

NOMBRE DE LA ASIGNATURA

Año 2020

Carrera/ Plan:

Licenciatura en Informática Plan 2015
Licenciatura en Sistemas Plan 2015
Analista Programador Universitario Plan 2015
Licenciatura en Informática Plan 2003-07/Plan 2012
Licenciatura en Sistemas Plan 2003-07/Plan 2012
Analista Programador Universitario Plan 2007

Año: 3ero**Régimen de Cursada:** *Semestral***Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:** SI203 / SI207 / SI208**Profesor/es:** Mg. Viviana Harari / Lic. Dalila Romero**Hs. semanales:****FUNDAMENTACIÓN**

El conocimiento profundo de los conceptos intrínsecos de los lenguajes de programación permite adquirir herramientas para construir criterios de evaluación de los lenguajes. Esto hace posible consolidar los conocimientos sobre los lenguajes ya conocidos e incorporar elementos para abordar los nuevos. Como formación complementaria se alienta y evalúa el trabajo colaborativo y la capacidad de presentar y exponer trabajos en forma escrita y oral.

OBJETIVOS GENERALES

Introducir, analizar, comparar y evaluar los conceptos subyacentes de los Lenguajes de Programación en los distintos Paradigmas.
Adquirir la capacidad de evaluar lenguajes de programación desde distintos puntos de vista, ya sea como su diseñador, implementador o como usuario del lenguaje.

COMPETENCIAS

- CGS1- Desempeñarse de manera efectiva en equipos de trabajo, con capacidad para organizarlos y liderarlos.
- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT10- Capacidad para realizar investigaciones bibliográficas y de diferentes fuentes de información a fin de obtener conocimiento actualizado en temas de la disciplina.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS - CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.



CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

- Sintaxis y semántica.
- Semántica operacional.
- Entidades atributos y ligaduras.
- Sistemas de tipos.
- Encapsulamiento y abstracción.
- Intérpretes y Compiladores.
- Paradigmas de lenguajes (imperativo, orientado a objetos, funcional, lógico).
- Programación basada en scripting.

PROGRAMA ANALÍTICO

- 1- Lenguajes de Programación como herramientas para producir software. Objetivo del estudio de sus conceptos. Distintos puntos de vista. Capacidad de comunicación. Relación e integración de los lenguajes de programación respecto a otros conceptos.
- 2- Evaluación de los lenguajes de programación a través de las características del software que producen. Principios de diseño de los lenguajes.
Perspectiva histórica de los lenguajes de programación motivación, herencia, características, evolución. Lenguajes funcionales. Lenguajes Orientados a objetos. Abstracción: qué papel juega. Necesidad de estandarizar: ¿cuándo y cómo hacerlo?
- 3- 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. Sintaxis abstracta y concreta.
Formas de definir la sintaxis. Gramáticas. (Backus Naum Form). Árboles sintácticos y de derivación. Gramáticas recursivas. Subgramáticas. Gramáticas ambiguas. EBNF. Diagramas sintácticos (CONWAY)
Semántica estática.
- 4- Semántica. Tipos de semánticas. Formas de definir la semántica de un lenguaje de programación. Nociones de semántica formal.
Procesamiento de un lenguaje interpretación y traducción. Tipos de traductores. Comparación entre Traductor e Intérprete. Combinación de ambas técnicas. Compiladores. Etapas de Análisis y Síntesis. Optimización.
- 5- Semántica Operacional.
Ligadura. Descriptores. Momentos de ligadura. Estabilidad.
Variables. Arquitectura Von Newman. Atributos. Momentos y estabilidad. Nombre: características. Alcance: visibilidad, reglas. Tipo: definición, clasificación. L-valor: tiempo de vida, alocaión. R-valor: constantes, inicialización. Alias
- 6- Unidades. Atributos. Representación en ejecución. Elementos. Unidades recursivas. Unidades genéricas. Alias y sobrecarga. Procesador abstracto: elementos, instrucciones.
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.
- 7- Compartir Datos. Ambiente común. Acceso al ambiente no-local. Parámetros. Ventajas. Evaluación de los parámetros reales y ligadura con los parámetros formales. Clase de parámetros: Datos y Subprogramas. Modos de pasaje de parámetros datos. Pasaje de Rutinas como parámetros.
- 8- Sistema de tipos. Tipos predefinidos, tipos definidos por el usuario, tipos estructurados, tipos abstractos. Implementación de datos, su representación.
Sistema de tipos: lenguajes seguros y fuertemente tipados. Seguridad en el manejo de tipos. Encapsulamiento y abstracción. Evolución de los tipos. Tipos Abstractos. Equivalencia de tipos.
- 9- Abstracción de Control. Estructuras de control: Definición de estructuras de control a nivel de sentencia y a nivel de unidad. Tipos de estructuras de control a nivel de sentencia. Diferencia entre sentencia de asignación y expresión. Evolución de las sentencias de selección e iteración en diferentes lenguajes, desde PL/1 a Java.
- 10- Abstracción de control a nivel de unidad. Excepciones: Definición. Modelos de Terminación y Reasunción. Distintos modelos de manejo de excepciones. Comparación.

