

Plan Anual de Actividades Académicas

CARRERA: TÉCNICO SUPERIOR EN SISTEMAS INFORMÁTICOS

ASIGNATURA: PROGRAMACIÓN AVANZADA 1

AÑO DE UBICACIÓN DEL CURSO	CUATRIMESTRAL
1 ^{er} (Primero)	1 ^{er} cuatrimestre

CONSTITUCIÓN DEL EQUIPO DOCENTE:		
APELLIDO Y NOMBRE	TÍTULO HABILITANTE	CARGO
GUCCIONE, Leonel	Ingeniero de Sistemas	Adjunto Interino Simple

Planificación de la asignatura

Fundamentación

La asignatura está ubicada dentro del área de la Programación. Es fundamental para el desarrollo profesional del egresado ya que brinda los conocimientos teórico-prácticos para la implementación de sistemas informáticos acordes con las exigencias actuales. En esta materia se enseña el mejor paradigma de programación disponible en este momento para enfrentar las cada vez más complejas tareas de la programación.

Objetivos

Capacitar al estudiante en la construcción de aplicaciones, usando el enfoque de objetos para su modelamiento y diseño, y para su desarrollo e implementación, un lenguaje de programación orientado a objetos.

Al finalizar la materia, el estudiante estará en capacidad de:

- Aplicar las técnicas y herramientas de desarrollo de un lenguaje orientado a objetos para la construcción de programas.
- Usar una metodología para el desarrollo de soluciones a problemas, mediante el enfoque de objetos, en su modelamiento y diseño, y la utilización de un lenguaje de programación orientado a objetos para su implementación.
- Analizar y explicar el comportamiento de programas sencillos, que involucren todas las construcciones de control fundamentales.
- Abordar un problema, descomponiéndolo en pasos elementales, para solucionarlo.
- Desarrollar algoritmos para resolver problemas sencillos. Implementar, verificar y depurar esos algoritmos en un lenguaje orientado a objetos.
- Conocer la filosofía de la programación orientada a objetos. Diseñar, codificar, verificar y depurar programas sencillos en un lenguaje orientado a objetos.
- Identificar las estructuras de datos básicas que sirven para representar información y discutir cuando se debe emplear cada una.
- Un estilo de programación que el estudiante deberá emplear de ahora en adelante en todas las demás asignaturas que requieran programación.

Contenidos

Los contenidos de esta materia Programación Avanzada 1 completamente relacionados con los contenidos de la materia Laboratorio V. En la materia Programación Avanzada 1 se brindan los contenidos teóricos y en Laboratorio V se desarrollan las guías de trabajos prácticos.

Unidad 1: Introducción a la Programación Orientada a Objetos [REF. 1, cap. 1]

Nuevo paradigma. Responsabilidades. Mensajes y métodos. Clases y ejemplares. Herencia. Mecanismos de abstracción.

Unidad 2: Clases y Objetos [REF. 1, cap. 2]

Clases. Campos (atributos). Variables de instancia y de clase. Control de acceso y herencia. Creación de Objetos. Constructores. Métodos. Parámetros. Métodos como control de acceso. Sobreescribiendo métodos. Miembros "static" (campos, bloques, métodos). Colector de basura y "finalize". Main. Método toString. Métodos nativos.

Unidad 3: Extendiendo Clases [REF. 1, cap. 3]

Una clase extendida. Miembro de clase “protected”. Constructores. Sobreescribiendo métodos y ocultando campos. La palabra “super”. Haciendo métodos y clases “final”. La clase Object. Clases y métodos abstractos. Clonando objetos. Cómo y cuando extender clases. Diseñando una clase para que sea extensible.

Unidad 4: Interfaces [REF. 1, cap. 4]

Herencia simple versus múltiple. Extendiendo “interfaces”. Conflicto de nombres. Implementando “interfaces”.

