



INSTITUTO/S: Tecnología e Ingeniería

CARRERA/S: Tecnicatura Universitaria en Programación / Tecnicatura Universitaria en Redes y Operaciones Informáticas.

MATERIA: Introducción a la Lógica y problemas computacionales

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: -----

EQUIPO DOCENTE: -----

CUATRIMESTRE: Tecnicatura Universitaria en Programación: 1^{ro} / Tecnicatura Universitaria en Redes y Operaciones Informáticas: 1^{ro}

AÑO: Tecnicatura Universitaria en Programación: 1^{ro} / Tecnicatura Universitaria en Redes y Operaciones Informáticas: 1^{ro}

PROGRAMA N°: Tecnicatura Universitaria en Programación: 2 / Tecnicatura Universitaria en Redes y Operaciones Informáticas: 2

(Aprob. Por Cons.Directivo 04/06/2022)

Instituto/s: Tecnología e Ingeniería

Carrera/s: Tecnicatura Universitaria en Programación / Tecnicatura Universitaria en Redes y Operaciones Informáticas.

Nombre de la materia: Introducción a la lógica y problemas computacionales

Responsable de la asignatura y equipo docente: —-----

Cuatrimestre y año: Tecnicatura Universitaria en Programación: 1^{ro} del 1^{er} año / Tecnicatura Universitaria en Redes y Operaciones Informáticas: 1^{ro} del 1^{er} año

Carga horaria semanal: 4 hs

Programa N°: Tecnicatura Universitaria en Programación: 2 / Tecnicatura Universitaria en Redes y Operaciones informáticas: 2

Código de la materia en SIU: 789

Introducción a la Lógica y problemas computacionales

1. Fundamentación

Esta materia actúa como puente entre la formación previa de la mayoría de las personas que acceden a las carreras de informática en la universidad, y las materias iniciales con contenidos fuertemente arraigados a la carrera puntual. Es la materia recomendada para estudiantes que no logran demostrar una base sólida en matemática, lógica e informática básica en su transcurso por el Curso de Preparación Universitaria.

La materia busca proveer un vocabulario mínimo sobre conceptos de informática, trabajar la idea de programación desde una concepción básica, mediante elementos que minimicen los problemas asociados a la sintaxis dura y enfocada en la idea de subtarear como herramienta y metodología de trabajo para solucionar problemas de complejidad mínima. Además, la materia trabaja elementos de comprensión lectora y lógico-matemática mediante la incorporación de enunciados no triviales que plantean problemas no triviales de conceptualizar, aunque simples de resolver.

2. Propósitos y/u objetivos

Propósitos

- Acercar a los/as estudiantes a un vocabulario mínimo sobre informática y ciencias de la computación, a la vez que proveer conceptos elementales sobre cómo funcionan los programas de computadoras y los diferentes paradigmas de programación.

- Ilustrar al estudiantado sobre las herramientas y conceptos de programación básicas del paradigma estructurado en un entorno de programación mediante bloques encastrables.
- Promover la correcta resolución de problemas computacionales mediante la técnica de división en subtarefas, fomentando el reuso y la claridad del código.

Objetivos

Que el/la alumno/a:

- Sea consciente de qué implican las disciplinas informáticas, su pasado, presente y posibles futuros.
- Entienda los elementos básicos de un lenguaje de programación del paradigma estructurado y las distintas formas de organizar su código en el mismo (secuencia, repetición y alternativa).
- Sea capaz de comprender las expresiones de tipo booleano, tanto en forma simple como en formas compuestas mediante la utilización de conectivas de conjunción, disyunción y negación.
- Pueda estructurar su programa en partes pequeñas y sencillas que interactúan entre sí para solucionar problemas más complejos, explotando la idea de procedimiento.
- Sea capaz de poder comprender un problema planteado en términos de un enunciado descriptivo, con ejemplos y operaciones asociadas, así como conceptualizar soluciones a problemas.
- Sea capaz de utilizar las estructuras de control más elementales de la programación estructurada, incluyendo la repetición simple, la alternativa condicional y la repetición condicional, así como expresiones numéricas simples y booleanas complejas, incluyendo conectivas lógicas.

3. Programa sintético:

Qué es la informática: hardware vs. software, historia de las computadoras, presente, posibles escenarios futuros. Historia del software y los lenguajes de programación: qué son los paradigmas de programación: imperativo, orientado a objeto y funcional.

Qué es un programa. Entornos de desarrollo y ejecución. Principios de la programación imperativa: comandos (acciones), estructuras de control de flujo de programas (secuencia, repetición simple, repetición condicional, alternativa condicional en comandos). Sensores booleanos. Conectivas booleanas. Sensores numéricos. División en subtarefas como metodología para la resolución de problemas complejos, y necesidad de dar estructura a un programa no trivial.

