

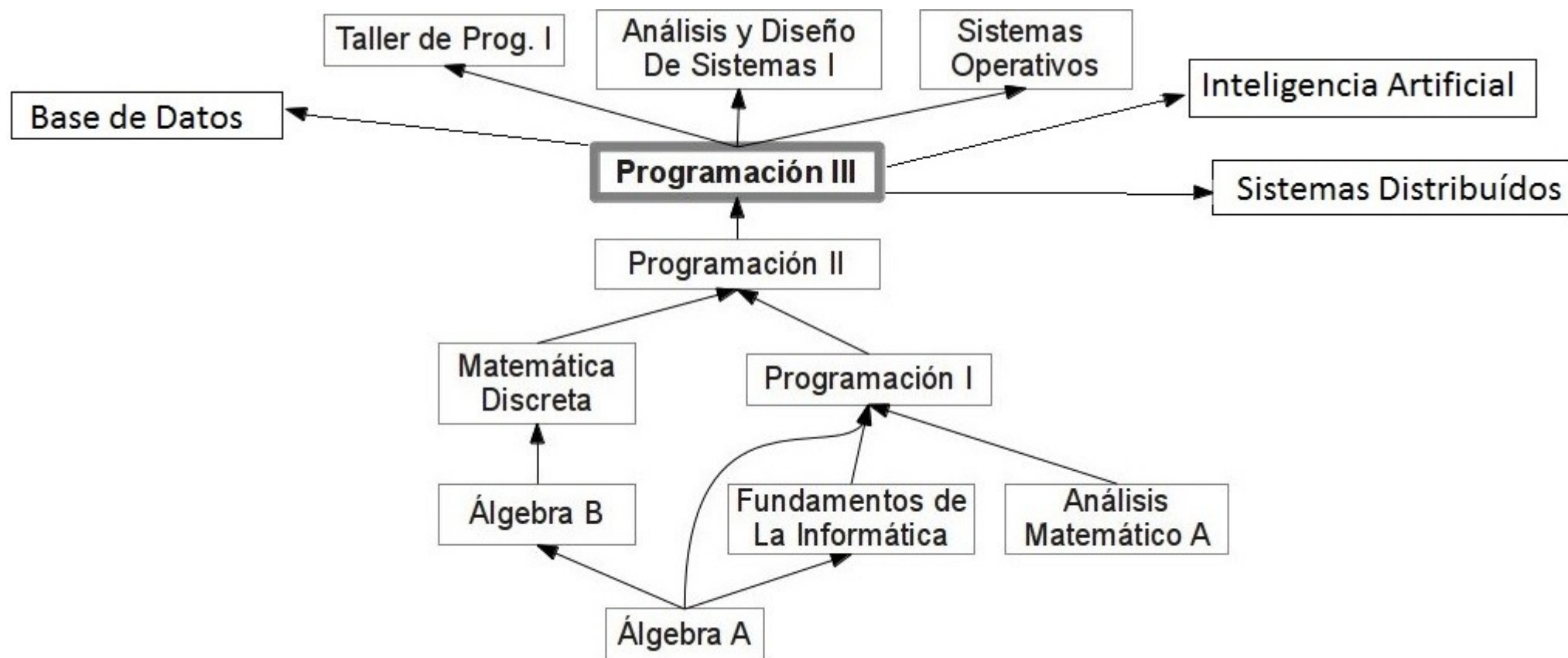
Programación III

```

public class vehiculo {
    String patente;
    int velocidad;
    public void VerDatos()
    {
        System.out.println("Patente: "+patente);
        System.out.println("Velocidad: "+velocidad);
    }
}

```





La Cátedra

Composición de la Cátedra

- Profesor Adjunto: Ing. Leonel Guccione.
- Jefe de Trabajos Prácticos: Gellon Ivonne.
- Ayudante Graduado: Prof. Guillermo Lazzurri.

CAMPUS



URL del Campus:

[https://ingenieria.campus.mdp.edu.ar/course/
view.php?id=1264](https://ingenieria.campus.mdp.edu.ar/course/view.php?id=1264)

Programa de la Asignatura

Unidad 1: Introducción a la Programación Orientada a Objetos

Introducción a la Programación Orientada a Objetos. Principales características del paradigma. Concepto de Objeto, Instancia, Clase, Método y Mensaje. Herramientas de desarrollo. Programación dinámica e interactiva. Técnicas de prueba y depuración.

Unidad 2: Objetos y Clases

Jerarquía de Clases. Variables de Clase. Inicialización. Concepto de Metaclasses. Concepto de Encapsulamiento. Ocultamiento de la información.

Propiedades de los objetos. Encapsulamiento. Ocultamiento de la Información. Modos de acceso. Métodos como control de acceso. Acceso a los atributos privados.

El problema del atributo público. Control de acceso por clase. Métodos getXXX() y setXXX(). Sobrecarga de métodos. Miembros “static” (atributos, métodos). Constantes. Constantes primitivos. Constantes con referencia a objeto. Constantes – cambiar el estado del objeto referido. Colector de basura y “finalize”. La relación de asociación: agregación y composición. Diferencias entre agregación y composición

Unidad 3: Reutilización

Concepto de Herencia. Clases Abstractas y Concretas. Redefinición. Concepto de Delegación. Herencia vs. Delegación. Concepto de Patrón de diseño (Design Pattern). Ejemplos de Patrones. Concepto de Framework.

Programa de la Asignatura

Unidad 4: Polimorfismo

Concepto de Polimorfismo. Concepto de enlace tardío (Late Binding). Concepto de doble envío. (Double Dispatch). Implementación de polimorfismo.

