Tipos de Datos y su Estructura

Mgr. Sharon Arandia

Universidad Privada Boliviana

2023

Tipos de Datos y su Estructura

Fuentes del Big Data

Si las tecnologías del Big Data se alimentan de datos, ¿cuáles son sus fuentes?

Las fuentes del Big Data son diversas y pueden incluir datos estructurados, semiestructurados y no estructurados, provenientes de diferentes fuentes, como:

- **Dispositivos IoT:** sensores, dispositivos de seguimiento, dispositivos de medición y otros dispositivos IoT generan grandes volúmenes de datos.
- **Redes sociales:** las plataformas de redes sociales generan grandes cantidades de datos no estructurados, como publicaciones, comentarios, mensajes, fotografías y videos.
- **Transacciones en línea:** las transacciones en línea generan grandes cantidades de datos estructurados, como los registros de ventas, transacciones financieras, entre otros.
- **Datos de sensores y sistemas de monitoreo:** estos pueden incluir datos de dispositivos de seguimiento de vehículos, sistemas de monitoreo ambiental y sistemas de monitoreo de infraestructura
- **Datos de medios digitales:** incluyen datos de publicidad en línea, videos, música y otras formas de contenido digital.
- **Datos de archivos y registros históricos:** incluyen registros de clientes, historiales médicos, registros de educación, registros gubernamentales, entre otros.
- **Datos de la web:** incluyen datos de sitios web y motores de búsqueda
- **Datos de la nube:** incluyen datos almacenados en servicios de almacenamiento en la nube.
- **Datos de dispositivos móviles:** incluyen datos generados por aplicaciones móviles, historiales de navegación web y registros de llamadas.

En resumen, las fuentes del Big Data son diversas y pueden provenir de casi cualquier fuente de información digital.

Procedencia

Los datos procesados por las soluciones Big Data pueden ser generados por humanos o por máquinas, aunque en última instancia es responsabilidad de las máquinas generar los resultados analíticos.

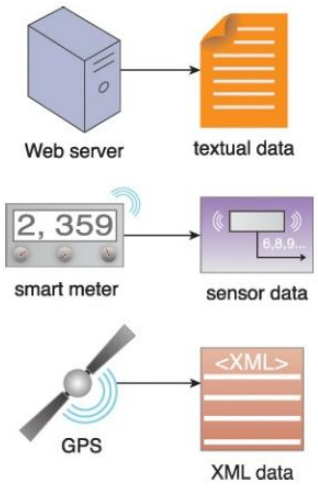
Los datos generados por humanos son el resultado de la interacción humana con sistemas, como en línea a través de servicios y dispositivos digitales.

Datos generados por humanos: son aquellos que son creados o producidos por seres humanos. Un ejemplo son los comentarios de reseñas de productos: los comentarios que se publican en sitios web de comercio electrónico como Amazon, eBay y otros sitios similares. En general, los datos generados por humanos son valiosos para los análisis y las investigaciones, ya que proporcionan una perspectiva única sobre los comportamientos, opiniones y preferencias humanas.

Ejemplo de una reseña de hotel recopilado de Trip Advisor:

Datos generados por máquinas: son generados por programas de software y dispositivos de hardware en respuesta a eventos del mundo real. Por ejemplo, un archivo de registro (log file) captura una decisión de autorización realizado por un servicio de seguridad.

Ejemplos de datos generados por máquinas:



Clasificación de los Datos según su Estructura

Los datos pueden clasificarse según su nivel de estructuración.

Se tiene la siguiente clasificación primaria:

- Datos estructurados
- Datos sin estructura
- Datos semiestructurados

Estos tipos de datos se refieren a la organización interna de los datos y, a veces, se denominan formatos de datos. Aparte de estos tres tipos de datos fundamentales, otro tipo importante de datos en los entornos Big Data son metadatos.

Datos estructurados

Son datos organizados y almacenados en un formato definido. Los datos estructurados se almacenan en bases de datos relacionales o en hojas de cálculo y tienen un formato específico y definido. Estos datos se pueden buscar, analizar y procesar fácilmente con herramientas de análisis de datos. Ejemplos de datos estructurados incluyen tablas de bases de datos, hojas de cálculo y archivos de texto con formato.

Los datos estructurados se ajustan a un modelo o esquema de datos y, a menudo, se almacenan en forma tabular. Se utilizan para capturar relaciones entre diferentes entidades y, por lo tanto, es más frecuentemente almacenada en una base de datos relacional. Usualmente los sistemas de ERP (Enterprise Resource Planning) y CRM (Customer Relationship Management) usan estas estructuras.

