Aprobado por resolución N°113 del CFI de fecha 04.07.2017

FACULTAD DE INGENIERIA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Programa de Programación 2

1. NOMBRE DE LA UNIDAD CURRICULAR

Programación 2

2. CRÉDITOS

12 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

Capacitar al estudiante en metodologías de diseño de programas de tamaño pequeño y mediano, utilizando técnicas que consisten en la formulación e implementación de abstracciones de datos como, listas, pilas, colas, árboles binarios, colecciones y otras.

Se busca que los estudiantes adquieran experiencia de programación en un lenguaje específico que les permita ejercitar las técnicas de modularización aprendidas y que aprendan y apliquen buenas prácticas de programación.

Se espera también que los estudiantes ejerciten técnicas de composición de datos, programas por recurrencia y que adquieran experiencia en el uso de punteros y estructuras dinámicas.

Se espera que los estudiantes conozcan algoritmos clásicos de recorridas y ordenamiento sobre las estructuras aprendidas.

Se pretende que los estudiantes aprendan los conceptos básicos del análisis del tiempo de ejecución de los programas dependientes del tamaño de la entrada y los puedan emplear para comparar la eficiencia de los algoritmos y de las representaciones contemplando tanto nociones de tiempo de ejecución como de espacio de almacenamiento.

Finalmente se espera que los estudiantes puedan diseñar sus propias abstracciones de datos y elegir representaciones apropiadas.

4. METODOLOGÍA DE ENSEÑANZA

La enseñanza está centrada en clases teóricas, prácticas y en el trabajo de laboratorio.

Las clases teóricas presentan los temas abordados en el curso, mientras que las clases prácticas permiten realizar ejercicios y consultas, finalmente los trabajos de laboratorio tienen como objetivo que los estudiantes adquieran experiencia de programación en un lenguaje específico que les permita ejercitar los temas aprendidos.

Dedicación horaria:

- 2 hs semanales de exposiciones teóricas
- 4 hs semanales de clases prácticas
- 6 hs semanales de dedicación a los trabajos de laboratorio

durante las 15 semanas del curso

Las clases prácticas requieren que los estudiantes lean y resuelvan ejercicios con la guía del docente, una parte se realiza en las clases y el resto lo deben resolver los estudiantes con la posibilidad de consultar.
Los trabajos de laboratorio se realizan por cuenta propia, de forma individual o grupal, con ayuda de los docentes a través de clases de consultas y participación en foros del curso.

5. TEMARIO

1. Introducción y lenguaje: repaso conceptos de programación adquiridos en anteriormente, aplicando éstos en el lenguaje de programación usado.
2. Recursión: repaso sobre nociones sobre tipos inductivos, presentación de la programación mediante recursión.
3. Estructuras lineales en memoria dinámica: estructuras lineales en memoria dinámica. Repaso sobre pasaje de parámetros a funciones y procedimientos. Funciones y procedimientos totales y como parciales. Conceptos relacionados con Listas y algunas variantes como pueden ser Listas Doblemente Encadenadas, Listas Circulares y Listas Indizadas. .
4. Estructuras arborecentes: estructuras arborescentes de memoria dinámica Conceptos relacionados con Árboles y algunas variantes como pueden ser Árboles Binarios, Árboles Binarios de Búsqueda y Árboles Finitarios o Generales.
5. Introducción al análisis de algoritmos: conceptos básicos del análisis del tiempo de ejecución de los programas dependientes del tamaño de la entrada. Como la velocidad de crecimiento del tiempo de ejecución de los programas determina el tamaño de los problemas que se puede resolver y permite comparar diferentes algoritmos.
6. Tipos abstractos de datos: Tipo Abstracto de Datos Lista, reforzando el desacople entre especificación e implementación. Repaso de implementaciones vistas anteriormente. Aplicar concepto de "Orden de ejecución" para evaluar la

eficiencia contemplando nociones de tiempo de ejecución y de espacio de almacenamiento.

7. TAD Lista, Pila, cola: variantes de listas, tales como Pilas y Colas, que restringen las políticas de inserción, supresión y obtención de elementos. Desarrollo y análisis de implementaciones de Pilas y Colas eficientes. Uso Listas, Pilas y Colas para la resolución de problemas.
8. Colecciones: tipos abstractos de datos Diccionario, Conjunto, Multiconjunto, Tabla y Cola de Prioridad. Desarrollo y análisis de implementaciones eficientes. Uso para la resolución de problemas.
9. TAD Árbol: tipos abstractos de datos Árbol Binario de Búsqueda, Árbol Finitario y Árbol n-ario. Desarrollar y analizar implementaciones. Uso para la resolución de problemas simples y complejos.
10. Diseño de tipos abstractos de datos: Profundizar en diferentes especificaciones e implementaciones para TADs. Análisis de implementaciones de versiones acotadas y no acotadas, uso y conveniencia de y estructuras estáticas y dinámicas. Practicar en diseñar TADs adecuados para resolver nuevos problemas, tomando en cuenta requerimientos de eficiencia en tiempo o en espacio. Multiestructuras.

