

CARRERA: Ciencias de Datos	CURSO LECTIVO: 2024
CÁTEDRA: Programación Funcional	CURSO: 2º año - 2º semestre
DURACIÓN: Semestral	Hs. TOTALES: 48
SEMANAS: 15	Hs. TEÓRICAS: 16 Hs. PRÁCTICAS: 32

PROFESORA ASISTENTE: Crespo Colina, Diliana

1. OBJETIVOS DE LA ASIGNATURA

Que los alumnos logren:

- Dominar los elementos fundamentales del paradigma de programación funcional.
- Comprender la importancia de la programación funcional en el procesamiento y la analítica de datos.
- Relacionar el Cálculo Lambda de Church con el enfoque de Máquina de Turing.
- Reconocer la importancia del enfoque de programación funcional en Big Data.
- Comprender los mecanismos intrínsecos del funcionamiento de la plataforma Spark.
- Dominar las acciones y transformaciones distribuidas típicas del enfoque funcional en Spark.
- Poder operar transformaciones de datos de manera eficiente en ambientes distribuidos desde el conocimiento intrínseco de la plataforma Spark.

2. UNIDADES TEMÁTICAS

1. Introducción.

Programación Funcional. Objetivo. Distinción del paradigma Estructurado, Orientado a Objetos y Orientado a Datos. Características principales. Lenguajes funcionales. Vinculación con Datos. Presentación plataforma Spark en nube AWS para práctica.

II. Fundamentos teóricos.

La Máquina de Turing y el Entscheidungsproblem (problema de decisión). Máquina de Turing. La tesis de Church-Turing. Cálculo Lambda. Abstracción Lambda y Funciones Anónimas. Sintaxis del Cálculo Lambda. Reglas de Evaluación del Cálculo Lambda. Variables vinculadas y variables libres. Reglas de Evaluación. Regla delta. Reducción beta. Conversión alfa. Reducción eta. Teoremas de Church-Rosser.

III. Nociones de Programación Funcional.

Principio de Inmutabilidad. Principio de Transparencia Referencial (Referential Transparency). Funciones de órdenes superiores (Higher Order Functions). Lazy Evaluation. El pensamiento computacional desde el paradigma Funcional Beneficios de la Programación Funcional. Patrones de Diseño funcionales. The Functor Pattern. Monoids. Natural Transformations. Monads

IV. Lenguaje Scala.

Introducción al lenguaje Scala. Variables y valores. Clases y Objetos. Funciones. Case Classes. Declaración de Funciones. Currying. Funciones Anónimas. Higher Order Functions. Expresiones Regulares. Pattern Matching. Traits. Diferencia entre clases abstractas y Traits. Lazy Evaluation. Type Parameters. El tipo "Option". Future. Higher Order Functions Relevantes.

V. Estructuras de Datos Funcionales.

Concepto. La estructura de datos "Option". La estructura de datos "Try". La estructura de datos "Either". Funciones de órdenes superiores. "For comprehensions". Estructuras de Datos tradicionales. Inmutabilidad e historia. Lazyness. Variables mutables e Inmutables. Recursión. Tail Recursion. Conexión entre "fold" y "monoids". Map. FlatMap.

VI. Spark en el espacio Big Data.

¿A qué se refiere Big Data? Antecedentes. ACID-BASE. El sistema de Archivos Distribuido (HDFS). Procesamiento con Map-Reduce. Introducción a Spark y PySpark. Ecosistema Spark 3.0. Uso básico del Shell de Spark. Introducción al Data Analysis con PySpark. Arquitectura de Spark. RDD (Resilient Distributed Data). Operaciones con RDD. Transformaciones (map, filter, flatMap, mapPartitions, mapPartitionsWithIndex, sample, union, intersection, distinct, groupByKey, reduceByKey, aggregateByKey, sortByKey, join, cogroup, cartesian, pipe, coalesce, repartition). Acciones (reduce, collect, count, first(), take, takeSample, takeOrdered, saveAsTextFile, saveAsSequenceFile, saveAsObjectFile, countByKey, foreach).

VII. Spark SQL, Dataframes y Datasets

Creación de DataFrames distribuidos en Spark. Untyped Dataset Operations (DataFrame operations). Corriendo SQL en Spark programáticamente. Global Temporary View. Creación de Datasets. Interacción con RDD. Funciones Escalares y Agregadas. Fuentes de Datos (Parquet, ORC, JSON, CSV, Text Files, Hive Tables, JDBC, Avro Files, Protobuf data). Motor de SQL Distribuido (integración con otras plataformas y nubes).

