



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

"2022 – LAS MALVINAS SON ARGENTINAS"

SAN LUIS, **07 MAR 2022**

VISTO:

El EXP-USL: 0000958/2022, mediante el cual se solicita la protocolización del Curso optativo "Análisis Comparativo de Paradigmas de Programación", correspondiente a la Carrera de Posgrado: "Especialización en Ingeniería de Software" (OCD-3-003/07); y

CONSIDERANDO:

Que el mencionado Curso tiene un crédito total de 80 horas y corresponde al Plan de Estudios de la Especialización.

Que el Profesor Responsable es el Dr. Mario Marcelo BERON y como colaboradores la Dra. Lorena Soledad BAIGORRIA FERNANDEZ y el Dr. Manuel PEREZ COTA.

Que el Comité Académico aconseja su aprobación, tal como consta a fs. 4 del presente expediente.

Que corresponde su protocolización.

Por ello y en uso de sus atribuciones

LA DECANA DE LA FACULTAD DE CIENCIAS
FISICO MATEMATICAS Y NATURALES

RESUELVE:

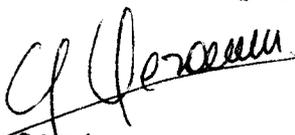
ARTICULO 1º.- Aprobar el programa del Curso: "Análisis Comparativo de Paradigmas de Programación" correspondiente a la Carrera de Posgrado "Especialización en Ingeniería de Software" (OCD-3-003/07) según Anexo único.-

ARTICULO 2º.- Designar como Profesor Responsable al Dr. Mario Marcelo BERON (D.N.I N° 22.140.238) y como colaboradores al Dra. Lorena Soledad BAIGORRIA FERNANDEZ (D.N.I N° 26.668.365) y al Dr. Manuel PEREZ COTA (D.N.I N° 35.281.375).-

ARTICULO 3º.- Comuníquese, notifíquese, publíquese, insértese en el Libro de Resoluciones y archívese.-

RESOLUCION D N° **073 22**

ndc


Dra. G. VERÓNICA GIL COSTA
Secretaria de Ciencia y Técnica
FCFMN - UNSL


Dra. A. Marcela Printista
Decana
FCFMN - UNSL



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

“2022 – LAS MALVINAS SON ARGENTINAS”

1
ANEXO

FORMULARIO DE PRESENTACIÓN DE UN CURSO DE POSGRADO

IDENTIFICACIÓN DEL CURSO

FACULTAD: Ciencias Físico Matemáticas y Naturales

DENOMINACIÓN DEL CURSO: “Análisis Comparativo de Paradigmas de Programación”

CATEGORIZACIÓN DEL CURSO: Perfeccionamiento

FECHA DE DICTADO: 2do año, 1er cuatrimestre

MODALIDAD DE DICTADO (Presencial o a Distancia): Presencial

CRÉDITO HORARIO TOTAL: 80 hs

Horas teóricas: 30 hs

Horas de prácticas de Aula (si correspondiese): 15 hs

Horas de prácticas de Laboratorio (si correspondiese): 20hs

Horas de prácticas de Campo (si correspondiese): 15 hs

APELLIDO Y NOMBRE DEL COORDINADOR: Mario Peralta

Documento de Identidad: 20137343

e-mail: mperalta@unsl.edu.ar Teléfono:4520300 Int 2101

EQUIPO DOCENTE

APELLIDO Y NOMBRE DEL RESPONSABLE: Berón, Mario Marcelo

Título/s: Doctor en Ciencias de la Computación

Documento de Identidad: 22140238

Institución laboral actual: Universidad Nacional de San Luis

e-mail: mberon@unsl.edu.ar Teléfono: 2664202249

Categoría en el Programa de Incentivos: II

APELLIDO Y NOMBRE DEL COLABORADOR: Baigorria Fernández, Lorena

Título/s: Magister en Ingeniería de Software

Documento de Identidad: 26668365

Correponde Anexo RD N° **073 22**

Dra. A. Marcela Printista
Decana
FCFMYN - UNSL

Dra. G. VERÓNICA GIL COSTA
Secretaría de Ciencia y Técnica
FCFMYN - UNSL



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

“2022 – LAS MALVINAS SON ARGENTINAS”

2

Institución laboral actual: Universidad Nacional de San Luis
e-mail: flbaigor@unsl.edu.ar Teléfono: 4520300 int 2101
Categoría en el Programa de Incentivos: III

