

## Programa de curso

<b>Profesor:</b>	: Juan Felipe Calderón
<b>Clases sección 1:</b>	: Martes y jueves, módulo 1 (8:30 - 9:50) - sala BC23
<b>Clases sección 2:</b>	: Martes y jueves, módulo 2 (10:00 - 11:20) - sala BC21
<b>Ayudantías ambas secciones:</b>	: Lunes, módulo 6 (17:00 - 18:20) - sala B16
<b>Sitio Web</b>	: DCC cursos (Moodle)
<b>Requisitos</b>	: IIC1103 Introducción a la Programación

## 1 Objetivos

Al finalizar el curso, el alumno estará capacitado para:

- Resolver problemas de diversa índole utilizando técnicas avanzadas de programación orientada a objetos.
- Utilizar herramientas modernas para el desarrollo de software.

## 2 Contenido

La siguiente es una lista (no necesariamente cronológica) de los tópicos que se revisarán durante el transcurso del semestre.

1. **Repaso IIC1103.** Revisión de los contenidos terminales del curso Introducción a la Programación.
2. **Programación Funcional.** Definición de conceptos. Diferencias con programación procedural. Introducción a F#.
3. **Programación Orientada al Objeto (POO).** Definición de clase y objeto. Diferencia entre clase y estructura. Elementos básicos de modelación OO: herencia, contratos (implementación de interfaces) y polimorfismo.
4. **Introducción a .Net** Aspectos básicos del Framework .NET, utilizando los lenguajes C# y F#. Uso del ambiente de desarrollo (Visual Studio).
5. **Estructuras de datos.** Definición, construcción y uso de estructuras de datos dinámicas simples: listas, pilas, colas y árboles.
6. **Streaming y archivos.** Periféricos de entrada y salida. Sistemas de archivos. Definición de streaming. Uso de streaming en archivos. XML. JSON. Serialización.
7. **Programación dirigida por Eventos.** Definición de delegación y manejo de eventos. Uso de estos modelos en .NET. Aplicación en programas con interfaz gráfica de usuario usando el componente Windows Presentation Foundation (WPF) de .NET.
8. **Generics.** Definición de tipos genéricos en .NET. Aplicación a eventos y estructuras de datos.

9. **Programación Multithread.** Funcionamiento de la CPU. Definición de proceso y thread. Concurrency y primitivas de sincronización.
10. **Comunicación por red.** Conceptos básicos de internet. Protocolos de comunicación. Sockets para la comunicación interprocesos. Webservices.
11. **Reflection.** Inspección de tipos, carga dinámica de bibliotecas (DLLs).

### 3 Metodología

El curso se desarrolla en base a clases expositivas y prácticas, ayudantías, y resolución de tareas.

Las clases presentan los contenidos del curso desde una perspectiva de explicación y aplicación de los conceptos a problemas prácticos.

Las ayudantías tienen como objetivo ejemplificar los contenidos vistos en clases y revisar contenidos complementarios de apoyo, además de aclarar dudas sobre las tareas.

Las tareas tienen como objetivo asegurar la resolución individual y grupal de problemas de programación aplicados a un dominio específico, siendo parte central dentro de la dinámica del curso.

De forma complementaria a las ayudantías existirá una sala de ayuda, donde cada semana un grupo de ayudantes del curso atenderá consultas particulares de los estudiantes.

Los apuntes del curso, las pautas de corrección de interrogaciones y material relevante de años anteriores, y toda la información pertinente al desarrollo del curso estarán disponibles en forma electrónica en el sitio Web del curso.

El canal oficial para la adquisición de los contenidos evaluados en el curso es la cátedra. Si bien se proveerán ejemplos y las *slides* usadas, éstas **no reemplazan** la asistencia a clases.

### 4 Tareas

Se encuentran programadas 7 tareas **individuales** de acuerdo al siguiente calendario:

- **Tarea 0:** martes 26 de agosto (23:59)
- **Tarea 1:** martes 09 de septiembre (23:59)
- **Tarea 2:** jueves 25 de septiembre (23:59)
- **Tarea 3:** jueves 09 de octubre (23:59)
- **Tarea 4:** martes 28 de octubre (23:59)
- **Tarea 5:** martes 11 de noviembre (23:59)
- **Tarea 6:** martes 02 de diciembre (23:59)

Su entrega se realizará en cuentas de repositorio Subversion (SVN), las que serán informadas a la brevedad.

