```
proc ordenar (inout r: reunion, in freq: ℤ, in prof: ℤ)  {
        Pre {esReuniónVálidaAux(r, prof, freq) ∧ r_0 = r}
        Post {
          esReuniónVálidaAux(r, prof, freq) ∧
            ordenadaDeMayorAMenorPorTonoDeVoz(r) ∧
            esUnaPermutación(r_0, r)
        }
}
    pred ordenadaDeMayorAMenorPorTonoDeVoz (r: reunion) {
      (∀i : ℤ) 1 ≤ i < |r|  ⟶_L tonoDeVoz(r[i − 1]_0) ≥ tonoDeVoz(r[i]_0)
    }

    pred esUnaPermutación (x: reunion, y: reunion) {
      |x| = |y| ∧_L
        (∀i : ℤ) 0 ≤ i < |x|  ⟶_L(
          (∃j : ℤ) 0 ≤ j < |y|  ∧_L(x[i]_1 = y[j]_1 ∧ x[i]_0 = y[j]_0))
    }
```