**✍️ Parte práctica (90 minutos)**

**1. Ejemplos guiados paso a paso (45 min)**

**Ejemplo 1 – Hashing simple sin colisiones**

Supongamos:

* Tabla de tamaño 10.
* Claves: 23, 45, 66

def hash\_simple(k): return k % 10

hash\_simple(23) = 3

hash\_simple(45) = 5

hash\_simple(66) = 6

**Resultado**: cada valor va a una posición distinta. ¡Perfecto!

**Ejemplo 2 – Hashing con colisiones y saturación progresiva**

Claves: 21, 31, 41 (todas % 10 = 1)

**Inserciones:**

* 21 → posición 1 ✔️
* 31 → posición 1 ❌ → probar 2 ✔️
* 41 → 1 ❌, 2 ❌ → 3 ✔️

**Búsqueda de 41**: se busca desde la base (1) hasta que se encuentra.

**Ejemplo 3 – Área de desbordes**

* Tabla principal: posiciones 0 a 4.
* Área de desbordes: 5 a 7.

Claves: 12, 22, 32 → todas hash = 2

* 12 y 22 van a posición 2 (capacidad 2).
* 32 va al área de desbordes (posición 5).

Encadenamiento: 2 → 5

**2. Ejercicios prácticos para los estudiantes (45 min)**

Proponé que los resuelvan individualmente o en pares. Luego los corregís en conjunto.

**Ejercicio 1**

Usá la función hash H(x) = x % 5 y una tabla de tamaño 5.

Insertar: 7, 12, 17, 22, 27  
Resolver usando **saturación progresiva**.

👉 ¿Dónde se insertará cada valor? ¿Cuántos accesos se requieren para buscar el 22?

**Ejercicio 2**

Usá una tabla con cubetas de tamaño 2 y un área de desbordes de 3 posiciones.

Insertar: claves 15, 25, 35, 45, 55 (todas tienen hash = 0).

👉 Simulá el proceso de inserción con **área de desbordes separada**. Dibujá la estructura.

**Ejercicio 3**

Explicar en palabras simples cómo funciona la **doble dispersión** y cuál es su ventaja frente a la saturación progresiva.