

## **PROGRAMA de Introducción a la Programación**

**Carreras:** Tecnicatura Universitaria en Programación Informática - Licenciatura en Informática - Licenciatura en Bioinformática

**Asignatura:** Introducción a la Programación

**Núcleo al que pertenece:** Básico

**Profesores:** Pablo E. Martínez López, Pablo Tobia, Valeria De Cristófolo, Gonzalo P. Fernández Florio, Leonardo Volinier.

**Asignaturas Correlativas:** Elementos de Lógica y Programación

### **Objetivos:**

Que la persona que cursa la materia:

- Pueda pensar, implementar y depurar un programa que resuelva problemas sencillos.
- Entienda las ideas de secuencia y estado, y las pueda usar al pensar sus programas.
- Pueda estructurar su programa en bloques que se invocan entre sí, entienda los conceptos de parámetro y valor de retorno.
- Entienda el concepto de estructura de datos, y pueda aprovecharlo para organizar el estado de los programas que construye.
- Maneje ciertos principios, prácticas y metodologías básicas que resulten en programas más robustos y mantenibles, p.ej. comentarios, elección de nombres, precondiciones, terminación.

### **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.
- Terminación y parcialidad. Precondiciones como metodología para el desarrollo de software robusto.
- Principios de la programación estructurada: funciones y procedimientos. Necesidad de darle una estructura a un programa no trivial.

- Resolución de pequeños problemas mediante programas.
- Estructuras de datos básicas: listas y registros.

**Carga horaria semanal: 8 hs**

### **Programa analítico:**

- Conceptos básicos de la programación: programa, código fuente, comandos, parámetros y argumentos, bloque, estado, indentación, efecto de un programa. Uso del ambiente de desarrollo.
- Procedimientos y funciones: efecto vs. denotación; expresiones vs. comandos; uso de expresiones y comandos. División en subtarefas (divide and conquer): enfoque top-down y bottom-up. Uso de nombres correctos.
- Propósito de un programa y precondition. Funciones y procedimientos totales vs. parciales. Diferencia entre programación defensiva y preconditiones. Concepto de error por fallo de precondition.
- Estructuras de control de flujo de un programa. Alternativa condicional: expresiones booleanas y concepto de circuito-corto. Repetición indexada: concepto de rango de objetos. Repetición condicional: parcialidad por no terminación, concepto de invariante de ciclo, casos de borde.
- Tipos de datos básicos: colores, direcciones, números, booleanos. Operaciones básicas de cada tipo. Tipo de una expresión, tipo de un parámetro. Error de tipado vs. error de ejecución.
- Recorridos secuenciales: esquemas de distintos recorridos usuales para problemas de procesamiento y búsqueda. Recorrido por filas, columnas, tableros, caminos, etc. Recorridos sobre objetos complejos como regiones de un mapa.
- Variables: concepto de variable, inicialización, comando de asignación, propósito de una variable. Construcciones usuales: acumulador, contador, etc.
- Registros: concepto de tupla. Modelado del dominio de aplicación con registros. Operadores de acceso y modificación de campos, inicialización, variables de registro. Registros con listas, listas con registros.
- Listas: tipo estructurados de las listas. Operaciones básicas, recorridos, esquemas usuales de recorridos sobre listas (filtro, búsqueda, mapeo, etc). Lista de listas. Tipo de una lista. Programación de funciones genéricas.

### **Bibliografía obligatoria:**

- P.E.Martínez López. Las bases conceptuales de la programación. Universidad Nacional de Quilmes, 2013.
- P.E.Martínez López y E. Bonelli. Tutorial de Gobstones. Universidad Nacional de Quilmes, 2010.
- F. Soullignac. Diapositivas de clases. Universidad Nacional de Quilmes, 2011-2015.
- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. y Stein, C. Introduction to algorithms, MIT Press, Cambridge, MA, 2009.
- Brian W. Kernighan y Rob Pike, The Practice of Programming, Addison Wesley Professional Computing Series, 1 edition (February 14, 1999). ISBN 978-0201615869



### **Bibliografía de consulta:**

- Aho, Hopcroft and Ullman. Data Structures and Algorithms, Addison Wesley, 1987.

