

# Análisis de Algoritmos y Estructura de Datos

**TDA tabla hash  
Refuerzo y ejercicios**

Prof. Violeta Chang C

Semestre 2 – 2023



# TDA Tablas Hash

- **Contenidos:**

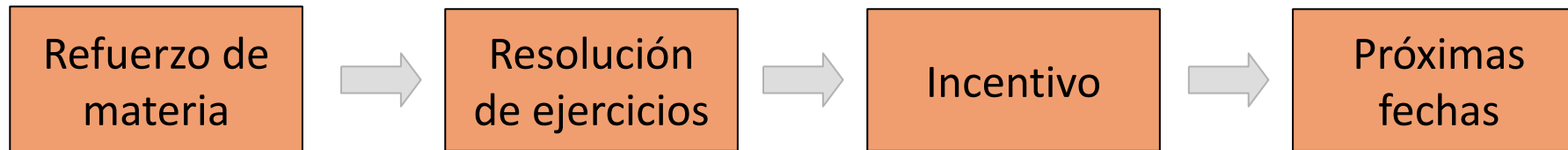
- Estructura de datos TDA tabla hash
- Operaciones de TDA tabla hash

- **Objetivos:**

- Comprender estructura de datos de TDA tabla hash
- Conocer y aplicar operaciones y estrategias de resolución de colisiones en tablas hash



# TDA tabla hash



Ruta de la sesión

# Refuerzo de materia



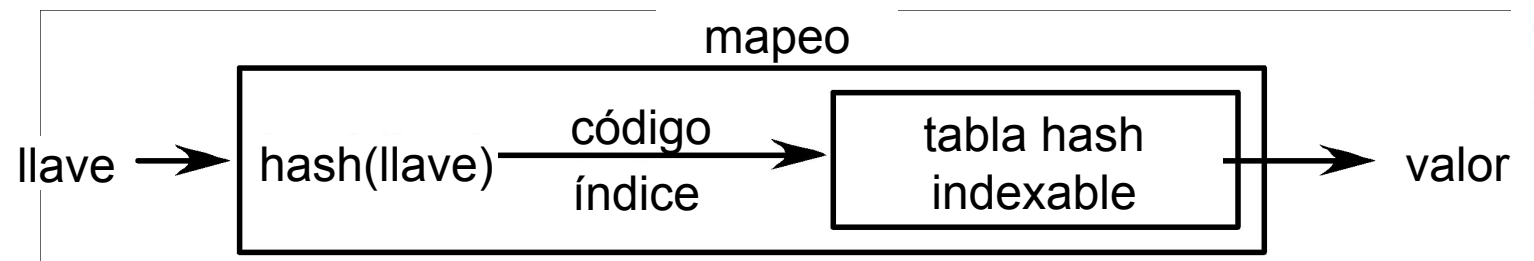
# TDA tabla HASH

## • Estructura de datos:

- Una **tabla hash** es un colección indexable, donde cada posición de la tabla se conoce como **bucket** (o celda) y puede tener una o más entradas
- Una **función hash** convierte una llave de entrada en un índice que es usado para acceder a un bucket en la tabla hash y debe ser determinística, eficiente y uniforme

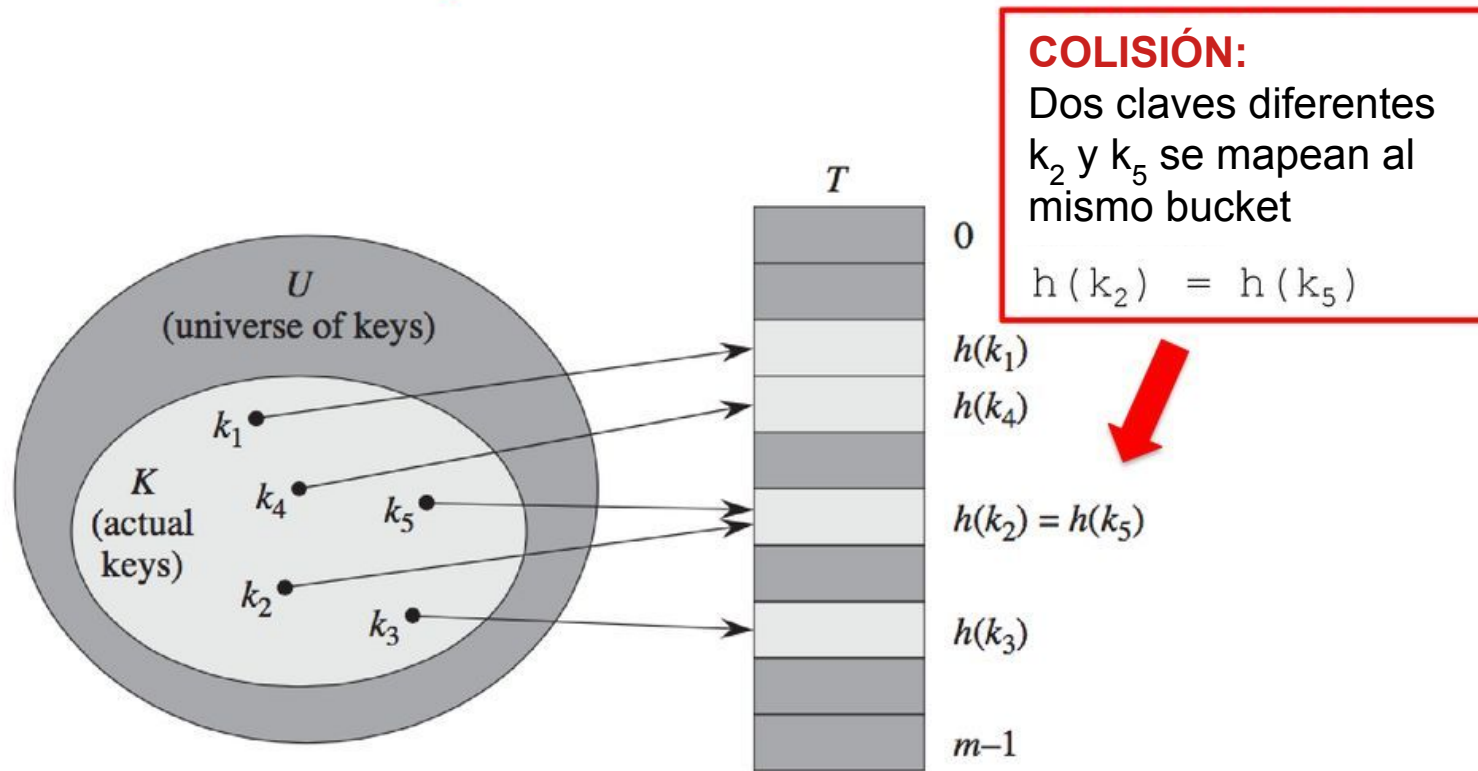
## • Operaciones:

- buscar(T,dato)
- insertar(T,dato)
- eliminar(T,dato)





# Resolución de colisiones



Estrategias de resolución de colisiones:

- **Hashing abierto**
  - encadenamiento
- **Hashing cerrado**
  - prueba lineal



DEPARTAMENTO DE  
**INGENIERÍA  
INFORMÁTICA**  
UNIVERSIDAD DE SANTIAGO DE CHILE

# Ejercicios



# Ejercicio en parejas

- **Ejercicio1**: Considerando una tabla hash de 9 buckets, una función  $\text{hash}(x) = (4x+2) \bmod 9$ , y las siguientes operaciones:

a) insertar(30)

b) insertar(43)

c) insertar(11)

d) insertar(16)

e) insertar(51)

f) eliminar(16)

g) insertar(47)

h) insertar(21)

i) eliminar(47)

j) eliminar(11)

k) insertar(17)

l) eliminar(21)

Mostrar la tabla hash resultante después de realizar las operaciones en el orden establecido usando las siguientes estrategias de resolución de colisiones:

- Prueba lineal
- Encadenamiento





# Ejercicio en parejas

- **Ejercicio2**: Escribir un algoritmo en pseudocódigo para insertar una llave (con sus datos asociados) en una tabla hash abierto, usando encadenamiento como estrategia de resolución de colisiones.

OBS: asumir que se cuenta con la función *hash(llave)* que recibe una llave y devuelve un valor entero que representa al bucket correspondiente a la llave ingresada.



# Incentivo - 15+5+5 minutos

- Escribir un algoritmo en seucodódigo para insertar una llave (con sus datos asociados) en una tabla hash cerrado, usando prueba lineal como estrategia de resolución de colisiones. Calcular y justificar la complejidad del algoritmo propuesto.

OBS: asumir que se cuenta con la función *hash(llave)* que recibe una llave y devuelve un valor entero que representa al bucket correspondiente a la llave ingresada.



# Incentivo - 15+5+5 minutos

## Puntaje

I.	El algoritmo propuesto apunta a resolver el problema planteado. (SI 1 punto / NO 0 punto)	
II.	El algoritmo resuelve correctamente el problema planteado. (SI 1 punto / NO 0 punto)	
III.	El algoritmo está escrito en pseudocódigo ordenado (SI 1 punto / NO 0 punto)	
IV.	El algoritmo está escrito en el formato establecido (SI 1 punto / NO 0 punto)	
V.	El algoritmo identifica entradas correctamente (SI 1 punto / NO 0 punto)	
VI.	El algoritmo identifica y declara salidas de manera correcta. (SI 1 punto / NO 0 punto)	
VII.	Calcula correctamente la complejidad (SI 1 punto / NO 0 punto)	
VIII.	Justifica la complejidad del algoritmo propuesto. (SI 1 punto / NO 0 punto)	
	PUNTOS	



DEPARTAMENTO DE  
**INGENIERÍA  
INFORMÁTICA**  
UNIVERSIDAD DE SANTIAGO DE CHILE

# Actividad de cierre



código: 6624 6856



# Próximas fechas...

- Resumen de la semana:
  - TDA tabla HASH

~~cátedra~~ – ~~refuerzo~~ – **laboratorio**

- Próxima semana:
  - PEP3**
  - Entrega de Tarea 3**
  - Pruebas recuperativas**

U4 - S13

Diciembre 2023						
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
					1	2
3	4	5 ✓	6	7 ✓	8	9
10	11	12 ●	13 ●	14 ●	15	16
17	18	19 ●	20	21	22	23
24	25	26	27	28	29	30
	Navidad					