

Esquemas de recorrido sobre tablero y direcciones

Esquema de recorrido genérico

```
Iniciar recorrido
while (quedan elementos para recorrer) {
    Procesar elemento
    Avanzar al siguiente elemento
}
Finalizar recorrido
```

Esquema de recorrido de procesamiento (transformación en realidad, todos procesan)

Recorriendo celdas de tablero

```
IrAlInicioEnUnRecorridoAl_YAl_(Este, Norte)
while (haySiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)) {
    ProcesarCeldaActual()
    IrASiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)
}
ProcesarCeldaActual()
```

ProcesarCeldaActual es una subtarea que transforma la celda actual de una forma particular. Puede implicar un procesamiento condicional.

Recorriendo direcciones

```
direcciónActual := minDir()
while (direcciónActual /= maxDir()) {
    ProcesarHaciaDirección_(direcciónActual)
    direcciónActual := siguiente(direcciónActual)
}
ProcesarHaciaDirección_(direcciónActual)
```

ProcesarHaciaDirección_ es una subtarea que dada una dirección, procesa uno o más elementos relacionados a dicha dirección (ej. la celda lindante, todas las de la fila/columna hacia esa dirección, etc.).

Esquema de recorrido de búsqueda

Recorriendo celdas de tablero

```
IrAlInicioEnUnRecorridoAl_YAl_(Este, Norte)
while (haySiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)
    && not esLaCeldaBuscada()) {
```

```
    IrASiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)
}
```

esLaCeldaBuscada es una subtaska que indica sí la celda donde está al cabezal es la celda que se está buscando.

Recorriendo direcciones

```
direcciónActual := minDir()
while (direcciónActual /= maxDir()
      && not esLaDirecciónBuscada_(direcciónActual)) {
    direcciónActual := siguiente(direcciónActual)
}
```

esLaDirecciónBuscada_ es una subtaska que dada una dirección indica sí la dirección es la buscada.

Esquema de recorrido de búsqueda parcial

Este tipo de recorridos tiene por precondition que efectivamente existe el elemento buscado.

Recorriendo celdas de tablero

```
IrAlInicioEnUnRecorridoAl_YAl_(Este, Norte)
while (not esLaCeldaBuscada()) {
    IrASiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)
}
```

esLaCeldaBuscada es una subtaska que indica sí la celda donde está al cabezal es la celda que se está buscando.

Recorriendo direcciones

```
direcciónActual := minDir()
while (not esLaDirecciónBuscada_(direcciónActual)) {
    direcciónActual := siguiente(direcciónActual)
}
```

esLaDirecciónBuscada_ es una subtaska que dada una dirección indica sí la dirección es la buscada.

Esquema de recorrido de acumulación

Recorriendo celdas de tablero

```
IrAlInicioEnUnRecorridoAl_YAl_(Este, Norte)
totalAcumulado := cantidadAContarAquí()
while (haySiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)) {
    IrASiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)
    totalAcumulado := totalAcumulado + cantidadAContarAquí()
}
```

cantidadAContarAquí es una subtarea que describe cuántos elementos se cuentan en la celda actual.

Recorriendo direcciones

```
direcciónActual := minDir()
totalAcumulado := cantidadAContarHacia_(direcciónActual)
while (direcciónActual /= maxDir()) {
    direcciónActual := siguiente(direcciónActual)
    totalAcumulado := totalAcumulado + cantidadAContarHacia_(direcciónActual)
}
```

cantidadAContarHacia_ es una subtarea que dada una dirección, describe la cantidad de elementos que se cuentan hacia dicha dirección.

Esquema de recorrido de máximo-mínimo

Este tipo de recorridos tiene por **precondición** que hay al menos un elemento que es el más chico o más grande. Además se debe comenzar sobre un elemento (y se evita el ir al inicio) o debe haber uno en cada celda/dirección.

Recorriendo celdas de tablero

```
IrAlInicioEnUnRecorridoAl_YAl_(Este, Norte)
mínimoOMáximoAlMomento := cantidadAContarAquí()
while (haySiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)) {
    IrASiguienteCeldaEnUnRecorridoAl_YAl_(Este, Norte)
    mínimoOMáximoAlMomento :=
        mínimoOMáximoEntre_Y_(mínimoOMáximoAlMomento, cantidadAContarAquí())
}
```

cantidadAContarAquí es una subtarea que describe el valor de los elementos que se cuentan en la celda actual.

mínimoOMáximoEntre_Y_ es una subtarea que dados dos valores describe entre ambos el más pequeño o más grande.

Notar que puede que en ocasiones no se desee el valor, sino algún otro criterio, como algo que identifique la celda de forma unívoca, su coordenada, etc.

Recorriendo direcciones

```
direcciónActual := minDir()
minimoOMaximoAlMomento := direcciónActual
while (direcciónActual /= maxDir()) {
    direcciónActual := siguiente(direcciónActual)
    minimoOMaximoAlMomento :=
        direcciónMínimaOMáximaEntre_Y_(direcciónActual, minimoOMaximoAlMomento)
}
```

direcciónMínimaOMáximaEntre_Y_ es una subtarea que dadas dos direcciones, describe la dirección hacia la cual se cuentan menos o más elementos.