1. PRESENTACIÓN DEL ESPACIO CURRICULAR

Espacio curricular: Programación I					
Código SIU-guaraní: 902		**Horas Presenciales:** 45		**Ciclo lectivo:** 2024	
Carrera:	Licenciatura en Ciencias de la Computación		**Plan de Estudio:**	Ord. 04/23	
Dirección a la que pertenece	Licenciatura en Computación		**Bloque/ Trayecto**	Algoritmos y Lenguajes	
Ubicación curricular:	1er Sem	**Créditos** 4	**Formato Curricular**	Teoría/práctica	
Equipo docente					
Cargo: Adjunto	**Nombre:** Elina Pacini		**Correo:** elina.pacini@ingenieria.uncuyo.edu.ar		
Cargo: JTP	**Nombre:** Leandro Spadaro		**Correo:** leandro.spadaro@ingenieria.uncuyo.edu.ar		
Cargo: JTP	**Nombre:** Silvina Manganelli		**Correo:** silvina.manganelli@ingenieria.uncuyo.edu.ar		
Cargo: JTP	**Nombre:** Laura Noussan		**Correo:** laura.noussan@ingenieria.uncuyo.edu.ar		

Fundamentación

La asignatura de Programación I se justifica en un contexto donde la tecnología y el software desempeñan un papel crucial en la sociedad moderna y, además, es un componente fundamental en la formación de profesionales de la Licenciatura en Ciencias de la Computación. En un mundo cada vez más digitalizado, la capacidad de programar y desarrollar software se ha convertido en una habilidad esencial que deben adquirir los estudiantes.

En esta asignatura se comienza a aprender a programar desde cero, es decir, desde las bases de la programación. Dado el carácter teórico práctico de la asignatura, el alumno logrará desarrollar el pensamiento lógico y adquirir las habilidades necesarias para enfrentar los desafíos de la programación y la resolución de problemas propios de esta disciplina.

Por tanto, el sentido de esta asignatura es proveer una metodología de enseñanza pensada en aquellos alumnos que no han programado antes y contribuir a que los profesionales de la Licenciatura en Ciencias de la Computación estén preparados para abordar problemas complejos propios de la informática. Este programa se alinea con la visión de formación por competencias y el aprendizaje centrado en el estudiante, reconociendo la importancia de integrar estos principios en la educación superior.

Se propone, para ello, trabajar con guías sencillas que permitan a los alumnos comprender y aplicar adecuadamente conceptos básicos de programación, comprendiendo además la manera en que las computadoras y dispositivos interpretan las órdenes que se programan mediante un correcto uso de las estructuras. En este sentido, se analizarán las principales sentencias, operadores y sintaxis común a la mayoría de los lenguajes de programación, todo esto mediante ejemplos y ejercicios.

Al finalizar el cursado el estudiante podrá resolver problemas sencillos del mundo real y contará con todas las bases para comenzar a aprender cualquier lenguaje de programación, esto gracias a que la lógica de programación que veremos es universal, y se aplica prácticamente a cualquier lenguaje.

Aportes al perfil de egreso (De la Matriz de Tributación)		
CE - Competencias de Egreso Específicas	*CE-GT* Competencias Genéricas Tecnológicas	*CE-GSPA* Competencias Sociales – Político - Actitudinales
CE 1.1. Especificar, proyectar y desarrollar sistemas de información. CE 1.3. Especificar, proyectar y desarrollar software.	CE-GT 1: Identificar, formular y resolver problemas de informática. CE-GT 2:Concebir, diseñar y desarrollar proyectos de informática. CE-GT 5: Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	CE-GSPA 6: Comunicarse con efectividad. CE-GSPA 9: Aprender de forma continua y autónoma.