Datos semiestructurados

Los datos semiestructurados tienen un nivel definido de estructura y consistencia, pero no son relacionales por naturaleza. En cambio, los datos semiestructurados son jerárquicos o basados en gráficos.

Tienen una cierta organización o etiquetado para permitir su análisis. Los datos semiestructurados pueden ser almacenados en formatos como JSON o XML, y contienen elementos etiquetados y valores. Estos datos se utilizan a menudo para almacenar información de documentos, registros y aplicaciones web. Ejemplos de datos semiestructurados incluyen documentos HTML, archivos XML y registros de servidor de aplicaciones web.

Debido a la naturaleza textual de estos datos y su conformidad con algún nivel de estructura, es más fácil procesarlos que a los datos no estructurados.

Datos no estructurados

Son datos sin estructura definida, por lo general sin etiquetas ni organización. Estos datos no se pueden almacenar en bases de datos relacionales o en hojas de cálculo y requieren herramientas de análisis de datos avanzadas para procesarlos. Ejemplos de datos no estructurados incluyen archivos de audio, archivos de video, correos electrónicos, publicaciones en redes sociales y documentos de texto sin formato.

Se estima que los datos no estructurados representan el 80 % de los datos dentro de cualquier empresa. Los datos no estructurados tienen una tasa de crecimiento más rápida que los datos estructurados.

Esta forma de datos es textual o binaria y, a menudo, se transmite a través de archivos que son autónomos y no relacionales.

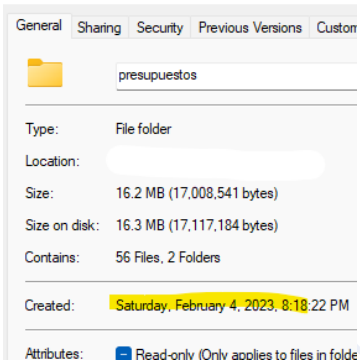
Un archivo de texto puede contener el contenido de varios tweets o publicaciones de blog.

Los archivos binarios suelen ser medios archivos que contienen datos de imagen, audio o video. Técnicamente, tanto los archivos de texto como los binarios tienen un estructura definida por el propio formato de archivo, pero este aspecto se pasa por alto, y la noción de ser desestructurado está en relación con el formato de los datos contenidos en el propio archivo.

En resumen, los datos estructurados, semiestructurados y no estructurados se diferencian en su organización y en su capacidad de ser procesados y analizados. Los datos estructurados son organizados y definidos, los datos semiestructurados tienen una cierta organización y etiquetado, mientras que los datos no estructurados no tienen una estructura definida y no se pueden procesar fácilmente con herramientas de análisis de datos convencionales.

Metadata

Son datos metadatos que proporcionan información sobre las características y la estructura de un conjunto de datos. La mayoría de los datos son generados por máquinas y se pueden agregar a los datos. El seguimiento de metadatos es importante para el procesamiento, almacenamiento y análisis de Big Data porque proporciona información sobre el pedigrí de los datos y su procedencia durante el procesamiento. Un ejemplo son las etiquetas XML que proporcionan el autor y la fecha de creación de un documento.

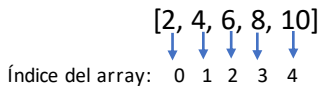


Estructura de los Datos

Una estructura de datos es una colección de datos que se caracterizan por su organización y las operaciones que se definen en ellos. Esta estructura vendrá caracterizada por relaciones entre los datos que la constituyen y por las operaciones posibles en ella, expresadas mediante un conjunto de reglas.

Estas son algunas de las estructuras más comunes que pueden adoptar los datos (si están estructurados) y sus ejemplos usando números:

Arreglos (Arrays): Una colección de elementos del mismo tipo de datos que se almacenan en una ubicación contigua de memoria. Se accede a los elementos del arreglo mediante un índice numérico.



Es recomendable usar arrays cuando el acceso a estos datos se realiza de manera aleatoria, en caso contrario es recomendable usar las listas.

Los arreglos se utilizan para almacenar datos que se pueden acceder de forma rápida y eficiente mediante su posición en la lista. También se utilizan para realizar operaciones matemáticas y manipular grandes conjuntos de datos de manera eficiente en aplicaciones como análisis de datos, inteligencia artificial y aprendizaje automático.