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
1 al 5	1,2	3,4
6 al 10	1,2	4

6.1 Básica

1. Weiss, Mark Allen Weiss. Data Structures and Algorithm Analysis in C (Second Edition) (1996). Perason. ISBN-13: 978-0201498400 ISBN-10: 0201498405
2. Aho, Alfred V., Ullman, Jeffrey D. Y Hopcroft, John E. (1983). Data Structures and Algorithms 1st Edition. Addison-Wesley. ISBN-13: 978-0201000238 ISBN-10: 0201000237

6.2 Complementaria

3. Deitel, Paul y Deitel Harvey (2012). C How to Program. Prentice Hall College. ISBN-10: 013299044X ISBN-13: 978-0132990448
4. Horton, Ivor. Beginning C (2013). Apress, 5th edition.

http://link.springer.com.proxy.timbo.org.uy:443/book/10.1007/978-1-4302-4882-8

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Son indispensables conocimientos básicos de programación, por ejmplo definición y uso de variables, estructuras de control condicionales e iterativas, procedimientos, funciones, pasaje de parámetros por valor y referencia, punteros y memoria dinámica.

7.2 Conocimientos Previos Recomendados: Conocimientos de Matemática Discreta.

ANEXO A
Para todas las Carreras

A1) INSTITUTO

Instituto de Computación

A2) CRONOGRAMA TENTATIVO

Consiste en un cronograma de avance semanal con detalle de las horas de clase asignadas a cada tema.

Semana 1	Tema 1 en el teórico.
Semana 2	Tema 2 en el teórico y tema 1 en el práctico. Lab Tarea 1.
Semana 3	Tema 3 en el teórico y tema 2 en el práctico. Lab Tarea 1.
Semana 4	Tema 4 en el teórico y tema 3 en el práctico. Lab Tarea 2.
Semana 5	Tema 5 en el teórico y tema 4 en el práctico. Lab Tarea 2.
Semana 6	Parcial
Semana 7	Tema 6 en el teórico y tema 5 en el práctico.Lab Tarea 3.
Semana 8	Tema 7 en el teórico y tema 6 en el práctico. Lab Tarea 3.
Semana 9	Tema 7 en el teórico y tema 7 en el práctico. Lab Tarea 3.
Semana 10	Tema 8 en el teórico y tema 7 en el práctico. Lab Tarea 4.
Semana 11	Tema 8 en el teórico y tema 8 en el práctico. Lab Tarea 4.
Semana 12	Tema 9 en el teórico y tema 8 en el práctico. Lab Tarea 4.
Semana 13	Tema 10 en el teórico y tema 9 en el práctico. Lab Tarea 4.
Semana 14	Tema 10 en el teórico y tema 10 en el práctico. Lab Tarea 5.
Semana 15	Repaso temas del curso. Lab Tarea 5.

El cronograma es tentativo y depende de los períodos de parciales, feriados y otros factores.

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

La unidad se evaluará por medio de dos pruebas parciales y trabajos de laboratorio. Las dos pruebas parciales tendrán una distribución 40% de los puntos para la primera y 60% para la segunda. Los trabajos de laboratorio son obligatorios y elminatorios. Los trabajos de laboratorio podrán asignar puntos para la aprobación o exoneración del curso.

Para exonerar el estudiante debe cumplir los siguientes requisitos:

- Aprobar el laboratorio.
- Reunir al menos un 60% del puntaje total del curso, sumando el puntaje de los parciales y de los trabajos de laboratorio.
- Obtener al menos un 25 % en cada uno de los dos parciales.

Para aprobar el curso el estudiante debe cumplir los siguientes requisitos:

- Aprobar el laboratorio.
- Reunir al menos un 25% del puntaje total del curso, sumando el puntaje de los parciales y de los trabajos de laboratorio.

No aprueban o exoneran el curso aquellos estudiantes que no cumplen con el mínimo en los trabajos de laboratorio o no obtienen el mínimo de 25% del puntaje total del curso.

A4) CALIDAD DE LIBRE
No se adhiere a la resolución de calidad de libre.

A5) CUPOS DE LA UNIDAD CURRICULAR
No corresponde.

ANEXO B para la carrera Ingeniería en Computación (plan 97)

B1) ÁREA DE FORMACIÓN

Programación.

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: Curso de Programación 1

Para el Examen: Curso de Programación 2