Expectativas de logro (del Plan de Estudios)

Al acreditar el espacio curricular, las y los estudiantes serán capaces de:

- Analizar, formular y resolver problemas de informática, pudiendo identificar datos de entrada y de salida, restricciones, tipos de datos y procesos a realizar para su solución; incluyendo, si es necesario, la división en subproblemas para facilitar su resolución.
- Comprender y aplicar las estructuras básicas de programación aplicando los fundamentos teórico prácticos para resolver problemas algorítmicos.

Contenidos mínimos (del Plan de Estudios)

Introducción al paradigma de lenguajes imperativos. Definición de un algoritmo. Estrategia de resolución de problemas. Elementos de un programa. Tipos de datos simples. Definición de variables y constantes. Tipos de expresiones: aritméticas, relacionales y lógicas. Estructuras de control de un programa: secuencial, condicional, cíclica. Subprogramas. Pasaje de parámetros por valor y por referencia. Arreglos.

Correlativas (Saberes previos/ posteriores del Plan de Correlatividades)

Saberes previos: No requeridos

Saberes posteriores: Programación II, Matemática Discreta, Sistemas Operativos y Álgebra y Geometría Analítica y, Algoritmos y Estructuras de Datos I.

2. RESULTADOS DE APRENDIZAJE

RA1 Desarrolla habilidades de resolución de problemas aplicando razonamiento lógico y creativo para abordar desafíos propios de la programación.

RA2 Aplica correctamente las estructuras básicas de programación para resolver problemas de manera eficaz.

RA3 Identifica y aplica los pasos esenciales para diseñar, programar y ejecutar programas, detectar posibles errores y corregirlos de tal manera que los programas resuelvan un problema determinado.

RA4 Diseña un algoritmo que divida un problema en subproblemas más pequeños y resuelva cada subproblema de manera independiente, desarrollando habilidades esenciales para la resolución de problemas algorítmicos del mundo real, con el propósito de mejorar la capacidad de abordar problemas complejos.

RA5 Comprende y aplica pasaje de parámetros entre el programa principal y los subprogramas diferenciando pasaje de parámetros por valor y por referencia, para facilitar la escritura de programas que sean más versátiles, eficientes y mantenibles.

3. CONTENIDOS/SABERES (Organizados por unidades, ejes u otros)

PROGRAMACIÓN I

UNIDAD 1: INTRODUCCIÓN

1.A Conceptos básicos: Computación e informática. Computadora: componentes, organización física, dispositivos de entrada, dispositivos de salida

1.B Resolución de Problemas: Contexto de un problema. Datos asociados. Comprensión de problemas y metodología general de resolución de un problema mediante el uso de computadoras (análisis, diseño, codificación y ejecución).

1.C Algoritmos y Programas: Definición de algoritmo y características. Tipos de algoritmos. Pseudocódigo. Diagrama de flujo. Definición de Programa. Instrucción. Elementos. Introducción al Paradigma Imperativo.

1.D Lenguajes de programación: Tipos de lenguajes de programación (lenguaje de máquina, lenguaje ensamblador, lenguajes de alto nivel). Traductores de lenguajes (compiladores e intérpretes). Fases de compilación.

UNIDAD 2: ESTRUCTURA GENERAL DE UN PROGRAMA

2.A Programa: Partes constitutivas de un programa: Instrucciones. Identificadores. Variables y constantes. Comentarios.

2.B Instrucciones: Palabras reservadas. Expresiones. Asignación. Lectura y escritura

2.C Datos: Clasificación de los datos: Tipo, dimensión, complejidad y mutabilidad.
Tipos de datos elementales: entero, real, lógico y cadena.

2.D Tipos de Expresiones: Expresiones aritméticas, relacionales y lógicas.

2.E Conceptos de estructura secuencial: Representación. Declaración de variables, asignación, operaciones de lectura y escritura.

UNIDAD 3: FLUJOS DE CONTROL I: ESTRUCTURAS DE DECISIÓN

3.A Estructura de Decisión: Concepto. Representación. Aplicación.
3.B Tipos de estructuras de decisión: Composición condicional (decisión simple). Composición alternativa (decisión doble). Decisiones múltiples (composición por alternativas anidadas o por composición selectiva).