Listas Enlazadas (Linked Lists): Una estructura de datos dinámica en la que cada elemento (nodo) contiene un valor y un puntero al siguiente nodo. Las listas enlazadas pueden ser simples, dobles o circulares.

2 -> 4 -> 6 -> 8 -> 10

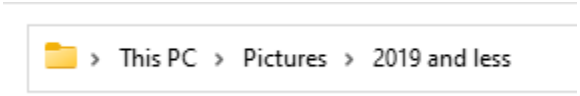
Por ejemplo, una lista enlazada de tareas pendientes que incluyen "Llamar al cliente", "Enviar un correo electrónico" y "Programar una reunión", se vería así:

Llamar al cliente --> Enviar un correo electrónico --> Programar una reunión

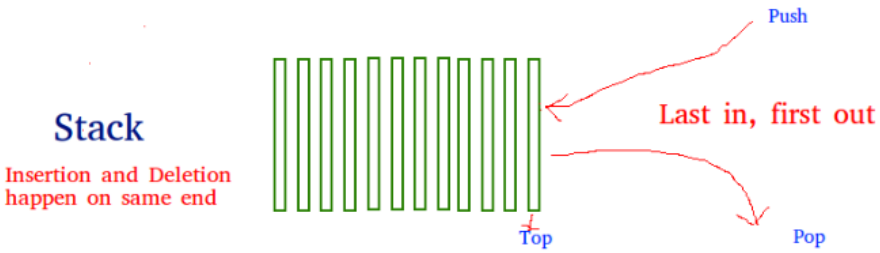
Se puede acceder a cada elemento de la lista de manera secuencial a través de los punteros que los enlazan. También es posible agregar o eliminar elementos de la lista de manera eficiente moviendo los punteros.

Las listas enlazadas se utilizan en muchas aplicaciones, como en la implementación de editores de texto, bases de datos, sistemas de archivos y otras estructuras de datos complejas.

En Python, las listas enlazadas no están incluidas en la biblioteca estándar, pero es posible crearlas mediante algoritmos.



Pilas (Stacks): Una colección de elementos ordenados en la que se permite el acceso solo al último elemento agregado (LIFO - Last In First Out). Las operaciones principales para el último elemento son push (añadir) y pop (eliminar).



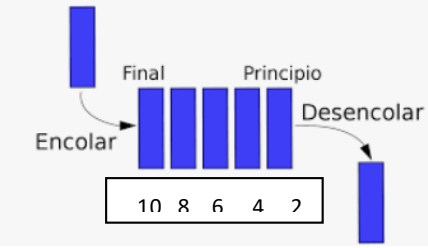
Para añadir elementos a la pila: Push(2) -> Push(4) -> Push(6) -> Push(8) -> Push(10)

Con una pila creada, se pueden eliminar los elementos, uno a uno: Pop() -> 10 -> Pop() -> 8 -> Pop() -> 6

Ejemplo de utilidad: el botón "Atrás" en un navegador web. Cada vez que visitas una página web nueva, se agrega a una pila en el navegador. Si deseas volver a la página anterior, presionas el botón "Atrás" y se elimina la página actual de la pila, y la página anterior se convierte en la parte superior de la pila y se muestra en la pantalla.

Esto se debe a que las páginas web se cargan en orden secuencial y, por lo tanto, se pueden acceder a través de una estructura de pila. Es gracias a las pilas que podemos tener las funciones de anular/rehacer en las aplicaciones.

Colas (Queues): Una colección de elementos ordenados en la que se permite el acceso solo al primer elemento agregado (FIFO - First In First Out). Las operaciones principales son enqueue (añadir) y dequeue (eliminar).

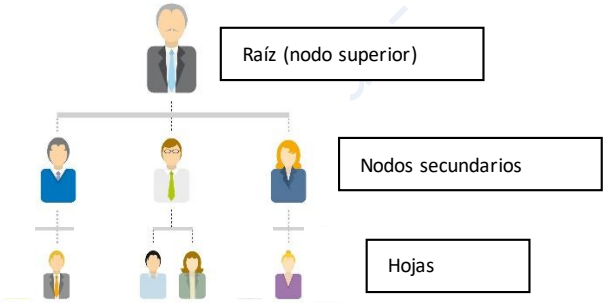


Enqueue(2) -> Enqueue(4) -> Enqueue(6) -> Enqueue(8) -> Enqueue(10)

Dequeue() -> 2 -> Dequeue() -> 4 -> Dequeue() -> 6

Las colas son una estructura de datos muy útil en el manejo de datos para organizar y procesar grandes cantidades de información. Un ejemplo sencillo en el que se pueden utilizar las colas para manejar datos es en un sistema de atención al cliente en el que se deben procesar solicitudes en orden de llegada.