- 11- Paradigma funcional. Características. Comparación lenguaje imperativo con lenguaje funcional. Definiciones de funciones. Script. Expresión y valor. Transparencia referencial. Evaluación de las expresiones, mecanismo de reducción o simplificación. Orden aplicativo, orden normal (lazy evaluation). Tipos de datos básicos y derivados. Tipos de funciones. Expresiones polimórficas. Currificación. Cálculo Lambda. Dominios de Aplicación.
- 12- Paradigma Orientado a Objetos. Características. Elementos básicos de la programación orientada a objetos: objetos, mensajes, métodos, clases. Conceptos de generalización, especificación y herencia. Diferentes tipos de herencia. Lenguajes híbridos, características principales. Dominios de aplicación.
- 13- Paradigma lógico. Características. Elementos de la programación lógica: variables, constantes, términos compuestos, listas. Cláusulas y predicados. Reglas y hechos. Dominios de aplicación.
- 14- Programación basada en scripting. Definición. Introducción histórica. Características. Tipos. Dominios de interés. Los lenguajes de scripting y la WWW. Aspectos innovadores.

BIBLIOGRAFÍA

BIBLIOGRAFIA OBLIGATORIA

- GHEZZI C. – JAZAYERI M.: Programming language concepts. John Wiley and Sons. (1998) 3er. Ed
- SEBESTA: Concepts of Programming languages. Benjamin/Cumming. (2010) 9a. Ed.
- LOUDEN K.C.: Programming languages: principles and practices (2011)
- PRATT: Programming Lenguajes. Design and Implementation. Prentice Hall (2001) 4ta. Ed.
-
- SETHI R.: Programming lenguajes: concepts and constructs. Addison – Wesley (1996) 2nd. Ed.
- Programming Language Pragmatics (3ed., Elsevier, 2009) Scott M.L

BIBLIOGRAFIA COMPLEMENTARIA

- FRIEDAMAN D.: Essentials of programming languages (2008)
- HOROWITZ: Fundamentals of Programming Languages. Spring- Verlar (1984)
- SCOTT M.L.: programming language pragmatics (3er. Ed.) (2009)
- TURCKER A.: Programming languages (2006)
- WATT D. Programming Language Design Concepts (2004)

METODOLOGÍA DE ENSEÑANZA

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real desde la visión del desarrollador, basados en la evaluación, comportamiento y características de los conceptos que atraviesan a los lenguajes de programación.

Las actividades se desarrollan bajo la modalidad de teorías y prácticas distribuidas equitativamente de acuerdo con el Plan de Estudios vigente.

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. En la teoría, que toma sentido y se fortalece con las prácticas, se realiza un análisis vertical, indagando sobre los conceptos y bajando ejemplos de la aplicación del mismo en diferentes lenguajes. En la práctica, que se fundamenta sus procedimientos en la teoría.

Los alumnos desarrollan actividades individuales y grupales (entre 2 y 3 alumnos) que debieran conducirlos a un seguimiento y avance gradual sobre los temas abordados. En las prácticas, se presentan ejercicios integradores, cuya entrega es obligatoria, en donde se plantean diferentes actividades de investigación y desarrollo que permiten que los alumnos pongan en juego sus propios conocimientos. La metodología de investigación que propone la cátedra propicia que el alumno realice las siguientes actividades:

- Búsqueda de bibliografía actualizada sobre el tema.
- Consulta a profesionales y catedráticos sobre temas investigados
- Discusión de alternativas tecnológicas para resolver el tema propuesto en diferentes lenguajes de programación.

Desde la cátedra se pretende fomentar el trabajo en grupo y la adquisición de experiencia en la comunicación escrita y oral de sus trabajos.

Los alumnos también cuentan con "cuestionarios online" que les dan la oportunidad de autoevaluarse para saber si han comprendido los conceptos enseñados.

Se ofrecen cuatro horarios de práctica distribuidos en la semana en distintas franjas (mañana, tarde y noche) a fin de cubrir todas las posibilidades que faciliten la asistencia.

Los materiales didácticos son los tradicionales, haciendo uso del cañón y PC disponible en las aulas para apoyar el desarrollo de los temas a través de aplicaciones de presentación.

Se pone a disposición de los/as estudiantes un entorno educativo web con la que se mantiene una comunicación dinámica. Se hace uso de la misma como:

- Medio informativo: Se coloca el cronograma de toda la cursada, el programa de la materia, bibliografía, guías de clases teóricas, ejercicios integradores y trabajos prácticos.
- Medio de comunicación: Se colocan novedades de la cursada, fechas de los parciales y entregas y, resultados de las correcciones de las diferentes evaluaciones.
- Medio de evaluación: Se utiliza el recurso para armar encuestas, cuestionarios autoevaluativos y, entrega de tareas con calificación.

Los/as estudiantes también cuentan con la posibilidad de comunicarse con los profesores y jefes de trabajos prácticos a través de sus cuentas de e-mail.

