UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS** **FACULTAD DE INGENIERÍA** **SYLLABUS** **PROYECTO CURRICULAR DE INGENIERÍA ELÉCTRICA**	

Nombre del Docente

ESPACIO ACADÉMICO (Asignatura): **PROGRAMACIÓN ORIENTADA A OBJETOS**						**Código:**	
Obligatorio	☒	Básico	☒	Complementario		**010**	
Electivo		Intrínseco		Extrínseco			
Número de Estudiantes						**Grupo**	
Número de Créditos		**TRES (3)**					
TIPO DE CURSO:		Teórico		Práctico		Teórico - Práctico	☒

Alternativas Metodológicas:

Clase Magistral	☒	Seminario		Seminario-Taller		Taller	☒	Teórico - Práctico	☒
Proyectos Tutoriados	☒	Otros							

HORARIO

DÍA	HORAS	SALÓN

I. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

Competencias del perfil a las que contribuye la asignatura:

- Esta asignatura contribuye al desarrollo de la competencia “Resuelve problemas computacionales algorítmicamente” que se encuentra en el dominio de “programación” del área “básicas de ingeniería” del proyecto curricular de ingeniería Eléctrica.
- Contribución a la formación:
- Este espacio académico contribuye al perfil de “Desarrollo de software y aplicaciones” el cual se consolida en la línea de “programación”, introduciendo al estudiante a la Programación Orientada a Objetos y brindándole las herramientas para aplicar sus principios, características y objetivos Puntos de apoyo para otras asignaturas
- Solución de ecuación lineales – Algebra lineal
- Solución de problemas prácticos sobre circuitos – Circuitos
- Aprendizaje sobre el uso de librerías matemáticas, para reutilizar y aplicar en la solución de problemas para el área de física y matemáticas.
- Aprendizaje de procesos metodológicos para resolver problemas de ingeniería, mediante el análisis de los requerimientos – Todas las materias de la carrera.
- Aplicación de lo aprendido en modelado, para efectuar representación gráfica del dominio y la solución de los problemas – Todas las asignaturas.
- Aprendizaje del uso de los puertos del computador para captura y procesamiento de señales – Asignaturas relacionados con ingeniería eléctrica aplicada – Instrumentación basada en Computador.

Conocimientos Previos:

Programación Básica

II. PROGRAMACIÓN DEL CONTENIDO

OBJETIVO GENERAL

Este curso tiene por objeto dar a conocer a los estudiantes del Proyecto Curricular de Ingeniería Eléctrica los conceptos básicos e intermedios sobre los lenguajes de Programación Orientada a Objetos, utilizando

procesos metodológicos y modelados que le permitan ver al estudiante la importancia de estas estrategias en la solución de los problemas propios de la Ingeniería Eléctrica. De tal forma que el estudiante adquiera disciplina en lo relacionado con gestión del conocimiento del problema, modelado del mismo e implementación de una solución utilizando lenguajes de programación orientados a objetos, que le permitan capturar datos, provenientes de diferentes fuentes, realizando el almacenamiento en una base de datos para su posterior procesamiento con el fin de obtener información sobre las variables y el estado de un sistema eléctrico.

OBJETIVOS ESPECÍFICOS

- Al finalizar la materia el alumno estará en capacidad de:
- Entender y aplicar el proceso de generación y declaración de clases (atributos y métodos) que definan las representaciones y comportamiento de los objetos.
- Asimilar las ideas orientadas a objetos manipulando a través de aplicaciones sencillas las diferentes etapas de la Programación Orientada a Objetos.
- Explorar la estructura sintáctica de las funciones de un lenguaje de Programación Orientado a Objetos.
- Realizar comunicación entre objetos a través del envío de mensajes y conocer las propiedades de los objetos y los tipos abstractos de datos.
- Interactuar con requerimientos, modelado y codificación para la resolución de problemas de software.
- Tener presente las sentencias de control, sintaxis y la semántica básica en la Programación Orientada a Objetos.
- Generar aplicaciones con soporte de almacenamiento secundario tanto en archivos planos como en bases de datos.
- Realizar lectura y manejo de puertos de comunicaciones.
- Explorar los principios básicos de software para comunicaciones.
- Efectuar diseños de interfaz gráfica de usuario y lógica de aplicación.
- Efectuar el diseño e integración de una aplicación completa con soporte de bases de datos (persistencia)

COMPETENCIAS DE FORMACIÓN

Competencias de Contexto:

- Análisis, Diseño, Desarrollo de software y aplicaciones, en donde se debe analizar e implementar un programa aplicando Programación orientado a objetos, empleando el modelo de tres capas

Competencias Básicas:

