



RIDAA
Repositorio Institucional
Digital de Acceso Abierto de la
Universidad Nacional de Quilmes



Universidad
Nacional
de Quilmes

Rodas Bonjour, Alan

Introducción a la programación



Esta obra está bajo una Licencia Creative Commons Argentina.
Atribución - No Comercial - Compartir Igual 2.5
<https://creativecommons.org/licenses/by-nc-sa/2.5/ar/>

Documento descargado de RIDAA-UNQ Repositorio Institucional Digital de Acceso Abierto de la Universidad Nacional de Quilmes de la Universidad Nacional de Quilmes

Cita recomendada:

Cálcena, E., Rodas Bonjour, A., Tobía, P., Salina, M., Cuoco, C., Cherchyk, F., Sauczuk, M., Rodríguez, A., Español, D., Sabaliauskas, P., Fernandez Florio, G.P. y Marrero, P. (2026). *Introducción a la programación. (Programa)*. Bernal, Argentina: Universidad Nacional de Quilmes. Disponible en RIDAA-UNQ Repositorio Institucional Digital de Acceso Abierto de la Universidad Nacional de Quilmes
<http://ridaa.unq.edu.ar/handle/20.500.11807/6486>

Puede encontrar éste y otros documentos en: <https://ridaa.unq.edu.ar>



PROGRAMA ANALÍTICO DE LA ASIGNATURA
INTRODUCCIÓN A LA PROGRAMACIÓN
Modalidad Regular

Departamento de Ciencia y Tecnología

Carrera: Lic. en Bioinformática

Asignatura: Introducción a la Programación

Asignaturas correlativas: Elementos de Programación y Lógica

Carga horaria total: 144 hs

Docentes:

Alan Rodas Bonjour, Pablo Tobía, Mauro Salina, Carlos Cuoco, Federico Cherchyk, Martin Sauczuk, Alan Rodriguez, Douglas Español, Pablo Sabaliauskas, Pablo Gonzalo Fernandez Florio, Pablo Marrero, Eugenio Cálcena.

Año lectivo:

2025

Objetivos:

Los objetivos para quienes cursen la asignatura son:

- Comprender las distintas herramientas que forman parte de los lenguajes de programación y poder usarlas de forma apropiada para pensar, implementar y depurar programas que resuelvan problemas sencillos.
- Ser capaz de razonar y justificar sobre las herramientas, técnicas y estrategias elegidas al momento de escribir un programa.
- Lograr abstraer el uso de las herramientas de programación y ser capaz de aplicarlos en diversos entornos o lenguajes de programación específicos.
- Manejar ciertos principios, prácticas y metodologías básicas que resulten en programas robustos y mantenibles, p.ej. comentarios, elección de nombres, precondiciones, terminación.

- Conocer las ideas de secuencia y estado, y poder usarlas al pensar programas.
- Poder estructurar programas en bloques que se invocan entre sí, entendiendo los conceptos de parámetro y valor de retorno.
- Entender el concepto de estructura de datos, y poder aprovecharlo para organizar el estado de los programas que se construyan.

Contenidos mínimos:

- Qué es un programa.
- Las herramientas de programación: entornos de ejecución y de desarrollo.
- Principios de la programación imperativa: acciones y comandos, valores y expresiones, tipos, estado.
- Formas de estructuración de código: secuenciación, repeticiones y alternativas.
- Terminación y parcialidad. Precondiciones como metodología para el desarrollo de software robusto. Estrategias para garantizar la terminación de programas.
- Principios de la programación estructurada: funciones y procedimientos. Necesidad de darle una estructura a un programa no trivial.
- Parámetros y valores de retorno.
- Buenas prácticas de desarrollo de software robusto: comentarios, elección de nombres, legibilidad, indentación.
- Resolución de pequeños problemas mediante programas.
- Estructuras de datos básicas: listas y registros.

Programa analítico:

Unidad 1. Programas, comandos y contratos

Qué es un programa. Lenguajes de programación y código fuente. Entorno de ejecución y de desarrollo: concepto y uso. Estado y efecto de un programa. Elementos del lenguaje. Comandos y acciones. Argumentos y valores literales. Definición de programa. Palabras clave. Cuerpo y secuenciación de comandos. Errores de sintaxis. Indentación. Comentarios. Contratos: propósito y precondición. Totalidad y parcialidad. Concepto de error por fallo de precondición.

Unidad 2. Procedimientos, representación y estrategia

Procedimientos: motivación, definición y uso. Dominio y universo de discurso. Abstracción y representación de información. División en subtarefas (divide and conquer): enfoque top-down y bottom-up.

Comparativa de estrategias. Reutilización. Uso de nombres adecuados y legibilidad. Contratos en procedimientos.

