

| Facultad de Ingeniería - Universidad Nacional de Cuyo                       |                                                   |                            |                        |
|-----------------------------------------------------------------------------|---------------------------------------------------|----------------------------|------------------------|
| P1- PROGRAMA DE ASIGNATURA                                                  |                                                   |                            |                        |
| Adecuación a las modalidades presencial y a distancia por Pandemia COVID-19 |                                                   |                            |                        |
| <b>Asignatura:</b>                                                          | <b>PARADIGMAS DE PROGRAMACIÓN</b>                 |                            |                        |
| <b>Docente Responsable:</b>                                                 | <b>PROFESOR ADJUNTO: DR. PABLO JAVIER VIDAL</b>   |                            |                        |
| <b>Carrera:</b>                                                             | <b>Licenciatura en Ciencias de la Computación</b> |                            |                        |
| <b>Año: 2020</b>                                                            | <b>Semestre: 2DO</b>                              | <b>Horas Semestre: 112</b> | <b>Horas Semana: 7</b> |

### OBJETIVOS

- Reconocer los fundamentos, características y diferencias de los distintos paradigmas de programación que son utilizados por los lenguajes de programación actuales.
- Identificar el paradigma que resulta más adecuado de aplicar para resolver diferentes tipos de problemas.
- Reconocer el modelo formal o semiformal subyacente de cada paradigma y la forma en que el mismo es incorporado en un lenguaje de programación correcto
- Adquirir las nociones básicas de los modelos de programación orientada a objetos, lógico y funcional.
- Aplicar los diferentes paradigmas en la resolución de problemas.

### CONTENIDOS

Concepto de paradigma de programación. Necesidad de la existencia de diferentes paradigmas de programación. Concepto de programa: definiciones generales y específicas. 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. Declaratividad: importancia de la separación del control de flujo de la lógica del dominio a modelar. Abstracción y modularización: definición y mecanismos de implementación. Concepto de ligaduras. Tipos de Ligaduras. Orden superior: concepto e implicancias en el desarrollo de programas. Utilización de las variantes del polimorfismo en los diferentes paradigmas. Comparación entre los diferentes paradigmas de programación.

#### UNIDAD2: PARADIGMA LÓGICO

Características de la programación declarativa y la programación lógica. Áreas de aplicación. Fundamentos Teóricos. Lógica de Primer Orden. Lógica de Cláusulas Definidas (Horn). Unificación. Algoritmo de Robinson. Aritmética y Operadores. Principio de resolución SLD. Tipos de Derivaciones SLD. Árbol de derivación SLD. Retroceso Cronológico. Operador de corte. Negación. Functores.

**Lenguaje asociado:** Implementación de programas en Prolog. Entorno de trabajo. Características del lenguaje. Sintaxis y semántica del lenguaje. Estructura general de programas. Hechos y Reglas. Realización de consultas. Trabajo con listas. Variables. Variables anónimas. Aritmética y Operadores.

#### UNIDAD 3: 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. Funciones recursivas. Manejo de listas. Listas por comprensión/extensión. Funciones de orden superior. Currificación y aplicación parcial de funciones. Evaluación diferida y listas infinitas. Sistemas de tipos. Polimorfismo y tipos genéricos. Tuplas. Expresiones lambda. Comparación con otros paradigmas.

**Lenguaje asociado:** Haskell. Entorno de trabajo, definición de programas, uso del intérprete. Notación bidimensional. Módulos. Notación de listas [n..m]. Notación de listas por comprensión. Operadores infijos y prefijos. Reglas de precedencia. Prelude de Haskell. Funciones incorporadas en el prelude para manejo de listas, de tuplas, de funciones de orden superior. Entidades y ligaduras.

#### **UNIDAD 4: PARADIGMA ORIENTADO A OBJETOS**

Concepto de Objeto. Concepto de mensaje, estado y comportamiento. 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 Clase como modelo/molde de objetos. Introducción a UML. 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. Autoreferencia y referencia a superclases. Subclases. Redefinición de métodos. Clases abstractas. Concepto de polimorfismo. 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. Persistencia de objetos. Colecciones.

**Lenguaje asociado:** Java. Características generales, sintaxis y semántica del lenguaje. Estructura de programas. Clases. Constructores. Encapsulamiento y modularidad. Instanciación y destrucción de objetos. Compilación y ejecución. Variables y métodos. Modificadores de acceso. Lanzamiento, captura y tratamiento. Excepciones. Herencia e Interfaces. Colecciones. Entidades y ligaduras.

