

HENRY



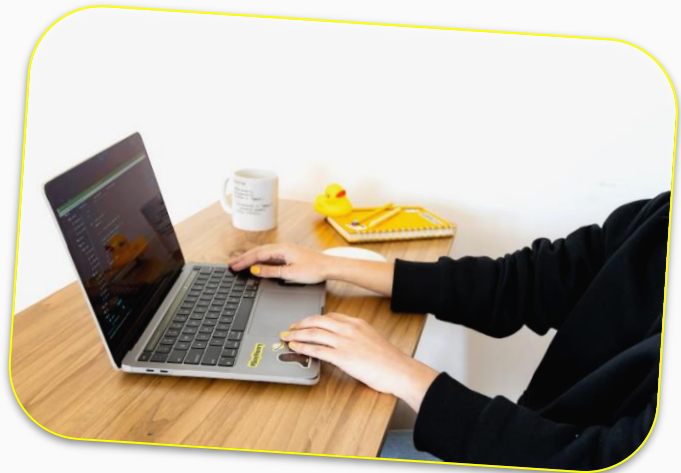
Estructuras de Datos II

Data Science





Agenda



- Listas Enlazadas
- Lista Enlazada simple
- Listas Doblemente Enlazadas
- Operaciones en una lista
- Tablas Hash (Hash tables)



OBJETIVOS DE LA CLASE

Al finalizar esta lecture estarás en la capacidad de...

- Entender la estructura de datos de Lista y sus variantes con principales aplicaciones
- Conocer el concepto de tablas Hash



Listas **Enlazadas**





¿Qué es?

- Secuencia de nodos enlazados entre ellos.
- Cada nodo contiene datos y además uno o más links a otros nodos.
- Según las restricciones que tengan la cantidad de links, estas pueden ser simplemente enlazadas, dobles o múltiples





Operaciones en una lista

- **Iterar sobre la lista:** Recorrer la lista viendo sus elementos o hasta que encontremos el elemento deseado.
- **Insertar un nodo:** La operación va a cambiar según el lugar donde queramos insertar el nodo nuevo:
 - Al principio de la lista.
 - En el medio de la lista.
 - Al final de la lista.
- **Sacar un nodo:**
 - Del principio de la lista.
 - Del medio de la lista.



Lista enlazada simple

Es una lista enlazada de nodos, donde cada nodo tiene un único campo de enlace.

Se puede recorrer en un sólo sentido.





Lista doblemente enlazada

Es un tipo de lista más sofisticado.

Cada nodo tiene dos enlaces: uno apunta al nodo anterior, y otro apunta al nodo siguiente.





Tablas **Hash**





¿Qué es?

Es una estructura que guarda los datos de una manera asociativa, o sea con un par clave-valor o key-value.

Los datos son guardados como en un arreglo, pero los índices tienen que ver con lo qué está guardado adentro.

¡Esto hace que sean muy rápidas para buscar datos!

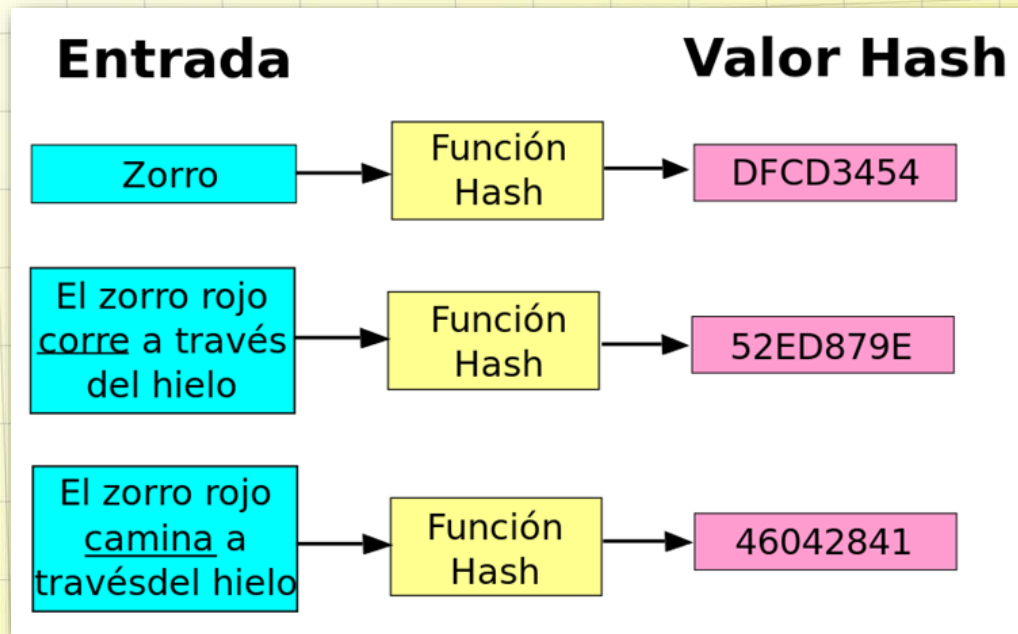


Para construir una Tabla Hash se necesita...

- **Una estructura de datos:** Acá se guardan los datos y se buscan por el índice. Puede ser un arreglo, o un árbol, etc...
- **Una función hashadora:** Se necesita una función que transforme lo que elegimos de key a un hash que será nuestro índice.
- **Una política de resolución de colisiones:** Es la política que se define para decidir qué pasa cuando dos keys distintas generan dos hash iguales (las funciones no son perfectas !).



Ejemplo



HENRY