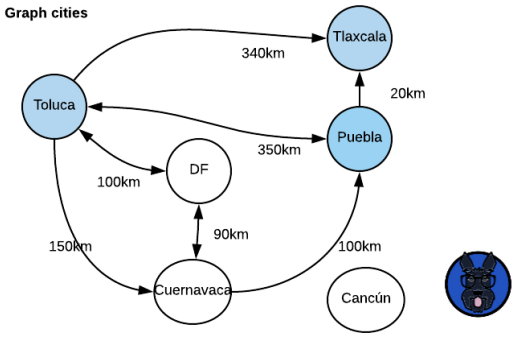
Árboles (Trees): Una estructura de datos jerárquica en la que cada elemento (nodo) tiene un valor y cero o más nodos secundarios. El nodo superior se llama raíz y los nodos sin hijos se llaman hojas.



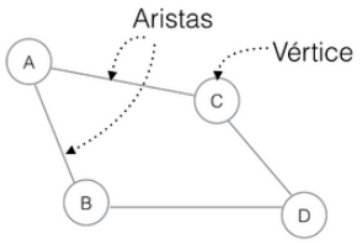
Los árboles son útiles para tomar en cuenta jerarquías. Un ejemplo en el que se pueden utilizar los árboles es en la representación de una estructura jerárquica como una organización empresarial.

Abuelo: Martin, Hijos: Andrés (nietos: Mateo), Christian (nietos: José, Melany) y Estefany (nietos: Andrea)

Grafos (Graphs): Una estructura de datos que representa un conjunto de objetos (vértices) conectados por enlaces (aristas). Los grafos pueden ser dirigidos o no dirigidos y se utilizan en muchas aplicaciones, como redes sociales y mapas. A diferencia de los árboles, estos no tienen una estructura jerárquica



Componentes:



Los vértices son también llamados nodos y las aristas conexiones.

Las estructuras de grafos son una forma poderosa y versátil de modelar problemas y relaciones en una variedad de campos, desde la planificación de la cadena de suministro y la logística hasta la ingeniería de software y la seguridad cibernética. Un ejemplo en el que una empresa puede utilizar grafos es en la representación de una red de transporte para planificar rutas de entrega y logística.

Supongamos que una empresa tiene una flota de camiones que necesitan entregar productos a diferentes ubicaciones en una ciudad. Para planificar la ruta de cada camión, se puede utilizar un grafo para representar las carreteras y las intersecciones de la ciudad.

Tablas Hash (Hash Tables): Son una estructura de datos muy versátil que se utiliza en una amplia variedad de aplicaciones para almacenar y recuperar información de manera eficiente, estas asocian llaves o claves con valores.

Permiten la búsqueda, inserción y eliminación de elementos en tiempo constante (O(1)), lo que significa que el tiempo necesario para encontrar un elemento no depende del tamaño de la tabla. Esto las hace muy útiles para implementar bases de datos, buscadores, índices de archivos, entre otros.

La operación principal que soportan de manera eficiente las tablas hash es la búsqueda.

Los elementos se almacenan en un arreglo y se accede a ellos mediante una función hash.

{0: 2, 1: 4, 2: 6, 3: 8, 4: 10}

Usos comunes:

- *Búsqueda y recuperación de datos*
- *Almacenamiento de contraseñas de manera segura. En lugar de almacenar las contraseñas en texto plano, se guarda una versión hash de la contraseña. Cuando un usuario intenta iniciar sesión, la contraseña que ingresó se convierte en una versión hash y se compara con la hash almacenada en la tabla. Si son iguales, el usuario se autentica con éxito.*
- *Detección de duplicados: las tablas hash se utilizan para detectar duplicados en conjuntos de datos.*
- *Caché de datos: las tablas hash se utilizan para implementar cachés de datos, que son mecanismos que almacenan datos en memoria temporal para mejorar el rendimiento.*

Conjuntos (Sets): Una colección de elementos únicos y no ordenados. Las operaciones principales son agregar, eliminar y comprobar la pertenencia de un elemento.

{2, 4, 6, 8, 10}

Usos comunes:

Eliminación de duplicados y búsqueda rápida: los conjuntos se implementan como tablas hash, lo que significa que la búsqueda de elementos en un conjunto es muy rápida.