#### **METODOLOGÍA DE ENSEÑANZA**

Por tratarse de una asignatura con importante respaldo conceptual, y para facilitar la apropiación de los conocimientos, se aplicará la ejemplificación y demostración de manera metódica. Los contenidos del programa se presentan y analizan en las clases teóricas y se consolidan con los trabajos prácticos. Existe una estrecha relación entre la teoría y práctica, a su vez la práctica fundamenta sus procedimientos en la teoría.

Los ejercicios están orientados al análisis o entendimiento de las soluciones propuestas y a la concepción o diseño para responder a nuevos requerimientos y problemas. Para ello el material de práctica es seleccionado de manera de cubrir un amplio espectro temático y muestra diferentes niveles de dificultad. Algunos ejercicios son resueltos completamente en clase, y otros son encaminados para ser completados por los alumnos fuera de clase.

##### Estrategias metodológicas y de enseñanza

Las siete horas semanales que tiene asignada la materia se distribuyen de la siguiente manera:

- Clases teóricas de cuatro horas para presentación y exposición de los temas. Dichas



clases incluyen gran contenido de forma práctica y serán participativas y de debate de forma virtual. Se desarrollarán de manera expositiva los conceptos fundamentales, con presentación de casos de estudio prácticos de aplicación inmediata del tema presentado. Se promoverá la investigación y la búsqueda de información por parte del alumno

- Clases prácticas de tres horas de ejercitación para resolución de problemas y consultas trabajando en el aula. Estas clases incluirán presentación del tema, casos de estudio abiertos a debate y de construcción colectiva, y consulta individual de ejercicios. Se incluyen actividades de laboratorio para aplicar diferentes lenguajes de programación (Python, Java, entre otros) como herramienta instrumental de resolución de problemas de acuerdo al paradigma que se esté estudiando. Para ello el material de práctica es seleccionado de manera de cubrir un amplio espectro temático y muestra diferentes niveles de dificultad. Algunos ejercicios son resueltos completamente en clase a distancia, y otros son encaminados para ser completados por los alumnos fuera de clase.

En el contexto de pandemia COVID 19, la modalidad a distancia adoptada utiliza la plataforma de Aula Abierta, meet y/o jitsi para el desarrollo de las clases teóricas en vivo en los horarios habituales de cursada sumado a herramientas como diapositivas (PowerPoint/PDF) para ayudar al alumno en el seguimiento de la clase. Además, se prevé grabar el desarrollo de ejercicios y subirlos a youtube o google drive a fin de brindar a los alumnos más ejemplos fuera del horario de habitual de clases.

### ***DISTRIBUCIÓN DE LA CARGA HORARIA***

| <b>Actividad</b>                          | <b>Carga horaria por semestre</b> |
|-------------------------------------------|-----------------------------------|
| Teoría y resolución de ejercicios simples | 64                                |
| Formación práctica                        |                                   |
| Formación Experimental – Laboratorio      | 0                                 |
| Formación Experimental - Trabajo de campo | 0                                 |
| Resolución de problemas de ingeniería     | 48                                |
| Proyecto y diseño                         | 0                                 |
| <b>Total</b>                              | <b>112</b>                        |

|                                         |                |
|-----------------------------------------|----------------|
| <b>Porcentaje de Horas Presenciales</b> | 0 % del Total  |
| <b>Porcentaje de Horas a Distancia</b>  | 100% del Total |

### ***Bibliografía básica***

| Autor                                         | Título                                                        | Editorial         | Año  | Ejemplares en biblioteca |
|-----------------------------------------------|---------------------------------------------------------------|-------------------|------|--------------------------|
| T.W. Pratt, M.V. Zelkowitz                    | Programming Languages - Design and Implementation, 4º edición | Prentice Hall     | 2000 |                          |
| Deitel&Deitel                                 | Como Programar en Java, 7ma edición                           | Prentice Hall     | 2008 |                          |
| R. Bird                                       | Introducción a la Programación Funcional con Haskell          | Prentice Hall     | 2000 |                          |
| W.D. Burnham, A.R. Hall                       | Prolog, Programación y Aplicaciones                           | Limusa            | 1990 |                          |
| Allen B. Tucker, Rober E. Noonan              | Programming Languages Principles and Paradigms                | McGraw-Hill       | 2007 |                          |
| Aho, Alfred V., Ravi Sethi, Jeffrey D. Ullman | <i>Compiladores: principios, técnicas y herramientas</i>      | Pearson Educación | 2008 |                          |