4. Programa analítico

4.1 Organización del contenido:

UNIDAD 0: Informática e historia

- Conceptos elementales sobre computadoras:
 - Qué es una computadora. Formas de las computadoras actuales.
 - Hardware: partes básicas de toda computadora. Periféricos.
 - Software: conceptos elementales sobre código fuente y código máquina.
 - Relaciones entre software y hardware.
- Historia de la informática
 - Repaso rápido sobre la historia de las computadoras.
 - Historia de la evolución de los lenguajes de programación, sistemas operativos y software.
 - Concepto de software libre vs. software privativo.
 - Dependencia de las sociedades modernas de los sistemas informáticos.
 - Nuevas tendencias en el desarrollo de la informática: computación biológica, computación cuántica, inteligencia artificial.

UNIDAD 1: Introducción a los conceptos elementales de programación

- Conceptos básicos sobre programación:
 - Concepto de programa (Cómo descripción de solución a problemas computacionales).
 - Concepto de lenguajes de programación.
 - Concepto de problemas computacionales.
 - Conceptos y ejemplos de código fuente.
 - Ambiente de desarrollo (IDEs).
 - Entorno de programación vs. Entornos de ejecución.
- Elementos básicos de los lenguajes de programación imperativos (y generales):
 - Comandos (Como descripción de acciones)
 - Programas simples y cuerpos (bloques) de código.
 - Secuenciación como forma de organización del código.
 - Concepto de estado (inicial, final e intermedios).
 - Concepto de transformación de estados.
 - Equivalencia de programas.

- Procedimientos. Enfocado como:
 - Herramienta para definir nuevos comandos.
 - Forma de comunicar claramente las ideas, tanto en estrategia como en abstracción.
 - Forma de separar el problema en partes más pequeñas que permitan plantear de forma clara la estrategia de solución elegida.
 - Forma de reutilización de código.
 - Elección de buenos nombres como forma de comunicar mejor las ideas elaboradas.
 - Conceptualización de procedimientos como programas independientes.
 - Conceptualización de procedimientos como múltiples programas que colaboran en la solución de un problema.
 - Procedimentación como forma de encapsulamiento de transformaciones.
- Repetición simple:
 - Repetición simple (cantidad fija y finita de veces).
 - Comandos compuestos (Comandos que esperan un cuerpo o bloque).
 - Expresiones numéricas literales.
 - Repetición como nueva forma de organización de código.
 - Casos de borde de una repetición.

UNIDAD 2: Alternativas y sensores

- Alternativa condicional y sensores:
 - Sensores como forma de obtener información del ambiente.
 - Alternativa condicional en comandos.
 - Alternativas simples (una rama, if-then) vs. alternativas completa (dos ramas, if-then-else)

UNIDAD 3: Repetición condicional y conectivas lógicas

- Conectivas lógicas:
 - Conjunción, disyunción y negación
 - Conectivas lógicas aplicadas a sensores.
- Repetición condicional:
 - Parcialidad por no terminación.
 - Recorridos secuenciales, como estrategia para garantizar la terminación.

4.2 Bibliografía y recursos obligatorios:

Factorovich, P.; Sawady O'Connor, F. (2015). *Actividades para aprender a programar*, volumen 1. Argentina, Ciudad Autónoma de Buenos Aires: Fundación Sadosky.

Czemerinski, H.; Dabbah, J.; Floris, C.; Frizzo, F.; Leonardi, M.C; . (2018). *Ciencias de la computación para el aula: 1er. ciclo de primaria: libro para docentes, 1ra ed.* Argentina, Ciudad Autónoma de Buenos Aires: Fundación Sadosky.

4.3 Bibliografía optativa:

Martínez López, P. E. (2013). *Las bases conceptuales de la programación: Una nueva forma de aprender a programar.* 1ra Ed. Argentina, La Plata; Ebook.

5. Metodología de enseñanza:

Dinámica de clases

La materia se organiza en tres partes bien definidas:

- Una parte de introducción, corta, que invita a reflexionar sobre la importancia de las computadoras en el día a día, entender los componentes mínimos de estas y su funcionamiento como caja negra, así como la historia de las mismas, el estado del arte y las perspectivas de las disciplinas informáticas a futuro, así como de distintas áreas del software y los lenguajes de programación. Esta sección es principalmente teórica, dadas por charlas expositivas y charlas debate, con posibles pequeños espacios de investigación sobre temáticas diversas.
- Una parte de programación, mediante bloques encastrables, donde se exploran los conceptos elementales del paradigma estructurado. Esta sección está dada por el descubrimiento por indagación, mediante la resolución de actividades, con una parte de contenidos teóricos consistente en puestas en común y debate, guiado por el docente.
- Una tercera parte, también de programación, en donde se agregan los conceptos de conectivas lógicas y de repetición condicional a los ya aprendidos, para combinarlos y solucionar problemas más complejos.

