**MODULO 1**

* Las categorías de tareas de la ciencia de datos incluyen:
* Gestión de datos - almacenamiento, gestión y recuperación de datos
* Integración y transformación de datos - agilizar los conductos de datos y automatizar las tareas de procesamiento de datos
* Visualización de datos - proporcionan una representación gráfica de los datos y ayudan a comunicar las percepciones
* Modelización - permiten construir, desplegar, supervisar y evaluar modelos de datos y de aprendizaje automático
* Las tareas de ciencia de datos dan soporte a lo siguiente
* Gestión de activos de código - almacenar y gestionar el código, realizar un seguimiento de los cambios y permitir el desarrollo colaborativo
* Gestión de activos de datos - organizar y gestionar los datos, proporcionar control de acceso y realizar copias de seguridad de los activos
* Entornos de desarrollo - desarrollan, prueban y despliegan el código
* Entornos de ejecución - proporcionan recursos computacionales y ejecutan el código

El ecosistema de la ciencia de datos consta de muchas opciones comerciales y de código abierto, e incluye tanto aplicaciones tradicionales de escritorio como herramientas basadas en servidores, así como servicios basados en la nube a los que se puede acceder mediante navegadores web e interfaces móviles.

**Herramientas de gestión de datos**: incluyen bases de datos relacionales, bases de datos NoSQL y plataformas de Big Data:

* MySQL y PostgreSQL son ejemplos de sistemas de gestión de bases de datos relacionales (RDBMS) de código abierto, e IBM Db2 y SQL Server son ejemplos de RDBMS comerciales y también están disponibles como servicios en la nube.
* MongoDB y Apache Cassandra son ejemplos de bases de datos NoSQL.
* Apache Hadoop y Apache Spark se utilizan para el análisis de Big Data.

Herramientas de **integración y transformación** de datos: incluyen Apache Airflow y Apache Kafka.

Herramientas de**visualización** de datos: incluyen ofertas comerciales como Cognos Analytics, Tableau y PowerBI y pueden utilizarse para construir cuadros de mando dinámicos e interactivos.

**Herramientas de gestión de activos de código:** Git es una herramienta esencial de gestión de activos de código. GitHub es una popular plataforma basada en web para almacenar y gestionar el código fuente. Sus características la convierten en una herramienta ideal para el desarrollo colaborativo de software, incluyendo el control de versiones, el seguimiento de incidencias y la gestión de proyectos.

**Entornos de desarrollo:** Los entornos de desarrollo más populares para la Ciencia de Datos incluyen Jupyter Notebooks y RStudio.

* Jupyter Notebooks proporciona un entorno interactivo para crear y compartir código, texto descriptivo, visualizaciones de datos y otros artefactos computacionales en una interfaz basada en un navegador web.
* RStudio es un entorno de desarrollo integrado (IDE) diseñado específicamente para trabajar con el lenguaje de programación R, que es una herramienta popular para la computación estadística y el análisis de datos.

**MODULO 2**

Debe seleccionar un lenguaje para aprender en función de sus necesidades, los problemas que intenta resolver y para quién los resuelve.

* Los lenguajes más populares son Python, R, SQL, Scala, Java, C++ y Julia.
* Para la ciencia de datos, puede utilizar las bibliotecas de computación científica de Python como Pandas, NumPy, SciPy y Matplotlib.
* Python también puede utilizarse para el Procesamiento del Lenguaje Natural (PLN) utilizando el Kit de Herramientas de Lenguaje Natural (NLTK).
* Python es de código abierto, y R es software libre.
* La sintaxis orientada a matrices del lenguaje R facilita la traducción de las matemáticas al código a los alumnos sin conocimientos de programación o con conocimientos mínimos.
* SQL es diferente de otros lenguajes de desarrollo de software porque es un lenguaje no procedimental.
* SQL se diseñó para gestionar datos en bases de datos relacionales.
* Si aprende SQL y lo utiliza con una base de datos, podrá aplicar sus conocimientos de SQL con muchas otras bases de datos fácilmente.
* Las herramientas de ciencia de datos construidas con Java incluyen Weka, Java-ML, Apache MLlib y Deeplearning4.
* Para la ciencia de datos, un programa popular construido con Scala es Apache Spark que incluye Shark, MLlib, GraphX, y Spark Streaming.
* Los programas construidos para la Ciencia de Datos con JavaScript incluyen TensorFlow.js y R-js.
* Una gran aplicación de Julia para la Ciencia de Datos es JuliaDB.

