

```

proc esSeñal (in s: seq⟨ℤ⟩, in prof: ℤ, in freq: ℤ, out result: Bool) {
  Pre {|s| ≥ 0 ∧ prof > 0 ∧ freq > 0}
  Post {result = true ↔ esSeñalAux(s, prof, freq)}
}

pred esSeñalAux (s: seq⟨ℤ⟩, prof: ℤ, freq: ℤ) {
  |s| > 0 ∧
  frecuenciaEnRango(freq) ∧
  duraMasDeUnSegundo(s, freq) ∧
  profundidadCorrecta(s) ∧
  ningunaMuestraSuperaLaProfundidad(s, prof)
}

pred frecuenciaEnRango (freq: ℤ) {freq ∈ [8, 32]}

pred profundidadCorrecta (prof: ℤ) {prof ∈ [8, 16, 32]}

pred duraMasDeUnSegundo (s: seq⟨ℤ⟩, freq: ℤ) {duraciónEnSegundos(s, freq) > 1}

pred ningunaMuestraSuperaLaProfundidad (s: seq⟨ℤ⟩, p: ℤ) {
  (∀i : ℤ) 0 ≤ i < |s| →L (-2)p-1 ≤ s[i] ≤ 2p-1 - 1
}

fun duraciónEnSegundos (s: seq⟨ℤ⟩, freq: ℤ) : ℤ = |s| div (freq · 1000);

```