Respecto a su evaluación, se tiene que:

$$T' = \frac{1}{7} \sum_{i=0}^6 T_i$$

entonces:

$$\bar{T} = \begin{cases} T' & \text{si } T_6 \geq 4.0 \\ \min\{3.9, P'\} & \text{si } T_6 < 4.0 \end{cases}$$

Como se puede apreciar en la fórmula, la tarea 6 es reprobatoria.

## 5 Evaluación

El desempeño de los estudiantes en el curso será evaluado mediante 2 interrogaciones (pruebas escritas), un examen y las tareas. Todas estas evaluaciones son **obligatorias**.

Las fechas de las evaluaciones escritas serán las siguientes:

- **Sección 1**

- $I_1$ : viernes 05 de septiembre (18:30)
- $I_2$ : miércoles 08 de octubre (18:30)
- **Examen**: lunes 01 de diciembre (8:30)

- **Sección 2**

- $I_1$ : viernes 12 de septiembre (18:30)
- $I_2$ : martes 14 de octubre (18:30)
- **Examen**: jueves 04 de diciembre (8:30)

**Nota: los estudiantes deben rendir las evaluaciones inscritas en las fechas correspondientes a la sección en las que están inscritos.**

Durante algunas clases se pasará asistencia, previo aviso en el sitio web del curso. Quienes tengan asistencia completa en estas revisiones, podrán reemplazar la nota de la peor interrogación con la nota del examen.

Para quienes hayan faltado a una interrogación, pueden ocurrir las siguientes situaciones:

- Quienes hayan justificado su inasistencia (mediante el procedimiento regular en la Dirección de Pregrado) automáticamente reemplazan la nota de la interrogación faltante por la del examen, sin necesidad de cumplir con el requisito de asistencia.
- Quienes NO hayan justificado su inasistencia, DEBEN cumplir con el requisito de asistencia antes mencionado. Si no lo hacen, quedan con un 1.0 en dicha interrogación.

En ambos casos, se pierde el derecho de reemplazar la nota de la peor interrogación por la del examen. La nota final del curso ( $N_f$ ) se calcula usando:

$$N'_f = 0.15I_1 + 0.15I_2 + 0.2E^1 + 0.5\bar{T}$$

---

<sup>1</sup>No hay eximición

Las condiciones para aprobar son las siguientes:

$$(0.3I_1 + 0.3I_2 + 0.4E) \geq 4.0, \bar{T} \geq 4.0, N'_f \geq 4.0$$

Considerando esto, la nota final será:

$$N_f = \begin{cases} N'_f & \text{si se cumplen todos los requisitos} \\ \min\{3.9, N'_f\} & \text{en caso contrario} \end{cases}$$

## 6 Contacto

- El medio de información oficial del curso es el sitio web, por lo que los alumnos están OBLIGADOS a consultarlo periódicamente.
- Cada alumno es responsable de revisar todas sus notas. Si existe algún desacuerdo, éstos sólo serán atendidos hasta cinco días hábiles después de que las notas hayan sido publicadas. Reclamos posteriores no serán considerados, aunque sean justificados.

## 7 Bibliografía Recomendada

- MSDN - Microsoft Development Network (<http://msdn.microsoft.com/>)
- Bertrand Meyer. Object-Oriented Software Construction. Second Edition. Prentice Hall, 1997.
- Sebesta, Robert. Concepts of programming languages. Addison-Wesley, 2002.
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides .Design Patterns: Element of Reusable Object Oriented Software. Addison-Wesley, 1995.
- Ivar Jacobson, Grady Booch, James Rumbaugh. The Unified Modeling Language: reference manual. Addison-Wesley, 1999.
- John Sharp. Microsoft Visual C# .NET: Step by Step. Microsoft Press, 2003.
- Bjarne Stroustrup. The C++ Programming Language. Addison Wesley. 2002

## 8 Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería:

[http://www.ing.puc.cl/esp/alumpregrado/procedimientos/integridad\\_academica.html](http://www.ing.puc.cl/esp/alumpregrado/procedimientos/integridad_academica.html)).

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los

efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno copia un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral. Por “copia” se entiende incluir en el trabajo presentado como propio partes hechas por otra persona.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile ([http://dsrd.uc.cl/images/stories/pdf/AlumnosUC/Info\\\_Academ/Reglamentos/alumnopregrado.pdf](http://dsrd.uc.cl/images/stories/pdf/AlumnosUC/Info\_Academ/Reglamentos/alumnopregrado.pdf)). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.