**MODULO 3**

* Python ofrece un variado ecosistema de librerías para la Ciencia de datos, que abarca la computación científica (Pandas, NumPy), la visualización (Matplotlib, Seaborn) y el Aprendizaje automático de alto nivel (Scikit-learn). Estas bibliotecas ofrecen herramientas para la manipulación de datos, operaciones matemáticas y desarrollo simplificado de modelos de aprendizaje automático.
* Las interfaces de programación de aplicaciones (API) facilitan la comunicación entre componentes de software. Las API REST, en concreto, facilitan la comunicación por Internet y el acceso a recursos como el almacenamiento. Los términos clave de las API son cliente (usuario o código que accede a ella), recurso (servicio o datos) y punto final (URL de la API).
* Los modelos de aprendizaje automático analizan los datos e identifican patrones para hacer predicciones y automatizar tareas complejas: los tres tipos fundamentales de aprendizaje automático son el aprendizaje supervisado, el no supervisado y el de refuerzo. El Aprendizaje supervisado incluye modelos de regresión y clasificación para el modelado predictivo y el reconocimiento de patrones. El Aprendizaje profundo, un subconjunto avanzado del aprendizaje automático, imita el procesamiento cerebral y permite resolver problemas complejos en diversos ámbitos.
* El Acuerdo de Licencia de Datos Comunitarios (CDLA) facilita la puesta en común de datos abiertos proporcionando condiciones de licencia claras para su distribución y uso, y el sitio IBM Data Asset eXchange (DAX) contiene conjuntos de datos abiertos de alta calidad.
* Model Asset eXchange (MAX) proporciona una gran cantidad de modelos de Aprendizaje profundo preentrenados, lo que permite a los desarrolladores desplegar fácilmente soluciones para diversos retos empresariales.

**MODULO 4**

* Los cuadernos Jupyter se utilizan en Ciencia de Datos para registrar experimentos y proyectos.
* Los Cuadernos Jupyter son compatibles con muchos archivos y lenguajes de Ciencia de Datos.
* Hay diferentes formas de instalar y utilizar los Cuadernos Jupyter.
* Cómo ejecutar, borrar e insertar una celda de código en los cuadernos Jupyter.
* Cómo ejecutar varios cuadernos al mismo tiempo.
* Cómo presentar un cuaderno utilizando una combinación de Markdown y celdas de código.
* Cómo cerrar sus sesiones de cuaderno una vez que haya terminado de trabajar en ellas.
* Jupyter implementa un modelo de dos procesos con un núcleo y un cliente.
* El servidor de cuadernos se encarga de guardar y cargar los cuadernos.
* El núcleo ejecuta las celdas de código contenidas en el cuaderno.
* La arquitectura Jupyter utiliza la herramienta NB convert para convertir archivos a otros formatos.
* Jupyter implementa un modelo de dos procesos con un núcleo y un cliente.
* El servidor de cuadernos se encarga de guardar y cargar los cuadernos.
* La arquitectura Jupyter utiliza la herramienta de conversión NB para convertir archivos a otros formatos.
* La interfaz gráfica de usuario del Navegador Anaconda puede lanzar varias aplicaciones en un dispositivo local.
* Los entornos Jupyter en el Navegador Anaconda incluyen JupyterLab y VS Code.
* Puede descargar los entornos Jupyter por separado del Navegador de Anaconda, pero es posible que no estén configurados correctamente.
* La GUI del Navegador Anaconda puede lanzar múltiples aplicaciones.
* Otros entornos Jupyter de código abierto son JupyterLab, JupyterLite, VS Code y Google Colaboratory.
* JupyterLite es una herramienta basada en navegador.