Unidad 3. Repetición simple

Repetición simple. Casos de borde.

Unidad 4. Parámetros

Parámetros. Argumentos. Pasaje de datos por valor. Concepto de alcance: alcance global y alcance local. Generalización de soluciones. Documentación de parámetros.

Unidad 5. Expresiones y tipos

Concepto de tipo de datos. Categorización de tipos simples: numéricos y enumerativos (colores y direcciones). Operadores aritméticos. Operadores sobre enumerativos. Sobrecarga de operadores. Sensores. Errores de tipos.

Unidad 6. Alternativas condicionales

Expresiones booleanas. Operadores de comparación. Operadores lógicos. Circuito corto y circuito largo. Alternativa condicional en comandos: Alternativa completa, alternativa simple y multi alternativa. Diferencia entre programación defensiva y precondiciones.

Unidad 7. Funciones simples

Funciones simples: motivación, definición y uso. Valor de retorno. Denotación vs. operacionalidad. Contratos en funciones. Tipos de funciones.

Unidad 8. Repeticiones condicionales

Repetición condicional. Parcialidad por no terminación. Idea de recorrido secuencial. Esquemas de recorridos de procesamiento y de búsqueda. Concepto de invariantes de ciclo. Concepto de post-condición de un bucle.

Unidad 9. Variables y funciones con procesamiento

Conceptos de variable: definición y uso. Inicialización y alcance. Definición vs. inicialización. Comando de asignación. Usos comunes de variables: contador, acumulador, etc. Malas prácticas en uso de variables: variables innecesarias y flags. Funciones con procesamiento: motivación, definición y uso. Esquemas de recorridos que utilizan variables: acumulaciones, máximos y mínimos. Alternativa condicional en expresiones (operador ternario). Mapeo de booleanos a enteros.

Unidad 10. Tipos personalizados

Otros tipos de datos: strings, tuplas. Tipos de datos definidos por el usuario. Tipos variantes o enumerativos: motivación, definición y definición de valores literales. Tipos registros o con estructura: motivación,

definición. Campos, constructores y funciones observadoras o de acceso. Modificación de campos y de registros. Modelado del dominio de aplicación con tipos personalizados.

Unidad 11. Listas

Tipos de datos de las listas: motivación y uso. Constructores de listas. Operaciones básicas sobre listas. Tipo interno de una lista y tipo completo de la misma (nociones sobre tipos paramétricos). Programación de funciones genéricas sobre listas. Listas de registros y registros con listas. Listas de listas. Esquemas usuales de recorridos sobre listas: búsquedas, acumulaciones, transformaciones, filtros y máximos y mínimos.

Bibliografía

Bibliografía obligatoria

- Módulos de Trabajo elaborados por el equipo docente.
- Martínez López, P. E. Las bases conceptuales de la programación. Universidad Nacional de Quilmes, 2013.
- Martínez López, P. E. et. al. Ciencias de la computación para el aula: 1er ciclo de secundaria. Fundación Sadosky. 2019.
- Martínez López P. E., Bonelli E. Tutorial de Gobstones. Universidad Nacional de Quilmes, 2010.
- Dahl O. J., Dijkstra E. W., Hoare C. A. R.. Structured Programming. Academic Press. London-New York, 1972.
- Soulignac F. Diapositivas de clases. Universidad Nacional de Quilmes, 2011- 2015.
- Kernighan B. W., Pike R. The Practice of Programming.
Addison Wesley Professional Computing Series, 1st edition, 1999.

Bibliografía de consulta:

- Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. Addison Wesley, 1987.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. Introduction to algorithms. MIT Press, Cambridge, MA, 2009.

Organización de las clases:

El curso está estructurado en una serie de unidades que se presentan siempre con una misma dinámica de tres partes:

En la primera parte el estudiantado deberá realizar una serie de ejercicios que desafían los conocimientos aprendidos al momento, y que requerirán de la indagación para descubrir por su cuenta un nuevo concepto o herramienta. Para esto se utilizarán herramientas de programación visual mediante bloques encastrables, que permiten la indagación y la solución de problemas nuevos de forma independiente y con nula o escasa asistencia del equipo docente. Se espera que los estudiantes realicen estas actividades en su casa de forma autónoma.

Acto seguido, en la segunda parte, contará con material teórico que expandirá sobre ese nuevo concepto o herramienta, profundizando en su comprensión, conceptualización, análisis de casos, y otros. Aquí se optará por una dinámica de aula invertida, con el uso de videos teóricos que el estudiantado puede consumir de forma independiente, así como material escrito y bibliográfico. Así, se espera que el estudiantado arribe a las clases con los conceptos teóricos ya vistos y parcialmente aprendidos. Al inicio de las clases se realizará un repaso de la teoría, así como resolución de dudas y consultas respecto a los nuevos temas vistos.

