Actividad 2 En el diseño de interiores de oficinas o hogares, necesitamos un programa que imprima las combinaciones posibles de la ubicación de escritorios y sillas, en una habitación de 4x4. Las restricciones son: garantizar que ningún elemento esté en la misma "fila" (por ejemplo, en una misma fila de escritorios) o en la misma "columna" (por ejemplo, alineación en una pared) para optimizar el uso del espacio y facilitar la circulación.

public class DisenoInteriores {

private static final int TAMANO\_HABITACION = 4;

private static char[][] habitacion = new char[TAMANO\_HABITACION][TAMANO\_HABITACION];

public static void main(String[] args) {

System.out.println("Combinaciones válidas de escritorios (E) y sillas (S) en una habitación de 4x4:");

encontrarCombinacionesValidas(0);

}

private static void encontrarCombinacionesValidas(int fila) {

if (fila == TAMANO\_HABITACION) {

imprimirHabitacion();

return;

}

for (int columna = 0; columna < TAMANO\_HABITACION; columna++) {

if (esPosicionValida(fila, columna, 'E')) {

habitacion[fila][columna] = 'E'; // Colocar escritorio

encontrarCombinacionesValidas(fila + 1);

habitacion[fila][columna] = '\0'; // Retroceder (Backtracking)

}

if (esPosicionValida(fila, columna, 'S')) {

habitacion[fila][columna] = 'S'; // Colocar silla

encontrarCombinacionesValidas(fila + 1);

habitacion[fila][columna] = '\0'; // Retroceder (Backtracking)

}

}

}

private static boolean esPosicionValida(int fila, int columna, char elemento) {

// Verificar si la fila o columna ya tienen un elemento

for (int i = 0; i < TAMANO\_HABITACION; i++) {

if (habitacion[fila][i] != '\0' || habitacion[i][columna] != '\0') {

return false;

}

}

return true;

}

private static void imprimirHabitacion() {

for (char[] fila : habitacion) {

for (char celda : fila) {

System.out.print((celda == '\0' ? "-" : celda) + " ");

}

System.out.println();

}

System.out.println();

}

}