PROGRAMA de *Introducción a la Programación*

Carrera/s: Tecnicatura Universitaria en Programación Informática / Licenciatura en Informática

Asignatura: *Introducción a la Programación*

Núcleo al que pertenece: *Programación*

Profesor/es: *Francisco Soulignac – Flavia Saldaña*

Asignaturas previas necesarias para favorecer el aprendizaje:

Elementos de Lógica y Programación

Objetivos:

Que el estudiante:

- *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 básicos que resulten en programas más robustos y mantenibles, p.ej. comentarios, elección de nombres, precondiciones, terminación.*

Contenidos mínimos:

- *Qué es un programa.*
- *Las herramientas del programador: 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 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 subtareas (divide and conquer): enfoque top-down y bottom-up. Uso de nombres correctos.*

• *Propósito de un programa y precondición. Funciones y procedimientos totales vs. parciales. Diferencia entre programación defensiva y precondiciones. Concepto de error por fallo de precondición.*

• *Estructuras de control de flujo de un programa. Alternativa condicional: expresiones booleanas y concepto de cortocircuito. 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 procesa-miento y búsqueda. Recorrido por filas, columnas, tableros, caminos, etc. Recorridos sobre obje-tos complejos como regiones de un mapa.*

• *Variables: concepto de variable, inicialización, operador de asignación (concepto de lvalue), propósito de una variable (invariante que cumple en un ciclo). Construcciones usuales: acumulador, contador, etc.*

• *Listas: tipo abstracto lista. Operaciones básicas, recorridos, esquemas usuales de recorridos sobre listas (filtro, búsqueda, mapeo, etc). Lista de listas. Tipo de una lista. Programación de funciones genéricas.*

• *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.*

Bibliografía obligatoria:

• *P. Martínez López. Las bases conceptuales de la programación. Universidad Nacional de Quilmes, 2013.*

• *P. Martínez López y E. Bonelli. Tutorial de Gobstones. Universidad Nacional de Quilmes, 2010*

• *F. Soulignac. Diapositivas de clases. Universidad Nacional de Quilmes, 2011-2015.*

Bibliografía de consulta:

• *Aho, Hopcroft y Ullman. Data Structures and Algorithms, Addison Wesley, 1987.*

• *Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. y Stein, C. Introduction to algorithms, MIT Press, Cambridge, MA, 2009.*

Organización de las clases:
La materia se divide en clases teoricas (3hs. semanales), en las que se introducen los conceptos fundamentales, y clases de practica/laboratorio (5hs semanales), en las que se aplican los conceptos fundamentales para la resolución de distintos problemas de programación en la computadora.

Modalidad de evaluación:
La materia cuenta con dos parciales teórico-prácticos (con recuperatorio) y dos examen integradores. Para promocionar la materia, el estudiante debe (a) obtener al menos 7 puntos en el segundo parcial o su recuperatorio o, (b) obtener al menos 4 puntos en el segundo parcial/recuperatorio y en alguna de las instancias de integración. La nota final se compone de la nota de los parciales e integradores.

CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Condiciones de cursada. Instalacion y manejo de Gobstones?				* previo	No corresponde
1	Teórica 1: Qué es programar. Elementos de Gobstones (tablero, cabezal bolitas, estado). Comandos y valores primitivos. Bloque. Program. Comentarios. Efecto de un programa. Propósito de un programa. Diferencia entre efecto (ejecucion) y propósito (intención). Noción de corrección. Documentación.	*				Ambos parciales, recuperatorio e integradores
1-2	Practica de la clase teórica 1		*	*		Idem
2	Teórica 2: Parcialidad y precondición. Contrato (precondición + signatura + propósito). Equivalencia de programas. Procedimientos. División en subtareas. Parámetros y argumentos (noción conceptual). Mecánica de una invocación. Alcance. Consideraciones de nomenclatura (cómo poner un nombre); concepto de idioma.	*				Idem
2-3	Practica de la clase teórica					Idem
3	Teórica 3: Repetición simple (repeat). Valores, expresiones y tipos: evaluación. Funciones primitivas. Tipos dirección, color y numero. Formalización de parámetros, argumentos e invocación en el marco conceptual de valores, expresiones y tipos. Repetición indexada (foreach). Concepto de rango e índice. No-anidación	*				Idem
3-4	Práctica de la clase teórica 3		*	*		Idem

4	Teórica 4: Alternativa condicional. Valores booleanos. Primitivas de comparación. Programación defensiva vs. precondiciones (noción de precondición en presencia de un if). Anidación. Calculo del tipo de una expresión. Errores de tipo. Noción revisada de operación total. Funciones simples y comando return. Funciones con comandos. Mecánica de la invocación a función (desaparición de efectos). Tipo de una función. Cortocircuitos.	*				Idem
4-5	Práctica de la teórica 4		*	*		Idem
5	Teórica 5: Repetición condicional. Parcialidad por no-terminación. Concepto de operación total con no-terminación. Esquema de recorridos de procesamiento y búsqueda. Diferencia entre recorridos con orden predeterminado y recorrido con orden desconocido.	*				Idem
5-6	Práctica de la teórica 5		*	*		Idem
6	Teórica 6: 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). Modelado de problemas en Gobstones	*				Idem
6-7	Práctica de la teórica 6		*	*		Idem
7-8	Modelado de un problema usando Gobstones: ejercicio integrador pre-parcial	*	*			Idem
8	Primer parcial				* instacia de evaluación	Parcial 1
8-9	Resolución e implementación del parcial		*	*		Parcial 2, recuperatorio e integradores

9	Teórica 7: introducción a Xgobstones. Listas: definición, operaciones de acceso, tipo de una lista, repetición sobre listas con foreach y while.	*				Idem
9-10	Práctica de la teórica 7		*	*		Idem
10	Teórica 8: patrones de resolución de problemas sobre listas: filtro, transformación y búsqueda	*	*			Idem
10-11	Practica de la teórica 8		*	*		Idem
11	Teórica 9: Registros. Definición, tipos de un registro, valores de un registro, constructores, operaciones de acceso, tipo de las operaciones. Uso de registros para el modelado del dominio de aplicación.	*				Idem
11-12	Práctica de la teórica 9		*	*		Idem
12	Teórica 10: Listas con registros. Modelado de problemas que requieren listas de listas, listas de registros y registros de listas. Estrategias de resolución de problemas	*	*			Idem
12-13	Práctica de la teórica 10		*	*		Idem
13-14	Modelado de un problema real usando XGobstones: ejercicio integrador pre-parcial	*	*			Idem
14	Segundo parcial				* instancia de evalución	Parcial 2
14-16	Resolución e implementación del segundo parcial y consultas pre-recuperatorio		*	*		
16	Recuperatorio				* instancia de evaluación	Recuperatorio
18	Integrador				* instancia de evaluación	Integrador

***INDIQUE CON UNA CRUZ LA MODALIDAD**