Unidad 5: Colecciones

Protocolos de Colecciones. Adición, inserción y eliminación de elementos. Clases de colecciones. Secuenciales y no secuenciales. Diccionarios. Conjuntos. Comparación. Ordenamiento. Conversión entre clases de colecciones.

Unidad 6: Excepciones

Excepciones. Definición. Buenas prácticas. Por qué usar excepciones. Qué es una excepción. Jerarquía de excepciones. Esquema básico en el uso de excepciones. Excepciones más específicas. Cómo lanzar excepciones? Flujo de control. Qué hacer cuando un método lanza una excepción? Cláusula throws y redefinición de métodos. Uso de finally. Cómo crear nuevas excepciones?

Programa de la Asignatura

Unidad 7: Diseño de Interfaces gráficas de Usuario

Interacción humano-computadora (HCI). Concepto de usabilidad. Interacción y manejo de Eventos. Introducción a los gráficos en computadora. Conceptos geométricos, estéticos y de usabilidad.

Unidad 8: Introducción al Diseño Orientado a Objetos

Documentación. Introducción a UML (Unified Modelling Language). Diagrama de Clases. Casos de Uso. Técnicas de Diseño Orientado a Objetos. CRC (Classes-Responsibilities-Collaborations). TDD (Test Driven Design)

Unidad 9: Eventos y Concurrency

Concepto de Proceso. Múltiples procesos independientes. Planificación. Prioridades. Semáforos. Exclusión mutua. Recursos compartidos. Interrupciones de Hardware.

Forma de Trabajo



Las **clases teóricas** serán de tipo ***Centrada en el Profesor - Clase Expositivas***, con algunas adaptaciones. Se utilizará el espacio para que el docente presente y explique cada tema, sintetizando la información que el alumno necesita para comprender los conceptos abordados.

Las **clases teórico-prácticas** serán de tipo ***Centrada en el Profesor - Demostración***, con algunas adaptaciones. Se utilizará el espacio para que el docente desarrolle un ejemplo completo y fundamentado que sirva de modelo al alumno para el trabajo de resolución de ejercicios, haciendo énfasis en el proceso de razonamiento requerido para lograr la solución. ??

Las **clases prácticas** serán de tipo ***Centrada en el Alumno - Aprendizaje Basado en Problemas***, con algunas adaptaciones. Se utilizará el espacio para que el alumno resuelva problemas prácticos y desarrolle proyectos a mediano plazo de creciente complejidad, en donde deberá consultar el material necesario que le permita solucionar el problema planteado, elaborar los algoritmos/programas requeridos utilizando el lenguaje de programación propuesto. El alumno trabajará en algunos casos en forma individual y en otros en forma grupal.



Evaluación

Requisitos de aprobación

Los conocimientos de los alumnos serán evaluados a través de 2 exámenes y la presentación y aprobación de un Trabajo Final (TF) en equipos. El Trabajo Final consiste en el desarrollo de un software.

La aprobación de cada examen se puede lograr en la instancia del parcial o de recuperatorio.

Examen	Instancia	Quiénes lo rinden	Condiciones particulares
Examen 1	Parcial 1	Todos los alumnos	
	Recup 1	Quienes desaprobaron el Parcial 1	
Examen 2	Parcial 2	Todos los alumnos que aprobaron Examen 1	
	Recup 2	Quienes rindieron y desaprobaron el Parcial 2	
Totalizador		Los alumnos habilitados	TF aprobado

Evaluación

Régimen de Promoción

El alumno rinde dos Exámenes y un Trabajo Final (TF). Nota mínima de aprobación: 5, para cada una de las instancias de evaluación y PF.

Al final del cuatrimestre se calcula la nota de la cursada de la siguiente forma:

$$\text{Nota Cursada} = \frac{\frac{\text{Examen 1} + \text{Examen 2}}{2} + \text{Trabajo Final}}{2}$$

Condición	Descripción	Consecuencia
Desaprobado	Si $0 \leq \text{Nota Cursada} < 5$	Recurso
Habilitado	Si $5 \leq \text{Nota Cursada} < 7$ y todas las notas de Exámenes y TF mayores o iguales a 5	Rinde Totalizador
Promocionado	Si $7 \leq \text{Nota Cursada} \leq 10$ y todas las notas de Exámenes y TF mayores o iguales a 5	
Todo otro caso particular no contemplado, será analizado por la cátedra		

Evaluación

En cuanto al Trabajo Final

El Trabajo Final deberá ser aprobado con nota ≥ 5 . El alumno deberá entregar el proyecto a término y defenderlo frente a un docente (por ejemplo: responder preguntas, hacer modificaciones en el código, etc.).

Evaluación

Criterios de aprobación del trabajo final (TF)

La calificación de este trabajo es individual.

En el momento de la defensa del TF, cada integrante del grupo deberá:

- Presentar un trabajo que cumpla con todas las consignas solicitadas. Sin errores de compilación ni ejecución.
- En la división de tareas ser participe activo tanto en la programación como en el diseño.
- Aplicar los contenidos teóricos y prácticos vistos durante la cursada.
- Investigar temas necesarios para el desarrollo de la aplicación disponibles en Internet.
 - Demostrar dominio individual del código desarrollado
 - explicar en forma individual el código desarrollado,
- modificar parte del código a solicitud del docente
- Respetar el aporte de cada uno de los integrantes del grupo.
- Usar de manera equitativa el tiempo asignado con sus compañeros de grupo.
- Usar el lenguaje tecnológico apropiado durante la presentación individual de la aplicación.

Durante el desarrollo del trabajo, el alumno deberá:

- Mostrar avances del trabajo a pedido de los docentes.

Cronograma de actividades