### **Organización de las clases:**

La materia se divide en clases teóricas (3hs. semanales), en las que se introducen los conceptos fundamentales, y clases de práctica/laboratorio (5hs semanales), en las que se aplican los conceptos fundamentales para la resolución de distintos problemas de programación en la computadora. Previo a las clases teóricas se presentan problemas para que las personas que cursan la materia descubran las ideas mediante el denominado “aprendizaje por indagación”.

### **Trabajos Prácticos**

#### **Práctica 1: Contratos y procedimientos**

**Objetivos:** Confeccionar los primeros programas y conocer el entorno de ejecución y de desarrollo. Presentar las ideas de comandos y procedimientos. Presentar las nociones de terminación y parcialidad y su expresión en contratos.

#### **Práctica 2: División en subtareas**

**Objetivos:** Trabajar en la necesidad de darle una estructura a un programa no trivial a través de la división en subtareas y su expresión en código mediante procedimientos. Practicar la utilización de precondiciones como metodología para el desarrollo de software robusto.

#### **Práctica 3: Parámetros y repetición**

**Objetivos:** Enriquecer la expresividad de los procedimientos mediante el uso de parámetros. Trabajar la combinación de comandos mediante repetición simple (repeat). Practicar la resolución de pequeños problemas mediante programas.

#### **Práctica 4: Expresiones y tipos**

**Objetivos:** Extender el repertorio de herramientas del lenguaje para incluir expresiones y tipos. Expresar datos simples mediante expresiones y comprender la necesidad y utilidad de los tipos de datos.

#### **Práctica 5: Alternativas y funciones**

**Objetivos:** Trasladar los principios aprendidos para procedimientos en el mundo de los comandos a las funciones en el mundo de las expresiones. Presentar la combinación de comandos mediante alternativa condicional (if-then-else). Practicar la resolución de pequeños problemas mediante programas.

#### **Práctica 6: Repetición condicional**

**Objetivos:** Presentar la combinación de comandos mediante repetición condicional (while). Practicar la resolución de pequeños problemas mediante programas.

#### **Práctica 7: Variables y funciones con procesamiento.**

**Objetivos:** Presentar la noción de estado, su expresión mediante variables y extender la expresividad de funciones para manejar estado.

**Práctica 8: Registros**

**Objetivos:** Modelar estructuras simples mediante registros y practicar su utilización en la confección de programas simples.

**Práctica 9: Listas**

**Objetivos:** Incorporar las listas como una estructura de datos requerida para expresar volúmenes no conocidos de información. Practicar la definición de listas y la confección de programas simples que involucren listas de elementos de tipos simples.

**Práctica 10: Recorridos sobre listas**

**Objetivos:** Combinar todas las herramientas presentadas antes en la confección de programas simples que integren las ideas.

**Modalidad de evaluación:**

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los artículos del Régimen de estudios de la UNQ (Res. CS 201/18). En la modalidad de libre se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral, un desarrollo de programa en el entorno de trabajo e instancias de evaluación similares a las realizadas en la modalidad presencial.

**CRONOGRAMA TENTATIVO**

Semana	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Condiciones de cursada. Instalación y manejo de Gobstones. Indagación: Elementos de Gobstones (tablero, cabezal bolitas, estado). Comandos y valores primitivos. Bloque. Programa.				* previo	No corresponde
1	Teórica 1: Qué es programar. Elementos de Gobstones (tablero, cabezal bolitas, estado). Comandos y valores primitivos. Bloque. Programa. Comentarios. Diferencia entre programar con bloques y con texto. Efecto de un programa. Propósito de un programa. Diferencia entre efecto (ejecución) y propósito (intención). Noción de corrección. Documentación.	*				Ambos parciales y sus recuperatorios, cuestionarios de avance, integradores
1-2	Práctica de la clase teórica 1. Indagación de procedimientos y representación.		*	*	*indagación	Idem
2	Teórica 2: Parcialidad y precondiciones. Contrato (nombre + propósito + precondiciones). Equivalencia de programas. Procedimientos. División en subtarear. Consideraciones de nomenclatura (cómo poner un nombre); concepto de idioma.	*				Idem
2-3	Práctica de la clase teórica 2. Indagación de parámetros y repeticiones simples.		*	*	*indagación	Idem
3	Teórica 3: Parámetros y argumentos (noción conceptual). Mecánica de una invocación. Alcance. Repetición simple (repeat). No-anidación. <b>Parcial 1, parte 1.</b>	*			* instancia de evaluación	Idem
3-4	Práctica de la clase teórica 3. Indagación de expresiones y tipos.		*	*	*indagación	Idem

