

```

proc trayectoriaDeLosFrutosIndividualesALargoPlazo (inout trayectorias : seq<seq<ℝ>>,
in cooperan : seq<Bool>, in apuestas : seq<seq<ℝ>>, in pagos : seq<seq<ℝ>>, in eventos : seq<seq<ℕ>>))
  requiere {|trayectorias| = |cooperan| = |apuestas| = |pagos| = |eventos|}
  requiere {(∀i : ℤ) ((0 ≤ i < individuosLen) ∧L (|pagos[i]| = |apuestas[i]|))}
  requiere {(∀i : ℤ) ((0 ≤ i < individuosLen) ∧L (|trayectorias[i]| = 1 ∧L trayectorias[i][0] > 0))}
  requiere {(∀i, k : ℤ) ((0 ≤ i < individuosLen, 0 ≤ k < pagosLen) ∧L (|pagos[i]|, |pagos[i][k]| > 0))}
  requiere {(∀n : ℤ) ((0 ≤ n < |eventos|) ∧L |eventos[n]| > 0)}
  asegura {(∀i, p : ℤ) (recorreIndividuosPasos(i, p) ∧L if cooperan[i] then trayectorias[i][p] = ufc(p) else trayectorias[i][p] =
    ufc(p) fi)}

  pred recorreIndividuosPasos (in i : ℤ, in p : ℤ) {
    0 ≤ i < individuosLen ∧L 0 ≤ p < pasosLen
  }

  aux cuantoGano (in i : ℤ, in p : ℤ) : ℝ = trayectorias[i][p] * apuestas[i][eventos[i][p]] * pagos[i][eventos[i][p]];

donde individuosLen = cooperan.len
donde pasosLen = eventos[0].len
donde pagosLen = pagos[0][0].len

```