



Código	Nombre			
MDS7202	Laboratorio de Programación Científica para Ciencia de Datos			
Nombre en inglés				
Scientific Programming Laboratory for Data Science				
CT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
(EL4106/MA5204/MDS7104),(MA3402/IN3401/IN3242/MDS7101/CC6104)			Obligatorio Magíster en Ciencia de Datos	
Resultados de Aprendizaje				
<p>El propósito del curso es que los estudiantes del Magíster lleven a cabo un proceso investigativo de Ciencia de Datos, logrando consolidar correctamente los contenidos de los cursos de Estadísticas y Aprendizaje de Máquinas. Podrán investigar proactivamente sobre un tema de interés, logrando plantear una pregunta de investigación relevante, encontrar literatura pertinente, evaluar métodos y herramientas para cada problema y ser capaz de comunicar modelos y estimaciones.</p> <p>Dentro de las competencias que se espera que el estudiante desarrolle están:</p> <ul style="list-style-type: none"> - Contará con el conocimiento y dominio de variados recursos computacionales existentes para análisis de datos. - Identificará qué herramienta computacional utilizar para cada problema de análisis de datos y cómo adaptarla para el problema en cuestión. - Será capaz de diseñar nuevas herramientas computacionales para resolver problemas de análisis de datos en caso de ser necesario. - Comprender, analizar y aplicar técnicas estadísticas avanzadas y herramientas para el análisis y tratamiento de datos <p>Al finalizar el curso, el estudiante:</p> <ul style="list-style-type: none"> - Podrá comunicar efectivamente las soluciones encontradas a través de visualización de datos y modelos. - Será capaz de investigar y evaluar herramientas y técnicas más allá de los contenidos del curso para proponer soluciones a problemas reales. - Contará con conocimientos iniciales de sistemas de Computación de Alto Rendimiento. - Será capaz de investigar por su cuenta la literatura de vanguardia con respecto a su tema de interés. 				



Metodología Docente	Evaluación General
<p>Este curso tiene una connotación teórica y práctica. Está compuesto por cátedras y demostraciones del uso de las herramientas de programación.</p> <p>En paralelo a las cátedras, el curso cuenta con un trabajo de investigación a lo largo del semestre. Tal trabajo de investigación es individual y propuesto por el(la) propio(a) estudiante, proponiendo la pregunta de investigación, antecedentes, bibliografía y metodología.</p> <p>Se espera que el(la) estudiante sea capaz de construir modelos para su propio problema, más allá de los enseñados en el curso, a partir de la revisión de literatura.</p> <p>Es importante destacar que el trabajo de investigación se centra en la proactividad del (de la) estudiante, teniendo una relación de tutoría con el(la) profesor(a).</p> <p>Al final del semestre se realiza una defensa del trabajo investigativo de cada estudiante.</p>	<p>El curso se evalúa a partir de hitos de entrega que medirán el cumplimiento de los objetivos propuesto al comienzo del semestre a los(as) estudiantes.</p> <p>El cálculo de esas notas se efectúa de la siguiente forma:</p> <ul style="list-style-type: none"> • NT = Promedio de las entregas parciales $(\sum w_i \cdot P_i)/n$, donde P_i son las notas de las tareas y w_i la ponderación que tiene cada una de ellas. <p>NP= Nota de participación final</p> <p>Nota Final= $0,1 \cdot NP + 0,9 \cdot NT$</p> <ul style="list-style-type: none"> • La condición para aprobar el curso es: NP \geq 4.0, NT \geq 4.0

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Introducción a la Programación Científica	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>6. Elementos Básicos de programación en Python</p> <p>7. Programación sin librerías y orientada a objetos</p> <p>8. Paquetes para programación científica: Numpy, Scipy, etc.</p>	<p>Los(as) estudiantes se familiarizan con el lenguaje de programación Python, sus métodos generales y la programación orientada a objetos. Luego, los estudiantes aprenden a utilizar los paquetes básicos para la computación científica.</p>	1-3



Número	Nombre de la Unidad	Duración en Semanas
2	Visualización De Datos	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
12. Matplotlib: gráficos 1D/2D 13. Histogramas y estimación de densidades 14. Graficar datos, estimaciones y modelos 15. Altas dimensiones vía reducción de dimensionalidad	Los estudiantes aprenden cómo comunicar modelos y estimaciones con el fin de presentar soluciones de alta complejidad.	1-5

Número	Nombre de la Unidad	Duración en Semanas
3	Manejo De Datos	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
3. Pandas: cargar y limpiar datos 4. Operaciones básicas con datos 5. Agrupar y exportar datos 6. Visualización	Los(as) estudiantes	1-5

Número	Nombre de la Unidad	Duración en Semanas
4	Modelos de Programación Científica	4
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Construcción de modelos 2. Optimización de modelos 3. ScikitLearn: librería de modelos	Los(as) estudiantes son capaces de construir modelos optimizados, evaluando su rendimiento e implementando las soluciones a problemas reales.	1-5

Número	Nombre de la Unidad	Duración en Semanas
5	Construcción Avanzada de Modelos	4
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Programación simbólica: TensorFlow 2. Modelos generativos e inferencia 3. Redes neuronales y entrenamiento	Los(as) estudiantes son capaces de conocer las estructuras estándares de redes neuronales y técnicas de entrenamiento. Además, los estudiantes aprenderán los	1-5



	paradigmas de programación probabilista y simbólica.	
--	--	--

Número	Nombre de la Unidad	Duración en Semanas
6	Computación de Alto Rendimiento	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Arquitectura de un sistema de HPC 2. Sistemas de colas y job scriping 3. Python paralelo y distribuido	Los(as) estudiantes se introducen en HPC, empezando a comprender la arquitectura de un sistema y las interdependencias entre la evolución de la tecnología y la arquitectura de estos procesadores	6-7

Bibliografía General
1. Joel Grus, Data Science from Scratch, First Principles with Python, O'Reilly, 2015 2. Wes McKinney, Python for Data Analysis, Data Wrangling with Pandas, NumPy, and IPython, O'Reilly, 2012 3. Jake VanderPlas, Python Data Science Handbook Essential Tools for Working with Data, O'Reilly, 2016 4. Cameron Davidson-Pilon, Bayesian Methods for Hackers, Probabilistic Programming and Bayesian Inference, Addison-Wesley, 2016 5. Sebastian Raschka, Python Machine Learning, Packt, 2016 6. J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012. 7. K.N. King, C Programming: A Modern Approach, W. W. Norton & Company. 2008 8. B. Chapman and G. Jost and R. van der Pas, Using OpenMP: Portable Shared Memory Parallel Programming, 2007. 9. P.S. Pacheco, An Introduction to Parallel Programming. Burlington. MA: Elsevier. 2011. 10. G.E. Karniadakis, R.M. Kirby, Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation, Cambridge University Press. 2003. 11. M. Snir, S.W. Otto, S. Huss-Lederman, D.W. Walker, J. Dongarra, MPI: The Complete Reference, MIT Press. 1995.

Vigencia desde:	Diciembre 2018
Elaborado por:	Felipe Tobar
Validado por:	Comité Académico Magíster en Ciencia de Datos
Revisado por:	Pablo Estévez y Jorge F Silva