

# CONCEPTOS DE BASES DE DATOS

CLASE 8



- Dispersión con espacio de direccionamiento estático
  - Parámetros que afectan la eficiencia
    - Función de dispersión
    - Tamaño de los compartimentos
    - Densidad de empaquetamiento
    - Método de tratamiento de saturación

- Tipos
  - Técnicas con espacio de direccionamiento **estático**
    - El espacio disponible esta **prefijado**
  - Técnicas con espacio de direccionamiento **dinámico**
    - El espacio disponible **aumenta o disminuye** según las necesidades en cada momento

# Dispersión

## Trat. de overflow

- Aún con algoritmos de dispersión muy buenos y tamaños de compartimento grandes → existe la posibilidad de que haya **registros en saturación**
  - Se debe incorporar algún método para tratar con estos registros que no pueden entrar en su dirección base
- **Métodos de tratamiento de overflow**
  - Saturación Progresiva
  - Saturación Progresiva encadenada
  - Saturación Progresiva encadenada (en áreas separadas)
  - Dispersión doble

# Dispersión

Trat. de overflow

- Saturación progresiva
  - **Inserción:** cuando se completa una dirección de memoria se busca en las siguientes direcciones en secuencia, hasta encontrar una vacía para almacenarlo
  - **Búsqueda:** comienza en la dirección base y continúa buscando en localidades sucesivas
    - Al llegar a la clave buscada → se finaliza con éxito
    - Al llegar a una dirección vacía → la clave buscada no está en el archivo
    - Al llegar al final del archivo, se continúa por el inicio → circularidad
    - Al llegar al lugar de comienzo → la clave buscada no está en el archivo (DE = 1)

# Dispersión

Trat. de overflow

- Saturación progresiva
  - **Eliminación**
    - Debe ser posible **utilizar el espacio liberado** para posteriores inserciones
    - Pero el espacio liberado por una eliminación **no debe obstaculizar** las búsquedas posteriores
      - La búsquedas finalizan al encontrar una dirección vacía → **no se debe dejar direcciones vacías**



# Dispersión

Trat. de overflow

- Saturación progresiva
  - **Eliminación**
    - Se marca el espacio liberado → Ej: #####
      - No se rompe las secuencias de búsqueda
      - El espacio liberado queda disponible para posteriores adiciones
  - **Ventaja:** simplicidad
  - **Desventaja:** tiende a agrupar en zonas contiguas
    - Búsquedas largas con DE que tienden a 1
  - Ejemplo

# Dispersión

Trat. de overflow

- Saturación progresiva → Función: **clave mod 11**

Inicio			+47, +57, +75			-47			+25		
Dir. base	Reg. 1	Reg. 2	Dir. base	Reg. 1	Reg. 2	Dir. base	Reg. 1	Reg. 2	Dir. base	Reg. 1	Reg. 2
0			0	75		0	75		0	75	
1			1			1			1		
2	35	46	2	35	46	2	35	46	2	35	46
3	58		3	58	47	3	58	###	3	58	25
4			4	57		4	57		4	57	
5			5			5			5		
6			6			6			6		
7			7			7			7		
8			8			8			8		
9	86	97	9	86	97	9	86	97	9	86	97
10	21	32	10	21	32	10	21	32	10	21	32



# Dispersión

Trat. de overflow

- Saturación progresiva encadenada
  - Técnica para evitar los problemas causados por la acumulación de registros
  - Similar a la sat. progresiva, excepto que las **claves sinónimo se enlazan** con apuntadores
    - Ya no es necesario buscar de forma secuencial
  - Cada dirección base contiene un **número** que indica el lugar del **siguiente sinónimo**
    - Si la dirección no contiene sinónimos → -1

- Saturación progresiva encadenada
  - **Ventaja:** sólo se necesita acceder a las direcciones que contienen registros con claves sinónimo
    - Mejora el número de accesos promedio
    - Ya no son necesarias las marcas de eliminación
  - **Desventaja:** debe agregarse un campo de enlace a cada registro
    - Requiere mayor espacio de almacenamiento

- Saturación progresiva encadenada
  - **Problema:** acceder a una dirección base ocupada con un registro que no es de ese lugar. Ejemplo:
    - Se insertan: Alfa (22), Epsilon (23)
    - Luego: Delta (22), Beta (24), Gamma (22)

Dir. base	Clave	Encadenado
22	Alfa	-1
23	Epsilon	-1
24		
25		
26		

# Dispersión

## Trat. de overflow

- Saturación progresiva encadenada
  - **Problema:** acceder a una dirección base ocupada con un registro que no es de ese lugar. Ejemplo:
    - Delta es sinonimo de Alfa pero se inserta en dir 24
    - Beta tiene dir. base 24 pero se carga en dir 25
    - Gamma también es sinónimo de Alfa, pero se inserta en dir 26
    - ¿Cuál es el siguiente del 24?