- Conocer el tipo de problemas de desarrollo software que soluciona un uso correcto de la programación orientada a objetos.
- Conocer las capacidades de la programación orientada a objetos para la reutilización del software.
- Entender los conceptos de clase, atributo, operación, interfaz y objeto.
- Entender el mecanismo de paso de mensajes.
- Comprender el modo en que se deben implementar los caminos de comunicación entre clases para permitir el paso de mensajes entre ellas.
- Entender y ser capaz de implementar los distintos tipos de relaciones que se pueden establecer a nivel de objeto entre dos clases: asociaciones, agregaciones y composiciones.
- Entender el concepto de estado de un objeto.
- Entender la relación entre diagramas de clase y código de implementación de dichos diagramas.
- Entender el mecanismo de abstracción de la herencia.
- Plantear jerarquías de herencia bien definidas.
- Entender el concepto y la utilidad del polimorfismo.
- Deducir la relación a nivel de implementación entre herencia y polimorfismo.
- Saber identificar los distintos tipos de polimorfismo: sobrecarga, sobre escritura, variables

- Polimórficas y genericidad.
- Entender los mecanismos de gestión de errores que ofrecen algunos lenguajes de programación.
- Entender el concepto de concurrencia.
- Entender y aplicar el concepto de persistencia.
- Generar el almacenamiento de variables en una base de datos para su posterior procesamiento con el fin de obtener información para las diferentes partes de un sistema eléctrico.

Competencias Laborales:

PROGRAMA (UNIDADES TEMÁTICAS)

I. Repaso de programación básica:

- Constantes, tipos de datos.
- Operadores
- Tipos de estructuras de decisión, If-else, switch-case
- Tipos de estructuras de iteración: for, while, do-while.
- Funciones, tipos de funciones, retorno y parámetros.
- Manejo de excepciones
- Comentarios y documentación
- Metodología RUP
- Diagrama de Casos de Uso
- Diagrama de Clases
- Diagrama de Objetos
- Diagrama de secuencia y de actividades
- Diseño de interfaces graficas de usuario

II. Introducción a la Programación Orientada a Objetos

- Reseña histórica
- Características de la Programación Orientada a Objetos.
- Evolución de los lenguajes de programación.
- Tipos de lenguajes. Traductores: intérpretes y compiladores.
- Clases, objetos o instancias de una clase
- Métodos y atributos
- Definición, características y uso de constructores
- Tipos de constructores, por defecto y por argumento
- Abstracción y encapsulamiento
- Herencia y polimorfismo
- Superclases y subclases
- Herencia simple
- Interfaces
- Herencia múltiple mediante interfaces
- Sobrecarga de operadores
- Comportamiento polimórfico
- Ejercicios de aplicación e integración

III. Flujos, Arreglos, Colecciones y Algoritmos

- Visión general de los flujos de entrada y salida
- Definición, características y usos de arreglos
- Arreglos multidimensionales
- Operaciones básicas de colecciones
- Listas
- Algoritmos de colección
- Búsqueda y ordenamiento

- Algoritmos de búsqueda
- Algoritmos de ordenamiento

IV. Diseño de aplicaciones con arquitectura de tres capas: presentación (GUI), lógica de aplicación (Lógica del negocio) y persistencia (Bases de Datos).

- Diseño de Interfaz gráfica de Usuario - GUI
- Conceptos básicos de diseño y usabilidad
- Controles gráficos básicos
- Diseño de la lógica de aplicación – Aplicación de la metodología RUP, modelado estructural básico UML y de comportamiento básico UML
- Fundamentos de diseño de Bases de Datos
- Modelo entidad relación
- Modelo relacional
- Modelado de datos con UML
- Implementación de una base de datos.
- Conexión a bases de datos empleando ADO.net
- Captura de datos a través de Puertos de comunicación para su almacenamiento en una base de datos.
- Diseño de aplicaciones completas que incluyen bases de datos.

III. ESTRATEGIAS

Metodología Pedagógica y Didáctica

- Asistencia a clases expositivas y de discusión donde se expone de forma magistral las técnicas y metodologías de la Programación Orientada a Objetos
- Elaboración y lectura de paper (documentación).
- Se debe procurar incentivar el trabajo de grupo más que el trabajo individual. (se recomienda trabajar en grupos de dos o tres estudiantes)
- Laboratorios con guías que permitan evaluar cada uno de los contenidos presentados en el curso.
- Implementación y prueba de prototipos (programas) en laboratorio de computación.

	Horas			Horas profesor/ semana	Horas Estudiante/ semana	Horas Estudiante/ semestre	Créditos
Tipo de Curso	TD	TC	TA	(TD + TC)	(TD + TC+TA)	X 16 semanas	TD
Teórico	2	4	3	6	9	144	3

Trabajo Directo (TD): Trabajo de aula con plenaria de todos los estudiantes.
Trabajo Cooperativo (TC): Trabajo de tutoría del docente a pequeños grupos o de forma individual a los estudiantes.
Trabajo Autónomo (TA): Trabajo del estudiante sin presencia del docente, que se puede realizar en distintas instancias: en grupos de trabajo o en forma individual, en casa o en biblioteca, laboratorio, etc.