UNIDAD 4: FLUJOS DE CONTROL II: ESTRUCTURAS REPETITIVAS

4.A Estructura Repetitiva: Concepto. Representación. Aplicación.
4.B Tipos de Estructuras Repetitivas: Bucles: Mientras, Hacer-Mientras, Repetir, Desde/Para.
4C Diseño de Bucles: Cuerpo del bucle, sentencias de inicialización, condiciones de terminación del bucle. Contadores y acumuladores. Bucles anidados.

UNIDAD 5: SUBPROGRAMAS: FUNCIONES Y PROCEDIMIENTOS

5.A Introducción a Subalgoritmos o Subprogramas: Estrategias de diseño. Subprograma o acción con nombre: concepto, declaración, argumentos, parámetros y variables.
5.B Comunicación con subprogramas: Pasaje de argumentos por valor y por referencia. Noción de llamada a procedimientos y funciones. Precondiciones (estado inicial requerido) y postcondiciones (estado final provocado) de cada una de ellas.
5.C Ámbito (alcance): Variables locales y globales.

UNIDAD 6: ARREGLOS (VARIABLES DIMENSIONADAS)

6.A Variables Unidimensionales o Vectores: Concepto: dimensión e índice. Declaración. Inicialización. Aplicación. Operaciones asociadas: lectura, escritura, recorrido, copia, modificación de uno o más elementos, búsqueda del valor mayor, menor o bien uno determinado. Utilización en Subprogramas.

6.B Variables Multidimensionales: Concepto: dimensiones e índices. Matrices. Declaración. Inicialización. Aplicación. Operaciones asociadas a matrices: lectura, escritura, recorrido, copia, modificación de uno o más elementos, búsqueda del valor mayor, menor o bien uno determinado. Utilización en Subprogramas.

4. MEDIACION PEDAGOGICA (metodologías, estrategias, recomendaciones para el estudio)

Se propone una metodología de clase invertida ya que permite contar con más tiempo para poder aplicar de manera práctica los conocimientos en clase presencial. El objetivo consiste en lograr que los estudiantes gestionen su aprendizaje interactuando con el material audiovisual provisto por la cátedra y trabajando de manera colaborativa. Para reforzar el aprendizaje, las clases incluyen demostraciones en computadora y el uso del proyector, siguiendo el viejo precepto que “una imagen vale más que mil palabras”. Durante las clases se realizará motivación continua. El inicio en programación puede ser frustrante para el alumno por no obtener la resolución deseada en los ejercicios. Por lo tanto, para motivar a los estudiantes se presentan ejemplos que pueden encontrar en el mundo real. Ej.:

Videos Juegos, Sitios Web, Aplicaciones de Escritorio y Aplicaciones Móviles. Es importante que el alumno sea consciente de lo que puede ser capaz con los conocimientos que adquiera en la materia. Además, se desea fomentar el intercambio de opiniones entre los alumnos introduciéndoos en formas de trabajo reales de equipos de programación. Previo a los encuentros presenciales se proveerá a los alumnos con todo el material necesario para que puedan ir avanzando en las unidades de manera autónoma en los tiempos que ellos tengan disponibles. Para cada unidad se les habilita en el aula abierta la siguiente información:

- Un documento con un resumen teórico del tema que se complementa con el libro de referencia de la cátedra
- Las diapositivas de la clase/unidad
- Uno o más videos realizados por los profesores de la cátedra donde se explica el tema y se proveen diferentes ejemplos prácticos. Los videos están colgados de *YouTube* y los alumnos pueden acceder a los mismos tantas veces como sea necesario.
- Una guía de trabajos prácticos obligatoria
- Una guía de trabajos prácticos con ejercicios complementarios