Dir. base	Clave	Encadenado
22	Alfa	24
23	Epsilon	-1
24	Delta	¿25 o 26?
25	Beta	-1
26	Gamma	-1

- Saturación progresiva encadenada
  - Una alternativa de solución podría ser **cargar el archivo en dos pasos**
    - 1) Sólo cargar los registros con direcciones base. Los registros que no entran en su dir. base se guardarán en un archivo separado → **se garantiza que ninguna dir. base estará ocupada por registros en saturación**
    - 2) Cargar los reg. separados en direcciones libres
  - Pero esta solución **no garantiza** que las eliminaciones y/o inserciones posteriores no tendrán problemas

- Saturación progresiva encadenada
  - La solución es **dar prioridad a los registros base**: cuando un registro no tiene lugar en su dirección base, pero dicha dirección está ocupada por un registro que no pertenece a esa dirección (tiene otra dirección base):
    - Se **reemplaza el registro actual** por el que sí corresponde a esa dirección base
    - Se **busca un nuevo lugar** para el registro reemplazado
  - Ejemplo



# Dispersión

## Trat. de overflow

- Saturación progresiva encadenada → Función:  
**clave mod 11**

Inicio			+75			+31			+12		
Dir. base	Reg. 1	Enlace	Dir. base	Reg. 1	Enlace	Dir. base	Reg. 1	Enlace	Dir. base	Reg. 1	Enlace
0		-1	0	75	-1	0	75	-1	0	75	-1
1		-1	1		-1	1	31	0	1	12	-1
2	35	-1	2	35	-1	2	35	-1	2	35	-1
3	58	-1	3	58	-1	3	58	-1	3	58	-1
4		-1	4		-1	4		-1	4	31	0
5		-1	5		-1	5		-1	5		-1
6		-1	6		-1	6		-1	6		-1
7		-1	7		-1	7		-1	7		-1
8		-1	8		-1	8		-1	8		-1
9	86	-1	9	86	0	9	86	1	9	86	4
10	21	-1	10	21	-1	10	21	-1	10	21	-1

# Dispersión

Trat. de overflow

- Saturación progresiva con encadenamiento en área separada
  - Al conjunto de direcciones base se le llama **área principal de datos**
  - Al conjunto de direcciones en saturación se le llama **área de saturación**
  - Cuando se agrega un registro nuevo:
    - Si hay lugar en la dirección base → **área principal**
    - Si no hay lugar → **área de saturación**

# Dispersión

Trat. de overflow

- Saturación progresiva con encadenamiento en área separada
  - **Ventaja:** se mejora el tratamiento de inserciones y eliminaciones
  - **Desventaja:** si el área de saturación separada está en un cilindro diferente del de la dirección base, toda búsqueda de registro en saturación implicará un desplazamiento → **incrementa el costo**
  - Ejemplo

# Dispersión

Trat. de overflow

- Saturación progresiva con encadenamiento en área separada → Función: **clave mod 11**

Area Principal

Dir. base	Reg. 1	Enlace
0		-1
1		-1
2	35	-1
3	58	-1
4		-1
5		-1
6		-1
7		-1
8		-1
9	86	-1
10	21	-1

Area Sat.

Dir. base	Reg. 1	Enlace
0		-1
1		-1
2		-1
3		-1

+75

Dir. base	Reg. 1	Enlace
0		-1
1		-1
2	35	-1
3	58	-1
4		-1
5		-1
6		-1
7		-1
8		-1
9	86	0
10	21	-1

Dir. base	Reg. 1	Enlace
0	75	-1
1		-1
2		-1
3		-1

# Dispersión

## Trat. de overflow

- Saturación progresiva con encadenamiento en área separada → Función: **clave mod 11**

+31

Dir. base	Reg. 1	Enlace
0		-1
1		-1
2	35	-1
3	58	-1
4		-1
5		-1
6		-1
7		-1
8		-1
9	86	0
10	21	-1

