

```
/**
 * Esta clase representa un cuadrante de una tabla, delimitado por cuatro de sus
 * campos, que son los límites de ese cuadrante.
 */
class Cuadrante {
    /**
     * Límite vertical inferior del cuadrante
     */
    final int lvi; // limite vertical inferior del cuadrante
    /**
     * Límite vertical superior del cuadrante
     */
    final int lvs; // limite vertical superior del cuadrante
    /**
     * Límite horizontal inferior del cuadrante
     */
    final int lhi; // limite horizontal inferior del cuadrante
    /**
     * Límite horizontal superior del cuadrante
     */
    final int lhs; // limite horizontal superior del cuadrante
    /**
     * Registro de coordenadas que guardara la casilla esquina del cuadrante mas
     * cercana al centro de su cuadrante "padre", es decir, el que lo creó
     * dividiendose en 4 subcuadrantes
     */
    final Coordenadas EsquinaCentral;
    /**
     * Constructor de la clase Cuadrante para el primer cuadrante padre de todos, que
     * equivale a toda la tabla de N filas x N columnas
     * @param N Dimensión de la tabla, es decir, número de filas y de columnas de la
     * tabla
     */
    Cuadrante(int N) {
        lvi = lhi = 0; lvs = lhs = N - 1; EsquinaCentral = null;
    }

    /**
     * Funcion que genera el subcuadrante numero "nhijo" del objeto Cuadrante que lo
     * invoca. Calcula sus limites y las coordenadas de EsquinaCentral para el hijo.
     * @param nhijo Número del subcuadrante que queremos generar (primero, segundo,
     * tercero ó cuarto).
     * @return El subcuadrante número "nhijo" del objeto Cuadrante que lo invoca.
     */
    Cuadrante Hijo(int nhijo) {
        return new Cuadrante(this, nhijo);
    }

    /**
     * Constructor sobrecargado de la clase Cuadrante, que usaremos para crear los
     * subcuadrantes hijos de un cuadrante. Es privado ya que sólo lo usan los
     * cuadrantes para generar sus hijos.
     */
    private Cuadrante (Cuadrante padre, int nhijo) {
        switch (nhijo) {
            case 1: // primer subcuadrante: el de arriba a la izquierda
                lvi = padre.lvi;
                lvs = padre.lvi + ((padre.lvs - padre.lvi) / 2);
                lhi = padre.lhi;
```

```
        lhs = padre.lhi + ((padre.lhs - padre.lhi) / 2);
        EsquinaCentral = new Coordenadas(lvs, lhs);
        break;
    case 2: // segundo subcuadrante: el de arriba a la derecha
        lvi = padre.lvi;
        lvs = padre.lvi + ((padre.lvs - padre.lvi) / 2);
        lhi = padre.lhi + ((padre.lhs - padre.lhi) / 2) + 1;
        lhs = padre.lhs;
        EsquinaCentral = new Coordenadas(lvs, lhi);
        break;
    case 3: // tercer subcuadrante: el de abajo a la izquierda
        lvi = padre.lvi + ((padre.lvs - padre.lvi) / 2) + 1;
        lvs = padre.lvs;
        lhi = padre.lhi;
        lhs = padre.lhi + ((padre.lhs - padre.lhi) / 2);
        EsquinaCentral = new Coordenadas(lvi, lhs);
        break;
    default: // (case 4); cuarto subcuadrante: el de abajo a la derecha
        lvi = padre.lvi + ((padre.lvs - padre.lvi) / 2) + 1;
        lvs = padre.lvs;
        lhi = padre.lhi + ((padre.lhs - padre.lhi) / 2) + 1;
        lhs = padre.lhs;
        EsquinaCentral = new Coordenadas(lvi, lhi);
        break;
    }
}
```