En la tercera y última parte se presentará una serie de ejercicios prácticos en donde los estudiantes deberán aplicar el nuevo concepto, sumándose a los ya existentes, para resolver problemas de diversas características. Estos problemas deben poder ser resueltos en una herramienta de programación mediante texto, acercándose a la forma industrial de programación. También existen ejercicios que invitan al análisis y la reflexión por parte del estudiantado, así como ejercicios de programación en papel. En las clases, el principal foco del equipo docente está puesto en acompañar en la resolución de estas actividades, realizar puestas en común, mostrar y debatir diversas soluciones para los problemas presentados.

Esta estructura de indagación-teoría-práctica se repite para cada unidad.

Adicionalmente existen clases dedicadas al repaso, en donde se presentarán ejercicios integradores, con enunciados complejos, planteando situaciones de problemas realistas, que fomentan el análisis y la concepción de soluciones con la participación activa del estudiantado.

Los recursos didácticos empleados en la asignatura son:

La materia requiere el uso de computadoras con conexión a internet en los espacios áulicos, a razón de una por estudiante. En los casos de cursos netamente virtuales, el estudiante deberá contar con computadora de escritorio, notebook o netbook (no siendo tablet y teléfonos celulares una alternativa).

La materia utiliza un campus virtual en donde se presentan los materiales, así como actividades de seguimiento y autoevaluación. Adicionalmente al campus virtual la materia utilizará la herramienta de programación Gobstones (que provee tanto un entorno para la propuesta de indagación como para la propuesta práctica).

En el caso de cursos completamente virtuales, se puede requerir el uso de herramientas adicionales para el manejo de comunicaciones grupales de texto y voz (Discord o similar) y eventualmente otras.

Formas de evaluación y acreditación:

La modalidad de evaluación y aprobación se regirá según el Régimen de Estudios vigente.

Para poder acceder a las instancias de evaluación calificadas, el estudiantado deberá acreditar:

- La asistencia de al menos al 75% de las clases presenciales o encuentros sincrónicos.
- La realización de al menos el 75% de las actividades de autoevaluación presentadas en el campus virtual, realizadas en tiempo y forma.

Las instancias evaluativas calificadas constan de:

- Dos evaluaciones parciales a realizar por la totalidad de los estudiantes independientemente de la modalidad de cursada. Estas serán presenciales, de forma escrita y en formato papel. Cada una consta de su correspondiente recuperatorio en caso de no haber aprobado el examen.
- Una evaluación virtual únicamente para los estudiantes que cursen en modalidades virtuales asincrónicas, que se realizará por el campus virtual. Esta evaluación tendrá su correspondiente recuperatorio.
- Una evaluación integradora para quienes no hayan promocionado la materia. La misma se realiza de forma escrita y en formato papel, con defensa oral del trabajo realizado.

Cronograma tentativo:

Clase	Tema	Tipo de actividad
1	Presentaciones y condiciones de cursada. Unidad 1: Instalación y manejo de Gobstones. Indagación: Elementos de Gobstones (tablero, cabezal bolitas, estado). Comandos y valores primitivos. Bloque. Programa.	Clase expositiva / Resolución de problemas y ejercicios.
2	Unidad 1: Qué es un programa. Comandos y valores primitivos. Cuerpo. Programa. Efecto de un programa. Propósito de un programa. Diferencia entre efecto (ejecución) y propósito (intención). Noción de corrección. Totalidad y parcialidad. Documentación: Propósitos y precondiciones. Diferencia entre programar con bloques y con texto. Indentación. Comentarios. Equivalencia de programas.	Clase expositiva / Resolución de problemas y ejercicios.
3	Unidad 2: Procedimientos. División en subtarear. Consideraciones de nomenclatura (cómo poner un nombre). Documentación de procedimientos. Procedimientos de representación. Procedimientos como forma de reutilización.	Clase expositiva / Resolución de problemas y ejercicios.
4	Unidad 2: Estrategias de solución. Metodología top-down y bottom-up. Comparativa de estrategias. Estrategias correctas vs. estrategias adecuadas.	Clase expositiva / Resolución de

		problemas y ejercicios.
5	Unidad 3: Repetición simple. Casos de borde. Buenas prácticas: indentación de comandos compuestos, no anidación. Unidad 4: Parámetros y argumentos (noción conceptual). Mecánica de una invocación. Alcance. Alcance global vs. alcance local.	Clase expositiva / Resolución de problemas y ejercicios.

