

SPARK JAVA-Parte I



Objetivos

Spark es un framework basado en Java (utiliza la versión JDK 8 en adelante) sencillo y muy ligero que permite desarrollar aplicaciones web. El objetivo de este trabajo práctico es utilizar dicho framework para construir servicios REST (microservicios) de mínima complejidad bajo el concepto arquitectónico Modelo-Vista-Controlador (MVC del inglés), utilizando los conocimientos básicos del lenguaje JAVA que ya posee.

Las actividades correspondientes al taller de Spark, se dividen en tres secciones. Esta sección I tiene con objetivo particular construir los modelos e implementar servicios web REST básicos de forma que pueda ser subida a la nube y ser invocado desde cualquier lugar, con cualquier plataforma y cualquier lenguaje de programación.



Herramientas / Material de Consulta

Para llevar a cabo los ejercicios, se debe:

- Tener instalado [Java Development Kit 8](#) (JDK 8). Se puede instalar en Windows, Mac o Linux y descargar de la página oficial de Oracle
- Tener instalado un entorno de desarrollo (IDE) para Java, tal como Eclipse, IntelliJ o NetBeans. La cátedra sugiere la versión 8.0.2 de [NetBeans para Java EE](#) o superior.

Material de consulta:

- Documentación online: <http://sparkjava.com/documentation>
- Material Taller Spark Parte I: Disponible online en el campus en el espacio compartido para ambas materias.
- Consultas: Utilicen las clases prácticas para realizar consultas específicas de Spark Java para la realización de cada ejercicio. Fuera del horario de las

mismas, pueden consultar en el canal "Spark-Parte I", de Slack, para lo cual, deben estar suscripto al mismo con anterioridad.



Organización

Grupal de 2 a 3 personas.



Fecha de entrega / presentación

Fecha tope de entrega y presentación: **14/09/2022**



Ejercicios

Para los ejercicios 1 al 12, se pide: (1) realizar el modelo de diseño (utilizando un diagrama de clases) para cada enunciado y (2) crear servicios REST que resuelvan cada consigna y sigan el diseño realizado.

Nota: A partir del ejercicio 6 los resultados se piden que sean devueltos en formato JSON.

- **Ejercicio 1:** Dada una palabra ingresada por el usuario, devolver si es palíndromo o no.
- **Ejercicio 2:** Dada una palabra ingresada por el usuario, devolver su longitud.
- **Ejercicio 3:** Dado el radio de una circunferencia ingresados por el usuario, calcular su perímetro y área.
- **Ejercicio 4:** Dados dos números ingresados por el usuario, realizar la suma y resta de ambos.
- **Ejercicio 5:** Dada una cantidad de segundos ingresados por el usuario, convertirlos en hora, minutos y segundos.
- **Ejercicio 6:** Devolver la sucesión Fibonacci de una orden ingresada por el usuario.

- **Ejercicio 7:** Registrar en memoria un conjunto de personas, sus hijos y los nietos (utilice listas parametrizadas). Obtener las personas registradas y los hijos y los nietos de alguna de esas personas.
- **Ejercicio 8:** Dado un equipo de fútbol solicitado por el usuario, devolver los partidos de fútbol jugados por el equipo, en qué campeonatos y el resultado obtenido. Por ejemplo, al consultar sobre “Boca”, se deberá devolver:
 - `[{"equipo1": "River", "equipo2": "Boca", "resultado": "Ganó River 3-1", "ronda": "Final Libertadores", "jugado": true}]`
- **Ejercicio 9:** Devolver los teléfonos para emergencias, según el dato solicitado por el usuario. Por ejemplo, si solicita “policía”, que devuelva “911”. Lo mismo con los bomberos, violencia de género y otros, cargados en memoria durante la ejecución del programa.
- **Ejercicio 10:** Dado un monto en dinero ingresados por el usuario, devolver el valor que tendrá el mismo luego de un plazo indicado por el usuario (ej.: 1 año, 5 meses, etc), de acuerdo con la inflación propuesta por el gobierno Argentino.
- **Ejercicio 11:** Dado un artículo por código solicitado por el usuario, retornar la descripción, el precio neto, el IVA y el total (neto+IVA) del mismo.
- **Ejercicio 12:** Dado un autor ingresado por el usuario, devolver los libros escritos por él.