

```

void Mergesort(String[] x, int ip, int iu) {
    if (ip >= iu)
        return;
    int im = (ip + iu) / 2; ← índice de la mitad del arreglo
    Mergesort(x, ip, im); ← ordenar la primera mitad
    Mergesort(x, im + 1, iu); ← repetir con la segunda mitad
    String[] aux = new String[iu + 1]; ← 1ª mitad comienza en ip
    int i = ip, i1 = ip, i2 = im + 1; ← 2ª mitad comienza en la mitad
    while (i1 <= im && i2 <= iu) {
        aux[i++] = x[i1].compareTo(x[i2]) < 0 ? x[i1++] : x[i2++];
    } ← compara los elementos, asigna el menor y avanza 1 en esa misma mitad
    while (i1 <= im) { ← recorre los que quedan desde el último i1
        aux[i++] = x[i1++]; ← asigna los restantes 1ª mitad
    }
    ← a partir del i que quedó en la otra iteración
    while (i2 <= iu) { ← recorre desde el i2 que quedó
        aux[i++] = x[i2++]; ← asigna los restantes 2ª mitad
    }
    ← a partir del i que quedó de la iteración anterior
    for (int j = ip; j <= iu; ++j) {
        x[j] = aux[j]; ← copia el arreglo auxiliar ordenado en el original
    }
}

```