4	Teórica 4: Valores, expresiones y tipos: evaluación. Expresiones primitivas. Tipos dirección, color y número. 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. Operadores aritméticos y operadores de enumeración.	*				Idem
4-5	Práctica de la teórica 4. Indagación de alternativas y funciones.		*	*	*indagación	Idem
5	Teórica 5: Alternativa condicional. Valores de verdad (booleanos). Operadores de comparación y operadores lógicos. Programación defensiva vs. precondiciones (noción de precondiciones en presencia de un if). Anidación. Funciones simples. Tipo de una función. <b>Recuperatorio de parcial 1, parte 1.</b>	*			* instancia de recuperación	Idem
5-6	Práctica de la teórica 5. Indagación de repetición condicional y funciones con procesamiento.		*	*	*indagación	Idem
6	Teórica 6: Repetición condicional. Parcialidad por no-terminación. Concepto de operación total con no-terminación. Esquema de recorridos de filas y de celdas para procesamiento y búsqueda. Diferencia entre recorridos con orden predeterminado y recorrido con orden desconocido. Evaluación por circuito-corto. <b>Parcial 1, parte 2.</b>	*			* instancia de evaluación	Idem
6-7	Práctica de la teórica 6. Indagación de funciones con procesamiento.		*	*	*indagación	Idem
7	Teórica 7: Funciones con procesamiento. Mecánica de la invocación a función (sin efectos visibles sobre el estado del tablero). 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).	*				Idem
7-8	Práctica de la teórica 7. Indagación de registros y variantes.		*	*	*indagación	Idem
8	Teórica 8: Registros y variantes. Definición, tipos de un registro, valores de un registro, constructores, operaciones de acceso, tipo de las operaciones. Definición de variante, valores de un variante, formas de uso de un variante. Uso de registros y variantes para el modelado del dominio de aplicación. <b>Recuperatorio de parcial 1, parte 2.</b>	*			* instancia de recuperación	Parcial 1



8-9	Práctica de la teórica 8.		*	*		Parcial 2, recuperatorio e integradores
9	Indagación de listas. <b>Parcial 1, parte 3.</b>				* indagación * instancia de evaluación	Idem
9-10	Práctica de la teórica 8		*	*		Idem
10	Teórica 9: Listas: definición, operaciones de construcción y acceso, tipo de una lista. Procedimientos sencillos para construir listas.	*	*			Idem
10-11	Práctica de la teórica 9.		*	*		Idem
11	Teórica 10: Procesamiento de listas por repetición con foreach y while. Esquemas de resolución de problemas sobre listas: totalización, transformación, filtrado y búsqueda; otras operaciones avanzadas. <b>Recuperatorio del parcial 1, parte 3.</b>	*			* instancia de recuperación	Idem
11-12	Práctica de la teórica 10.		*	*		Idem
12	Modelado de un problema real usando Gobstones: ejercicio integrador pre-parcial	*	*			Idem
12-13	Práctica de la teórica 10 y ejercicios integradores		*	*		Idem
13	<b>Parcial 2.</b>				* instancia de evaluación	Parcial 2
13-15	Resolución e implementación del segundo parcial y consultas pre-recuperatorio	*	*			Idem
15	<b>Recuperatorio del parcial 2.</b>				* instancia de evaluación	Recuperatorio
16	Integrador				* instancia de evaluación	Integrador