Unidad 5: Clases Colección, clases genéricas [REF. 1, cap. 6, cap 11, cap 21]

Clases especiales para el manejo de colección de datos: arreglo, conjunto, lista, enumerado, tabla de hashing, etc.

Clases cuyos atributos son de un tipo de datos genérico que se define en el momento de la instanciación.

Unidad 6: Excepciones [REF. 1, cap. 12]

Creando tipos excepción. Throw. Throws. Try, catch y finally. Cuándo usar excepciones.

Unidad 7: Threads [REF. 1, cap. 14]

Creación de Threads. Sincronización. Comunicación entre threads. Wait y Notify. Semaforos. Suspendiendo Threads. Interrumpiendo Threads. Finalizando la ejecución de un Thread. La interfaz Runnable.

Unidad 8: JDBC [REF. 2]

Acceso a Bases de Datos. Seleccionar una base de datos. Establecer una conexión, cargar los drivers, hacer la conexión. Seleccionar una tabla. Crear sentencias JDBC. Ejecutar sentencias. Introducir datos en una tabla. Obtener datos desde una tabla. Recuperar valores desde una hoja de resultados. Actualizar tablas. Utilizar uniones. Utilizar transacciones. Procedimientos almacenados. Crear aplicaciones JDBC completas

Unidad 9: MVC [REF. 1, cap. 22]

Arquitectura Modelo – Vista – Controlador. Interfaz Observer. Clase Observable.

Unidad 10: Introducción al uso de formularios [REF. 3, cap. 13]

Uso de los componentes básicos de un formulario. La clase Swing. Programación de eventos.

Bibliografía

Bibliografía básica

- Referencia 1: Gosling, “El lenguaje de Programación Java”

Bibliografía complementaria

- Referencia 2: Leonel Guccione, “Apuntes de cátedra”
- Referencia 3: Bruce Eckel, “Pensando en Java”

Metodología de Trabajo

Forma de trabajo

1. Resolución de algoritmos (escritos en papel y lápiz) de creciente nivel de complejidad.

2. Elaboración de programas en computadora de creciente nivel de complejidad.
3. Participación activa de las clases orales, respondiendo a las preguntas del docente y preguntando cada vez que no comprenda los conceptos vertidos.
4. Participación en grupos de investigación y ejercitación.
5. Resolución de las evaluaciones propuestas por el docente.
6. Diálogo e intercambio de opiniones.

Procesos de intervención pedagógica.

- Clases teórico prácticas que comprenden lo siguiente puntos:
 - Organización de los contenidos curriculares y los recursos utilizados para cada clase.
 - Alineamiento de la enseñanza, o sea, relacionar el objetivo de la clase con las actividades llevadas a cabo.
 - Elaborar el foco introductorio y el sensorial para cada clase. O sea, establecer la forma de captar la atención de los alumnos.
 - Retroalimentación. Interacción continua con los alumnos valorizando cada uno de sus aportes.
 - Monitoreo de los logros de los alumnos y de sus conductas, para valorar el progreso de su aprendizaje.
 - Revisión y cierre. Finalización de la clase con una síntesis de lo aprendido y una conexión con el siguiente tema a tratar.
 - Indagación mediante preguntas, que permitan a los alumnos reforzar lo aprendido y permitir la interacción de aquellos que son menos participativos.

Evaluación

Forma de evaluación

Se establecen dos evaluaciones parciales con sus respectivos recuperatorios para la aprobación de la cursada y un examen final para la aprobación de la materia. Se permite rendir un segundo recuperatorio en la misma fecha del segundo parcial para quienes desaprobaban la primera instancia (recuperatorio del primer parcial).

La primera evaluación comprende las primera cinco unidades y la segunda evaluación las restantes.

Se establece una clase de revisión integradora antes de cada examen parcial y una clase de muestra y realización de la evaluación después de cada examen parcial, para que el alumno vea sus errores y se oriente en el proceso de aprendizaje.

