**1.OBJETIVOS**

* Transmitir a los alumnos conceptos fundamentales vinculados al diseño de lenguajes de programación para facilitar el análisis de los distintos paradigmas.
* Hacer operativos los conceptos básicos, mediante procesos de reflexión, creación y ejercitación.
* Desarrollar conocimientos y competencias vinculadas con el diseño y criterios de diseño de lenguajes de programación.
* Desarrollar competencias para la identificación y selección de paradigmas.
* Integrar los conocimientos particulares, vinculados al diseño de lenguajes de programación, con otros conocimientos pertinentes, ya logrados durante el curso de la carrera.
* Ajustar la visión individual conformada desde la teoría y la práctica realizada durante la carrera sobre la programación, los lenguajes de programación y su implementación a su marco teórico actual; completarla.

Al finalizar el curso los alumnos estarán preparados para

* Analizar y juzgar críticamente cualquier lenguaje de programación en función de su aplicabilidad, estructuras, componentes e implementación.
* Clasificar lenguajes de programación en función de su uso, aplicabilidad e implementación.
* Vincular las características de los lenguajes de programación con los criterios que determinaron su diseño (epistemología: conocimientos teóricos, sintaxis, semántica) y a las posibilidades de implementación (estado del arte: hardware, software).
* Vincular los distintos paradigmas con su modelo semántico.
* Aplicar los contenidos de la asignatura a la resolución de problemas del mundo real.
* Transferir los conocimientos adquiridos en el contexto teórico a otras situaciones de la práctica.

**2. CONTENIDOS MINIMOS**

Evolución de los principales lenguajes de programación. Criterios de implementación. Sintaxis y semántica. Nociones básicas de semántica formal. Análisis léxico y sintáctico. Entidades y ligaduras. Sistemas de tipos. Verificación de tipos y ámbitos. Tipos de datos. Polimorfismos. Niveles de polimorfismos. Expresiones y sentencias de asignación. Estructuras de control. Abstracción y encapsulamiento. Subprogramas y su implementación. Tipos de datos abstractos. Soporte para programación orientada a objetos. Concurrencia. Manejo de excepciones. Paradigmas de lenguajes de programación: Imperativo, Orientado a Objetos, Funcional, Lógico

**3. CONTENIDOS**

**Unidad 1. Introducción a los lenguajes de programación**

Objetivos y razones para estudiarlos. Concepto. El arte de diseñarlos. Orígenes. Historia. Taxonomías. Estudio de los lenguajes: sintácticos, semánticos y pragmáticos. Desarrollo de los lenguajes de programación. Características deseables. Eficiencia. Medición de la eficiencia. Lenguajes de programación con mayor demanda.

**Unidad 2 Paradigmas**

Concepto. Clasificación. Paradigma Imperativo. Paradigma Orientado a Objetos. Paradigma Lógico. Paradigma Funcional. Características. Lenguajes. Ejemplos. Integración de paradigmas.

**Unidad 3 Sintaxis**

Concepto. Características. Elementos. Estructura sintáctica. Tipos de sintaxis. Gramáticas. BNF. EBNF. BNF que utiliza el MROC (Manual de Referencia Oficial de ANSI C). Arboles sintácticos. Parsing descendente y ascendente. Precedencia. Producciones recursivas. Diagramas sintácticos (CONWAY). Ejemplos.

**Unidad 4 Semántica**

Concepto. Tipos de semántica: estática y dinámica. Modelos semánticos: semántica axiomática (verificación formal de programas, terna de Hoare), semántica composicional u operacional y semántica denotacional (funciones recursivas, cálculo lambda).

**Unidad 5 Ligadura**

Definición. Momentos: Tipos de Ligadura: estática y dinámica. Variables. Atributos (nombre, alcance, tipo, l-value, r-value). Alias.

**Unidad 6 Unidades**

Abstracción. Concepto. Atributos. Definición vs. Declaración. Comunicación entre rutinas. Parámetros. Rutinas genéricas. Pasaje de funciones como parámetros. Pasaje de arreglos como parámetros. Sobrecarga. Unidades en memoria. Closures.

**Unidad 7 Tipos**

Abstracción de datos. Evolucion del concepto de tipo: tipo básico, tipo definido por el usuario, tipo de dato abstracto. Sistema de tipos. Tipado fuerte. Tipado débil. TDA. Constructores de encapsulamiento. Diseño de TDA. TDA parametrizados. Extensiones de tipo. Herencia. Polimorfismo. Implementación de la programación OO. Encapsulamiento de nombres. Compatibilidades y conversiones.

**Unidad 8 Excepciones**

Concepto. Tipos. Mecanismos no específicos. Mecanismos específicos. Ejemplos. Lenguajes que proveen manejo de excepciones. Eventos.

**Unidad 9 Ejecución**

Introducción. Expresiones. Asignación. Evaluación. Control de flujo no estructurado. Control de flujo estructurado. Estructuras de control estructuradas: secuencia, condición e iteración. Iteradores. Recursión. No determinismo.

**4. BIBLIOGRAFÍA**

***4.1. Básica***

|  |  |  |  |
| --- | --- | --- | --- |
| **AUTOR** | **Título** | **Editorial** | **Año** |
| PRATT, T | Lenguajes de programación. Diseño e implementación. | Prentice Hall | 1998 |
| TUCKER, A. | Lenguajes de programación | McGraw Hill | 1990 |
| SETHI, R. | Lenguajes de programación. Conceptos y constructores. | Addison Wesley | 1992 |

***4.2. Herramientas de trabajo:***

* Compilador Pascal o C
* Intérprete Lisp, Haskell
* Intérprete Prolog
* Entorno de trabajo Java

**5. METODOLOGÍA DE LA ENSEÑANZA**

Enfoque teórico práctico, con énfasis en la aplicación de los conocimientos teóricos a la resolución de problemas.

