AÑO ACADÉMICO 2022

DEPARTAMENTO: **MATEMÁTICA**

PROGRAMA DE CÁTEDRA: **MÉTODOS COMPUTACIONALES EN INGENIERÍA I**

OBLIGATORIA/OPTATIVA: **OBLIGATORIA**

CARRERA/S: **ING. ELECTRÓNICA, ING. MECÁNICA, ING. ELÉCTRICA, ING. CIVIL, ING. EN PETROLEO e ING. QUÍMICA.**

ÁREA: **CIENCIAS BÁSICAS**

ORIENTACIÓN: -

PLAN DE ESTUDIOS – ORD. Nº:
Ing. Mecánica: 806/97,510/00, 536/00,76/05, 1046/05,1069/06, 192/10, 1537/14 y 844/21
Ing. Eléctrica: 807/97,482/04, 1069/06, 1537/14, 1578/14 y 993/22
Ing. Civil: 805/97,536/00,1069/06, 0192/10, 1537/14 y 853/21
Ing. En Petróleo: 804/97, 534/00, 536/00,940/01,1069/06, 192/10, 1537/14 y 845/21
Ing. Química: 803/97,536/00,1069/06, 0192/10, 1563/14, 1537/14, 0436/15, 762/20 y 868/21
Ing. Electrónica: 802/97, 536/00, 481/04,1005/05, 1069/06, 1537/14, 977/17 y 869/21

CARGA HORARIA SEMANAL: **6HS**

CARGA HORARIA TOTAL: **96HS**

RÉGIMEN: ***CUATRIMESTRAL***

CUATRIMESTRE: ***PRIMERO***

OBLIGATORIA

EQUIPO DE CÁTEDRA:

Apellido y Nombres	Cargo
CEJAS BOLECEK, NÉSTOR RENÉ	**PAD-3**
VILUGRON, RENÉ MARTÍN	**AYP-3**

ASIGNATURAS CORRELATIVAS (*S/Plan de Estudios*):

REGULARIZADA:

ÁLGEBRA Y GEOMETRÍA II.

* La materia que se exige regularizada para cursar deberá estar aprobada para **aprobar o promocionar** MÉTODOS COMPUTACIONALES EN INGENIERÍA I.

1. FUNDAMENTACIÓN:

El modelado computacional de sistemas como los de las organizaciones (industrias, universidades, gobiernos, proyectos y otros) y los de ingeniería (sistemas o

productos físicos que requieran de habilidades para aplicar principios matemáticos y científicos al diseño orientado a desarrollar soluciones) permite obtener una descripción estructural y dinámica de su comportamiento. Estas herramientas resultan de utilidad en la toma de decisiones basadas en la optimización, calidad y reducción de costos. En el ámbito de la ingeniería o de la medicina se utilizan en el diseño, el análisis y en la resolución problemas de optimización. La realidad virtual es otro ámbito de desarrollo actual para el modelado y estudio de sistemas y consiste en generar ambientes virtuales con la computadora para cuyas situaciones reales análogas serían muy riesgosas o de un elevado costo. Ejemplos de estos son la cirugía virtual para capacitación médica o la simulación de diversas condiciones de un choque de un automóvil para la optimización de la seguridad.

El presente es un curso introductorio al campo de la informática que aborda la comprensión y el análisis de sistemas desde un punto de vista general y en donde desarrollamos elementos de la programación para implementar diseños de sistemas. Comenzamos haciendo una revisión de las ideas más revolucionarias que llevaron a la creación de la computadora digital que hoy conocemos. Varias de estas ideas permitieron realizar los avances más importantes en el campo de las Neurociencias, la Biología molecular, las Telecomunicaciones y también en el desarrollo de nuevas tecnologías que constantemente cambian nuestro modo de percibir la realidad. Las unidades siguientes tienen como objetivo desarrollar y analizar el concepto de sistemas desde un punto de vista del diseño. Además, se abordarán conceptos de programación desde el punto de vista del ciclo de desarrollo de software y práctico. Esto le permitirá al estudiante diseñar y escribir programas que puedan representar el comportamiento de algún sistema. Finalmente, en la última unidad se estudian la organización estructurada de las computadoras digitales y los componentes básicos de un sistema operativo, particularizando estos conceptos al sistema operativo LINUX.

Las herramientas y conocimientos computacionales del curso le permitirán al estudiante abordar otros lenguajes de programación que complementado con cursos adecuados, le permitirán ampliar sus conocimientos en temas más avanzados y de gran interés actual como: el cálculo numérico, la simulación, la computación visual y la inteligencia artificial, entre algunos de los más relevantes.

2. OBJETIVOS:

Se pretende que el estudiante:

- Comprenda el valor y la importancia de la información y las tecnologías en nuestra sociedad.
- Adquiera conocimientos básicos para evaluar un sistema general mediante la utilización de diagramas UML: componentes básicos, criticidad, modularidad y testeo.
- Adquiera habilidades básicas de programación: ciclo de desarrollo del software y el uso de un lenguaje base (*Python*) de alto nivel.
- Utilice *git* como herramienta colaborativas en línea.
- Aprenda a diseñar un sistema y reproducir su comportamiento con un programa de computadora.

