Análisis formal de complejidad para `liberar_tabla_hash`

Sean:

- H = `HASH_SIZE` (tamaño de la tabla hash)
- N = número total de nodos en todas las listas de la tabla hash

1. Planteamiento y desarrollo de $T(H, N)$

La función recorre cada posición de la tabla hash (buckets), y libera todos los nodos encadenados de cada bucket.

$$T(H, N) = t_1 + \sum_{i=0}^{H-1} \left(t_2 + \sum_{j=1}^{n_i} t_3 \right)$$

donde n_i es la cantidad de nodos en el bucket i.

$$T(H, N) = t_1 + Ht_2 + t_3 \sum_{i=0}^{H-1} n_i$$

$$T(H, N) = t_1 + Ht_2 + t_3 N$$

ya que $\sum_{i=0}^{H-1} n_i = N$ (total de nodos en la tabla).

Redefiniendo $c_1 = t_3$, $c_2 = t_1 + Ht_2$:

$$T(H, N) = c_1 N + c_2$$

---

2. Calculando los casos

Mejor caso: No hay nodos en ninguna lista ($N = 0$):

$$T_m(H, 0) = t_1 + Ht_2 = c_2$$

Por lo tanto, $T_m(H, 0) \in \Theta(H)$.

Peor caso: Todas las listas están llenas, el total de nodos a liberar es N:

$$T_p(H, N) = c_1 N + c_2$$

Por lo tanto, $T_p(H, N) \in \Theta(N + H)$, pero si $N \gg H$, $T_p(H, N) \in \Theta(N)$.

Caso promedio: Depende de la cantidad promedio de nodos por bucket, pero en general la complejidad es proporcional a $N + H$:

$$T_{pr}(H, N) = c_1 N + c_2$$

---

3. Comprobación de cotas por límites

$$\lim_{N\to\infty} \frac{T(H,N)}{N} = c_1$$

Por lo tanto:

$$T(H,N) \in \mathcal{O}(N)$$
$$T(H,N) \in \Omega(N)$$
$$T(H,N) \in \Theta(N)$$

4. Resumen Final:

- Mejor caso: $\Theta(N)$
- Peor caso: $\Theta(N)$
- Caso promedio: $\Theta(N)$

Las cotas están verificadas con límites y usando los nombres de las variables reales del código.