VIII. Temas de Analítica Avanzada en Spark

GraphX como gestor de datos en Grafos. ML en Spark. MLLIB. Caso de estudio: Understanding Wikipedia with LDA and Spark NLP.

3. BIBLIOGRAFÍA

3.1 BIBLIOGRAFÍA GENERAL OBLIGATORIA

- “Learning Functional Programming” - Jack Widman – O'Reilly – 2022 - Disponible en Biblioteca: <https://learning.oreilly.com/library/view/learning-functional-programming/9781098111748/>
- “Advanced Analytics with PySpark” - Akash Tandon, Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills – O'Reilly - 2022 - Disponible en Biblioteca: <https://learning.oreilly.com/library/view/advanced-analytics-with/9781098103644/>
- “Data Analysis with Python and PySpark” - Johnatan Rioux - O'Reilly – 2022 - Disponible en Biblioteca: <https://learning.oreilly.com/library/view/data-analysis-with/9781617297205/>
- Documentación oficial de Apache Spark. Disponible (actualizada) en <https://spark.apache.org/docs/latest/>

3.2 BIBLIOGRAFÍA ADICIONAL

- “La mente nueva del Emperador” - Roger Penrose - 1989 - Audiolibro disponible en Biblioteca: <https://learning.oreilly.com/videos/-/9781618033321/>
- “The implementation of Functional Programming Languages” - Simon Peyton Jones – 1987 - On-line en <https://www.microsoft.com/en-us/research/wp-content/uploads/1987/01/slpj-book-1987-small.pdf>

4. METODOLOGÍA

El curso está organizado en 8 unidades temáticas divididas en encuentros de 3hs reloj semanales, a realizarse en formato presencial. La modalidad adoptada para el dictado será **teórico-práctica**. En las clases se presentarán los temas de cada unidad, proponiendo espacios de intercambio con el docente y entre los estudiantes a partir de consignas específicas. Se facilitará material de lectura obligatoria y complementaria para complementar la comprensión de las unidades.

5. EVALUACIONES Y CRITERIOS PARA LA APROBACIÓN

La aprobación de la materia estará supeditada al cumplimiento de la condición de asistencia exigida por la Universidad, la aprobación de todas las actividades prácticas y la aprobación del examen integrador.

Los trabajos prácticos podrán ser individuales o grupales, debiéndose cargar a través de la plataforma de Entornos Virtuales de Aprendizaje en tiempo y forma, otorgándole una única instancia de revisión y recuperación. Las actividades prácticas deberán contar con su aprobación para acceder a la instancia de evaluación final.

Para los trabajos prácticos y la evaluación final se realizarán sesiones de consultas individuales y grupales, haciendo además puesta en común general si el caso lo requiera. A los estudiantes que presenten dificultades se les observará y se los guiará para resolver el conflicto.

La instancia de recuperación está prevista para estudiantes que no hayan aprobado el examen integrador o que hayan estado ausentes.

Criterios de Evaluación:

- Respeto de las consignas presentadas.
- Resolución correcta de los problemas planteados.
- Adecuada respuesta a los contenidos teóricos.

6. CRITERIOS y MODALIDAD PARA LA EVALUACIÓN DEL EXAMEN FINAL

El examen final consiste en una evaluación oral y escrita, presencial e individual, donde el alumno deberá demostrar conocimientos teóricos y prácticos. El examen final se diferencia en que abarca todos los temas del programa. Los ejercicios prácticos tendrán un carácter integrador, articulando los distintos contenidos vistos en la materia. Finalmente, en las preguntas teóricas se pretende que el alumno demuestre un conocimiento profundo de los temas, relacionando conceptos entre sí.

Criterios de Evaluación:

- Adecuada respuesta a los contenidos teóricos.
- Relación de conceptos pertinente.
- Resolución correcta de los problemas planteados.
- Fundamentación bibliográfica de los temas.
- Respeto de las consignas presentadas.

Se agrega la posibilidad de defender un trabajo final en la instancia del examen final. Esta opción se habilitará únicamente a los alumnos que hayan aprobado la cursada con una nota de 7 (siete) o mayor.