Durante los encuentros presenciales, y con el objetivo de reforzar los conocimientos adquiridos, se aplica gamificación al aula dando la posibilidad a los alumnos de competir por Bonus Cards, cartas virtuales que habilitan a los felices poseedores a hackear las reglas de la cátedra; o, dicho de otra manera, es similar a ganar un superpoder para utilizar cuando quieran/puedan. Las cartas se obtienen mediante desafíos que se van desarrollando durante el cursado y que exigen la pericia por parte de los competidores en el arte de “prestar atención” y “participación activa”. Los desafíos son anunciados durante las clases, así como las reglas y el premio por lograr lo cometido. La gamificación es de utilidad para resolver tanto dudas de los conceptos teóricos pertinentes a cada tema, como así también sobre resolución de los ejercicios prácticos.

Además de la sala virtual habilitada en el aula abierta, los alumnos tienen la posibilidad de realizar consultas puntuales a través de la aplicación Slack, una herramienta de comunicación en equipo. A través del Slack los alumnos pueden postear sus preguntas y tanto los profesores de la cátedra como el resto de los compañeros pueden ayudar a resolverlas. Una de las ventajas de esa aplicación es que puede instalarse en los dispositivos móviles y se reciben notificaciones cada vez que hay actividad.

5. INTENSIDAD DE LA FORMACIÓN PRACTICA

Ámbito de formación práctica	Carga horaria	
	Presencial	No presencial
Formación Experimental		
Resolución de problemas del mundo real	**30**	
Actividades de proyecto y diseño de sistemas informáticos		
Instancias supervisadas de formación en la práctica profesional		
Otras actividades		
Carga horaria total	**30**	

6. SISTEMA DE EVALUACIÓN

La materia es promocional y por lo tanto será evaluada de manera CONTINUA. Esto significa que en cada clase y a lo largo del cursado se considerarán lo siguientes criterios generales de evaluación:

- Participación: se evaluará que el alumno participe activamente en cada una de las actividades y desafíos propuestos tanto en clase como en el aula virtual.

- Dedicación a la materia: se evaluará la dedicación a la materia (repaso de los temas presentados en los videos y material propuesto por la cátedra para cada clase).

- Responsabilidad: Los alumnos deberán presentar *en tiempo y forma los trabajos prácticos obligatorios* y de acuerdo al calendario establecido para tal fin (el calendario de entregas es publicado al inicio del semestre en el aula virtual). Las fechas de entrega también se irán notificando a lo largo del semestre. Los trabajos prácticos consistirán en la resolución de diversos problemas y su implementación a través del software PSeInt. Los alumnos serán evaluados individualmente.

Las evaluaciones son teóricas-prácticas e incluyen la resolución de problemas algorítmicos donde se aplican los aprendizajes correspondientes a las unidades evaluadas. Las evaluaciones se realizan de acuerdo a los contenidos/saberes especificados en el programa.

Dado que la materia es promocional, durante el cursado se realizarán 2 evaluaciones obligatorias con contenidos globales y 2 controles. Dichas evaluaciones serán de carácter individual. La nota final *NF* se calcula como:

$$NF = NPC * 0{,}20 + NG1 * 0{,}20 + NG2 * 0{,}60$$

donde,

- *NPC* es la nota promedio correspondiente a los controles evaluativos.
- *NG1* corresponde al resultado del primer examen global. Este examen incluye la evaluación de las primeras 4 (cuatro) unidades del programa.
- *NG2* corresponde al resultado del segundo examen global e incluye a todas las unidades de la asignatura.

En caso de inasistencia a alguno de los exámenes se considerará la evaluación como desaprobada. En caso de enfermedad la inasistencia deberá ser debidamente justificada siguiendo el procedimiento establecido por la facultad.

