guide)”, Barnes & Noble, 2009
[2] David Powers, PHP Solutions: Dynamic Web Design Made Easy friends of ED; 3rd edition 2010
[3] Practical PHP Programming http://www.tuxradar.com/practicalphp

XI - Resumen de Objetivos

Al finalizar el curso se espera que el alumno sea capaz de:
Desarrollar aplicaciones en el Paradigma de Programación Lógica de programas de mediana complejidad. Aprovechar razonablemente las posibilidades del Paradigma de Programación Funcional y construir programas de mediano nivel de dificultades en este contexto. Reconocer las oportunidades de utilización de la Programación Multiparadigma utilizando correctamente sus dificultades. Desarrollar aplicaciones en entorno Web distinguiendo y aprovechando correctamente los conceptos y herramientas asociados.

XII - Resumen del Programa

Paradigma de Programación Funcional.
Paradigma de Programación Lógica.
Programación en entorno Web.
Programación Multiparadigma.

XIII - Imprevistos

Mail Cátedra: progunslcmn@gmail.com
Mail Prof. Responsable: cnecco@gmail.com

XIV - Otros