6	Unidad 4: Técnicas para parametrizar y generalizar problemas. Documentación de parámetros.	Resolución de problemas y ejercicios.
7	Unidad 5: Concepto de tipo de datos. Tipos simples: números y enumerativos (colores y direcciones). Expresiones literales vs. calculadas. Evaluación de una expresión. Expresiones primitivas: operadores de enumeración, operadores aritméticos. Formalización de parámetros, argumentos e invocación en el marco conceptual de valores, expresiones y tipos. Cálculo del tipo de una expresión. Errores de tipo. Noción revisada de operación total.	Clase expositiva / Resolución de problemas y ejercicios.
8	Unidad 5: Prácticas integradoras con los conceptos de expresiones y tipos.	Resolución de problemas y ejercicios.
9	Unidad 6: Valores de verdad (booleanos). Evaluación por circuito-corto. Operadores de comparación y operadores lógicos. Alternativa condicional. Diferentes formas de la alternativa (completa, simple y multi-alternativa). Programación defensiva vs. precondiciones (noción de precondiciones en presencia de un if). Anidación.	Clase expositiva / Resolución de problemas y ejercicios.
10	Unidad 7: Funciones simples. Tipo de una función. Documentación de funciones. Parcialidad en funciones.	Clase expositiva / Resolución de problemas y ejercicios.
11	Unidad 8: Repetición condicional. Parcialidad por no-terminación. Concepto de operación total con no-terminación. Esquema de recorridos para procesamiento y búsqueda sobre diferentes elementos.	Clase expositiva / Resolución de problemas y ejercicios.
12	Unidad 8: Ejercitación de repetición condicional. Concepto de invariante de ciclo y de post-condición de ciclo.	Clase expositiva / Resolución de problemas

		y ejercicios.
13	Unidad 9: Funciones con procesamiento. Mecánica de la invocación a función (sin efectos visibles sobre el estado). Variables. Propósito de una variable. Comando de asignación. Tipo de una variable (errores de tipo). Inicialización y alcance de una variable. Variables innecesarias. Ejemplos de distintos usos de las variables (contadores, acumuladores, etc).	Clase expositiva / Resolución de problemas y ejercicios.
14	Unidad 9: Alternativa condicional en expresiones. Recorridos utilizando variables: acumulaciones, máximos y mínimos.	Clase expositiva / Resolución de problemas y ejercicios.
15	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
16	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
	Evaluación parcial para estudiantes virtuales asincrónicos únicamente (vía campus)	Evaluación

17	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
18	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
	Recuperatorio de evaluación parcial para estudiantes virtuales asincrónicos únicamente (vía campus)	Evaluación

19	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
20	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
21	Evaluación parcial 1 para todos los estudiantes.	Evaluación.
22	Unidad 10: Necesidad de nuevos tipos de datos. Strings. Tuplas. Variantes: utilidad de los tipos enumerativos. Definición y uso de variantes. Registros: utilidad de los tipos con estructura. Definición y campos de un registro. Valores de un registro, constructores, operaciones de acceso, tipo de las operaciones. Uso de registros y variantes para el modelado del dominio de aplicación.	Clase expositiva / Resolución de problemas y ejercicios.
23	Unidad 10: Uso de registros y variantes para el modelado del dominio de aplicación.	Resolución de problemas y ejercicios.
24	Unidad 11: Listas: definición, operaciones de construcción y acceso, tipo de una lista. Procedimientos sencillos para construir listas.	Clase expositiva / Resolución de problemas y ejercicios.
25	Unidad 11: Práctica de manipulación y acceso a los elementos de una lista. Procesamiento de listas por repetición condicional. Repetición indexada (foreach). Procesamiento de listas por repetición indexada. Limitaciones de la repetición indexada.	Clase expositiva / Resolución de problemas y ejercicios.
26	Unidad 11: Esquemas comunes de recorridos sobre listas: acumulaciones, búsquedas, transformaciones, filtros y máximos y mínimos. Problemas sobre listas genéricas. Noción de tipo paramétrico.	Clase expositiva / Resolución de problemas y ejercicios.
27	Unidad 11: Ejercitación general con listas. Listas de listas. Listas de registros. Registros con listas.	Clase expositiva / Resolución

		de problemas y ejercicios.
28	Recuperatorio de evaluación parcial 1 para todos los estudiantes que hayan desaprobado la evaluación parcial 1.	Evaluación.
29	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
30	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
31	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
32	Repaso general de lo aprendido al momento con ejercicios integradores.	Resolución de problemas y ejercicios.
33	Evaluación parcial 2 para todos los estudiantes.	Evaluación.
34	Clase de repaso general.	Clase de consulta.

35	Clase de repaso general.	Clase de consulta.
36	Recuperatorio de evaluación parcial 2 para todos los estudiantes que hayan desaprobado la evaluación parcial 1.	Evaluación.
37	Clase de repaso general.	Clase de consulta.
38	Clase de repaso general.	Clase de consulta.
39	Evaluación integradora para quienes no hayan promocionado.	Evaluación.