6.1. Criterios de evaluación

En cada uno de los exámenes se tomarán en consideración a la hora de evaluar los siguientes criterios de evaluación:

- Analiza e interpreta correctamente el problema a resolver, identificando correctamente los tipos de datos y las operaciones asociadas para los diferentes datos de entrada y de salida de un problema, adquiriendo habilidades esenciales para la manipulación de información en la programación, en diversos contextos y problemas de programación. (RA1)
- Diseña un algoritmo aplicando buenas prácticas de programación en el uso de las diferentes estructuras de programación para obtener programas que sean fáciles de interpretar y mantener. (RA2)
- Realiza pruebas de escritorio que permitan testear un algoritmo contemplando diferentes conjuntos de datos de prueba para detectar posibles errores y corregirlos. (RA3)
- Analiza correctamente y divide el problema en subproblemas (subprogramas) que sean más fáciles de resolver de manera individual adquiriendo habilidad para resolver problemas más complejos. (RA4)
- Aplica correctamente el pasaje de parámetros entre el programa principal y los subrprogramas, comprendiendo la diferencia entre pasaje de parámetros por valor y por referencia, para obtener programas que sean más legibles, eficientes y mantenibles. (RA5)

6.2. Condiciones de regularidad

Un alumno quedará en condición de alumno regular cuando:

- haya cumplido *estrictamente* con cada uno de los criterios de evaluación continua, incluyendo la entrega de los trabajos prácticos y evaluación de controles *en las fechas establecidas* en el calendario de la asignatura.
- haya cumplido con el 75% de la asistencia
- haya presentado en tiempo y forma todos los trabajos prácticos completos
- haya obtenido al menos un 60% en la nota final (NF).

En caso de cumplir con los criterios de regularidad pero no alcanzar el 60% en la nota final, existirá una *instancia de recuperatorio* también con contenido *global* de similares características al segundo examen global**.** Este recuperatorio debe ser aprobado con nota mayor o igual al 60%.

6.3. Condiciones de promoción

El alumno promocionará la materia cuando haya cumplido con todas las condiciones establecidas para la condición de regularidad, y además:

- haya obtenido *al menos 60%* en cada uno de los controles
- haya obtenido *al menos 70%* en cada uno de los globales (NG1 y NG2)
- haya obtenido un promedio *mayor o igual* al 75%.

6.4. Régimen de acreditación para

- **Promoción directa:** Detallado en sección condiciones de promoción
- **Alumnos regulares:**

  Examen Final

  Aquellos alumnos que regularizan la materia deberán rendir un examen final en alguna de las fechas establecidas en el calendario académico habilitadas para tal fin. El examen consiste de dos instancias, una instancia práctica donde el alumno debe diseñar e implementar la solución algorítmica a un problema que abarque los aspectos fundamentales de la asignatura, y una instancia teórica donde el alumno debe explicar su algoritmo y responder preguntas complementarias que abarquen los contenidos del programa. El examen final se rinde a programa completo, exigiendo su aprobación con una nota mayor o igual a 6 (seis) de acuerdo a lo establecido en la ordenanza N° 108 en cada una de las instancias de evaluación.

- **Alumnos libres:**

  No se admiten alumnos libres.

7. BIBLIOGRAFIA

Titulo	Autor /es	Editorial	Año de Edición	Ejemplares Disponibles	Sitios digitales
Algoritmos y programación: mejores prácticas	Ayala San Martín, Gerardo	Universidad de las Américas Puebla (UDLAP	2020	Electrónica	Enlace
Introducción a la Programación	Juganaru Mathieu, Mihaela.	Larousse	2014	Electrónica	Enlace
Aprender a Programar. Ejemplos en PSeInt	José Luis, Rodríguez Muñoz	Independiente	2023	Online Gratuito	Online Gratuito
Diseño estructurado de algoritmos aplicados en PSEINT	Pedro, Vélez Duque	Editorial Grupo Compás	2021	Online Gratuito	Online Gratuito

7.1. Recursos digitales del espacio curricular (enlace a aula virtual y otros)

https://aulaabierta.ingenieria.uncuyo.edu.ar/course/view.php?id=1875

8. FIRMAS

Elina Pacini

V°B° DIRECTOR/A DE CARRERA
RESPONSABLE A CARGO
Fecha

DOCENTE
Fecha: 13/10/2023