

## PROGRAMA DE CURSO

Código	Nombre			
<b>CC5104</b>	<b>LENGUAJES DE PROGRAMACION II</b>			
Nombre en Inglés				
Programming Languages II				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1.5	5.5
Requisitos			Carácter del Curso	
(CC4101 Lenguajes de Programación, CC3002 Metodologías de Diseño y Programación) / Autor			Curso Electivo	
Resultados de Aprendizaje				
<p>El curso entrega los elementos técnicos y formales necesarios para entender la problemática de garantizar que los programas cumplen con sus objetivos especificados, limitando efectos indeseados, lo más temprano posible. Se estudian las nociones de programas válidos, errores y excepciones, contratos, y sobre todo, tipos. En particular, se describen los conceptos fundamentales de los sistemas de tipos, permitiendo entender porque los lenguajes de hoy siguen evolucionando en ese aspecto (por ejemplo, el sistema de tipos de Java), y porque no existe una respuesta única y definitiva al problema.</p> <p>El alumno será capaz de entender la semántica estática de los lenguajes de programación actuales en sus detalles, así como los desafíos actuales en el área. Sera capaz además de relacionar un sistema de tipo con la semántica del lenguaje, probando su coherencia. Además de proveer un entendimiento más profundo de la semántica de los lenguajes de programación, el curso habilita a los alumnos para diseñar (extensiones de) lenguajes que proveen garantías fuertes de buen comportamiento.</p>				

Metodología Docente	Evaluación General
<p>Clases expositivas del profesor de cátedra. De manera de relacionar lo conceptual con lo concreto, el profesor combina explicación teórica en la pizarra con demostración en vivo de la implementación de los conceptos vistos, a través de la proyección con data show de la programación de varios lenguajes y sistemas de tipos.</p> <p>Clases auxiliares dedicadas a repasar puntos delicados vistos en clase, explicar ejemplos más extensos, resolver ejercicios propuestos, y preparación pre y post controles.</p>	<p>Se realizan tres controles para evaluar si se han cumplido los objetivos. El examen final evalúa todo el contenido del curso. Tanto los controles como el examen se enfocan en evaluar que el alumno haya comprendido los conceptos como sus implicancias concretas. La Nota de Control se calcula de la siguiente manera: Promedio ponderado del examen (40%) y del promedio de los controles (60%)</p> <p>Se realizan varias tareas a lo largo del curso. Las tareas consisten en implementar sistemas de tipos e intérpretes de lenguajes. Las tareas son individuales y se promedian a partes iguales para formar la nota de tareas.</p> <p>Controles y tareas se aprueban por separado y deben ser igual o superior a 4.0. La nota final es 2/3 de la nota de controles y 1/3 de la nota de tareas.</p>

### Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas	
1	Introducción	2	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Problemática de los sistemas de tipos Tipos, contratos y computabilidad Errores Semántica de lenguajes: enfoques Coherencia en tipos		Entender como detectar errores en programas antes de ejecutarlos, la importancia de lenguajes seguros, y de sistemas de tipos coherentes.	PLAI TAPL

Número	Nombre de la Unidad	Duración en Semanas	
2	Lambda calculo	2	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
El lambda calculo Semántica operacional Tipos simples para el lambda calculo Normalización fuerte		Descripción formal de un lenguaje, tanto en su semántica estática como dinámica. Entender como los tipos pueden enforzar propiedades fuertes.	TAPL PLAI

Número	Nombre de la Unidad	Duración en Semanas	
3	Extensiones al lambda calculo	3	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Formas derivadas Tipear: pares, tuplas, records, sumas, variantes Recursión Referencias Excepciones Tipos recursivos		Entender como varios mecanismos de lenguajes pueden ser definidos precisamente, extendiendo el simple núcleo formal anterior. Ser capaz de demostrar la coherencia de un diseño.	TAPL

Número	Nombre de la Unidad	Duración en Semanas	
4	Conexión con lógica constructiva	1	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Bases de lógica constructiva Proposiciones como tipos		Entender la conexión fundamental entre programas y construcciones lógicas	TAPL PFPL

Número	Nombre de la Unidad	Duración en Semanas	
5	Polimorfismo	2	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Polimorfismo de subtipos Coerciones Polimorfismo paramétrico		Entender la noción de polimorfismo, central en los lenguajes. Diferenciar formas de polimorfismo.	TAPL

Número	Nombre de la Unidad	Duración en Semanas	
6	Objetos	2	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Calculo de objetos Modelo formal de Java: Featherweight Java		Describir formalmente lenguajes orientados a objetos. Entender como probar propiedades sobre esos lenguajes, en particular Java.	TAPL

Número	Nombre de la Unidad	Duración en Semanas	
7	Inferencia de tipos	2	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Inferir tipos: enfoques y desafíos		Entender como funciona la reconstrucción de tipos, y sus ventajas.	PLAI TAPL

Número	Nombre de la Unidad	Duración en Semanas	
8	Perspectivas	1	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Lenguajes estáticamente tipeados vs. dinámicamente tipeados Sistemas de tipos y efectos Análisis de programas, etc.		Tener una visión abierta sobre los desafíos actuales en el área, y elementos de áreas relacionadas.	PFPL TAPL

Bibliografía	
Types and Programming Languages (TAPL), B. Pierce Programming Languages: Application and Interpretation (PLAI), S. Krishnamurthi Practical Foundations for Programming Languages (PFPL), R. Harper	

Vigencia desde:	Primavera 2010
Elaborado por:	Éric Tanter