

```
Run  TallerEstructuras x
=== 1) Editor Deshacer/Rehacer ===
Texto final: Hola Mundo

=== 2) Sistema de Atención (PriorityQueue) ===
Atendido primero: Luis (prio=3)
En cola restante: 2

=== 3) Usuarios únicos (HashMap) ===
Registro repetido (ID=101) aceptado? false
Usuarios:
  101 -> Sofía
  102 -> Carlos

=== 4) Árbol Binario: recorridos ===
In-orden: [2, 4, 5, 7, 8, 9, 11]
Pre-orden: [7, 4, 2, 5, 9, 8, 11]
Post-orden:[2, 5, 4, 8, 11, 9, 7]

=== 5) AVL: inserciones y rotaciones detectadas ===
In-orden AVL: [10, 20, 25, 27, 28, 30]
Rotaciones ocurridas: [Rotación Simple Derecha (LL), Rotación Doble Izquierda-Derecha (LR), Rotación Simple Derecha (LL), Rotación Doble Derecha (RR)]
```

```
Run  TallerEstructuras x
otación Doble Izquierda-Derecha (LR), Rotación Simple Derecha (LL), Rotación Doble Derecha-Izquierda (RL), Rotación Simple Izquierda (RR)]
```

```
Run  TallerEstructuras x
In-orden AVL: [10, 20, 25, 27, 28, 30]
Rotaciones ocurridas: [Rotación Simple Derecha (LL), Rotación Doble Izquierda-Derecha (LR), Rotación Simple Derecha (LL), Rotación Doble Derecha (RR)]

=== 6) BST: búsqueda/insert/eliminar ===
BST contiene 12? true
BST in-orden (tras eliminar 10): [8, 12, 15, 17, 20, 25]

=== 7) Grafo: BFS y Dijkstra ===
BFS desde A: [A, B, C, D, E]
Camino más corto A->E (Dijkstra): [A, C, B, D, E]

=== 8) Complejidad ===
Complejidades (Big-O) resumidas:
- Pila (push/pop/peek): O(1) amortizado.
- PriorityQueue (insert/extraer máx): O(log n).
- HashMap (put/get): O(1) promedio, O(n) peor caso patológico.
- Árbol binario (recorridos): O(n).
- AVL (buscar/insertar/eliminar): O(log n) siempre (balanceado).
- BST (buscar/insertar/eliminar): promedio O(log n), peor O(n) si se desbalancea.
- Grafo BFS: O(V+E).
- Dijkstra con PQ binaria: O((V+E) log V).

Process finished with exit code 0
```