Por ser una materia de primer cuatrimestre, resulta de vital importancia la presencialidad, el seguimiento y exigencia de la asistencia a clases, con la intención de estimular el compromiso del alumnado con su formación.

Plan de trabajo en el campus:

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. El mismo se utilizará como espacio de seguimiento del estudiantado fuera de los espacios presenciales, repositorios de materiales y actividades auto-asistidas.

El espacio áulico virtual contará al menos con:

- Un canal de comunicación unidireccional con el estudiantado (Foro de avisos)
- Enlace a los materiales de bibliografía obligatorios y otros posibles materiales.
- Enlaces a las actividades prácticas.
- Enlaces a posibles videos teóricos.

En todos los casos mencionamos “enlaces” para referirnos a vínculos externos o a material embebido en el aula virtual, cualquiera fuera el caso, según conveniencias técnicas. Se priorizará siempre el contenido embebido en la medida de lo posible.

Las actividades en el campus se encontrarán organizadas en secciones semanales, y no serán discriminadas por comisiones o grupos de estudiantes. El acceso a las distintas unidades se encontrará limitado por fechas según la planificación calendárica para el cuatrimestre en curso.

Otras herramientas físicas y digitales requeridas

La materia requiere el uso de computadoras en los espacios áulicos para casi todas las clases de la segunda y tercera parte de la materia. El software a utilizar es un sitio web, por lo que se deberá contar con acceso a internet desde los dispositivos. Las clases en que sea necesario el uso de equipamiento especial serán coordinadas según disposición calendárica con la dirección de la carrera.

Los/Las estudiantes que cuenten con máquina propia en su casa podrán utilizarla para continuar las actividades en el hogar. También se dispondrán de actividades en papel, que cumplen un rol didáctico a la vez que de inclusión.

Para las evaluaciones se requerirá un aula grande, sin máquinas, con suficientes bancos para todos los/as estudiantes. En casos en donde haya múltiples comisiones en la misma banda horaria, se podrá utilizar un aula lo suficientemente grande para albergar a varias de las comisiones.

Los/as estudiantes deberán llevar siempre un cuaderno y lapicera para tomar notas, así como para trabajar en las actividades en papel.

6. Actividades de investigación y extensión

No aplica.

7. Evaluación y régimen de aprobación

El régimen de aprobación de la materia sigue el Régimen Académico vigente, adecuando la cantidad de exámenes, instancias de recuperación, y otros elementos de la materia al mismo. Los siguientes regímenes de aprobación de cursada y materia cumplen los requisitos

impuestos en el régimen académico al momento de elaboración de este programa, pero podrían cambiar si el régimen académico lo requiriese.

7.1 Aprobación de la cursada

La cursada se organiza en torno a la participación directa del estudiantado en los espacios presenciales. La participación de cada persona en esos espacios se considera obligatoria, debiendo asistir a al menos el 70% de los encuentros presenciales para poder obtener la regularidad.

En cuanto a los conocimientos, estos se deberán acreditar en el transcurso de la cursada durante dos exámenes parciales que presenta la materia, que serán en papel, a libro cerrado y de carácter individual. Cada examen contará con su correspondiente instancia de recuperación. El examen o su correspondiente recuperatorio deberá estar aprobado según el régimen académico vigente.

Así, para la aprobación de la cursada se requerirán los siguientes:

- Acreditación del 70% de asistencia a las clases presenciales.
- Acreditación de una nota igual o superior a 4 en cada instancia de examen o su correspondiente recuperatorio.

Dependiendo de los resultados en los exámenes se podrá acreditar la cursada o aprobar directamente la materia por promoción.

7.2 Aprobación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

Evaluación integradora: tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los

contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

7.3 Criterios de calificación

La calificación del examen individual se determinará en la escala 0 a 10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente.

8. Cronograma

El siguiente es un cronograma propuesto para la materia. El mismo deberá ser revisado y adaptado según el calendario académico en curso. Las actividades se conformarán de manera presencial, virtual o combinadas y se comunicarán al inicio de la cursada de manera de fijar los encuentros.