### ***Bibliografía complementaria***

| Autor                         | Título                                  | Editorial   | Año  | Ejemplares en biblioteca |
|-------------------------------|-----------------------------------------|-------------|------|--------------------------|
| John S. dean, Raymond H. Dean | Introducción a la Programación con Java | McGraw-Hill | 2009 |                          |
| H. Schildt                    | Java 7 - Oracle Press                   | ANAYA       | 2012 |                          |
|                               |                                         |             |      |                          |
|                               |                                         |             |      |                          |
|                               |                                         |             |      |                          |
|                               |                                         |             |      |                          |

### ***EVALUACIONES (S/ Ord. 108-10\_CS)***

Se propone una evaluación continua de alumno de acuerdo al desarrollo ejercicios entregables y un examen parcial a distancia. Se presenta a continuación las normas para regularizar y aprobar la materia.

#### ***Evaluaciones durante el cursado:***

- El alumno deberá entregar ejercicios especificados por el docente responsable de cada trabajo práctico. El alumno debe aprobarlos con nota aprobado o sus correspondientes re-entregas.
- Examen parcial. Se evaluará a distancia con un examen parcial a distancia aquellos conceptos teóricos que sean relevantes para alcanzar la regularidad de la asignatura.

#### ***Evaluación Recuperatoria y Global:***

- Cada etapa tendrá su correspondiente instancia de recuperación a través de la reentrega de la misma en caso de obtener nota desaprobado en una entrega. Los archivos y su

documentación quedarán a respaldo documental.

- El examen parcial tendrá su correspondiente recuperatorio a distancia.

### ***Condición de Regularidad***

En base a los resultados de las evaluaciones el alumno quedará como Alumno Regular, cuando haya aprobado cada entrega o re-entrega los ejercicios solicitados y el correspondiente examen parcial o su recuperatorio.

**Alumnos recursantes.** No hay régimen especial para alumnos recursantes.

### ***Criterios de evaluación:***

En cuanto a alumnos regulares: se valorará la precisión del alumno en las respuestas a las preguntas realizadas durante el examen oral. La coherencia de sus respuestas entre sí junto con el grado de conocimiento demostrado en la capacidad de relacionar los conceptos vistos durante la cursada. La madurez conceptual evidenciada en la capacidad de respuestas rápidas concisas y precisas a consultas realizadas. El conocimiento de los modelos y su potencialidad.

En cuanto a alumnos libres: se evaluará la capacidad del alumno para desarrollar soluciones a diferentes problemas partiendo de los conceptos fundamentales de cada paradigma. Se valorará la precisión del alumno en las respuestas a las preguntas realizadas durante el examen de preguntas vía plataforma web y luego un examen oral. La coherencia de sus respuestas entre sí junto con el grado de conocimiento demostrado en la capacidad de relacionar los conceptos de la asignatura. La madurez conceptual evidenciada en la capacidad de respuestas rápidas, concisas y precisas a consultas realizadas. El conocimiento de las fases del desarrollo de un compilador tanto a nivel teórico como las dificultades a nivel práctico.

### ***Condiciones para la acreditación de la asignatura:***

**Examen Final alumnos regulares.** El examen final es de tipo integrador teórico-práctico, de forma oral o escrita, sobre cualquiera de los temas desarrollados en la materia. El examen final se rinde a programa completo, exigiéndose el nivel superior correspondiente. independientemente que se hayan tomado o no en las evaluaciones parciales. Todos los temas evaluados deben conocerse en al menos un 60% del alcance desarrollado en la materia. Podrán rendir examen final aquellos alumnos regulares. La calificación del examen final considerará la totalidad del proceso de enseñanza aprendizaje.

**Examen final para alumnos libres.** El alumno deberá comunicarse con el profesor responsable quién le entregará una serie de ejercicios siguiendo las consignas dadas durante el año lectivo en curso que deberá resolver antes de presentarse al examen final. El examen consta de la evaluación de lo entregado y un examen oral. Los ejercicios deberán ser entregados a la docente responsable de la asignatura 72 hs. antes del examen final. En caso de que los ejercicios estén aprobados con una nota superior o igual a 6 (SEIS) el alumno libre deberá rendir un examen oral en el mismo momento y lugar según los criterios de evaluación mencionados previamente. El examen final libre se rinde a programa completo, exigiéndose el nivel superior correspondiente.

### ***Programa de examen:***

Contempla la totalidad de los temas del presente programa.



30/09/2020

Vidal, Pablo Javier

FECHA, FIRMA Y ACLARACIÓN TITULAR DE CÁTEDRA