**Ejemplo 3**: “Modelando la solución de mi proyecto final”

## Plantear el proceso genérico para solucionar el problema de su proyecto.

Al iniciar el problema se piden desde consola los siguientes datos al usuario:

* Un **int** que se pide al usuario que indica el **número de generaciones** que va a transcurrir el programa.
* Un **int** que indica el tipo de ambiente al que se tendrán que adaptar los organismos.

1. Se inicializa la población de forma aleatoria, esto es se les asignan valores aleatorios a todos los individuos.
2. Mientras el número de generaciones sea menor al que dio el usuario este proceso
   * 1. Se evalúan todos los individuos sumando los valores correspondientes a la tabla de evaluación.
     2. Se seleccionan los mejores individuos de dicha población, usando el número de tipo de ambiente que dio el usuario.
     3. Con los individuos seleccionados se generan nuevos individuos.
     4. Se usan estos individuos para la nueva población.
   1. Se repite este proceso hasta que esté llena la nueva población.
   2. Se remplaza la población antigua con la nueva.
   3. Guardo al mejor individuo.
   4. Despliego la población en pantalla.
   5. Incrementar la generación actual

## Dividir en sub-problemas su estrategia general, es decir, simplificar el problema grande dividiéndolo en problemas más pequeños.

### Sub-problema inicializar población:

Se inicializa la población de forma aleatoria, esto es se les asignan valores aleatorios a todos los individuos.

### Sub-problema evaluación:

Se evalúan todos los individuos sumando los valores correspondientes a la tabla de evaluación.

### Sub-problema selección:

Se seleccionan los mejores individuos de dicha población, usando el número de tipo de ambiente que dio el usuario.

### Sub-problema selección:

Con los individuos seleccionados se generan nuevos individuos.

### Sub-problema repoblación:

Se usan estos individuos para la nueva población.

Se repite este proceso hasta que esté llena la nueva población.

Se remplaza la población antigua con la nueva.

Guardo al mejor individuo.

Despliego la población en pantalla.

## Crear el algoritmo correspondiente para cada sub problema.

### Sub-problema inicializar población:

inicializarPoblación(**población**)

1. Mientras la falten individuos en la **población**
   1. Rellenar cada individuo con valores aleatorios 1 o 0

### Sub-problema evaluación:

evaluaPoblacion**(poblacion)**

1. Mientras la falten individuos en la **población**
   1. Multiplicar el valor de la posición correspondiente de la característica del individuo con su valor correspondiente en la tabla de características.
   2. Sumar todos los valores del individuo y guardar el valor en la calificación de individuo.
   3. Moverse al siguiente individuo.

### Sub-problema selección:

seleccionaIndiviuo**(poblacion)**

1. selecciona un individuo1 al azar
2. selecciona un individuo2 al azar
3. selecciona un individuo3 al azar
4. compara las calificaciones de los 3 individuos
5. devuelve el individuo con la calificación más grande

### Sub-problema generación:

generaHijo(**papa**, **mama, hijo**)

1 Mientras queden características en papa

* 1. Genera un numero aleatorio y guárdalo en **rand**
  2. si **rand** es mayor o igual que 5
     1. copia la característica de papa
  3. si **rand** es menor que 5
     1. copia la característica de mama

### Sub-problema repoblación:

repoblación(**poblacion**)

* 1. Guardar0 en **individuoActual**
  2. Declarar individuo **mama**
  3. Declarar individuo **papa**
  4. Mientras **individuoActual** sea menor que **numPoblación.**
     1. Llamar procedimiento evaluaPoblacion**(poblacion)**.
     2. Llamar función seleccionaIndiviuo**(poblacion)** guardar en **papa**
     3. Llamar función seleccionaIndiviuo**(poblacion)** guardar en **mama**
     4. Llamar función generaHijo(**papa**, **mama, hijo**) y guardar en hijo;
     5. Guardar individuo en **nuevaPoblación**.
     6. Incrementar **individuoActual** en 1.
  5. Se remplaza la población antigua con la nueva.

### Sub-problema mejor individuo:

mejorIndividuo(**poblacion**)

1. Guarda 0 en **califMax**
2. Mientras la falten individuos en la **población**
   1. Si calificación de individuo es mayor que **califMax**
      1. Guardar **calificación** del individuo en **califMax**
      2. Guardar individuo en **mejorIndividuo**
3. Devolver **mejorIndividuo**

### Sub-problema desplegar poblacion:

desplegarPoblacion(**poblacion**)

1. Mientras la falten individuos en la **población**
   1. Despliega individuo en consola.

1. **Crear el algoritmo general que mande a llamar a todos los demás algoritmos.**

Algoritmo General:

1. Pedir generaciones al usuario y guardarlo en **limiteGeneracion**
2. Pedir ambiente al usuario y guardarlo en **ambiente**
3. Declarar **población** como una matriz
4. Declarar **mejor** individuo
5. Llamar procedimiento inicializarPoblación(**población**).
6. Mientras el **generacionActual** sea menor que **limiteGeneracion**
   1. Llamar Procedimiento repoblación(**poblacion**)
   2. Llamar función mejorIndividuo(**poblacion**) y guardarlo en **mejor**.
   3. Llamar desplegarPoblacion(**poblacion**)