Dir. base	Reg. 1	Enlace
0	75	1
1	31	-1
2		-1
3		-1

+12

Dir. base	Reg. 1	Enlace
0		-1
1	12	-1
2	35	-1
3	58	-1
4		-1
5		-1
6		-1
7		-1
8		-1
9	86	0
10	21	-1

Dir. base	Reg. 1	Enlace
0	75	1
1	31	-1
2		-1
3		-1

# Dispersión

Trat. de overflow

- Dispersión Doble
  - Cuando sucede un caso de saturación se aplica una **segunda función de dispersión** a la clave, para producir un nuevo número
  - Dicho número se utiliza como **desplazamiento** → se suma a la dirección original **tantas veces como sea necesario** hasta encontrar una dirección con espacio para almacenar la clave



# Dispersión

Trat. de overflow

- Dispersión Doble
  - **Ventaja:** se evita acumulamiento
    - Los registros no quedan almacenados “localmente”, tienden a esparcirse en el archivo
  - **Desventaja:** aumenta el TAP a los registros
    - Sino se consiguen compartimentos de saturación con dirección al mismo cilindro de disco, el cambio de cilindro requerirá un desplazamiento → **incrementa el costo**
  - Ejemplo

# Dispersión

Trat. de overflow

- Dispersión Doble → F1: clave mod 11 , F2: clave mod 7

Inicio

Dir. base	Reg. 1
0	
1	
2	35
3	58
4	
5	
6	
7	
8	
9	86
10	21

+75

Dir. base	Reg. 1
0	
1	
2	35
3	58
4	
5	
6	
7	
8	75
9	86
10	21

+31

Dir. base	Reg. 1
0	
1	31
2	35
3	58
4	
5	
6	
7	
8	75
9	86
10	21

- Hash asistido por tabla
  - Necesita una estr. adicional → **tabla en memoria**
    - Una entrada por cada cubeta del archivo
    - Las entradas comienzan con una secuencia con valor infinito (todos unos)
    - Si hay **saturación** al realizar una inserción, el valor de secuencia de la entrada puede tener que actualizarse
  - Usa **tres** funciones de dispersión
    - Dirección física
    - Desplazamiento
    - Secuencia de K-Bits

- Hash asistido por tabla
  - Inserciones **lentas**, recuperaciones **rápidas** (1 acceso)
    - Sirve cuando se recupera más de lo que se inserta
  - **Inserciones**
    - La cubeta **tiene lugar**: queda el elemento
    - La cubeta **está llena**: **saturación** → comparación de secuencias de claves involucradas y selección de clave saliente
  - **Eliminaciones**
    - Consiste en reescribir el nodo sin el elemento a eliminar
    - Si se elimina una clave de una cubeta que estaba **saturada**, **será más compleja la inserción de una nueva clave** en dicha cubeta
  - Ejemplo

# Dispersión

Estático

- Hash asistido por tabla
  - Se **insertan** las primeras 6 claves, no hay saturación

Clave	F1H(Clave)	F2H(Clave)	F3H(Clave)
Alfa	50	3	0001
Beta	51	4	0011
Gamma	52	7	0101
Delta	50	3	0110
Epsilon	52	3	1000
Rho	51	5	0100
Pi	50	2	0010
Psi	50	9	1010

Tabla en memoria		Nodos en disco	
...	1111	50	Alfa Delta
50	1111	51	Beta Rho
51	1111	52	Gamma Epsilon
52	1111		
53	1111		
54	1111		
...	1111		

# Dispersión

Estático

- Hash asistido por tabla
  - Al insertar Pi, se **satura** un nodo

Clave	F1H(Clave)	F2H(Clave)	F3H(Clave)
Alfa	50	3	0001
Beta	51	4	0011
Gamma	52	7	0101
Delta	50	3	0110
Epsilon	52	3	1000
Rho	51	5	0100
Pi	50	2	0010
Psi	50	9	1010

Tabla en memoria		Nodos en disco	
...	1111	50	Alfa Pi
50	0110	51	Beta Rho
51	1111	52	Gamma Epsilon
52	1111	53	Delta
53	1111		
54	1111		
...	1111		



# Dispersión

Estático

- Hash asistido por tabla
  - Se elimina Pi

Clave	F1H(Clave)	F2H(Clave)	F3H(Clave)
Alfa	50	3	0001
Beta	51	4	0011
Gamma	52	7	0101
Delta	50	3	0110
Epsilon	52	3	1000
Rho	51	5	0100
Pi	50	2	0010
Psi	50	9	1010