Semana	Unidad	Tema
1	Unidad 0: Informática e historia	<p>Bienvenida a la materia (presentación de programa y contrato pedagógico)</p> <p>Informática (qué es una computadora) partes de una computadora, introducción a la historia de las computadoras. Tarea: lectura de historia de las computadoras y el software, incluido software libre.</p> <p>Historia de las computadoras y el software.</p> <p>Charla debate sobre paradigmas de programación y software libre.</p>
2	Unidad 1: Introducción a los conceptos elementales de programación	<p>¿Qué es un programa? ¿Qué es programar? Problemas computacionales como transformaciones de estado. ejemplos. Punto de entrada, comandos y secuencia. Ambiente de desarrollo (IDE) y ambiente de PilasBloques, ambiente de desarrollo vs. ambiente de ejecución. Infinidad de soluciones y equivalencia de programas.</p> <p>Procedimientos. Definición de nuevos comandos y su uso. Importancia de comunicar adecuadamente las ideas mediante subtarear.</p>

3	Unidad 1: Introducción a los conceptos elementales de programación	<p>Repetición simple: forma y utilidad. Expresiones (Números). Control de flujo de la repetición. Procedimiento como forma de expresar la estrategia.</p> <p>Legibilidad y comunicación. Importancia de no anidar estructuras. Elección de buenos nombres: propósito y verbos.</p>
4	Unidad 1: Introducción a los conceptos elementales de programación	<p>Programación en papel. Sintaxis para programar en papel. Lectura e interpretación de enunciados complejos.</p> <p>Clase de solución de ejercicios en papel</p>
5	Unidad 2: Alternativas y sensores	<p>Condición y sensores. Estado actual. Alternativa condicional como forma de realizar elecciones según el estado. Alternativa condicional completa y simplificada. Errores comunes: duplicación de código en ambas ramas.</p> <p>Solución de ejercicios de alternativa condicional en papel.</p>
6	Unidad 2: Alternativas y sensores	<p>Clase de guía práctica integradora y repaso general de temas.</p> <p>BUFFER: Clase de repaso general de todos los temas vistos.</p>
7	Examen Parcial 1 Contenidos de Unidad 1 y 2	<p>PRIMER PARCIAL: PROGRAMAS, COMANDOS, EXPRESIONES, PROCEDIMIENTOS, REPETICIÓN SIMPLE Y ALTERNATIVA CONDICIONAL</p> <p>PRIMER PARCIAL: SOLUCIÓN</p>
8	Unidad 3: Repetición condicional y conectivas lógicas	<p>¿Qué es lógica? Propositiones atómicas y compuestas. Conectivas lógicas: conjunción, disyunción y negación. Tablas de verdad, verificación de proposiciones compuestas. Pasaje de lenguaje natural a fórmula.</p> <p>La lógica en la programación. Conectivas en</p>

		condiciones booleanas. Lógica aplicada a los sensores. Sintaxis y forma de utilización.
9	Unidad 3: Repetición condicional y conectivas lógicas Recuperatorio Examen Parcial Contenidos de Unidad 1, 2 Y 3	Práctica de ejercicios con condiciones con conectivas. RECUPERATORIO DE PARCIAL 1
10	Unidad 3: Repetición condicional y conectivas lógicas	Repetición condicional. Utilidad y control de flujo. Fallos por no terminación. Condiciones antes y después de la repetición. Separación en subtareas para evitar fallos de no terminación. Sensores numéricos. Naturaleza de las expresiones (booleanos vs. números).
11	Unidad 3: Repetición condicional y conectivas lógicas	Ejercicios de repetición condicional y sensores numéricos. BUFFER: Clase de repaso general de todos los temas vistos y solución de actividades.
12	Unidad 3: Repetición condicional y conectivas lógicas	Clase de guía práctica integradora y repaso general de temas. BUFFER: Clase de repaso general de todos los temas vistos y solución de actividades.
13	Examen Parcial 2 Contenidos de Unidad 1, 2 y 3	SEGUNDO PARCIAL: CONDICIONALES CON CONECTIVAS LÓGICAS. REPETICIONES CONDICIONALES Y SENSORES NUMÉRICOS. SEGUNDO PARCIAL: SOLUCIÓN
14	Consulta y repaso: Programación	Clases de consulta para quienes deben realizar recuperatorio de parcial de programación. Clases de consulta para quienes deben realizar recuperatorio de parcial de programación.

15	Recuperatorio Examen Parcial Contenidos de Unidad 1, 2 Y 3	Clases de consulta para quienes deben realizar recuperatorio de parcial de programación. SEGUNDO RECUPERATORIO: PROGRAMACIÓN
16	Consulta y repaso: Programación	Clases de consulta para quienes deben realizar examen integrador. Clases de consulta para quienes deben realizar examen integrador.