APELLIDO Y NOMBRE DEL COLABORADOR: Perez Cota, Manuel

Título/s: Doctor Ingeniero Industrial
Documento de Identidad: Pass 35281375L
Institución laboral actual: Universidad de Vigo
e-mail: . mpc.unsl@gmail.com Teléfono: 986 471 495
Categoría en el Programa de Incentivos: no corresponde

PROGRAMA ANALÍTICO

FUNDAMENTACIÓN:

Los problemas que son resueltos con la Ingeniería de Software pueden ser abordados desde diferentes puntos de vista. No obstante, la elección del enfoque adecuado facilita la elaboración de la solución. En la actualidad existen diferentes paradigmas de programación que proporcionan distintas formas de concebir la tarea de programar y diversas metodologías de desarrollo que simplifican la elaboración de soluciones a problemas en dominios de aplicación específicos. Entre los paradigmas de programación reinantes se pueden mencionar el paradigma imperativo en el cual se encuentran inmersos el paradigma imperativo y el orientado a objetos y el paradigma declarativo en donde se destacan el paradigma funcional y el lógico. Como fue mencionado previamente, cada paradigma tiene su dominio de aplicación, por ejemplo el paradigma orientado a objetos es ampliamente utilizado para el desarrollo de aplicaciones empresariales en general y el paradigma funcional o lógico facilita la construcción de sistemas basados en conocimiento como así también, en el caso del paradigma funcional, la prueba de correctitud de los programas lo cual lo hace muy útil para el desarrollo de aplicaciones que requieren medidas de seguridad precisas y extremas. Cada paradigma tiene sus ventajas y desventajas las cuales, si son bien conocidas, permitirán que el ingeniero de software utilice el paradigma más apropiado para resolver una problemática determinada. Por esta razón, en este curso se brindará un enfoque amplio de los paradigmas de programación que dista sustancialmente de la mera presentación y descripción de un lenguaje representativo del paradigma. En este curso se abordarán los fundamentos de los paradigmas de programación se mostrará la

Correponde Anexo RD N° **073 22**


Dra. A. Marcela Primitista
Decana
FCFMyN - UNSL


Dra. G. VERÓNICA GIL COSTA
Secretaria de Ciencia y Técnica
FCFMyN - UNSL



materialización de los conceptos en los lenguajes tradicionales de cada paradigma y a partir de esa tarea se analizarán los lenguajes que se usan en la actualidad para el desarrollo de aplicaciones.

OBJETIVOS:

Posibilitar que los estudiantes de la Especialización en Ingeniería de Software comprendan y apliquen los conceptos fundamentales de los paradigmas de programación imperativo, funcional, lógico y orientado a objetos.

- Lograr que los estudiantes de la Especialización en Ingeniería de Software manejen concienzudamente lenguajes de programación representativos de los paradigmas de programación imperativo, orientado a objetos, funcional y lógico.
- Utilizar tecnologías de punta asociadas con los lenguajes de programación representativos de los paradigmas de programación imperativo, orientado a objetos, funcional y lógico.

CONTENIDOS MÍNIMOS:

Paradigmas de Programación. Paradigma Imperativo. Características del Paradigma Imperativo. Ventajas y Desventajas del Paradigma Imperativo. Programación Procedural. Lenguajes Procedurales. Programación Orientada a Objetos. Lenguajes Orientados a Objetos Paradigma Declarativo. Características del Paradigma Declarativo. Ventajas y Desventajas del Paradigma Declarativo. Programación Funcional. Lenguajes Funcionales. Programación Lógica. Lenguajes Lógicos. Aspectos innovadores y relevantes de la programación funcional y de la programación lógica. Desarrollo de casos comparando fortalezas y debilidades de los diversos paradigmas en espacios de problema diversos.

PROGRAMA DETALLADO:

Unidades del Curso

Unidad Nro. 1: Paradigmas de Programación

Lenguajes de Programación y Paradigmas. Línea del Tiempo de los Lenguajes de Programación. Historia de los Lenguajes de Programación. Generaciones de los Lenguajes

Correponde Anexo RD N° **073 22**


Dra. A. Marcela Prindista
Decana
FCFMN - UNSL


Dra. G. VERÓNICA GIL COSTA
Secretaria de Ciencia y Técnica
FCFMN - UNSL



de Programación: Primera Generación, Segunda Generación, Tercera Generación, Cuarta Generación. Características Comunes y Principales Diferencias de los Lenguajes de Programación. Estilos de Lenguajes: Procedurales, Funcionales, Lógicos, Orientados a Objetos. Paradigma de Programación. Paradigma Imperativo: Procedural, Orientado a Objeto. Paradigma Declarativo: Funcional y Lógico. Abstracciones. Mecanismos de Abstracciones. Metodologías de Programación. Programación Multiparadigma.