Tabla en memoria		Nodos en disco	
...	1111	50	Alfa
50	0110	51	Beta Rho
51	1111	52	Gamma Epsilon
52	1111	53	Delta
53	1111		
54	1111		
...	1111		

# Dispersión

Estático

- Hash asistido por tabla
  - Se inserta Psi

Clave	F1H(Clave)	F2H(Clave)	F3H(Clave)
Alfa	50	3	0001
Beta	51	4	0011
Gamma	52	7	0101
Delta	50	3	0110
Epsilon	52	3	1000
Rho	51	5	0100
Pi	50	2	0010
Psi	50	9	1010

Tabla en memoria		Nodos en disco	
...	1111	50	Alfa
50	10 10	51	Beta Rho
51	1111	52	Gamma Epsilon
52	1111	53	Delta
53	1111	59	Psi
54	1111		
...	1111		

# Dispersión

Estático

- Dispersión con espacio de dir. **estático**
  - ¿Qué hacer si se agota el espacio disponible?
  - Cuando se llega a una **DE mayor a 75%** se genera más saturación de la deseada
  - Cuando se completa:
    - Obtener más espacio
    - Actualizar la función de dispersión
    - Redispersar

# Dispersión

Estático

- Dispersión con espacio de dir. **estático**
  - **Costo de la redistribución → ALTO**
    - Se utiliza mucho tiempo
    - Mientras se realiza no es posible que los usuarios accedan al archivo
  - Por eso, al crear un archivo se puede optar por:
    - **1)** Analizar la tasa de crecimiento del archivo y optimizar la relación 'uso de espacio' **vs.** 'periodicidad de redistribución'
    - **2)** No establecer un tamaño prefijado, sino que el archivo crezca a medida que lo necesite → **dispersión dinámica**

# Dispersión

Dinámico

- Dispersión con espacio de dir. **dinámico**
  - Existen diferentes técnicas de dispersión con espacio de direccionamiento dinámico. Entre ellas están:
    - Hash Virtual
    - Hash Dinámico
    - Hash Extensible → se analizará esta técnica

# Dispersión

Dinámico

- Hash Extensible
  - Adapta el resultado de la función de hash de acuerdo al número de registros que tenga el archivo, y de las cubetas necesarias para su almacenamiento
  - **Estr. auxiliar** → **tabla en memoria principal** para direcciones físicas
  - **Función** → genera **secuencias de bits**
    - **32 bits**: millones de direcciones diferentes
    - Usando la secuencia retornada se accede a la tabla para recuperar la dirección física



# Dispersión

Dinámico

- Hash Extensible
  - Se utilizan **sólo los bits necesarios** de acuerdo a cada instancia del archivo
  - La tabla tendrá  **$2^i$  entradas** (direcciones de cubetas), siendo  **$i$**  el número de bits actuales para el sistema
  - Al intentar insertar en una cubeta llena → **saturación**
    - Se aumenta en uno el valor asociado a la cubeta saturada
    - Se genera una cubeta con el mismo valor de la saturada
    - La cantidad de celdas de la tabla se duplica, y también se duplica el valor asociado a la tabla
    - Se **redispersan todos los registros** de la cubeta saturada

# Dispersión

Dinámico

- Hash Extensible → Ejemplo
  - Se cuenta con los siguientes elementos, el resultado de  $f(h)$  y sabiendo que cada compartimento tiene capacidad para 2 registros

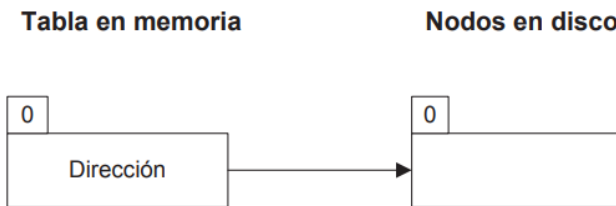
1	Argentina	...11101100
2	Brasil	...01001011
3	Uruguay	...10111010
4	Chile	...01101100
5	Colombia	...11110111
6	Paraguay	...00010001
7	Ecuador	...10010111
8	Perú	...11101111