Operaciones de conjunto: los conjuntos en Python tienen una serie de métodos para realizar operaciones de conjuntos como intersecciones, uniones, diferencias y comparaciones de subconjuntos. Estas operaciones pueden ser muy útiles en el procesamiento de datos y la resolución de problemas.

Compatibilidad con otras estructuras de datos: los conjuntos en Python son compatibles con otras estructuras de datos como listas, tuplas y diccionarios, lo que los hace ideales para trabajar en conjunto con otras estructuras de datos en proyectos de programación más grandes.

Mapas (Maps): Una estructura de datos que asocia claves únicas con valores. Las operaciones principales son agregar, eliminar y buscar elementos por clave.

Un mapa que asocia nombres de frutas con su cantidad en inventario:

{"manzanas": 20, "naranjas": 15, "plátanos": 30}

Usos comunes:

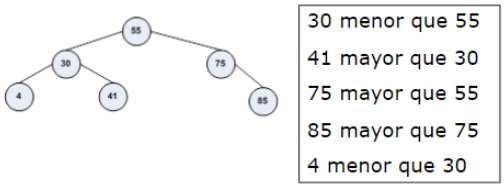
Los mapas se pueden utilizar para almacenar configuraciones y opciones de un programa o aplicación. Por ejemplo, un mapa podría ser utilizado para almacenar las preferencias de un usuario, como el idioma predeterminado o el color de fondo preferido.

Análisis de datos: los mapas se pueden utilizar para almacenar datos en bruto, así como para analizar y procesar datos. También pueden usarse para implementar cachés en memoria temporal de cálculos complejos.

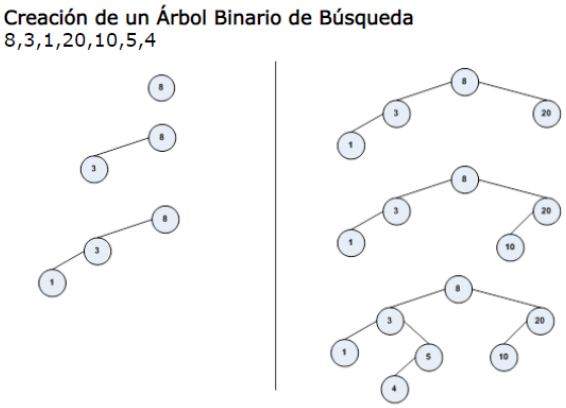
Indexación de datos: los mapas se pueden utilizar para indexar grandes conjuntos de datos, lo que permite acceder a los datos de forma eficiente. Por ejemplo, para indexar un conjunto de archivos de texto, permitiendo una búsqueda rápida de palabras o frases específicas.

Implementación de algoritmos: los mapas se pueden utilizar para implementar algoritmos que requieren una búsqueda eficiente de elementos, como aquellos que usan grafos.

Árboles de Búsqueda (Search Trees): Una estructura de datos que organiza los elementos en un árbol y se utiliza para la búsqueda eficiente de elementos. Los árboles de búsqueda comunes incluyen los árboles binarios de búsqueda y los árboles AVL. Dado un nodo, se sitúan los números menores al lado izquierdo y los mayores al lado derecho.



Ejemplo de su estructuración:



Usos comunes:

Diccionarios en línea: utilizan árboles de búsqueda para almacenar las palabras y sus definiciones, de manera que sea fácil y rápido encontrar una palabra específica.

Sistema de archivos de un sistema operativo: se utilizan para representar la jerarquía de archivos y directorios en un sistema operativo, lo que permite una búsqueda y navegación eficiente.

Búsqueda en motores de búsqueda: los motores de búsqueda utilizan árboles de búsqueda para indexar las páginas web y para realizar búsquedas rápidas en sus bases de datos

Implementación de algoritmos de ordenamiento: algunos algoritmos de ordenamiento como el árbol de búsqueda binario se basan en árboles de búsqueda para organizar y ordenar una lista de elementos.

Análisis de cadenas de ADN: en bioinformática, los árboles de búsqueda se utilizan para buscar y comparar secuencias de ADN.

Bibliografía

- Curto (2022). *Fundamentos y Usos del Big Data*
- *Estructura de Datos. Capitulo V. http://informatica.uv.es/docencia/fguia/TI/Libro/PDFs/CAPI5.pdf*
- *Instituto profesional Araucana – Estructura de datos*