

**Facultad de Ingeniería**  
**Escuela de Informática y Telecomunicaciones**

**PROGRAMA DE ASIGNATURA**  
*Programación avanzada*

**I. Identificación**

Código	: CIT-1010
Créditos	: 6
Duración	: Semestral
Ubicación en plan de estudio	: Semestre 2
Requisitos	: Programación (CIT-1000)
Sesiones semanales	: 2 cátedras, 1 ayudantía

**II. Objetivos Generales y Específicos**

El objetivo general del curso se relaciona con adquirir conceptos y elementos avanzados de programación que permitan usar el computador para resolver problemas.

Además, al finalizar el curso el alumno deberá ser capaz de:

- Utilizar el pensamiento algorítmico para resolver un problema planteado.
- Aprender los fundamentos y conceptos básicos de la programación orientada a objetos.
- Desarrollar capacidades de abstracción y modelamiento utilizando estructuras de almacenamiento avanzadas y paradigmas de programación modernos.

**III. Descripción de Contenidos.**

1. **Revisión de elementos de programación:** Elementos básicos de un lenguaje de programación, estructuras de control, funciones, paso de parámetros, funciones del lenguaje de programación, arreglos, búsqueda, archivos secuenciales.
2. **Orientación a objetos**
  - 2.1. **Introducción a objetos:** Definición y creación de objetos, atributos y ámbitos de estos (públicos y privados). Constructores y métodos.
    - 2.1.1. Conocer la sintaxis de definición de clases, atributos y métodos.
    - 2.1.2. Aplicar el concepto de clases para el modelado y creación de objetos.
    - 2.1.3. Comprender los ámbitos de llamado de métodos y atributos según el ámbito de acceso a estos.
    - 2.1.4. Transferencia de conocimiento a elaboración de modelado más complejo mediante ejemplos y ejercitación.
  - 2.2. **Manejo de instancias:** Creación de múltiples instancias de una clase, interacción entre objetos, objetos como valores de retorno, objetos como parámetros.

- 2.2.1. Comprender la diferencia entre distintas instancias de una misma clase.
  - 2.2.2. Comprender el mecanismo de acceso a métodos y atributos de objetos y la interacción entre ellos.
  - 2.2.3. Comprender el uso y aplicación de objetos como parámetros y valores de retorno.
  - 2.2.4. Transferencia de conocimiento a elaboración de modelado más complejo.
- 2.3. **Herencia y polimorfismo:** Concepto de herencia y polimorfismo, casting y sobrecarga de métodos.
- 2.3.1. Comprender el concepto e implicancias de herencia en el modelado de soluciones.
  - 2.3.2. Conocer la sintaxis de casting y sobrecarga de métodos.
  - 2.3.3. Transferencia de conocimiento a elaboración de modelado más complejo.
3. **Modelado utilizando contenedores de información:** Manejo de tipos de datos abstractos: vector, lista, cola, cola de prioridad, pila, diccionario, templates.
- 3.1. Conocer estructuras de almacenamiento y modelado avanzado.
  - 3.2. Modelado y métricas de eficiencia en base a las herramientas utilizadas.
  - 3.3. Comprender las ventajas y desventajas de la utilización de distintas estructuras de almacenamiento en términos de uso de memoria y tiempos de operación.
  - 3.4. Resolver problemas contextualizados simples de la vida cotidiana, utilizando las herramientas del curso.

#### **IV. Importancia del curso en el plan de estudios**

La importancia en este curso radica en permitir a los alumnos desarrollar el bagaje de conocimientos necesario para comprender y, eventualmente, utilizar las estrategias metodológicas más relevantes para el uso de los lenguajes de programación como herramienta de trabajo.

Así, este curso contribuye al cumplimiento del perfil de egreso a través del desarrollo del siguiente conjunto de objetivos de aprendizaje (vistos como una serie de conocimientos, habilidades, actitudes y valores):

- Modelar el comportamiento de sistemas, empleando lenguaje matemático, conceptos de física, lenguaje computacional y simulación, entre otros métodos.
- Elaborar y adaptar diseños en el campo de las Tecnologías de la Información y de las Comunicaciones, que permitan satisfacer necesidades detectadas mediante el diagnóstico y la modelación.
- Contribuir al mejoramiento de la calidad de los procesos de producción de software y/o de diseño de sistemas de telecomunicaciones.

- Comunicar ideas en forma oral y escrita
- Capacidad de pensar en forma analítica y racional
- Habilidad de procesar datos generados experimentalmente
- Capacidad de abstracción y modelación
- Habilidad de identificar, formular y resolver problemas complejos de forma autónoma, con enfoque sistémico
- Capacidad de integrar conocimientos
- Capacidad de aprender en forma autónoma y continua
- Capacidad de adaptarse a nuevas situaciones
- Capacidad de crítica y autocrítica.
- Motivación al logro y a la calidad
- Ética profesional acorde con los valores de la Universidad

#### **V. Metodología**

Se contempla la realización de 2 sesiones de cátedra en aula de clases. Además se dispone de tutoría en Laboratorio, donde se aplicará la técnica de estudio de casos, en las que las materias del curso se irán desarrollando en base a los ejemplos aplicados que en cada sesión se planteen.

El curso será apoyado con apuntes de clases almacenados en un servidor para acceso vía Internet, en el que se describirán las materias y los ejercicios que deberán realizarse clase a clase.

Puesto que esta es una asignatura con una orientación práctica, se ha contemplado que la evaluación del curso considere los desarrollos realizados por el alumno sesión a sesión.

#### **VI. Evaluación**

Se contempla la realización de evaluaciones parciales (controles, trabajos, prácticas de laboratorio, entre otros), dos pruebas solemnes de igual ponderación y un examen.

Nota de Presentación =  $(25\% \text{ Solemne 1} + 25\% \text{ Solemne 2} + 20\% \text{ Notas Parciales})/0.7$

Nota Final =  $25\% \text{ Solemne 1} + 25\% \text{ Solemne 2} + 20\% \text{ Notas Parciales} + 30\% \text{ Examen}$ .

Podrán eximirse aquellos alumnos cuyos promedios de notas parciales y de pruebas solemnes sean superiores a 5.0.

## **VII. Bibliografía básica de referencia**

### Bibliografía obligatoria

1. The C++ Programming Language, special edition, 2000, Bjarne Stroustrup.
2. C++: Manual de referencias con anotaciones. 1994, Margaret Ellis.
3. Programación en C++: Un enfoque práctico. 2006, Luis Aguilar.
4. Programación orientada a objetos, 1998, Luis Aguilar.
5. Apuntes de Cátedra.

#### **PAUTAS ETICAS BASICAS**

*El plagio es el uso de las ideas o trabajo de otra persona sin el adecuado consentimiento. El plagio puede ser intencional o no. El plagio intencional es el claro intento de hacer pasar el trabajo o ideas ajenas como el suyo propio para su beneficio. El plagio no intencional puede ocurrir si Ud. no conoce el mecanismo adecuado de referenciar la fuente de sus ideas e información. Si no está seguro de los métodos aceptados para referenciar, debería consultar con su profesor, tutor o personal de biblioteca.*

*El plagio comprobado es una actitud que puede resultar en severas sanciones disciplinarias y/o en la exclusión de la Universidad (Artículo 44, Reglamento del Estudiante de Pregrado).*

Elaborado por: Jonathan Frez, Roberto Konow.  
Fecha revisión: Enero 2016.  
Fecha vigencia: Agosto 2016.