# Descripción de las estructuras de datos desarrolladas:

Para la implementación del algoritmo genético

* Para modelar la población, la cual corresponde a los 22 jugadores; 11 en cada equipo se procedió a representarla como una lista con dos sublistas. Cada sublista es un equipo y cada equipo están compuestas por jugadores. Donde cada jugador se representa como una lista. La composición de un jugador consta de tres atributos codificados en binario y un ID para identificar si se trata de un portero, defensa, media cancha y delantero. En la figura 1 se puede apreciar la estructura de datos empleada para la población.

Diagrama

Descripción generada automáticamente

**Figura 1**. Estructura de datos para población

Para la implementación de movimientos

Para la implementación de jugadores en interfaz

# Descripción de los algoritmos desarrollados:

Algoritmo genético:

El algoritmo genético implementado en el proyecto consta de 5 etapas y una etapa inicial.

Primero se genera una población inicial con atributos aleatorios, una vez creada esta población inicia el algoritmo genético.

En la etapa “Population” se toma todos los elementos generados y les asigna un ID de player (1: portero, 2: defensa, 3: media cancha, 4: delanteros). Luego esta generación entra a la etapa de la aptitud donde se calcula su potencial.

En la etapa “Calculate fitness” se procede a recorrer cada equipo para analizar sus jugadores. La idea es que el jugador ideal sea aquel con la mayor cantidad de unos en sus atributos. Por consiguiente, se va a contar los bits

Diagrama

Descripción generada automáticamente

**Figura 2**. Diagrama de flujo de algoritmo genético

# Problemas encontrados:

1. Uno de los problemas encontrados es que el algoritmo genético variaba las características de los jugadores, mas no en el juego en sí, es decir; el algoritmo genético no refleja los movimientos de los jugadores. Por lo que la parte grafica debía proponer una manera de que se visualizara un partido de manera práctica
2. Para actualizar el movimiento de un jugador, se debe saber la posición de todos los demás jugadores. Ya que se debe repintar el canvas de la ventana y para ello, se deben volver a pintar los jugadores en sus posiciones antiguas. Esto implica tener un control mayor en la parte del frontend.
3. El movimiento de los jugadores se ve con una vibración a la hora de moverse, ya que nunca alcanzan sus puntos finales debido a que la cantidad de pixeles que avanzan no les permite llegar.