El equipo docente mantiene reuniones al comienzo de cada ciclo lectivo para delinear la planificación anual a la vista de la evaluación de los resultados precedentes y luego se mantiene en comunicación permanente para ajustes puntuales. Asimismo se realiza la revisión continua de los temas desarrollados, ya sea para modificar la metodología utilizada como para introducir o modificar conceptos y lenguajes nuevos.

EVALUACIÓN

Para la aprobación cursada de la materia, los alumnos deben aprobar un trabajo de investigación e integrador de las temáticas abordadas y, un examen teórico-práctico que cuenta con dos recuperatorios. Las fechas del examen y sus recuperatorios se establecen de acuerdo a la duración del semestre fijado

por el Calendario Académico de la Facultad, son publicados al principio del ciclo lectivo y, se organizan de forma tal de no entorpecer el normal desarrollo de las cursadas.

El ejercicio integrador es planteado en dos etapas y se presenta al iniciar la cursada. Es obligación su "entrega" y "aprobación" para poder rendir el examen teórico-práctico. La primera entrega cuenta con la posibilidad de una reentrega.

El desarrollo de investigación que propone el trabajo integrador debe ser presentado, por el grupo de alumnos, a través de un informe escrito que será evaluado por los docentes teniendo en cuenta el contenido técnico, la estructura, la organización, la sintaxis, la claridad conceptual y la y la bibliografía consultada citada rigurosamente.

Como parte de la evaluación del trabajo integrador se tendrá en cuenta el desempeño grupal e individual de cada uno de los integrantes, a través de un coloquio. Respecto a la parte grupal, los grupos responderán aspectos generales de las tareas asignadas y darán explicaciones individuales que permitirán calificar diferentes aptitudes de los miembros del equipo (conocimientos / modo de expresarse / predisposición al trabajo colaborativo). En lo que respecta a la parte individual, cada uno de los integrantes del grupo deberá exponer sobre la tarea realizada. El docente indagará sobre sus conocimientos, su claridad de la presentación, su organización y la forma de expresión.

A lo largo de la cursada se implementarán diferentes actividades, tanto en la teoría como en la práctica. Las mismas tendrán un puntaje asignado y el estudiante, que las realice en forma satisfactoria, podrá acumular una serie de puntos. Esos puntos los podrá utilizar en forma favorable en algunas de las instancias de evaluación.

Todas las evaluaciones realizadas a los estudiantes quedarán reflejadas en planillas, que la cátedra irá actualizando a lo largo de la cursada. Algunos de los datos que se registrarán son los siguientes: resultados de las evaluaciones, capacidad del estudiante para desarrollar su aprendizaje, claridad de las presentaciones realizadas, forma de organización y expresión en las diferentes instancias de evaluación oral, formulación de la solución de los diferentes desafíos en forma autónoma, entre otros.

Para la evaluación final de la materia se tomará una prueba teórica individual escrita. La misma se puede realizar en las mesas de finales correspondientes al calendario académico o, el alumno puede optar por realizarlo al finalizar la cursada.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	17/3	PRESENTACION DE LA ASIGNATURA: pautas Introducción, evaluación, historia de los lenguajes
2	31/3	ESTRUCTURA DE UN LENGUAJE: sintaxis y semántica. Sintaxis.
3	7/4	SEMANTICA. Procesamiento de un lenguaje: interpretación y compilación
4	14/4	SEMANTICA OPERACIONAL: entidades y ligaduras. Variables
5	21/4	SEMANTICA OPERACIONAL: unidades. Procesamiento de lenguajes estáticos
6	28/4	SEMANTICA OPERACIONAL: unidades recursivas, cadena estática. Acceso al ambiente no local
7	5/5	SEMANTICA OPERACIONAL: comportamientos mas dinámicos. Cadena dinámica
8	12/5	SEMANTICA OPERACIONAL: conclusiones, diferenciación y reconocimiento de atributos, repaso.
9	19/5	COMPARTIR DATOS: casos. Parámetros

10	26/5	SISTEMA DE TIPOS: características, tipos predefinidos y tipos definidos por el usuario.
11	2/6	SISTEMA DE TIPOS: tipos compuestos y tipos abstractos. Lenguajes seguros, equivalencia y compatibilidad.
12	9/6	ABASTRACION DE CONTROL: a nivel de sentencia y a nivel de unidad. Excepciones
13	16/6	PARADIGMAS. Funcional. Orientado a Objetos. Lógico
14	23/6	SCRIPTING: Programación basada en scripting
15	30/6	Repaso general

Evaluaciones parciales	Fecha
Parcial- 1er Fecha	semana del 8/6
Parcial - 2da Fecha	semana del 29/6
Parcial – 3er Fecha	Semana del 6/7

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

La comunicación en la asignatura se da a través de:

<https://catedras.info.unlp.edu.ar>

Mail de contacto de los profesores:

vharari@info.unlp.edu.ar

dalilar@info.unlp.edu.ar

Firma del/los profesor/es

Año del Centenario de la Reforma Universitaria