# Dispersión

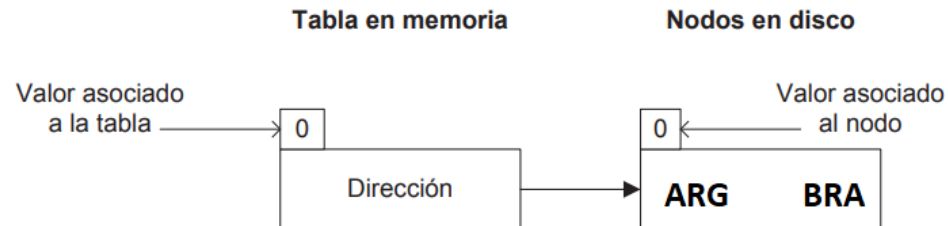
Dinámico

- Hash Extensible → Ejemplo

- Estado inicial (tamaño nodo = 2):



- Se inserta Argentina y Brasil:



1	Argentina	...11101100
2	Brasil	...01001011
3	Uruguay	...10111010
4	Chile	...01101100
5	Colombia	...11110111
6	Paraguay	...00010001
7	Ecuador	...10010111
8	Perú	...11101111

# Dispersión

Dinámico

- Hash Extensible → Ejemplo
  - Se inserta Uruguay → Overflow
    - Se aumenta en 1 el valor del nodo saturado
    - Se crea un nuevo nodo con el mismo valor
    - Se duplica el tamaño de la tabla
    - Se aumenta en 1 el valor de la tabla

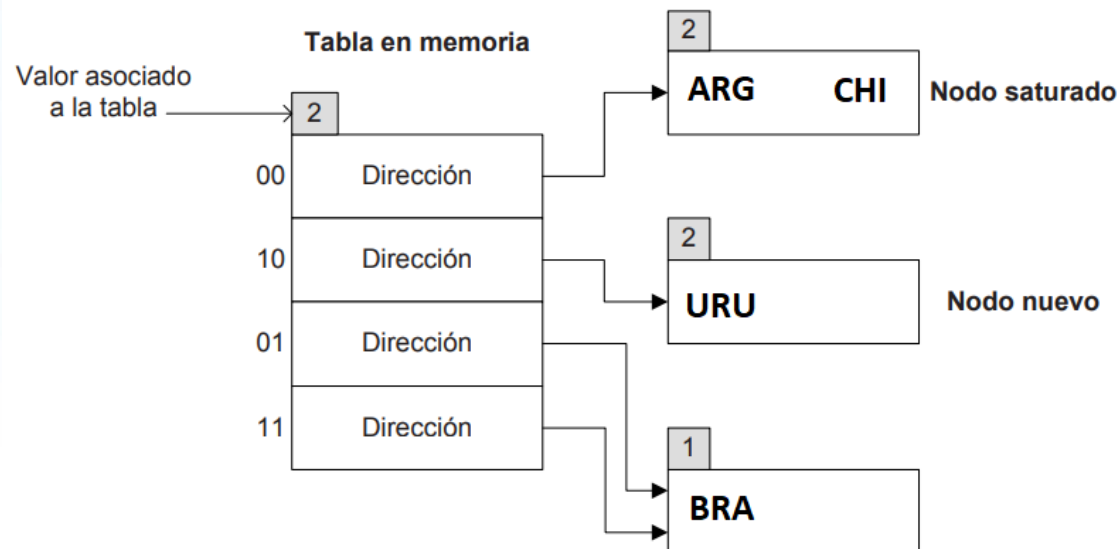


1	Argentina	...11101100
2	Brasil	...01001011
3	Uruguay	...10111010
4	Chile	...01101100
5	Colombia	...11110111
6	Paraguay	...00010001
7	Ecuador	...10010111
8	Perú	...11101111

# Dispersión

Dinámico

- Hash Extensible → Ejemplo
  - Se inserta Chile → Overflow
    - Se aumenta en 1 el valor del nodo saturado
    - Se crea un nuevo nodo con el mismo valor
    - Se duplica el tamaño de la tabla
    - Se aumenta en 1 el valor de la tabla

