**Método de ingeniería.**

**FASE 1: IDENTIFICACIÓN DEL PROBLEMA.**

El problema que debemos solucionar con base en el enunciado es que debemos modelar un sistema en el que se puedan generar una cantidad de personas con determinada por el usuario. Las personas se generan con un código aleatorio generado por el sistema, un nombre, apellido, sexo, fecha de nacimiento, estatura y nacionalidad. Este sistema debe permitir generar una cantidad grande de usuarios. También se debe poder buscar una persona por coincidencias, cada que se ingrese un carácter de la búsqueda se debe actualizar para poder identificar cada usuario creado. Además de esto, se debe permitir borrar una persona seleccionada en la lista, también editar una persona exceptuando el código y guardar esta información de la persona. Seguidamente, este programa debe permitir guardar la información de las personas generadas, para que cada vez que cierre y abra la aplicación no se tengan que generar las personas otra vez, además, se deben permitir cargar estos archivos cada que se inicie la aplicación para que se tenga la información de las personas que se crearon en otra oportunidad.

**FASE 2: RECOPILACIÓN DE INFORMACIÓN NECESARIA.**

Un árbol binario puede definirse como un árbol que en cada nodo puede tener como mucho grado 2, es decir, a lo más 2 hijos. Los hijos suelen denominarse hijo a la izquierda e hijo a la derecha, estableciéndose de esta forma un orden en el posicionamiento de los mismos. Todas las definiciones básicas que se dieron para árboles generales permanecen inalteradas sin más que hacer las particularizaciones correspondientes. En los árboles binarios hay que tener en cuenta el orden izqda-drcha de los hijos. Por ejemplo: los árboles binarios **a)** y **b)** de la figura 1(adoptamos el convenio de que los hijos a la izquierda son extraídos extendiéndonos hacia la izquierda y los hijos a la derecha a la derecha) son diferentes, puesto que difieren en el nodo 5. El árbol **c** por convenio se supone igual al **b)** y no al **a)**.

La búsqueda más eficiente se efectúa en un árbol binario balanceado. Desafortunadamente, la función Inserta no asegura que el árbol permanezca balanceado, el grado de balance depende del orden del orden en que son insertados los nodos en el árbol. La altura de un árbol binario es el nivel máximo de sus hojas (profundidad). La altura del árbol nulo se define como –1. Un árbol binario balanceado es un árbol binario en el cual las alturas de los dos subárboles de todo nodo difieren al o sumo en 1. El balance de un nodo en un árbol binario se define como la altura de su subárbol izquierdo menos la altura de su subárbol derecho. Cada nodo en un árbol binario balanceado tiene balance iguala1, -1 o 0, dependiendo de si la altura de su subárbol izquierdo es mayor que, menor que, o igual a la altura de su subárbol derecho.

**FASE 3: BÚSQUEDA DE SOLUCIONES CREATIVAS.**

1era idea: Usar un árbol binario para guardar los datos de las personas y que sea una manera rápida de realizarlas.

2da idea: Para generar los archivos usar un código autogenerado random para tener una menor probabilidad de que se repita.

3ra idea: Para los porcentajes de fechas y países tener un valor random con cantidades decimales para poder que se tenga la misma cantidad de personas por cada país como se pide en el enunciado.

4ta idea: Usar un árbol balanceado para poder que añadir y borrar una persona sea rápida y que la búsqueda de cada persona sea más rápida que el árbol binario.

5ta idea: Guardar los datos de las personas en un Json para que se pueda cargar cada información de personas al cerrar la aplicación sin perder un dato.

6ta idea: Guardar los datos de las personas en un archivo csv para poder leer ese archivo cada vez que se abra la aplicación y así no perder los datos.

**FASE 4: TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS PRELIMINARES.**

Las ideas factibles para este caso son las segunda, tercera, cuarta y quinta. Esto porque es más fácil tener una función que genere un id random para cada persona y no tener un tipo de orden para que las opciones no se acaben tan rápido. Además, tener un random para que se creen las personas correspondientes a cada país según la población de América en 2020 es algo factible porque este valor siempre estará dentro de los parámetros y es mucho más probable que se cree en donde el porcentaje tiene un rango mayor. Seguidamente, el árbol balanceado es la mejor opción ya que es un árbol que en tiempo de búsqueda es mucho más rápido que el árbol binario por su balanceo, esto hace que nunca pueda queda en una lista lo que haría que el tiempo fuera muy largo. Y por último, guardar los datos es un Json es lo más viable porque guardarlo en un csv se demoraría un tiempo muy largo para guardar y cargar, en cambio con el Json no y los datos no se pierden.

**FASE 5: EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN.**