Unidad Nro 2: Paradigma de Programación Imperativo - Programación Procedural

Paradigma de Programación Imperativo. Comandos. Datos. Modelo de Programación. Procedimientos. Funciones. Beneficios de la Programación Procedural. Características de la Programación Procedural. Desventajas de la Programación Procedural. Estudio detallado del soporte Python Imperativo.

Unidad Nro. 3: Paradigma de Programación Imperativo - Programación Orientado a Objetos

Tipo de Dato Abstracto. La Estructura Estática: Clases. La Estructura de Tiempo de Ejecución: Objetos. Administración de la Memoria. Genericidad. Herencia. Polimorfismo. Estudio detallado del lenguaje de programación Python y su soporte orientado a objetos. Estudio detallado de Javascript.

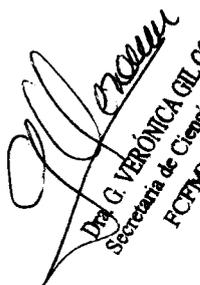
Unidad Nro. 4: Paradigma de Programación Declarativa - Programación Funcional

Computación sin Estado: Expresiones y Funciones, Computación como Reducción. Evaluación: Valores, Sustitución sin Captura, Estrategias de Evaluación, Comparación de Estrategias. Programación en Lenguajes Funcionales: Ambiente Local, Interactividad, Tipos, Concordancia de Patrones, Aspectos Imperativos. Estudio detallado del soporte funcional en los lenguajes Python, Ruby y Groovy.

Unidad Nro. 5: Paradigma de Programación Declarativa - Programación Lógica

Programación Lógica. Deducción como una Computación. Lógica de Primer Orden. Programas Lógicos. Teoría de la Unificación: Variable Lógica, Sustitución, Unificador Más General, Algoritmo de Unificación. Extensiones. Ventajas y Desventajas del Paradigma de Programación Lógica. Estudio detallado de Prolog. Estudio detallado de Prolog y su interfaz con otros lenguajes: Java y C.


Dra. A. Marcela Printista
Decana
FCFMYN - UNSL


Dra. G. VERÓNICA GIL COSTA
Secretaria de Ciencia y Técnica
FCFMYN - UNSL



Prácticos del Curso

Práctico Nro. 1: Resolución de ejercicios básicos con Python Imperativo. Implementación de un sistema pequeño usando las características de Python Imperativo. Tecnología a utilizar: Pycharm.

Práctico Nro. 2: Resolución de ejercicios básicos utilizando el soporte orientado a objetos de python. Implementación de un sistema pequeño utilizando el soporte orientado a objetos de Python. Análisis de ventajas y desventajas respecto del paradigma imperativo. Tecnología a utilizar: Pycharm.

Práctico Nro. 3: Estudio del soporte funcional provisto por el lenguaje de programación Python. Resolución de una guía de ejercicios donde se utilice el soporte funcional de python. Aplicaciones al desarrollo de sistemas en general. Tecnología a utilizar: Pycharm.

Práctico Nro. 4: Consta de dos partes a saber:

Parte A: Resolución de una guía de ejercicios utilizando el lenguaje de programación Prolog. La finalidad de esta actividad es afianzar los conceptos básicos de la programación lógica. Los ejercicios se resolverán con SWI Prolog.

Parte B: Estudio de módulos de Python que proporcionan soporte funcional. Ejemplo: Kanrem. Los ejercicios se resolverán utilizando al IDE Pycharm.

SISTEMA DE EVALUACIÓN (:

Para la aprobación del módulo se requerirá realizar en forma individual:

- 1) Presentación en tiempo y forma de los ejercicios indicados en cada práctica de la materia.
- 2) Evaluación formal integral mediante test de evaluación.