- Adquiera conocimientos de la organización estructurada de las computadoras con la finalidad de comprender cómo se llevan a cabo las operaciones de cálculo y el procesamiento de la información.
- Adquiera conocimientos de los componentes básicos de un sistema operativo para mejorar el uso de la computadora como herramienta de trabajo y como ejemplo de un sistema informático complejo.

3. CONTENIDOS SEGÚN PLAN DE ESTUDIOS:

Introducción a la informática. Arquitectura de computadoras. Nociones de sistemas. Sistemas operativos. Introducción a la programación. Análisis de sistemas. Técnicas de programación. Lenguaje de alto nivel.

4. CONTENIDO PROGRAMA ANALÍTICO:

Unidad 1: Introducción a la Informática

Introducción a la informática. Representación de la información. Breve descripción de la teoría de la información de Shannon. Sistemas de representación de la información. Análogo vs. digital. Definición de Tecnología. Evolución de las computadoras digitales. Ley de Moore. Conceptos básicos de las nuevas tecnologías de: Redes sociales, Blockchain, Inteligencia Artificial, Bots, Realidad Virtual y Aumentada y almacenamiento de información en la nube.

Unidad 2: Introducción a la programación

Definición de cómputo, instrucción y programa. Lenguajes compilados e interpretados. Estructuras de datos. Control de flujo y Ramificación. Variables. Diagrama de actividades (UML) de un programa. Aspectos generales de los lenguajes de programación.

Unidad 3: Introducción al lenguaje de programación de Python

Introducción a Python. Objetos y tipos de variables. Conversiones y Casteo. Operaciones simples. Control de flujo y ramificación. Descomposición, abstracción y funciones. Nociones de programación orientada a objetos, Clases, Polimorfismo, Herencia. Testeo y debugging de código. Empleo de git como software de control de versiones.

Unidad 4: Introducción a los sistemas

Definición y nociones de sistemas. Componentes básicos, criticidad, modularidad y testeo. Ciclo de desarrollo del software. Nociones de diagramas UML. Diseño de sistemas.

Unidad 5: Arquitectura de las computadoras y sistemas operativos

Introducción a la Organización estructuradas de las computadoras. Unidad Central de Procesamiento: organización y funcionamiento. Ejecución de instrucciones. Paralelismo y multiprocesos. Memoria y direcciones. Ordenamiento de bytes: Little Endian, Big Endian. Jerarquía de memorias y tiempos de acceso. Sistemas operativos: procesos y threads, memoria, sistema de archivos, entras y salidas de datos, virtualización y la nube. Caso de estudio: LINUX.

5. BIBLIOGRAFÍA BÁSICA Y DE CONSULTA:

1) Tanenbaum, Andrew S. Organización de computadoras: un enfoque estructurado. Pearson educación, 2000. ISBN-10: 9701703995

2) Abraham Silberschatz, Greg Gagne, Peter B. Galvin - Operating System Concepts-Wiley (2018).
3) McQuaid, Mike. *Git in practice*. Manning, 2015.
4) Sylvia Langfield, Dave Duddell - Cambridge International AS and A Level Computer Science Coursebook - Cambridge University Press (2016).
5) Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel. UML at Classroom: An Introduction to Object-Oriented Modeling. Springer (2014)

Se sugiere la utilización de la bibliografía mencionada y se trabajará con guías de trabajos prácticos y lecturas adicionales.

6. PROPUESTA METODOLÓGICA:

Los temas de la materia se desarrollarán tanto en el aula, donde se dará un abordaje teórico y práctico, y en el laboratorio de informática, donde se brindarán las herramientas computacionales y los trabajos prácticos a resolver. La primera unidad está focalizada en el desarrollo del pensamiento crítico del estudiante orientado a cuestionar constantemente las nuevas tecnologías de la informática. Los estudiantes deberán elaborar un trabajo de investigación sobre una de estas tecnologías y evaluar sus ventajas y desventajas, haciendo hincapié en la privacidad de los datos y las consecuencias transformadoras de la tecnología desde un punto de vista socio-económico. En la segunda unidad se desarrollarán los conceptos de sistemas y se emplearán diagramas UML para el diseño de algún sistema. En las unidades 2 y 3 se orientarán las prácticas en el desarrollo del pensamiento computacional del estudiante, esencial para conseguir las habilidades de programación que se requieren en el curso. Los programas se escribirán en el lenguaje de programación interpretado de alto nivel *Python*. Este lenguaje provee diversas bibliotecas de cálculo numérico, procesamiento de la información y visualización y ha sido ampliamente adoptado en diversas universidades y en en la industria del software. En la práctica los estudiantes tendrán la experiencia de utilizar el software de control de versiones, denominado git, con el fin de que adquieran experiencia en la utilización de herramientas colaborativas en línea empleadas actualmente en la industria del software. Para ello se propondrán actividades de elaboración de código en forma grupal y remota con git.

