

PROGRAMA DE CURSO

Código	Nombre			
CC5111	PROGRAMACIÓN AVANZADA EN SCALA			
Nombre en Inglés				
ADVANCED PROGRAMMING IN SCALA				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	0	7
Requisitos			Carácter del Curso	
CC4101 Lenguajes de Programación			Electivo para la ICC	
Resultados de Aprendizaje				
<p>El alumno será capacitado para entender y aplicar conceptos avanzados de programación, habiéndolos estudiados en el lenguaje Scala. En particular, el alumno tendrá un buen conocimiento de como la programación orientada a objetos y la programación funcional se complementan, y como explotar sus ventajas respectivas en contextos específicos con lenguajes que combinan ambos paradigmas. Además, el alumno profundizará el manejo de sistemas de tipos avanzados que proveen fuertes garantías estáticas, entendiendo las fortalezas y debilidades del sistema de tipos de Scala en comparación con el de otros lenguajes. El alumno manejará varios mecanismos avanzados de programación como traits, parámetros implícitos, continuaciones, reflexión, concurrencia y actores, que contribuyen al desarrollo de sistemas de software más escalables y robustos.</p>				

Metodología Docente	Evaluación General
<p>Clases expositivas del profesor de cátedra, que incluyen momentos dedicados a repasar puntos delicados vistos anteriormente, explicar ejemplos extensos, y resolver ejercicios.</p> <p>De manera de relacionar lo conceptual con lo práctico, el profesor combina explicación teórica en la pizarra con exposición en vivo de la implementación de los conceptos vistos, a través de la proyección con data show de la programación de varios ejemplos ilustrando dichos conceptos.</p> <p>Se incita a los alumnos a que lleven consigo un computador para que puedan programar los ejemplos y ejercicios de manera directa, fomentando así el aprendizaje activo.</p>	<p>Se realiza una evaluación continua a través de “mini-controles” regulares, en particular cada vez que se completa el estudio de una unidad temática. Los mini-controles tienen una duración de 15-30 min y se llevan a cabo en horario de clases. Apuntan a evaluar la buena asimilación de los conceptos, con preguntas enfocadas y pequeños ejercicios de programación. Las notas de controles se promedian para formar la nota de control.</p> <p>Se realizan varias tareas a lo largo del curso para validar la capacidad del alumno a aplicar los conceptos y mecanismos vistos en clase en ejemplos de complejidad mediana. Las tareas son individuales y se promedian a partes iguales para formar la nota de tareas.</p> <p>Para aprobar el curso, la nota de control y la nota de tareas deben ser ambas igual o superior a 4.0.</p> <p>Si el promedio final del alumno es igual o superior a 5.0, éste no tendrá que rendir un examen final. El examen final será personalizado y podrá tener una componente conceptual (control) y/o una componente práctica (tarea), según los resultados obtenidos por el alumno.</p>

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Introducción	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Intro a Scala Breve tour del lenguaje Herramientas de desarrollo	Entender la motivación y los objetivos del lenguaje Scala. Ser operacional para programar	[1] 1, 2, 3 [2] 1, 2, 3, 4

Número	Nombre de la Unidad	Duración en Semanas
2	Clases, Objetos, Tipos y Operaciones Básicas	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Clases y objetos Tipos y operaciones básicas Estructuras de control predefinidas	Desarrollar programas básicos con objetos Conexión con: programación básica en Java	[1] 4, 5, 6, 7 [2] 5, 6

Número	Nombre de la Unidad	Duración en Semanas
3	Funciones y Abstracciones de Control	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Funciones y clausuras Currificación Recursión por la cola Abstracciones de control Parámetros por nombre	Desarrollar programas básicos con funciones. Saber definir abstracciones de control específicas. Conexión con: programación funcional en Scheme, evaluación perezosa en Haskell, macros en Scheme.	[1] 8, 9 [2] 12

Número	Nombre de la Unidad	Duración en Semanas
4	Herencia y Traits	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Clases abstractas, sobrescritura, polimorfismo, enlace dinámico Jerarquía de clases de Scala Traits.	Saber organizar clases con herencia y traits. Entender cuando usar traits. Conocer las clases estándares de Scala. Conexión con: interfaces en Java, herencia múltiple en Python y C++, traits en Smalltalk.	[1] 10, 11, 12 [2] 8, 10