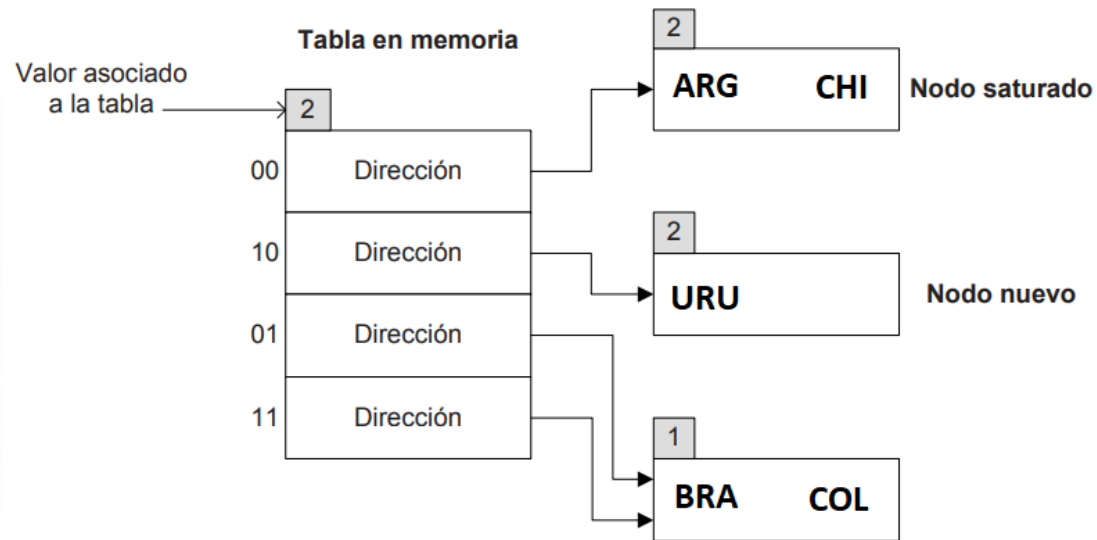


1	Argentina	...11101100
2	Brasil	...01001011
3	Uruguay	...10111010
4	Chile	...01101100
5	Colombia	...11110111
6	Paraguay	...00010001
7	Ecuador	...10010111
8	Perú	...11101111

# Dispersión

Dinámico

- Hash Extensible → Ejemplo
  - Se inserta Colombia



1	Argentina	...11101100
2	Brasil	...01001011
3	Uruguay	...10111010
4	Chile	...01101100
5	Colombia	...11110111
6	Paraguay	...00010001
7	Ecuador	...10010111
8	Perú	...11101111

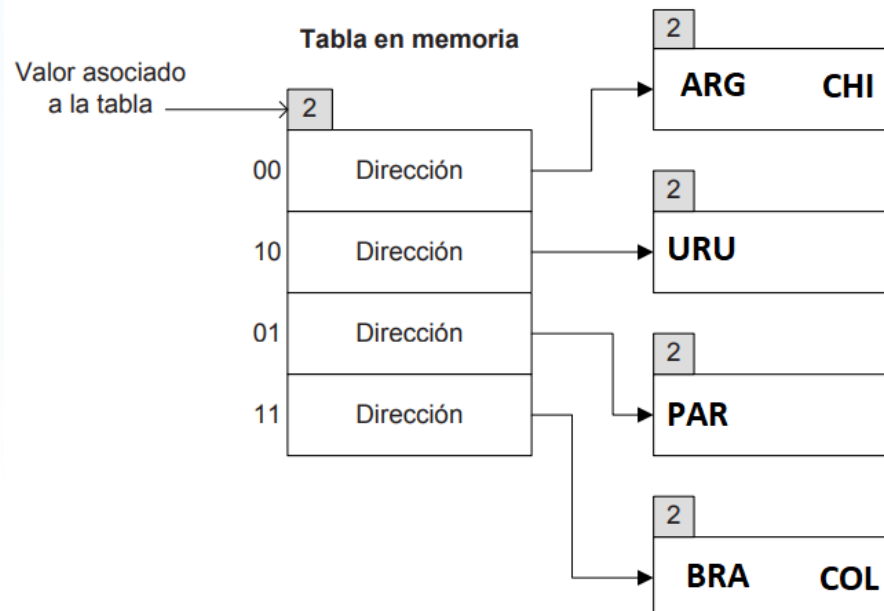


# Dispersión

Dinámico

- Hash Extensible → Ejemplo

- Se inserta Paraguay → Overflow
  - Se aumenta en 1 el valor del nodo saturado
  - Se crea un nuevo nodo con el mismo valor
  - En este caso no se duplica el tamaño de la tabla ni se aumenta su valor



1	Argentina	...11101100
2	Brasil	...01001011
3	Uruguay	...10111010
4	Chile	...01101100
5	Colombia	...11110111
6	Paraguay	...00010001
7	Ecuador	...10010111
8	Perú	...11101111