Para aprobar la materia se debe aprobar un examen final.

Las fechas de los exámenes parciales y sus recuperatorios se muestran en el cronograma estimado de clases.

Criterio de evaluación

Como criterio de evaluación, se tiene en cuenta la correcta resolución de las preguntas del examen. Para aprobar se deben responder correctamente al menos el 50% de las preguntas. También se considera el grado de participación en clase.

Descripción de las situaciones de pruebas a utilizar para la evaluación continua y final

- Cuestionarios

- Respuesta múltiple
- Resolución de problemas.

Recursos didácticos a utilizar como apoyo a la enseñanza.

Se utilizarán los recursos didácticos con los que cuenta la institución, de la siguiente forma:

▪ **Cañón para proyección**

Se utiliza para visualizar el material elaborado para las clases teóricas y mostrar la resolución de problemas. También para las clases de revisión y consulta grupal. En general su uso será con una frecuencia de una vez por semana.

▪ **Sala de informática con acceso a Internet**

Se utiliza para la resolución de ejercicios, elaboración de proyectos especiales y tareas de investigación.

Su uso será en cada clase.

▪ **Blog**

Se utiliza como medio de difusión del material teórico y ejemplos utilizados en la materia. También para comunicar al alumnado la fecha de los exámenes y cualquier otra novedad que sea necesaria.

Cronograma estimado de clases.

Las clases se dictan dos veces por semana con una duración de 2 hs cada una.

Clase 1: Introducción a la programación orientada a objetos.

Clase 2: Resolución de ejemplos en forma grupal.

Clase 3: Extensión de clases (herencia).

Clase 4: Clases abstractas. Clone().

Clase 5: Interfases.

Clase 6: Resolución de ejemplos en forma grupal.

Clase 7: La clase Lista ligada.

Clase 8: Clases genéricas. Clases de colección (Vector, Enumeration, Hashtable).

Clase 9: Excepciones.

Clase 10: Resolución de ejemplos en forma grupal.

Clase 11: Repaso para el primer parcial.

Clase 12: Primer parcial.

Clase 13: Programación concurrente. Creación de Threads. Sincronización.

Clase 14: Ejemplos de programación concurrente.

Clase 15: Comunicación entre threads. Wait y Notify. Semáforos.

Clase 16: Ejemplos de comunicación entre threads.

Clase 17: Resolución grupal del primer parcial y consulta para el recuperatorio.

Clase 18: Recuperatorio del primer parcial.

Clase 19: Model View Controller. El patrón Observer Observable.

Clase 20: Resolución de ejemplos en forma grupal.

Clase 21: Persistencia usando JDBC.

Clase 22: Resolución de ejemplos en forma grupal.

Clase 23: Uso de formularios.

Clase 24: Resolución de ejemplos en forma grupal.

Clase 25: Repaso para el segundo parcial.

Clase 26: Segundo parcial.

Clase 27: Revisión del segundo parcial.

Clase 28: Recuperatorio del segundo parcial

1	Jue 25/Mar 10
2	Vie 26/Mar 10
3	Jue 08/Abr 10
4	Vie 09/Abr 10
5	Jue 15/Abr 10
6	Vie 16/Abr 10
7	Jue 22/Abr 10
8	Vie 23/Abr 10
9	Jue 29/Abr 10
10	Vie 30/Abr 10
11	Jue 06/May 10
12	Vie 07/May 10
13	Jue 13/May 10
14	Vie 14/May 10
15	Jue 20/May 10
16	Vie 21/May 10
17	Jue 27/May 10
18	Vie 28/May 10
19	Jue 03/Jun 10
20	Vie 04/Jun 10
21	Jue 10/Jun 10
22	Vie 11/Jun 10
23	Jue 17/Jun 10
24	Vie 18/Jun 10
25	Jue 24/Jun 10
26	Vie 25/Jun 10
27	Jue 01/Jul 10
28	Vie 02/Jul 10