Los conocimientos teóricos serán presentados favoreciendo la participación del alumno, para favorecer la vinculación de los conocimientos específicos de la materia con conocimientos previos. Los conocimientos específicos de la materia serán llevados a la práctica mediante la resolución de problemas de la disciplina.

Una vez alcanzado el nivel teórico suficiente se ofrecerá a los alumnos una lista de problemas de la disciplina, de entre los que seleccionarán alguno para resolver como un trabajo práctico de proyecto y diseño. Este trabajo deberá ser desarrollado incrementalmente a lo largo de la materia. No se requiere que la resolución del trabajo sea individual.

La sección dedicada a la práctica de cada clase estará caracterizada por las siguientes modalidades: 1) la reflexión en la práctica 2) debates orientados a definir criterios para la toma de decisiones en el desarrollo del trabajo 3) las consultas individuales para satisfacer dudas de los alumnos.

Al finalizar la materia, los diferentes trabajos son puestos en común por los autores y discutidos con sus compañeros.

**5. EVALUACIÓN**

**5.1 Oportunidades y criterios de Evaluación**

*a. En cada clase.*

Mediante actividad práctica realizada en computadoras, se vincularán los contenidos introducidos en la teórica en los trabajos prácticos que desarrollan los alumnos. Los alumnos deberán entregar un informe parcial de dicho trabajo, que involucre la práctica desarrollada, al comienzo de la clase siguiente.

*b. Examen parcial*

Los alumnos serán evaluados con respecto a los contenidos presentados en clase mediante a) preguntas sobre contenidos teóricos y la resolución de ejercicios y b) sobre la autoría y dominio del trabajo que están desarrollando mediante preguntas orientadas a relacionar los contenidos de la materia con cada trabajo en particular; también se indagará sobre la participación del alumno en la producción de dicho trabajo.

*c. Desarrollo del trabajo práctico integrador*

Los alumnos acompañarán el desarrollo de la teoría de la materia mediante la resolución de un problema abierto, que involucra situaciones de diseño y programación, posible en la actividad profesional, ajustado al perfil del Licenciado en Sistemas de Información.

La extensión y profundidad del tratamiento del tema deberá ser adecuada al año de la carrera en que esté ubicada la asignatura.

*d. Examen final*

El examen final se desarrollará a partir del trabajo que el alumno produjo a lo largo de toda la materia. A partir del mismo se indagará sobre el dominio que tiene el alumno de los conceptos propios de la materia y su capacidad para ponerlos en juego.

**5.2 Requisitos para la aprobación**

*a. En cada clase.*

Mediante actividad práctica realizada en computadoras, se vincularán los contenidos introducidos en la teórica en los trabajos prácticos que desarrollan los alumnos. Los alumnos deberán entregar un informe parcial de dicho trabajo, que involucre la práctica desarrollada, al comienzo de la clase siguiente.

*b. Examen parcial*

Los alumnos serán evaluados con respecto a los contenidos presentados en clase mediante a) preguntas sobre contenidos teóricos y la resolución de ejercicios y b) sobre la autoría y dominio del trabajo que están desarrollando mediante preguntas orientadas a relacionar los contenidos de la materia con cada trabajo en particular; también se indagará sobre la participación del alumno en la producción de dicho trabajo.

*c. Entrega del trabajo práctico integrador*

Los alumnos acompañarán el desarrollo de la teoría de la materia mediante la resolución de un problema abierto, que involucra situaciones de diseño y programación, posible en la actividad profesional, ajustado al perfil del Licenciado en Sistemas de Información.

*d. Examen final*

El examen final se desarrollará a partir del trabajo que el alumno produjo a lo largo de toda la materia. A partir del mismo se indagará sobre el dominio que tiene el alumno de los conceptos propios de la materia y su capacidad para ponerlos en juego.

**5.2 Requisitos para la aprobación**

***Aprobación de la cursación de la asignatura.*** Para aprobar es necesario cumplir con:

* Asistencia mínima del 50%
* Aprobación del examen parcial con nota igual o superior a cuatro puntos:

Los parciales deben rendirse en las fechas estipuladas por la Facultad.

En el caso de que el alumno desapruebe el parcial cuenta con una instancia de recuperación.

El desaprobar o no asistir a la recuperación (teniendo el parcial desaprobado) tiene como consecuencia desaprobar el curso de la materia.

* Aprobación de los Trabajos prácticos con nota igual o superior a cuatro puntos:

En el caso de esta materia la nota final de los trabajos prácticos se calcula como una nota promedio de los trabajos requeridos. Existe una instancia de recuperación.

***Aprobación de la asignatura.*** Para aprobar es necesario aprobar la cursación y el examen final

* Para aquellos alumnos que no alcanzaran el 75% de asistencias el examen final escrito se aprueba con seis puntos.
* Para los alumnos que alcancen o superen el 75% el examen final se aprueba con cuatro puntos.

**6. DISTRIBUCION CARGA HORARIA**

**Carga Horaria - Modalidad de Enseñanza**

|  |  |
| --- | --- |
| **Modalidad** | **Horas** |
| **T**eóricas | **43** |
| Act. **P**rácticas | **35** |
| **E**valuaciones | **18** |
| **Total** | **96** |