IV. RECURSOS

Medios y Ayudas

- Aula normal con pizarrón para sesiones de cátedra y para sesiones de discusión.
- Disponibilidad para acceder a proyector multimedia.
- Laboratorio de computación, con un computador por alumno, para las sesiones de laboratorio; cada computador debe contar con el intérprete para el lenguaje de programación que se va a utilizar para validar los prototipos.
- Aula virtual o página web para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.

- Acceso fuera de clases a laboratorios de computación que cuenten con el intérprete para el lenguaje de programación que se va a utilizar para validar los prototipos, y con acceso a la página web del módulo.
- Acceso al material bibliográfico recomendado
- Asignación de una persona que tenga las plenas competencias del curso (monitor) para asesorar a los estudiantes en dudas durante las sesiones del laboratorio de computación

Bibliografía

Textos Guías

- ARAGON MESA, Francisco. Introducción a la programación Orientada a Objetos: Java.
- BARNES, David J. Programación Orientada a Objetos: Una introducción práctica usando BlueJ. ED Prentice Hall.
- RUMBAUGH, James. Modelado y diseño Orientado a objetos. ED Prentice Hall
- JOYANES Luis. Programación Orientada a Objetos. ED Mc Graw Hill.
- BOUD, Timothy. Introducción a la Programación Orientada a Objetos. ED Addison Weasley.
- RUMBAUGH, James. BOOCH Gaddy. El lenguaje Unificado de Modelado. Manual de referencia. ED Addison-Wesley. Ed 2003
- CEBALLOS, Javier. Curso de Programación, Microsoft C#. ED Alfaomega. Ed 2007
- CEBALLOS, Javier. Visual C#, Enciclopedia de Microsoft. ED Alfaomega. Ed Segunda 2007.
- ARCHER Tom, WHITECHAPEL Andrew. Inside C#. ED Microsoft Press. Ed Segunda 2002

Textos Complementarios

Revistas

- Wei, L. (2008). A software environment for practical graph applications.
- Computer And Information Technology, 2008. Cit 2008. 8th Ieee International Conference On.
- Digital Object Identifier: 10.1109/CIT.2008.4594644. Page(s): 24 - 29
- Chunyan, Q. (2010). Uml - Based Software Process Modeling. Computer, Mechatronics, Control And Electronic Engineering (CMCE), 2010 International Conference. Volume: 1 Digital Object Identifier: 10.1109/CMCE.2010.5610482. Page(s): 247 – 250.
- Omar, N. ; Razik, N. (2008). A.Determining the basic elements of object oriented programming using natural language processing. Information Technology, 2008. ITSim 2008. International Symposium on
- Volume: 3 Digital Object Identifier: 10.1109/ITSIM.2008.4632009. Page(s): 1 – 6

Direcciones de Internet

- Tutoriales de Visual Studio 2012 y Access
- Manuales y ayudas de Rational Rose y Microsoft

V. ORGANIZACIÓN / TIEMPOS

Espacios, Tiempos, Agrupamientos

PROGRAMA SINTÉTICO	SEMANAS ACADÉMICAS															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Repaso de programación básica																
Introducción a la Programación Orientada a Objetos																
Flujos, Arreglos, Colecciones y Algoritmos																
Diseño de aplicaciones con arquitectura de tres capas: presentación (GUI), lógica de																

aplicación (Lógica del negocio) y persistencia (Bases de Datos).																

VI. EVALUACIÓN			
	TIPO DE EVALUACIÓN	**FECHA**	**PORCENTAJE**
PRIMER CORTE		Semana 8	
SEGUNDO CORTE		Semana 16	
EXAMEN FINAL		Semana 17 -18	

ASPECTOS A EVALUAR DEL CURSO

1. Claridad y entendimiento de los conceptos.
2. Que se haya identificado correctamente el problema y que el modelo lo represente adecuadamente.
3. Que la solución diseñada resuelva el problema.
4. Apego a la formalidad y estándares requeridos.
5. Que el análisis de corrección sea exhaustivo.
6. Que el prototipo corresponda al algoritmo diseñado y no presente errores de sintaxis.
7. La asistencia a las clases magistrales y a los laboratorios.
8. El esfuerzo y dedicación en la resolución de problemas.
9. Que la documentación permita reconocer la forma en que se ha abordado el problema y la estructura del programa implementado.
10. En las pruebas escritas se consideran en forma parcial los aspectos considerados en proyectos de programación bajo problemas que requieren un menor tiempo de desarrollo y en
11. una modalidad que no requiere uso del computador, así como la comprensión conceptual.

DATOS DEL PROFESOR

Nombre:	
Pregrado:	
Postgrado:	
Correo Electrónico:	