La estructura general de las clases tendrá el siguiente esquema:
1) Introducción de conceptos fundamentales por parte del docente
2) Espacio para realizar consultas, conjeturas y responder dudas acerca de la temática presentada o de las herramientas de programación, según corresponda.
3) Asignación para la clase siguiente de material bibliográfico y ejercicios prácticos relacionados con los conceptos desarrollados.

Unidad 1:
4) Presentación y discusión de algún caso concreto relacionado a las nuevas tecnologías que involucre algunos de los conceptos explicados
5) Selección de una temática por el estudiante, búsqueda bibliográfica, análisis y elaboración de una presentación para exponer oralmente en clase y discutir.

Unidades 2 y 3 (Programación):
8) Presentación de herramientas de programación y orientación en el pensamiento computacional.
9) Profundización y resolución de los ejercicios en computadora por parte de los estudiantes en forma individual y grupal.
10) Utilización de la herramienta de control de versiones git para realizar proyectos colaborativos simultáneamente.

Unidad 4 (Diseño de sistemas):
6) Presentación de herramientas de diseño (UML) y orientación en el pensamiento sistémico.
7) Realización de ejercicios que involucren el diseño, análisis y estudio de criticidad de sistemas con diagramas UML.

Unidad 5 (integración de conceptos y profundización del funcionamiento de las computadoras):
11) En la etapa final de la cursada se solicitará a los estudiantes la elaboración de un proyecto integrador que consiste en la implementación con el lenguaje Python del comportamiento de un sistema de interés diseñado por ellos mismos.

7. EVALUACIÓN Y CONDICIONES DE ACREDITACIÓN:

ESTUDIANTES REGULARES. Se considerará regularizado el cursado de la materia con la aprobación de dos parciales escritos e individuales con nota de cada parcial igual o superior a 4 (cuatro). En caso de desaprobar alguno de los dos parciales, habrá un recuperatorio por parcial. En caso de no aprobar alguno de los exámenes en esta segunda instancia el estudiante quedará libre en la materia. Al regularizar la asignatura el estudiante aprobará la materia al obtener una nota equivalente o superior al 4 (cuatro) en un examen final escrito.

ESTUDIANTES PROMOCIONALES. Se considerará promocionada la materia con la aprobación de los dos parciales individuales y escritos, la entrega en tiempo y forma de todas las prácticas y la presentación de exposiciones a lo largo del cursado. Todas estas instancias deberán ser aprobadas con una nota igual o superior a 8 (ocho). La desaprobación de un parcial o exposición, la falta de entrega de alguna de las prácticas en tiempo y en forma implicarán la pérdida automática de la posibilidad de promoción. Al aprobar la promoción el estudiante quedará eximido de rendir el examen final. La nota final se obtiene del redondeo entero superior (según lo dispuesto en la Ordenanza 273/19, artículo 33) del valor obtenido computado como: 0.15*(promedio de la nota de exposiciones)+ 0,25*(promedio de notas de las prácticas) + 0,3*(nota del primer Parcial) + 0,3*(nota del segundo Parcial).

ESTUDIANTES LIBRES. Los estudiantes en esta situación deberán rendir el examen libre en las mismas fechas de examen final. Esta instancia consiste en la evaluación de un examen escrito práctico y una instancia de exposición oral. En el examen práctico se evaluará la implementación de algoritmos en Python, la utilización de la herramienta de git y la resolución de algún problema utilizando la computadora.

Esta instancia se deberá aprobar con una calificación igual o superior a 4 (cuatro) para pasar a la siguiente instancia de examen. En la instancia de exposición oral el estudiante deberá desarrollar el contenido que le sea solicitado y responder preguntas conceptuales de cualquier temática abordada en la materia. Esta instancia se aprueba también con una nota igual o superior a 4 (cuatro). En caso de aprobar ambas instancias la nota final se calcula a partir del promedio de estas dos notas.

8. DISTRIBUCIÓN HORARIA:

TEORÍA Y PRÁCTICA: LUNES Y JUEVES DE 19:00 A 20:30HS
PRÁCTICA: MIÉRCOLES DE 18:00 A 21:00HS
HORAS DE TEORÍA: 3
HORAS DE PRÁCTICA: 3

9. CRONOGRAMA TENTATIVO:

Fechas de desarrollo de temas, actividades y evaluaciones.

MES	UNIDAD
Marzo	I
Abril	II y III
Mayo	IV
Junio	V

Primer Parcial (y fecha de entrega de los prácticos 1, 2 y 3 para promocionar): 12 de Mayo de 2021
Recuperatorio Primer Parcial: 26 de Mayo de 2021
Segundo Parcial (y fecha de entrega de los prácticos 4 y 5 para promocionar): 16 de Junio de 2021
Recuperatorio Segundo Parcial: 30 de Junio de 2021

René Cejas Bolecek

Martha Ferrero

PROFESOR **CONFORMIDAD DEL DEPARTAMENTO**

CONFORMIDAD DEL CENTRO REGIONAL UNIVERSITARIO BARILOCHE

Mg. ALFONSO AGUILAR
Secretario Académico
Centro Regional Univ. Bariloche
Univ. Nacional del Comahue