Número	Nombre de la Unidad	Duración en Semanas
5	Clases de Casos, Pattern Matching y Extensibilidad	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Clases de casos Pattern matching Clases selladas Extensibilidad: OO vs. ADT.	Entender la motivación de tener clases de casos y selladas (ADT). Manejar el pattern matching. Entender los distintos beneficios y limitaciones de objetos y ADTs en cuanto a extensibilidad. Saber usar los mecanismos apropiados según el caso. Conexión con: tipos de datos algebraicos en Racket, Haskell, ML.	[1] 15 [2] 14 [4]

Número	Nombre de la Unidad	Duración en Semanas
6	Colecciones	1.5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Traits principales Colecciones mutables e inmutables Operaciones sobre colecciones Streams y vistas perezosas Interoperabilidad con Java.	Familiarizarse con las colecciones más importantes. Entender la diferencia entre colecciones mutables e inmutables. Saber manejar streams y otras colecciones avanzadas. Usar librerías Java desde Scala. Conexión con: colecciones de Java, estructuras infinitas en Haskell, streams en Shell.	[1] 16, 17, 24 [2] 13

Número	Nombre de la Unidad	Duración en Semanas
7	Tipos	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Inferencia de tipos Clases genéricas Funciones genéricas Límites de variables de tipo Restricciones de tipo Variancia.	Entender la inferencia de tipos de Scala. Saber definir programas genéricos de manera precisa. Conexión con: generics en Java, polimorfismo en Haskell.	[1] 19 [2] 17

Número	Nombre de la Unidad	Duración en Semanas
8	Tipos Avanzados	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Tipos singleton Proyecciones de tipos y caminos Tipos estructurales Tipos compuestos Tipos existenciales Tipos self Tipos abstractos Polimorfismo de familia Tipos superiores Tipos dinámicos.	Entender los múltiples mecanismos que el sistema de tipos de Scala soporta, y sus aplicaciones para definir librerías reusables y robustas. Evaluar el compromiso entre garantías estáticas y complejidad. Conexión con: Java, Haskell, lenguajes con tipos dinámicos.	[1] 20 [2] 18

Número	Nombre de la Unidad	Duración en Semanas
9	Conversiones Implícitas y Parámetros Implícitos	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Conversiones implícitas Reglas y aplicaciones Parámetros implícitos Aplicaciones.	Saber usar conversiones implícitas para enriquecer librerías existentes. Saber usar parámetros implícitos para mejorar la usabilidad de librerías . Entender el uso de implícitos en las librerías de Scala.	[1] 21 [2] 21

Número	Nombre de la Unidad	Duración en Semanas
10	Programación Modular	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Objetos como módulos Organizar módulos con traits Enlace dinámico de módulos	Saber definir programas modulares con objetos, que permitan configuración flexible, dinámica, y robusta. Conexión con: inyección de dependencia en frameworks como Spring y Guice.	[1] 29

Número	Nombre de la Unidad	Duración en Semanas
11	Actores y Concurrencia	1.5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Threads vs actores Actores y envío de mensajes Canales Futuros Diseño de aplicaciones con actores.	Entender las ventajas del modelo de actores en comparación con la programación con threads. Saber implementar aplicaciones basadas en actores. Conexión con: threads en Java, actores en Erlang.	[1] 32 [2] 20 [3]

Número	Nombre de la Unidad	Duración en Semanas
12	Continuaciones	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Capturar e invocar continuaciones reset/shift Aplicaciones Transformación CPS Inversión de control	Entender el interés de manipular continuaciones para programas reactivos (GUI, web) y calculos suspendibles. Conexión con: Racket, servidores web basados en continuaciones como Continue y Seaside, programación de GUIs.	[2] 22

Número	Nombre de la Unidad	Duración en Semanas
13	Macros y Reflexion	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Introducción a la reflexión Macros Reflexión y generación de programas.	Conocimiento básico de las herramientas de metaprogramación. Conexión con: Macros en Scheme, reflexión en Java y Smalltalk.	[5]

Bibliografía	
[1] M. Odersky, L. Spoon, B. Venners. Programming in Scala. 2nd Edition. Artima. 2010	
[2] C. S. Hortsman. Scala for the Impatient. Addison Wesley. 2012	
[3] P. Haller, F. Sommers. Actors in Scala. Artima. 2011	
[4] W. Cook. Understanding Data Abstraction, Revisited. Proc. Onward 2009	
[5] http://docs.scala-lang.org	

Vigencia desde:	PRIMAVERA 2013
Elaborado por:	Eric Tanter