BIBLIOGRAFÍA:

- Francesco Climeri, Kevin Hamlen, Nicola Leone. Practical Aspects of Declarative Languages. 20th International Symposium, PADL 2018, Los Angeles, CA, USA, January 8–9, 2018, Proceedings. Publisher: Springer, Chan. Print ISBN: 978-3-319-73304-3. Online ISBN: 978-3-319-73305-0. Print ISSN: 0302-9743. Online ISSN: 1611-3349. 2018
- Bertrand Meyer. Object-Oriented Software Construction. Second Edition. Prentice Hall. ISBN: 0-13-629155-4. Año de Publicación: 1997

Correponde Anexo RD N° **073 22**

Dra. A. Marceta Printista
Decana
FCFMN - UNSL

Dra. G. VERÓNICA GIL COSTA
Secretaria de Ciencia y Técnica
FCFMN - UNSL



- Elad Shalom. A Review of Programming Paradigms Throughout the History: With a Suggestion Toward a Future Approach. Enero 2018.
- Kenneth Louden; Kenneth Lambert. Programming Languages: Principles and Practices. Editorial: Cengage Learning. Enero 2011.
- Max Bramer. Logic Programming with Prolog. Editorial: Springer; 2nd edición noviembre 2013.
- Hammond, Kevin, and Greg Michaelson, eds. Research directions in parallel functional programming. Springer Science & Business Media, 2012.
- Van-Roy, Peter, and Seif Haridi. Concepts, techniques, and models of computer programming. MIT press, 2004.
- Mertz, David. Functional Programming in Python. O'Reilly. 2015.
- Beazley, David; Jones, Brian. Python Cookbook. O'Reilly. 2013
- Sweigart, Al. AUTOMATE THE BORING STUFF WITH PYTHON 2ND EDITION. No Starch Press. San Francisco. 2020.
- Thompson, Simon. Haskell The Craft of Functional Programming. Addison-Wesley.
- Serrano Mena, Alejandro. Practical Haskell : A Real World Guide to Programming. aPress. 2019.
- Evans, Jeremy. Polished Ruby Programming: Build better software with more intuitive, maintainable, scalable, and high-performance Ruby code. Packt Publishing. 2021.
- Davis, Adam. Learn Groovy. Apress. 2016.
- St.Lauren, Simon. Introducing Erlang: Getting Started in Functional Programming 2nd Edición. O'Reilly Media; 2nd edición. 2017

Complementaria

- Manolescu, Dragos-Anton, Markus Voelter, and James Noble, eds. *Pattern Languages of Program Design 5*. Vol. 5. Addison-Wesley Professional, 2006.
- Bramer, Max. Logic Programming with Prolog. Second Edition. Springer. 2013.
- Sebesta, Robert W. *Concepts of programming languages*. Boston: Pearson, 2012.
- Horowitz, Ellis. *Fundamentals of programming languages*. Springer Science & Business Media, 2012.

CARACTERÍSTICAS DEL CURSO

DESTINATARIOS Y REQUISITOS DE INSCRIPCIÓN: estudiantes inscriptos en la carrera Especialización en Ingeniería de Software que cumplen los requisitos de admisión de la misma.


Dra. A. Marcolle
Decana
FCFMN - UNSL


Dra. G. VERÓNICA GIL COSTA
Secretaría de Ciencia y Técnica
FCFMN - UNSL



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

"2022 – LAS MALVINAS SON ARGENTINAS"

7

CUPO (Número mínimo y máximo de estudiantes): 1 – 25 (por cuestiones de protocolo)

PROCESO DE ADMISIÓN: A cargo del comité académico de la carrera.

LUGAR DE DICTADO: Aula de posgrado del departamento de informática/sala 7 y 8
bloque 2, Universidad Nacional de San Luis

FECHA PREVISTA PARA ELEVAR LA NÓMINA DE ESTUDIANTES APROBADOS:
16 de agosto

Dr. A. Marcela Prindista
Decana
FCFMyN - UNSL

FINANCIAMIENTO DEL CURSO

COSTOS: se utilizan los recursos físicos/tecnológicos del departamento de informática.

FUENTES DE FINANCIAMIENTO: Autofinanciamiento y fondo de la carrera.

ARANCEL GENERAL: \$3500 (tres mil quinientos pesos)

BECA AL DOCENTE DE LA UNSL: 100% del arancel general.

BECA AL ESTUDIANTE DE LA UNSL : de acuerdo a la reglamentación de la carrera

OTRAS BECAS:

Dr. G. VERÓNICA GIL COSTA
Secretaria de Ciencia y Técnica
FCFMyN - UNSL

100 % DEL arancel para personal no docente de la UNSL

50% del arancel para personal de otras universidades con las cuales se tiene convenio.

Correponde Anexo RD N° **073 22**